Deep Embedded Clustering in Few-shot Representations (DECiFR)

Yasaman Esfandiari
HRL Laboratories
Malibu, CA 90265
yesfandiari@hrl.com

Rodolfo Valiente
HRL Laboratories
Malibu, CA 90265
rvalienteromero@hrl.com

Amir Rahimi
HRL Laboratories
Malibu, CA 90265
amrahimi@hrl.com

Abstract

Few-shot Learning (*FSL*) has been the center of attention in the deep learning community as it can potentially address the problem of data inaccessibility. Several approaches have been proposed to learn from a few samples efficiently, nevertheless, the majority of them use a large dataset to generalize the feature representation obtained from a single or pre-defined set of backbones before adapting to novel classes. In this paper, different from prior works that use a single best-performing backbone, we present a model-agnostic framework that does not require to "*decipher*" which backbone is more suitable for the specific FSL task. We propose the Deep Embedded Clustering in Few-shot Representations (DECiFR) algorithm that leverages Deep Embedded Clustering (DEC) to abstract discriminative information from the best combination of features from different backbones, by simultaneously mapping and clustering feature representations using deep neural networks. Subsequently, we propose a contrastive variant of KNN to enhance the cluster separation by propagating through the samples that minimize the inter-class distance and maximize the intra-class distance. Empirical results show that our approach not only enhances the feature embeddings but also boosts the classification accuracy, approaching or surpassing state-of-the-art performance on numerous datasets.

1 Introduction

The impressive performance of Deep Neural Network (DNN) models in a variety of tasks such as image classification, speech recognition, etc. has been explored by many researchers. However, these DNN models need thousands of training samples and will perform poorly when labeled training data is limited (20), expensive (50) or hard to be adapted to (32). To address this limitation, inspired by humans who learn from a small number of instances, various FSL algorithms have been proposed. Generally, FSL algorithms are divided into two broad classes, i.e., meta-learning (61) and non-meta-learning (56). In meta-learning approaches, a meta-model is trained to provide priors (e.g. feature extractors) and gets fine-tuned based on the few learning samples (aka support set). However, in non-meta-learning approaches, a pre-trained model is loaded and fine-tuned using the support set (9; 22). Researchers have been proposing different algorithms that exploit the feature space to come up with the best FSL accuracy (68; 56; 34). However, the results from the majority of these studies are heavily reliant on the pre-trained backbone, and there exists a gap in studying the importance of the feature extraction step in FSL tasks. Aside from some studies (59; 41) this area remains unexplored.

In this paper, the Deep Embedded Clustering in Few-shot Representations (DECiFR) algorithm is proposed which is a model-agnostic framework for FSL. In other words, unlike many other FSL approaches that use a single best-performing backbone (EsViT, ResNet18, etc), our framework does not require prior knowledge of backbone performance to "*decipher*" which backbone is more suitable for a specific FSL task. Additionally, DECiFR outperforms many of the previous works (Fig. 1) in

R0-FoMo: Workshop on Robustness of Few-shot and Zero-shot Learning in Foundation Models at NeurIPS 2023.

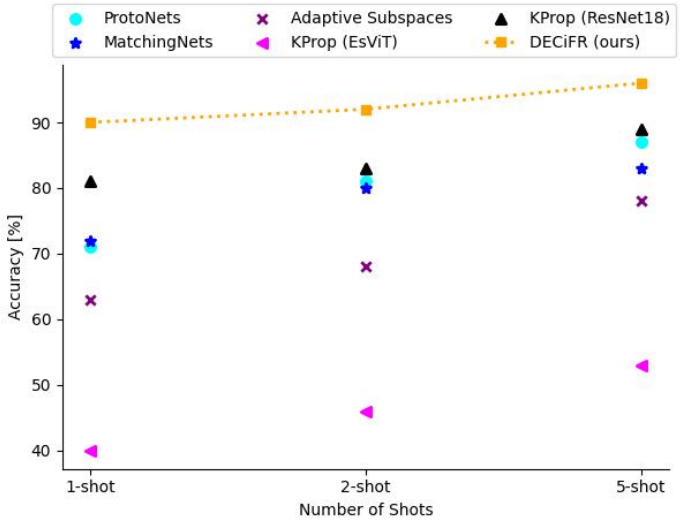


Figure 1: 5-way 5-shot accuracy using DECiFR compared with SOTA on CUB200 dataset. Our method, DECiFR, is shown in orange.

various $1, 2$, and 5-shot few-shot classification tasks. Our proposed pipeline consists of three major components: I) feature learning (module 1 in Fig. 2), II) label propagation (module 2 in Fig. 2), and III) classification (module 3 in Fig. 2). For the feature learning component, we are leveraging a feature preprocessing stage with the DUNN index along with Deep Embedded Clustering (DEC) (65) approach to achieve a better clustering representation based on a pre-selected ensemble of pre-trained model architectures. More specifically, the use of DEC is tailored to FSL tasks compared to other generative models as the confidence level of the feature space data points is taken into account with more emphasis on the data point with higher confidence. Additionally, the large-sized clusters which can distort the role of smaller hidden clusters in other generative models (e.g. autoencoders) are accounted for in DEC. Also, the designed target and input distributions are uniquely designed and shown to strengthen predictions (65).

To further improve our proposed algorithm, we refined the label propagation component of the pipeline which has been extensively used in FSL (17; 69). One of the common practices for label propagation is by using the k-nearest-neighbors (KNN) algorithm (31; 14). As authors in (14) showed, the accuracy of the down-stream few shot learning task is not improving by adding more pseudo-labels derived by KNN. This observation led us to fine-tune the KNN algorithm. Inspired by (8) we proposed a contrastive-KNN (denoted by cKNN) approach in which pseudo-labels are inferred by generating a graph of labeled points and selecting the label that minimizes the inter-class distance while maximizing the intra-class distance, propagating the labels in a contrastive KNN setting. Lastly, we adopt the KPCA method which was utilized in FSL by authors in (14). In KPCA, the principal components of the inputs in the feature space are learned and their distances are compared with unlabeled samples added by an error term. The unlabeled samples are then assigned to the class for with the least error compared to others.

To evaluate our proposed algorithm, we used a variety of few-shot image classification datasets (e.g. RESISC45, CUB200, Eurosat, etc.) and analyzed the performance of each of the components of the algorithm. We compared the cluster qualities corresponding to various backbones with 2-D TSNE plots and DUNN index metrics. Finally, we compared the few-shot classification accuracy of DECiFR with other state-of-the-art algorithms.

Our contributions can be listed as follows:

- We propose a novel algorithm (DECiFR) to improve the feature representation learning component of FSL. We evaluated our algorithm on a variety of FSL image classification datasets. Our results show that DECiFR improves the cluster quality of the features, resulting in superior performance compared to a variety of state-of-the-art FSL methods.

- We propose adding a contrastive flavor to KNN, by not only minimizing the inter-cluster distance but also maximizing the intra-cluster distance (cKNN).

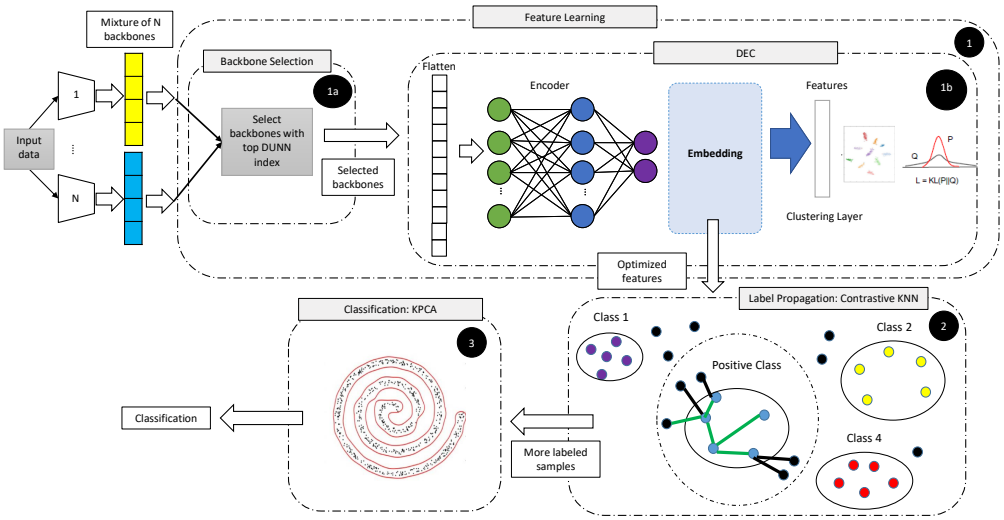


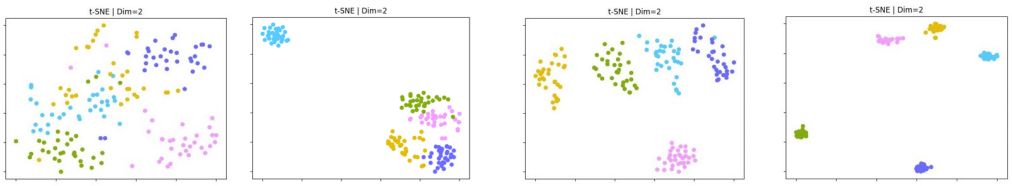
Figure 2: DECiFR architecture consists of three components: 1) Feature Learning, 2) Label Propagation, and 3) Classification. Given a mixture of backbones, 1a) Useful backbones are pre-selected based on their corresponding DUNN indexes. 1b) The flat versions of selected backbones are used to learn a more discriminative representation by simultaneously mapping and clustering features using DEC. 2) The optimized features are used in the cKNN module and more labeled samples are added in the feature space. 3) KPCA is applied to obtain the final classifications.

2 Framework

Generally, there are three components involved in most FSL algorithms. I) Mapping the inputs to the feature space II) Label Propagation III) Few-shot Classification. In this section, we are going to elaborate on each of the components and present our DECiFR algorithm.

Feature Extraction Component: In both meta-learning and non-meta-learning approaches to FSL, feature extraction is a key part to prevent noisy inputs from intruding into the clustering algorithms. Without any prior knowledge about the dataset, it is very challenging to know which pre-trained backbone provides the few-shot classifier with more useful clusters. Fig. 3 shows the significance of feature extraction in FSL. Fig. 3b shows that the data within similar clusters are more densely connected to each other, and the data corresponding to different clusters are located farther apart (better cluster quality) when ResNet18 architecture is used compared to Fig. 3a in which EsViT backbone is used. It is also useful to compare the corresponding 5-way 5-shot few-shot classification accuracy when the same classification method (e.g. DECiFR) is applied. The noteworthy accuracy reduction of 37.92% further motivates the need for devising an algorithm to improve the quality of the input feature embeddings as much as possible.

In a nutshell, our main idea is to train an unsupervised classifier to provide the users with model-agnostic backbones. To this goal, we utilize a mixture of backbones along with the unsupervised DEC framework which learns feature representations and cluster assignments using deep neural networks resulting in a more discriminative feature space. As a preprocessing module (module 1a in Fig. 2), we utilize the DUNN index (DI)- which is calculated as a ratio of the smallest distance within a cluster to the largest distance between two separate clusters- to select a few backbones with higher DI from a pool of backbones. A high DI translates to better clustering since samples in each cluster are densely connected, while separated clusters are further away from each other. Then, given a selected number of backbones, the DEC module (module 1b in Fig. 2) is implemented to learn more representative feature representations and cluster assignments through the training of an autoencoder



(a) few-shot accuracy = 52.94% (b) few-shot accuracy =90.86% (c) few-shot accuracy = 91.29% (d) few-shot accuracy = 96.76%

Figure 3: TSNE plots of feature spaces corresponding to CUB dataset for 5-way 5 shot few-shot classification using a) EsViT backbone, b) ResNet18 backbone, c) DEC autoencoder latent space using 4 sampled backbones (ResNet18, WideResNet101, EsViT, SimCLR), and d) DEC autoencoder latent space using 4 selected backbones by DUNN-index (ResNet18, ResNet20, WideResNet101, RegNet128).

that learns from a mixture of backbones as an input. DEC (65) learns a mapping from the data space to a lower-dimensional feature space by iterative optimizations of the cluster assignment and the underlying feature representation.

Basically, *DEC* learns the optimized representation in two steps: First, an autoencoder is pre-trained to minimize a reconstruction loss, measured by Mean Squared Error (MSE) as,

$$L_{rec} = \|x_i - f_{w_d}((f_{w_e}(x_i))\|^2 \tag{1}$$

where the mapping f_{w_e} maps the input x_i to the embedded space Z using learnable parameters ω. In other words, $z_i = f_{w_e}(x_i) \in Z$. Subsequently, the decoder attempts to recreate the original input from the lower-dimensional embedded space, to obtain the reconstruction of x_i, i.e, $y_i = f_{w_d}(z_i)$, w_e and w_d represent the encoder and decoder parameters respectively.

Second, after the autoencoder is trained, the decoder is removed and the encoder is fine-tuned by optimizing a KL divergence objective function between the soft cluster assignments q_i and the auxiliary distribution p_i, as,

$$L = \mathrm{KL}(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{2}$$

It should be noted that p_{ij} denotes the probability of sample i belonging to cluster j with distribution P. Also, q_{ij} quantifies the similarity between z_i (sample i) and the cluster center μ_j (probability of point i be assigned to the cluster j but with a distribution Q). using a Student's t-distribution following (60), q_{ij} is defined as:

$$q_{ij} = \frac{(1 + \|z_i - \mu_j\|^2)^{-1}}{\sum_j (1 + \|z_i - \mu_j\|^2)^{-1}}, \tag{3}$$

For DEC the selection of the auxiliary distribution P is critical and was one of the contributions of the original paper (65), similarly, we compute p_{ij} as:

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_j (q_{ij}^2 / \sum_i q_{ij})}, \tag{4}$$

Label Propagation Component: Based on the observations in previous research, adding more labeled samples using vanilla approaches like KNN, will not result in better FSL performance (14). To tackle this problem, inspired by (8), we propose a contrastive variant of KNN (denoted by *cKNN*) in which not only smaller distances from the labeled samples are favored, but also the propagated labels need to have maximal distances from other labeled samples (module 2 in Fig. 2).

In cKNN, to propagate a label of class c_n we first compute the K closest neighbors to data points $x_i \in c_n$ (x_i are the labeled points belonging to class c_n) from all the unlabeled points x_j, by running

K iterations of arg $\min_j \min_i ||x_i - x_j||$. Subsequently, we compute the distance of the selected K closest neighbors (x_k) to the remaining labeled data points x_l (x_l are the remaining labeled points belonging to other classes $x_l \notin c_n$), and select the data point that is farther to the closest x_l (intra-class distances), i.e, $\text{argmax}_k \min_l ||x_k - x_l||$. The selected x_k is then included in the set of labeled points assigned to class c_n. It should be noted that at each propagation step, we label one additional data point. We iteratively repeated this process to add a pre-defined number of additional labels for each class.

Few-shot Classification Component: As the last component in our pipeline (module 3 in Fig. 2), we use the labeled samples along with the pseudo labels from cKNN as inputs to the Kernel Principle Component Analysis (KPCA).

KPCA is a kernelized version of PCA, that extends the linear PCA to non-linear data mapping. First, with all the available labels (after cKNN label propagation) we compute a kernel matrix K for each class. For a kernel function, a Gaussian kernel is used, i.e, $K = k(x_i, x_j) = \exp(-||x_i - x_j||^2/(2\sigma^2))$, in which, x_i, x_j are the data point i, j respectively (data points in class c) and σ^2 is a hyperparameter that represents the width of the Gaussian kernel. Then, when a new test image is presented, the embedded space z of the query image is used to compute the reconstruction error $L^c_{RE}(z)$ for each class c (14), defined as:

$$L^c_{RE}(z) = k(z,z) - \frac{2}{n}\sum_{i=1}^{n} k(z, x_i) + \frac{1}{n^2}\sum_{i,j=1}^{n} k(x_i, x_j) - \sum_{l=1}^{q} f_l(z)^2 \quad (5)$$

in which n is the number of data points in class c, f_l are the projections of z onto the principal components (defined in (14)) and q the number of principal components. Finally, the test image is classified based on the smallest $L^c_{RE}(z)$. Algorithm 1 presents our DECiFR method.

3 Experimental Results

Experimental Setup: We evaluated our proposed framework on six few-shot classification benchmark datasets, i.e, CUB (62), EuroSat (26), Plantvillage (30), RESISC45 (12), Mars surface and MSL curiosity images (1). We retrieved pre-trained models (on ImageNet-1k) ranging from transformer-based models (EsViT (36)) to ResNet models (ResNet18,ResNet50,WideResNet101, etc. (21)), and self-supervised contrastive models (SimCLR (8)). Using the pre-trained backbones, we transformed the unlabeled data into the corresponding feature spaces. Using the DUNN index metric as a pre-processing stage we select the models for which the DUNN index is within 1-standard deviation range of the highest DUNN index value, selecting $2-5$ backbones from a pool of 14 backbones. We then used the embeddings corresponding to the selected backbones as inputs to the DEC autoencoder. We train the DEC autoencoder for 1000 epochs using ADAM optimizer with $lr = 1e-3$ and $wd = 1e-8$.

We then perform 1000 trials, in each picking either 5 classes at random. In each trial, we passed the optimized features provided by the DEC autoencoder corresponding to selected classes to the label propagation module, which finds the closest unlabeled samples (using Euclidean distance) to the labeled samples which at the same time are farthest away from other classes. In our experiments, we do the label propagation such that the number of labeled samples for each class is 5 (for 1-shot and 2-shot we add 4 and 3 samples respectively). For 5 shot experiments we add 2 more samples to each class.

Finally, the labeled samples for each class will be passed to KPCA as the classification component. KPCA learns the kernels, for each of the classes, transforms them, and uses them along with the eigenvectors corresponding to the n largest eigenvalues (set to 4 here) to construct the loss function as discussed in Eq. 5.

Evaluation Metrics: We looked at TSNE Plots along with the DUNN-index to evaluate the cluster qualities for each of the stages of our method. By comparing Fig. 3c, and Fig. 3a for which the DUNN-Indexes are 0.35 and 0.16 respectively, we can observe that a higher DUNN index accounts for better cluster quality and as a result, a higher few-shot accuracy. More importantly, a comparison between Fig. 3b , Fig. 3c, and Fig. 3d shows that DEC autoencoder is enhancing the cluster quality when a combination of features are passed through it, which eventually enhances the few-shot

Dataset Name	Best Single Backbone	Worst Single Backbone	DECiFR
Mars Surface Image	39.39% (SimCLR)	22.50% (RegNet128)	**39.54%** [a]
RESISC45	**91.51%** (EsViT)	27.96% (RegNet128)	89.18% [b]
MSL Curiosity Image	72.77% (SimCLR)	29.77% (RegNet128)	**73.33%** [c]
PlantVillage	92.81% (EsViT)	28.91% (RegNet128)	**93.34%** [d]
CUB	90.93% (WResNet50)	36.52% (EfficientNet)	**96.76%** [e]
EuroSat	**90.07%** (EsViT)	23.81% (RegNet128)	89.81% [f]
Fewshot-CIFAR100	**61.92%** (WResNet50)	39.64% (SimCLR)	60.68% [g]
miniImageNet	**84.33%** (WResNet50)	64.21% (SimCLR)	84.07% [h]

[a] (ResNet50 – ResNet101 – ResNext50 – ResNext101- WResNet101), [b] (EsViT – ResNet50 – ResNext50)
[c] (SimCLR - ResNet101 – ResNext50 – ResNext101- WResNet101), [d] (ResNext101 - RegNet128)
[e] (ResNet20- ResNet18- WResNet101- RegNet128), [f] (EsViT- ResNet20- ResNext101- WResNet50- WResNet101)
[g] (WResNet50- ResNet50), [h] (WResNet50- ResNet18)

Table 1: 5-way 5-shot few-shot accuracy using DECiFR with selected backbone architectures by DUNN-index pre-processing compared to using best and worst performing single backbones.

accuracy. This performance is further enhanced if the top backbones selected at the pre-processing stage are fed into the DEC module (see Fig. 3d) both in terms of cluster quality and subsequently, few-shot accuracy. More extensive TSNE plots are presented in Section 5.3.

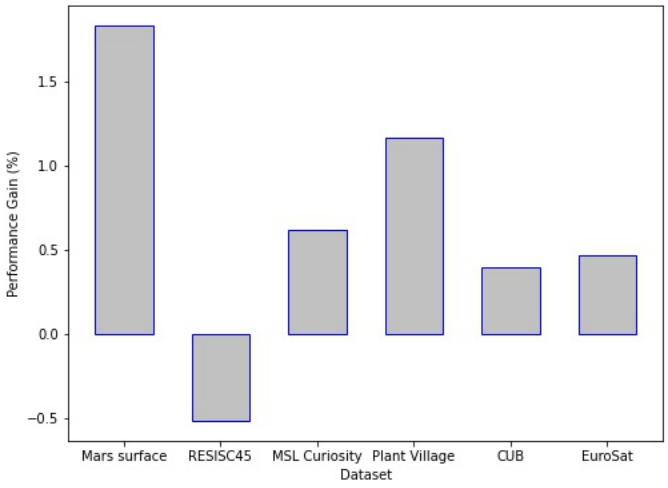


Figure 4: Accuracy gain by using DECiFR along with cKNN compared to KNN.

Next, we evaluated the effectiveness of our label propagation module by comparing the few-shot accuracy with and without cKNN. Fig. 4 shows that for most of the datasets, using cKNN, i.e, finding the unlabeled samples corresponding to the nearest positive classes and furthest negative classes, the few-shot accuracy improves. More comprehensive results on the effect of cKNN on the final few-shot accuracy is provided in Section 5.4.

To evaluate the accuracy of our method, we compared the accuracy of DECiFR compared to the best few-shot accuracy corresponding to using a single backbone. To make sure that our comparison also captures the feature abstraction effect alone, we also compared the performance of DECiFR using a single backbone along with our proposed cKNN label propagation module. Table 1 shows that DECiFR achieves comparable performance to using the best single backbones on most of the datasets, and shows superior performance on some of the datasets such as CUB, where the few-shot accuracy is about 6% more compared to using a single backbone. Based on our motivation, this is a great achievement to be as perfomant as or even better than a single best backbone without a prior knowledge about what the best backbone is. Also, Table. 1 shows that by using a single poor

model, the accuracy decreases substantially. As a result, by selecting a poor model, even the SOTA methods suffer substantially. However, DECiFR maintains its superior performance even if those poor models get selected in the pre-processing stage (refer to PlantVillage dataset where RegNet128 alone is performing unfavorably when used alone rather than the scenario where it is being used by DECiFR). It has also been observed that by passing features extracted from a single poor model to the DECiFR pipeline (even without combining it with other backbones) the few-shot accuracy is enhanced substantially (refer to Section 5.5). These observations along with Fig. 3, lead us to the conclusion that the DEC module alone is capable of abstracting useful features from a given feature embedding, and the combination of feature extraction backbones enhances the abstraction quality. The highest accuracies are highlighted in bold for each dataset.

4 Conclusion

In this paper, the DECiFR algorithm is proposed to improve the few-shot classification performance, by learning more discriminative representations from a mixture of backbones. We use a Deep Embedded Clustering (DEC) autoencoder to achieve better clustering quality corresponding to the feature extraction stage of the FSL pipeline given a mixture of pre-trained backbones. Experimental results show that using the DEC module alone enhances the feature abstraction quality even on single backbones, and can be further refined by using a combination of backbones selected at the pre-processing stage. We further improve our results by revising the label propagation module and showing its efficacy. We also compare the performance of DECiFR with state-of-the-art algorithms and show similar or in many cases superior performance without the need for prior knowledge about the quality of the backbone.

References

[1] Mars surface image (curiosity rover) labeled data set version 1.

[2] Bárbara C Benato, Jancarlo F Gomes, Alexandru C Telea, and Alexandre Xavier Falcão. Semi-supervised deep learning based on label propagation in a 2d embedded space. In *Iberoamerican Congress on Pattern Recognition*, pages 371–381. Springer, 2021.

[3] Yassir Bendou, Yuqing Hu, Raphael Lafargue, Giulia Lioi, Bastien Pasdeloup, Stéphane Pateux, and Vincent Gripon. Easy: Ensemble augmented-shot y-shaped learning: State-of-the-art few-shot classification with simple ingredients. *arXiv preprint arXiv:2201.09699*, 2022.

[4] Qi Cai, Yingwei Pan, Ting Yao, Chenggang Yan, and Tao Mei. Memory matching networks for one-shot image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4080–4088, 2018.

[5] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[7] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[9] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[10] Xiangyu Chen and Guanghui Wang. Few-shot learning by integrating spatial and frequency representation. In *2021 18th Conference on Robots and Vision (CRV)*, pages 49–56. IEEE, 2021.

[11] Yinbo Chen, Xiaolong Wang, Zhuang Liu, Huijuan Xu, and Trevor Darrell. A new meta-baseline for few-shot learning. 2020.

[12] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

[13] Tomáš Chobola, Daniel Vašata, and Pavel Kordík. Transfer learning based few-shot classification using optimal transport mapping from preprocessed latent space of backbone neural network. In *AAAI Workshop on Meta-Learning and MetaDL Challenge*, pages 29–37. PMLR, 2021.

[14] Joseph F Comer, Philip L Jacobson, and Heiko Hoffmann. Few-shot image classification along sparse graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4187–4195, 2022.

[15] Paras Dahal. Learning embedding space for clustering from deep representations. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 3747–3755. IEEE, 2018.

[16] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*, 2019.

[17] Matthijs Douze, Arthur Szlam, Bharath Hariharan, and Hervé Jégou. Low-shot learning with large-scale diffusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3349–3358, 2018.

[18] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Diversity with cooperation: Ensemble methods for few-shot classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3723–3731, 2019.

[19] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Selecting relevant features from a multi-domain representation for few-shot classification. In *European Conference on Computer Vision*, pages 769–786. Springer, 2020.

[20] Li Fei-Fei, Robert Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

[21] Vincent Feng. An overview of resnet and its variants. *Towards data science*, 2, 2017.

[22] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 06–11 Aug 2017.

[23] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[24] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *Ijcai*, pages 1753–1759, 2017.

[25] Xifeng Guo, Xinwang Liu, En Zhu, and Jianping Yin. Deep clustering with convolutional autoencoders. In *International conference on neural information processing*, pages 373–382. Springer, 2017.

[26] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

[27] Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. *Advances in Neural Information Processing Systems*, 32, 2019.

[28] Yuqing Hu, Stéphane Pateux, and Vincent Gripon. Squeezing backbone feature distributions to the max for efficient few-shot learning. *Algorithms*, 15(5):147, 2022.

[29] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017.

[30] David Hughes, Marcel Salathé, et al. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060*, 2015.

[31] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5070–5079, 2019.

[32] Ashraful Islam, Chun-Fu Richard Chen, Rameswar Panda, Leonid Karlinsky, Rogerio Feris, and Richard J Radke. Dynamic distillation network for cross-domain few-shot recognition with unlabeled data. *Advances in Neural Information Processing Systems*, 34:3584–3595, 2021.

[33] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, page 0. Lille, 2015.

[34] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10657–10665, 2019.

[35] SuBeen Lee, WonJun Moon, and Jae-Pil Heo. Task discrepancy maximization for fine-grained few-shot classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5331–5340, 2022.

[36] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. *arXiv preprint arXiv:2106.09785*, 2021.

[37] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1–10, 2019.

[38] Jialin Liu, Fei Chao, and Chih-Min Lin. Task augmentation by rotating for meta-learning. *arXiv preprint arXiv:2003.00804*, 2020.

[39] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. *arXiv preprint arXiv:1805.10002*, 2018.

[40] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, and Yi Yang. Transductive propagation network for few-shot learning. 2018.

[41] Jiawei Ma, Hanchen Xie, Guangxing Han, Shih-Fu Chang, Aram Galstyan, and Wael Abd-Almageed. Partner-assisted learning for few-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10573–10582, 2021.

[42] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

[43] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International Conference on Machine Learning*, pages 2554–2563. PMLR, 2017.

[44] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *International Conference on Machine Learning*, pages 3664–3673. PMLR, 2018.

[45] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems*, 31, 2018.

[46] Archit Parnami and Minwoo Lee. Learning from few examples: A summary of approaches to few-shot learning. *arXiv preprint arXiv:2203.04291*, 2022.

[47] 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.

[48] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

[49] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

[50] Mohammad Rostami, Soheil Kolouri, Eric Eaton, and Kyungnam Kim. Deep transfer learning for few-shot sar image classification. *Remote Sensing*, 11(11):1374, 2019.

[51] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.

[52] Tonmoy Saikia, Thomas Brox, and Cordelia Schmid. Optimized generic feature learning for few-shot classification across domains. *arXiv preprint arXiv:2001.07926*, 2020.

[53] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR, 2016.

[54] Daniel Shalam and Simon Korman. The self-optimal-transport feature transform. *arXiv preprint arXiv:2204.03065*, 2022.

[55] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4136–4145, 2020.

[56] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.
[57] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2019.
[58] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.
[59] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *European Conference on Computer Vision*, pages 266–282. Springer, 2020.
[60] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
[61] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.
[62] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.
[63] Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019.
[64] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7278–7286, 2018.
[65] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016.
[66] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *International Conference on Machine Learning*, pages 7115–7123. PMLR, 2019.
[67] Zhongjie Yu, Lin Chen, Zhongwei Cheng, and Jiebo Luo. Transmatch: A transfer-learning scheme for semi-supervised few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
[68] Ruixiang ZHANG, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial approach to few-shot learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
[69] Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16, 2003.
[70] Hao Zhu and Piotr Koniusz. Ease: Unsupervised discriminant subspace learning for transductive few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9078–9088, 2022.

5 Supplementary Material

5.1 Related Work

Few-shot Learning: Very broadly, research works addressing FSL can be divided into two categories: meta-learning (61; 56) and non-meta-learning (63; 59; 16). Meta-learning refers to algorithms that focus on learning priors from past experiences which can be effectively exploited in new learning tasks (46). The main approaches in meta-learning are metric-based (45; 33; 58; 66; 37; 61; 56; 55), optimization-based (22; 48; 57; 51), and model-based (42) approaches. While metric-based methods focus on developing algorithms that learn a distance function over data samples in prior tasks to generate discriminative embeddings, optimization-based methods try to learn the optimization model parameters for different tasks and communicate with the few-shot learner in an episodic manner.

On the other hand, model-based approaches involve developing model architectures that are specially designed for fast learning which can be memory-based (4; 53), rapid-adaptation-based (43; 44), etc. Additionally, some hybrid approaches to meta-learning include: Semi-supervised FSL(49), where they focus on using the few training samples in learning the meta-learner and improve the few-shot classifier; generative FSL algorithms (64) which try to generate more labeled samples from the limited support set; and transductive FSL (39; 27) proposes focusing on algorithms in which the relationship between members of the support set is analyzed. Rather than training a meta-learner, non-meta-learning algorithms focus on the use of pre-trained networks as feature extractors to learn distance metrics (63; 11; 14), train new classifiers (59), and perform transductive inference (16). Researchers are also interested in transfer learning methods to achieve more generalizable feature extractors(67).

Feature Importance: The performance of the classification task is demonstrated to be enhanced by a proper embedding that reduces the dimensionality of the data while maintaining the "relevant" information. Authors in (59) show that using a well-learned embedding might be more efficient than using complex meta-learning techniques. Similar works indicate that meaningful representations are more effective than other solutions for few-shot classification tasks (16; 47). Other methods that take advantage of the significance of features rely solely on feature extractors that have already been trained (52) or use multi-domain image representation to automatically choose the most relevant representation from a feature bank (19). In (47) it is shown that the reuse of features is the principal element behind the success of Model Agnostic Meta-Learning (MAML) (23).

Recent few-shot literature has also used distillation to take advantage of the availability of existing backbones (41; 59), by transferring previous knowledge from the teacher to the student model. Similarly, ensemble approaches, serve as an alternative to distillation, by using concatenated features obtained from various backbones (18; 38). However, as the use of multiple backbones increases the computational complexity, authors in (29) suggest the use of snapshots to improve efficiency. A different approach is to use generative models to learn a new embedding space for clustering. In that direction, DEC (65; 15; 24) proposed learning more meaningful features by jointly optimizing a reconstruction and a clustering loss. In (25) a convolutional version of DEC is presented that incorporates convolutional layers to preserve the local structure in the data. In this research, we focus on learning better features given existing backbones rather than learning a better backbone. We take inspiration from the DEC approaches, but differently, we used it for FSL classification to learn a more meaningful representation from pre-selected backbones that facilitate the classification tasks.

Label Propagation: For unlabeled data and self-supervised learning, label propagation has been widely utilized to infer pseudo-labels (69; 17; 5; 6). In that direction, label propagation through diffusion was introduced in a semi-supervised learning environment by (69). The authors in (7) introduce the transductive label propagation strategy and Liu *et al*.. (40) proposes a Transductive Propagation Network (TPN) that learns the label propagation parameters. Other methods investigated label propagation in feature space by building a k-nearest-neighbor network (31) or mapping the feature space onto a created 2-dimensional plane before propagating (2). Comer *et al*. (14) noted that closest neighbors are highly likely to belong to the same class and suggest utilizing label propagation to the closest unlabeled data before employing a kernel PCA reconstruction error as a decision boundary in the feature space. In contrast to previous work, we propose a contrastive version of KNN to generate a graph of labeled points by minimizing the inter-class distance while maximizing the intra-class distance. Our cKNN label propagation method is compatible with other FSL approaches.

5.2 DECiFR Algorithm

Algorithm 1 presents our DECiFR method. It received a set of images I, n_b pre-trained backbones from a set of backbones B ($b_n \subset B, n \leq n_b$), the DEC autoencoder architecture, the number of epochs to train the deep autoencoder ($epochs_{ae}$), and the number of epochs to optimize for the clustering ($epochs_c$), the label propagation algorithm(cKNN) and the FSL solver(e.g.KPCA). During backbone selection (1a) the DUNN Index for each backbone $b_n \subset B$ is computed and the top backbones $B_s \in B$ that are within 1σ of $max(dunn_n)$ are selected. DEC (1b) consists of two steps, i.e., training the autoencoder (step 1) and fine-tuning the clusters (step 2). First, an Autoencoder is trained to learn a non-linear mapping (f_{w_e}) from the concatenated features of selected backbones ($x = concat(B_s(I))$) to a lower-dimensional embedding space z_i by minimizing the reconstruction loss L_{rec} as in Eq. 1. After the autoencoder is trained, the decoder is discarded and only the encoder is used to compute the embedded space (z_i) where the clustering objective is then optimized. Subsequently, the embedded space is computed ($z_i = f_{w_e}(x_i)$) for all the data, and then K-means clustering is carried out in the feature space z_i to obtain k initial centroids μ_j. Second, given the trained autoencoder and initial centroids, the clustering objective is optimized. The soft assignment q_{ij} between μ_j and z_i as in Eq. 3 are computed, and the mapping f_{w_e} is updated by minimizing the KL divergence loss between the soft assignments q_{ij} and the target distribution p_{ij}. Finally, DEC learns a mapping to a lower-dimensional embedding space in which it optimizes the underlying feature representation and the cluster assignment $l_e = DEC(x_i)$. As the last components of the algorithm, Label propagation and Classification are performed as discussed thoroughly in corresponding sections.

Algorithm 1 Deep Embedded Clustering in Few-shot Representations (***DECiFR***).

Input: I, B, n_b, DEC, $cKNN$, $FSL\ solver(KPCA)$

for $n \leq n_b$ **do**
  $e_n = b_n(I)$ {calculate the feature embedding e_n}
  $dunn_n = DUNN(e_n)$
end for
Sort b_n based on $dunn_n$
Add top backbones within 1σ of $max(dunn_n)$ to B_s
$x = concat(B_s(I))$ concatenate selected backbones
Parameter initialization f_{w_e}: train the autoencoder
Compute the learned embedded space $z_i = f_{w_e}(x_i)$
Initialize cluster centroids μ_j using K-means
for $e \leq epochs_c$ **do**
  Compute q_{ij} and p_{ij} using Eq. 3, and Eq. 4
  $l_e = DEC(x_i)$ {refine the clusters}
end for
Label Propagation $pseudo = cKNN(l_e)$
Classification: Pass the labeled samples to KPCA

5.3 Cluster Quality Experimental Analysis

Table 8 presents the DUNN index values for all the 14 pre-trained backbones. The backbones for which the DUNN Index is within 1-standard deviation range of the highest DUNN index value are shown in bold, which were eventually selected to go into the DEC autoencoder at the pre-processing stage. The results are presented for 6 of the different few-shot image classification datasets we used, i.e., Mars Surface Images, RESISC45, MSL Curiosity, CUB200, PlantVillage, and Eurosat datasets. Additionally, Figures 5 to 10 present the 2-D TSNE plots for all the datasets, using a) Worst Single Backbone, b) Best Single Backbone, and c) DECiFR with selected backbones (shown in bold in Table 8). We can observe how the performance is enhanced if the top backbones selected at the pre-processing stage are fed into the DEC module, showing that the selected backbones with higher DUNN index account for better cluster quality and as a result, a higher few-shot accuracy. It is worth emphasizing that the goal here is to achieve the cluster quality comparable to the performance of the best backbone since the main goal here is to design a method that is capable of reasonable *FSL* learning performance without any prior requirement on the input backbone architecture.

Backbone	DECiFR	Without DECiFR
SimClr	72.63%	**72.91%**
ResNet101	**70.01%**	67.90%
ResNext50	**71.10%**	69.62%
ResNext101	69.42%	**70.78%**
WResNet101	**72.17%**	71.29%
Combination	**73.33%**	72.50%

Table 2: 5-way 5-shot accuracy of DECiFR vs KPCA alone using single backbone on MSL Curiosity Image dataset.

Backbone	DECiFR	KPCA alone
ResNext101	91.94%	**92.69%**
RegNet128	**29.10%**	28.91%
Combination	**93.34%**	86.47%

Table 3: 5-way 5-shot accuracy of DECiFR vs KPCA alone using single backbone on PlantVillage dataset.

Backbone	DECiFR	KPCA alone
ResNet20	**86.48%**	67.68%
ResNet18	86.96%	**89.40%**
WResNet101	89.18%	**90.04%**
RegNet128	**56.45%**	50.12%
Combination	**96.76%**	89.13%

Table 4: 5-way 5-shot accuracy of DECiFR vs KPCA alone using single backbone on CUB dataset.

Backbone	DECiFR	KPCA alone
ResNet50	38.99%	**40.51%**
ResNet101	**38.88%**	33.73%
ResNext50	34.19%	**35.49%**
ResNext101	38.66%	**38.79%**
WResNet101	**37.96%**	37.01%
Combination	**39.54%**	38.11%

Table 5: 5-way 5-shot accuracy of DECiFR vs KPCA alone using single backbone on Mars Surface Image dataset.

Backbone	DECiFR	KPCA alone
EsViT	**91.76%**	91.27%
ResNet50	86.27%	**87.59%**
ResNext50	85.01%	**86.40%**
Combination	**89.18%**	87.24%

Table 6: 5-way 5-shot accuracy of DECiFR vs KPCA alone using single backbone on RESISC45 dataset.

5.4 Contrastive K-Nearest-Neighbors

To further analyze the effect of cKNN module, we perform additional experiments in which we compare the few-shot accuracy with and without cKNN for different backbones. Table 9 shows the 5-way 5-shot accuracy of KPCA (the few-shot classification module only without DEC) using single backbones with vanilla KNN vs cKNN for label propagation. Results show that for the majority of the datasets, the few-shot accuracy improves by adding cKNN.

Backbone	DECiFR	KPCA alone
EsViT	90.13%	**90.92%**
ResNet20	80.53%	**81.64%**
ResNext101	82.90%	**85.14%**
WResNet50	82.30%	**86.28%**
WResNet101	**82.65%**	81.99%
Combination	**89.81%**	86.55%

Table 7: 5-way 5-shot accuracy with and KPCA alone with single backbone on EuroSat dataset.

	Mars Surface	RESISC	MSL Curiousity	Plant Village	CUB	EuroSat
EsViT	0.136	**0.290**	0.055	0.158	0.172	**0.215**
ResNet20	0.107	0.168	0.087	0.226	**0.308**	**0.225**
SimClr	0.103	0.205	**0.127**	0.222	0.233	0.199
ResNet50	**0.163**	**0.304**	0.107	0.206	0.201	0.193
ResNet18	0.108	0.168	0.087	0.198	**0.309**	0.115
ResNet101	**0.148**	0.230	**0.138**	0.203	0.120	0.178
ResNext50	**0.172**	**0.269**	**0.154**	0.227	0.182	0.163
ResNext101	**0.164**	0.256	**0.127**	**0.261**	0.202	**0.214**
WResNet50	0.140	0.235	0.111	0.222	0.194	**0.247**
WResNet101	**0.171**	0.248	**0.128**	0.249	**0.273**	**0.211**
RegNet128	0.145	0.205	0.088	**0.295**	**0.285**	0.131
RegNet8	0.124	0.207	0.110	0.216	0.146	0.107
EfficientNet	0.100	0.069	0.018	0.100	0.074	0.037
ConvNext	0.124	0.199	0.089	0.159	0.171	0.150

Table 8: DUNN indexes 14 various backbones. The backbones for which the DUNN Index is within 1-standard deviation range of the highest DUNN index value are shown in bold (selected $2-5$ backbones)

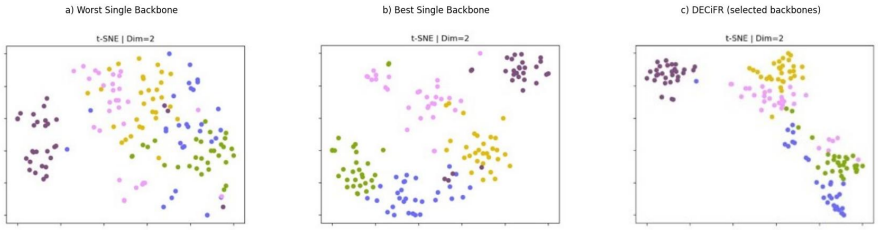


Figure 5: TSNE plots of feature spaces corresponding to Mars Surface Image dataset for 5-way 5 shot few-shot classification using a) Worst Single Backbone, b) Best Single Backbone, c) DECiFR with selected backbones

5.5 Experimental Results with DECiFR

We extend our few-shot classification results by comparing the few-shot classification accuracy of single backbones with and without DECiFR. In other words, to understand the effect of DEC module alone, we feed each single backbone to the DEC module and run KPCA afterward, and compare with the performance of single backbones which are directly fed into KPCA. Tables 2 to 7 present the 5-way 5-shot accuracy with and without DECiFR using a single backbone on different datasets. Results show that DECiFR maintains its superior performance across different datasets especially when the combinations of backbones are fed into it. This shows that the DEC module alone can boost the clustering performance of single models slightly which can be improved much further if a combination of backbones is fed into it. We should emphasize here that in the aforementioned experiments, we select single backbones and feed them into either DEC first or KPCA directly just to show the effect of DEC alone. However, the main contribution here is to design a method that can

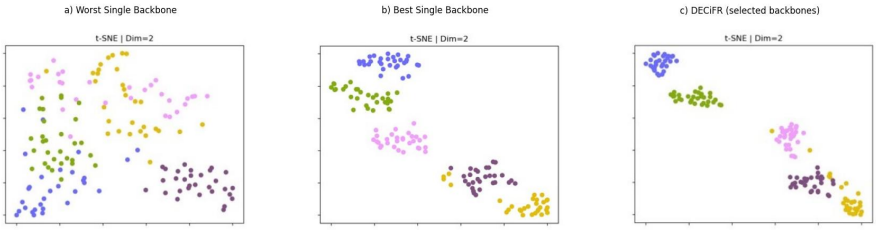


Figure 6: TSNE plots of feature spaces corresponding to EuroSat dataset for 5-way 5 shot few-shot classification using a) Worst Single Backbone, b) Best Single Backbone, c) DECiFR with selected backbones

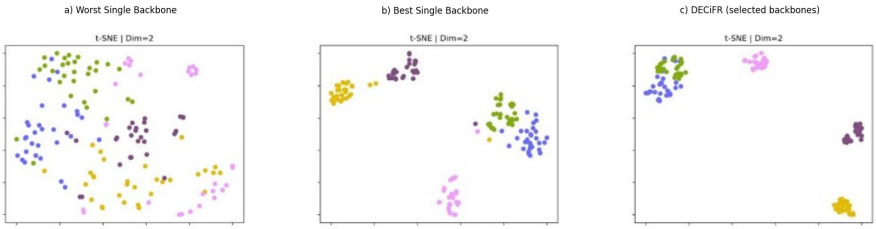


Figure 7: TSNE plots of feature spaces corresponding to MSL Curiosity Image dataset for 5-way 5 shot few-shot classification using a) Worst Single Backbone, b) Best Single Backbone, c) DECiFR with selected backbones

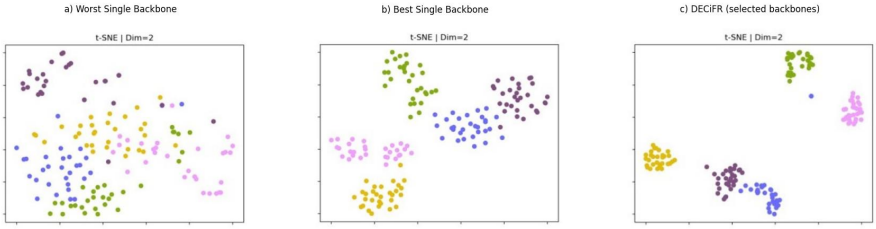


Figure 8: TSNE plots of feature spaces corresponding to PlantVillage dataset for 5-way 5 shot few-shot classification using a) Worst Single Backbone, b) Best Single Backbone, c) DECiFR with selected backbones

provide us with high FSL performance without any prior knowledge about the backbones. Obviously, SOTA methods (e.g. KPCA) may have much better performance if the "right" backbone model is fed into it, but their performance is very much dependent on the input backbones (shown in Tables 1 and 2 in (14)), whereas DECiFR can maintain its performance even if the worst model architectures are given to it (see PlantVillage results in Table 1 where RegNet128 has the worst performance by itself but the performance is enhanced by a large margin even if it is passed to DEC).

5.6 Additional Comparative Analysis

We evaluate the performance of our method by comparing it with SOTA methods on RESISC45 and EuroSat datasets. As these datasets are satellite imagery datasets, it is harder, even for human

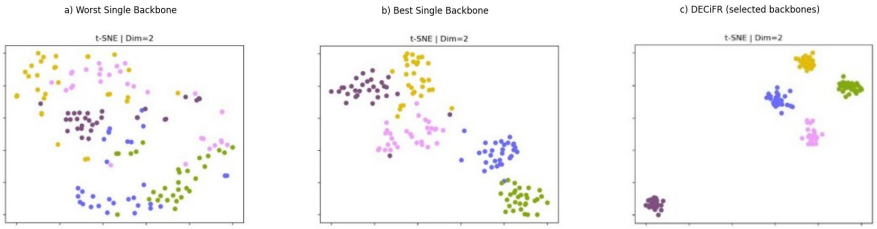


Figure 9: TSNE plots of feature spaces corresponding to CUB dataset for 5-way 5 shot few-shot classification using a) Worst Single Backbone, b) Best Single Backbone, c) DECiFR with selected backbones

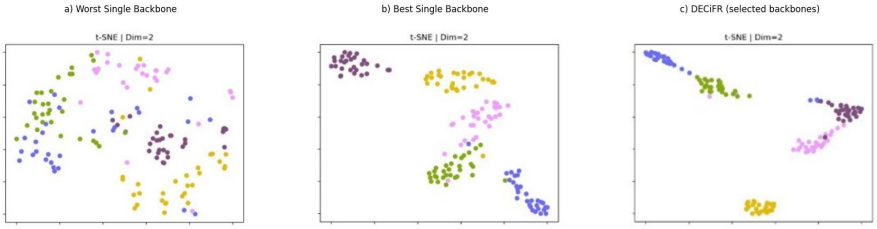


Figure 10: TSNE plots of feature spaces corresponding to EuroSat dataset for 5-way 5 shot few-shot classification using a) Worst Single Backbone, b) Best Single Backbone, c) DECiFR with selected backbones

eyes to learn the representations. Therefore, many of the well-known methods suffer from learning meaningful representations. Table. 10 and 11 show that DECiFR achieves comparable accuracy to state-of-art methods, and is very close to KProp in 5-way 5-shot experiments. It should be emphasized that KProp (along with most of the SOTA algorithms) uses the best-performing backbone for feature extraction without prior knowledge about the best-performing backbone the performance of these methods will degrade substantially (shown in Tables 1 and 2 in (14)). The main advantage of DECiFR is that it does not require prior knowledge of the backbone proficiency.

Backbone	Mars Surface	RESISC	MSL Curiosity	Plant Village	CUB	EuroSat
EsViT	34.02 / **35.18**	**91.51** / 91.27	68.13 / **70.96**	92.81 / **93.92**	**54.48** / 52.94	90.07 / **90.92**
ResNet20	**35.83** / 34.21	**86.70** / 84.71	69.25 / **69.32**	**92.71** / 91.12	90.64 / **90.86**	**83.06** / 81.64
SimClr	39.39 / **41.62**	**80.40** / 79.01	72.77 / **72.91**	90.82 / **91.15**	63.92 / **67.68**	81.29 / **85.17**
ResNet50	38.38 / **40.51**	86.07 / **87.59**	71.41 / **71.68**	**91.38** / 90.61	**90.61** / 88.07	**81.65** / 76.62
ResNet18	35.57 / **36.83**	86.00 / **87.14**	**70.96** / 68.94	**92.21** / 90.39	89.21 / **89.40**	82.71 / **83.23**
ResNet101	**35.94** / 33.73	84.18 / **87.83**	**72.38** / 67.90	91.67 / **91.90**	89.01 / **90.42**	81.73 / **83.74**
ResNext50	35.28 / **35.38**	84.38 / **86.40**	69.43 / **69.62**	**90.59** / 89.96	89.63 / **91.32**	81.60 / **81.80**
ResNext101	37.11 / **38.79**	84.61 / **85.33**	70.01 / **70.78**	91.46 / **92.69**	**89.91** / 89.37	82.57 / **85.14**
WResNet50	**41.52** / 41.15	86.35 / **89.54**	**71.22** / 71.16	**93.89** / 93.71	90.93 / **91.09**	82.62 / **86.28**
WResNet101	35.41 / **37.01**	**85.42** / 85.03	69.84 / **71.30**	93.21 / **93.71**	88.79 / **90.05**	**82.28** / 81.99
RegNet128	19.68 / **22.50**	27.88 / **27.96**	**31.48** / 29.77	26.13 / **28.91**	**50.15** / 50.12	**27.19** / 23.81
RegNet8	35.34 / **38.65**	84.60 / **86.84**	69.32 / **70.26**	**92.57** / 91.37	90.78 / **90.80**	81.39 / **83.49**
EfficientNet	30.90 / **31.80**	**61.81** / 57.57	**49.82** / 41.47	78.56 / **79.22**	36.36 / **36.52**	71.54 / **71.84**
ConvNext	38.17 / **40.61**	84.37 / **85.83**	70.32 / **71.20**	91.40 / **91.64**	88.89 / **90.89**	81.91 / **82.69**

Table 9: 5-way 5-shot accuracy using single backbones with KNN (left value) and cKNN (right value) for label propagation.

Method	1-shot	2-shot	5-shot
ProtoNets (56)	60.66%	72.18%	81.21%
MatchingNets (61)	60.21%	69.59%	75.75%
Adaptive Subspaces (55)	60.55%	69.17%	79.16%
KProp (14)	**83.17%**	**87.06%**	**92.08%**
DECiFR (ours)	65.93%	71.73%	89.18%

Table 10: 5-way few-shot accuracy using DECiFR compared with SOTA on RESISC45 dataset.

Method	1-shot	2-shot	5-shot
ProtoNets (56)	39.93%	44.97%	55.08%
MatchingNets (61)	37.40%	39.57%	44.86%
Adaptive Subspaces (55)	27.55%	36.85%	40.42%
KProp (14)	**77.20%**	**83.40%**	**90.44%**
DECiFR (ours)	64.54%	72.61%	89.81%

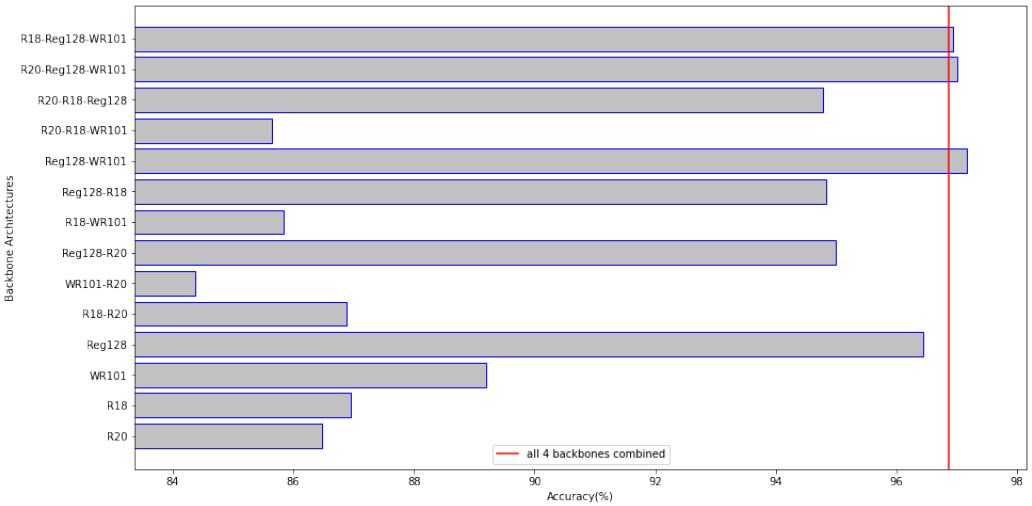
Table 11: 5-way few-shot accuracy using DECiFR compared with SOTA on EuroSat dataset.

To test the robustness of our proposed method, we also looked at the different combinations of selected backbones. Fig. 11a shows two interesting aspects of our algorithm: I) Incorporating the DEC autoencoder helps with the clustering performance, as RegNet128 used as a single backbone achieves few-shot accuracy of 96.45% after being passed to the DEC autoencoder whereas the accuracy is much less (52% which is shown in Section 5.5) if the features are used without using the DEC autoencoder. II) DECiFR is robust among all the different combinations of selected backbones at the pre-processing stage. To evaluate the robustness of DECiFR further, we selected 4 backbones at random, each of which is from a different class of backbones. We chose EsViT (transformer-based), SimCLR (contrastive self-supervision-based), ResNet20, and WideResNet50 model architectures. Fig. 11b shows that by combining the backbones based on the DECiFR pre-processing stage, the few-shot accuracy is higher, however, combining all of the features from these backbones still achieves higher accuracy compared to other combinations. In general, Fig. 11 shows that by combining all the 4 selected backbones from the pre-processing stage, we can achieve a comparably accurate model in comparison to other combinations. We also evaluated the robustness of DECiFR by using other backbone combinations which are discussed in Section 5.7.

Method	1-shot	5-shot
SOT (54)	**95.80%**	**97.12%**
PT+MAP (10)	**95.48%**	93.99%
PEMnE-BMS* (28)	94.87%	96.43%
LST+MAP (13)	91.68%	94.09%
EASE (70)	91.68%	94.12%
TDM (35)	84.36%	93.37%
ProtoNets (56)	71.88%	87.42%
MatchingNets (61)	72.36%	83.64%
Adaptive Subspaces (55)	63.30%	78.25%
KProp (14)	81.08%	89.39%
DECiFR (ours)	90.67%	**96.76%**

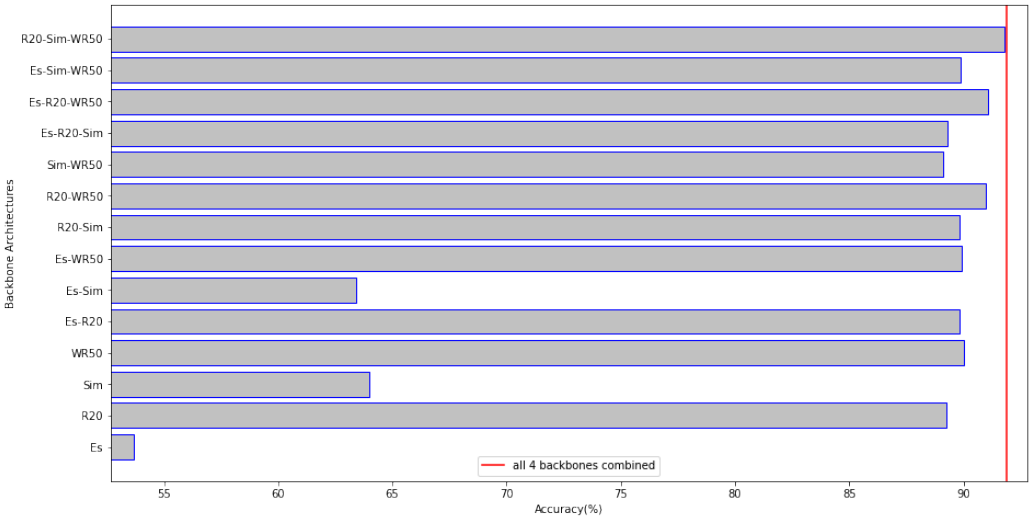
Table 12: 5-way few-shot accuracy using DECiFR compared with SOTA on CUB200 dataset. Top two accuracies are highlighted in bold

Following we compare the 1, 2, and 5 shot accuracy corresponding to DECiFR with other state-of-the-art methods on CUB, CIFAR100-FS, and MiniImageNet datasets when $n = 5$. Table 12 shows that DECiFR achieves the second-best 5-shot accuracy (also on the leaderboard) compared to the state-of-the-art methods. Additionally, it should be mentioned that the majority of methods in Tables 12, 10, and 11 use a single well-performing backbone. However, in real-world scenarios, it is challenging to *decipher* which backbone is more suitable for FSL, so the necessity for a model-agnostic FSL method is not addressed by many of the methods that DECiFR is compared against.



*Note: R18 = ResNet18, R20 = ResNet20, WR101= WideResNet101, Reg128 =RegNet128

(a)



*Note: Es = EsViT, R20 = ResNet20,Sim = SimCLR, WR50= WideResNet50

(b)

Figure 11: 5-way 5-shot few-shot accuracy of DECiFR using all the different combinations of 4 a) selected backbones by the pre-processing module (DUNN-index), b)randomly selected backbones on the CUB200 dataset.

Next, we evaluate the performance of our method by comparing it with SOTA methods on Fewshot-CIFAR100 and miniImageNet datasets. Table. 13 and 14 show that DECiFR achieves comparable accuracy to SOTA methods (second-best 1-shot accuracy). It should be noted that DECiFR does not rely on fine-tuning or prior knowledge of good-performing backbones, unlike the approach in (3), which currently achieves the best results. The backbone selection in (3) significantly impacts their performance, but our proposed algorithm does not require such prior knowledge. We do not require to "*decipher*" which backbone is more suitable for the specific FSL task, making DECiFR a more versatile approach.

5.7 Robustness

Finally, we present additional results on the robustness of our proposed method. Figures 12 to 16 present the 5-way 5-shot accuracy of DECiFR using all the different combinations of selected

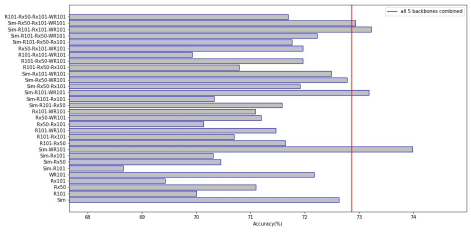
Method	1-shot	5-shot
ProtoNets (56)	41.54%	57.08%
KProp (14)	42.01%	57.91%
EASY Inductive (3)	47.94%	**64.14%**
EASY Transductive (3)	**54.47%**	**65.82%**
DECiFR (ours)	**48.35%**	60.68%

Table 13: 5-way few-shot accuracy using DECiFR compared with SOTA on the Fewshot-CIFAR100 dataset. The top two accuracies are highlighted in bold

Method	1-shot	5-shot
ProtoNets (56)	60.37%	78.02%
KProp (14)	71.95%	79.26%
EASY Inductive (3)	70.63%	**86.28%**
EASY Transductive (3)	**82.31%**	88.57%
DECiFR (ours)	**77.01%**	**84.07%**

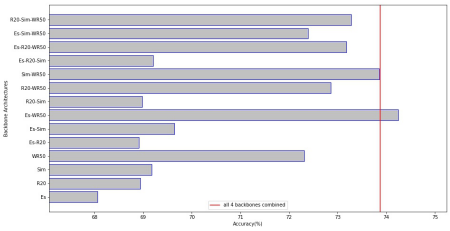
Table 14: 5-way few-shot accuracy using DECiFR compared with SOTA on the miniImageNet dataset. The top two accuracies are highlighted in bold

backbones by the pre-processing module (DUNN-index) versus 4 randomly selected backbones on various datasets. We can observe that by combining all the 4 selected backbones from the pre-processing stage, we can achieve a comparably accurate model in comparison to other combinations. An additional observation is that the accuracy using the selected backbones at the pre-processing stage is higher compared to a combination of 4 randomly selected models.



*Note: Sim = SimCLR, R101 = ResNet101, Rx50 = ResNext50, Rx101 = ResNext101, WR101= WideResNet101

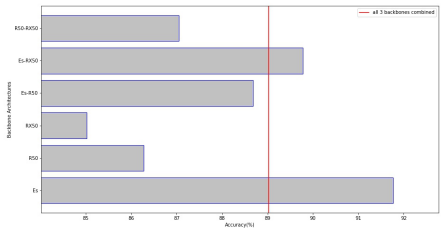
(a)



*Note: Es = EsViT, R20 = ResNet20, Sim = SimCLR, WR50= WideResNet50

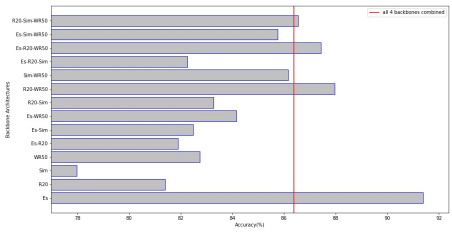
(b)

Figure 12: 5-way 5-shot few-shot accuracy of DECiFR using all the different combinations of a) 5 selected backbones by the pre-processing module (DUNN-index), b) 4 randomly selected backbones on the MSL Curiosity Rover dataset.



*Note: Es = EsViT, R50 = ResNet50, Rx50 = ResNext50

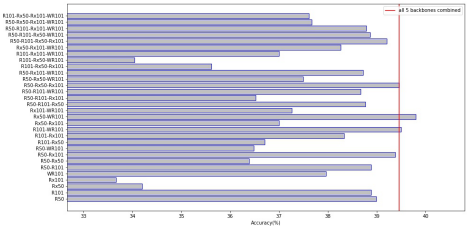
(a)



*Note: Es = EsViT, R20 = ResNet20,Sim = SimCLR, WR50= WideResNet50

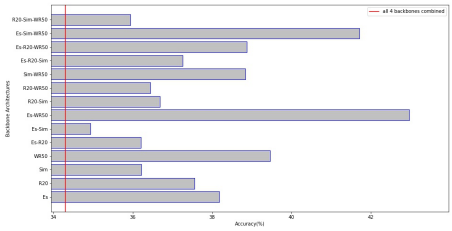
(b)

Figure 13: 5-way 5-shot few-shot accuracy of DECiFR using all the different combinations of a) 3 selected backbones by the pre-processing module (DUNN-index), b) 4 randomly selected backbones on the RESISC45 dataset.



*Note: R50 = ResNet50, R101 = ResNet101, Rx50 = ResNext50, Rx101 = ResNext101, WR101= WideResNet101

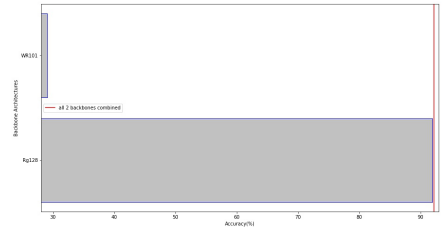
(a)



*Note: Es = EsViT, R20 = ResNet20,Sim = SimCLR, WR50= WideResNet50

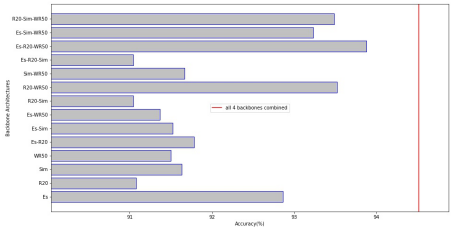
(b)

Figure 14: 5-way 5-shot few-shot accuracy of DECiFR using all the different combinations of a) 5 selected backbones by the pre-processing module (DUNN-index), b) 4 randomly selected backbones on the Mars surface images dataset.



*Note: WR101= WideResNet101, Reg128 =RegNet128

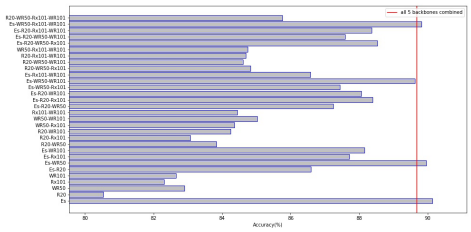
(a)



*Note: Es = EsViT, R20 = ResNet20,Sim = SimCLR, WR50= WideResNet50

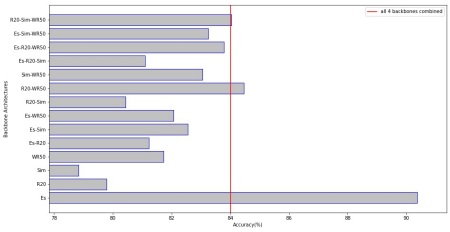
(b)

Figure 15: 5-way 5-shot few-shot accuracy of DECiFR using all the different combinations of a) 2 selected backbones by the pre-processing module (DUNN-index), b) 4 randomly selected backbones on the PlantVillage dataset.



*Note: Es = EsViT, R20 = ResNet20, WR50= WideResNet50, Rx101 = ResNext101, WR101= WideResNet101

(a)



*Note: Es = EsViT, R20 = ResNet20,Sim = SimCLR, WR50= WideResNet50

(b)

Figure 16: 5-way 5-shot few-shot accuracy of DECiFR using all the different combinations of a) 5 selected backbones by the pre-processing module (DUNN-index), b) 4 randomly selected backbones on the EuroSat dataset.