Experiments in Cross-Domain Few-Shot Learning for Image Classification

Hongyu Wang[a], Henry Gouk[b], Huon Fraser[a], Eibe Frank[1a], Bernhard Pfahringer[a], Michael Mayo[a], and Geoffrey Holmes[a]

[a]Department of Computer Science, University of Waikato, Hamilton, New Zealand

[b]School of Informatics, University of Edinburgh, Edinburgh, United Kingdom

ARTICLE HISTORY

Compiled February 14, 2022

ABSTRACT

Cross-domain few-shot learning for image classification has many practical applications. This paper attempts to shed light on suitable configurations of feature exactors and "shallow" classifiers to use in this machine learning setting. We apply ResNet-based feature extractors pretrained on two versions of the ImageNet (miniImageNet and ILSVRC2012) dataset to five target domains with different degrees of similarity to ImageNet, varying the size of the feature extractor, the stage of the network at which features are extracted, and the learning algorithm applied to the extracted features. We evaluate standard learning algorithms such as logistic regression and linear discriminant analysis, as well as variants thereof, and additionally consider the effect of normalising the feature vectors using various p-norms. We also apply multi-instance learning to improve utilisation of the training images. In our experiments, the cosine similarity classifier and ℓ_2-regularised 1-vs-rest logistic regression generally exhibit the best classification performance. We also find that algorithms such as linear discriminant analysis yield consistently higher accuracy using ℓ_2-normalised feature vectors. Features extracted from the penultimate stage of a ResNet-101 model, and multi-instance learning techniques, produce the highest accuracy for a majority of the target domains. Our results will inform practitioners who are considering the application of pretrained ImageNet feature extractors in cross-domain settings with small amounts of labelled training images in the target domain.

KEYWORDS

Cross-Domain Few-Shot Learning, Pretrained Feature Extractors, Normalisation, Transfer Learning, Multi-instance Learning

1. Introduction

Convolutional neural networks (CNN) have substantially improved the accuracy of supervised learning approaches for image classification, but large labelled sets of images are generally essential to obtain a high-accuracy classifier. Because of the cost of labelling data or other limitations, this may not be possible in all practical applications. This motivates the investigation of so-called "few-shot" learning approaches. Few-shot learning (FSL) tackles the problem of learning models in target domains that exhibit a very limited number of labelled training instances (Vinyals et al. 2016). To make this feasible, it is considered essential to exploit prior knowledge gleaned from a source domain that is related to the target domain and utilise it to compensate for the lack of labelled data in the target domain. In this manner, it is possible to obtain a model that is a more accurate predictor than a model trained only from the labelled data available for the target domain. Consequently, few-shot

[1] CONTACT Eibe Frank. Email: eibe@waikato.ac.nz

learning can be interpreted as a form of transfer learning (Pan and Yang 2010) from a domain with a large quantity of training data to another domain with much less data.

Many studies on FSL consider situations where there is only a small shift between the source and target domains (Vinyals et al. 2016, Snell et al. 2017). Commonly, experiments are performed based on a single dataset, using some classes as the source domain and employing the rest to form the target domain. In this paper, we consider the more realistic cross-domain few-shot learning (CD-FSL) scenario instead, which refers to few-shot learning tasks where the source domain data and the target domain data are from strictly different origins and not, e.g., derived from the same dataset (Guo et al. 2020). Compared with in-domain FSL, CD-FSL presents a potentially more challenging transfer learning problem, as its source and target domains are likely to differ significantly in input distribution. For example, while an FSL task that is purely based on ImageNet (Deng et al. 2009) has mutually exclusive class splits, images in both partitions still share common properties such as resolution, perspective, etc., since they are all natural images obtained in similar ways. On the other hand, in a CD-FSL scenario transferring from ImageNet to CIFAR-10 (Krizhevsky 2012), images in the source and target domains differ substantially in terms of resolution and perspective, yielding an inherently more challenging transfer learning task.

CD-FSL aims to enable efficient learning in one field with knowledge from another. This is consistent with the original intention of research into few-shot learning: achieving human-competitive sample efficiency with machine learning (considering that humans are often able to learn to perform a new task given a very small number of examples). Crucially, real-world scenarios requiring CD-FSL are far more common than those requiring methods for learning new classes in the same domain.

A standard approach to FSL is to train a "shallow" (e.g., linear) classifier on features provided by a convolutional network. Often, these methods employ a meta-learning strategy involving episodic training (Hospedales et al. 2021, Vinyals et al. 2016) to yield a feature extractor providing features that generalise well. In this context, it is common to propose new classification rules, such as the nearest centroid classifiers in prototypical networks (Snell et al. 2017) or the linear support vector machines (SVMs) in MetaOptNet (Lee et al. 2019), and employ them in an episodic training framework.

Our work is based on the observation by Guo et al. (2020) that in a single-source-domain CD-FSL setting, a simple transfer learning approach that pretrains a feature extractor in the source domain and builds a linear classifier on the extracted features in the target domain can outperform meta-learning methods developed for the standard FSL setting. Guo et al. (2020) show this using a new benchmark for CD-FSL that employs miniImageNet (a subset of the full ImageNet dataset) as source domain and a diverse set of other datasets as target domains.

With the work presented in this paper, we aim to provide an extensive evaluation of widely-used "shallow" classifiers applied to features extracted by a pretrained network in CD-FSL.[2] We employ the benchmark developed by Guo et al. (2020) to enable a comparison to the results presented in their paper. In our experiments, we additionally evaluate the effect of various data normalisation strategies and also consider a set of larger feature extraction networks trained on the full ImageNet dataset. As the data extracted by convolutional networks is high-dimensional, we include classifiers designed for processing such data, e.g., microarray data, in our experiments. More specifically, we apply random

[2] Our code and data are available at https://zenodo.org/record/5152448

projection ensembles of LDA models (McLachlan 1992) and the nearest shrunken centroid classifier (Tibshirani et al. 2003). In addition, we experimentally compare the effect of using feature vectors extracted from various stages of a wide range of ResNet models (He et al. 2015). Lastly, we convert each instance in CD-FSL into a bag of weakly augmented samples, transforming the original mono-instance task into a multi-instance one, and evaluate multi-instance learning methods on the transformed task.

Our main conclusions are that (i) variants of linear discriminant analysis can be effectively applied in CD-FSL, (ii) feature vector normalisation can help to increase predictive performance, (iii) standard regularised logistic regression is competitive with the cosine similarity classifier used by Chen et al. (2019), (iv) the penultimate stage of feature extractors can produce feature vectors that lead to better CD-FSL accuracy than those obtained from the final stage, and (v) simple multi-instance learning methods can be profitably applied by converting instances into weakly augmented bags of instances.

This paper extends our earlier conference publication available as (Wang et al. 2020). Extensions to the conference publication include evaluation of a greater range of ResNet models, comparison between different ResNet stages, and application of multi-instance learning techniques to CD-FSL tasks.

2. Related Work

Most recently developed approaches to few-shot learning make use of episodic training (Vinyals et al. 2016, Snell et al. 2017, Lee et al. 2019, Sung et al. 2018, Finn et al. 2017), where the objective function used to train the feature extractor is dynamically generated for each minibatch. The loss is dynamic in the sense that it depends on classifier weights that are generated after performing the forward propagation, via solving a second optimisation problem or using a learning algorithm with a closed form solution. A different few-shot learning problem is synthesised for each episode by selecting a random subset of classes from the training dataset, with the goal of training a network that will produce class-agnostic features. Due to this idea of nested learning, these techniques can be seen as a form of meta-learning. In practice, it has been shown that such episodic training methods learn features that may perform poorly compared to simple baselines when evaluated in a cross-domain setting (Guo et al. 2020, Chen et al. 2019).[3]

Prototypical networks (Snell et al. 2017) are the most popular example of episodic training in few-shot learning. In each episode of training, a nearest centroid classifier is constructed, and the cross-entropy loss of this classifier is used as a training signal for the feature extractor. It is instructive to relate this to the classification methods we consider in this paper. Nearest centroid classifiers can be seen as a simplification of linear discriminant analysis (LDA) (Hastie et al. 2009). While nearest centroid classifiers commonly use Euclidean distance, LDA makes use of Mahalanobis distance—a metric defined via the covariance matrix of the training data. The quadratic discriminant analysis (QDA) extension constructs a per-class covariance matrix, rather than a global pooled covariance matrix. We also consider the shrunken nearest centroid classifier (Tibshirani et al. 2003), a variant of LDA designed for high-dimensional datasets with only a few samples—a defining

[3] Note that in this paper, we consider a few-shot scenario where features are obtained from a single source domain. The meta-learning approach has been used to successfully combine information from multiple source domains [Triantafillou et al. (2020), Dvornik et al. (2020), Liu et al. (2021), Triantafillou et al. (2021)].

characteristic of few-shot learning. Finally, addressing high-dimensional problems with LDA, it is common to construct a committee of classifiers, where each member of the ensemble first randomly projects the high-dimensional features into a lower-dimensional space (Durrant and Kabán 2015). We include all these variants in our experiments.

The matching networks method for few-shot learning (Vinyals et al. 2016) uses the episodic training approach to learn an embedding function that considers multiple training instances from the few-shot learning problem simultaneously. This can be seen as a dynamic metric learning approach, to which a *k*-nearest neighbours (*k*-NN) classifier is then applied in order to make predictions. Relation networks (Sung et al. 2018) also make use of metric learning but use a Siamese network that considers pairs of instances rather than the recurrent network approach used in matching networks. Guo et al. (2020) demonstrate that these methods underperform in the cross-domain setting. In our paper, we apply a *k*-NN classifier directly to the features extracted by the pretrained network.

The MetaOptNet (Lee et al. 2019) method makes use of the implicit function theorem to embed a convex optimiser into the forward propagation of a convolutional network. This enables training of a linear SVM on a randomly generated few-shot learning problem in each episode before performing a backpropagation and weight update. The loss of this SVM is then used as a training signal for the feature extractor network. We include support vector machines in our comparison.

Two recent papers have pointed out the shortcomings of episodic training approaches to few-shot learning, and demonstrated the viability of transfer learning approaches in CD-FSL. The work of Chen et al. (2019) demonstrates that using a standard pretrained network and a simple classifier that makes predictions based on the cosine similarity between the features of a novel instance and a prototype learned from each class achieves competitive performance. In particular, this is accomplished without any sophisticated meta-learning process. Guo et al. (2020) propose a new set of benchmarks for cross-domain few-shot learning—the one we extend in this paper—which is shown to be much more challenging than previously used evaluation protocols, and demonstrate that the meta-learning approaches considered in their experiments are significantly outperformed by comparatively simple baselines based on pretrained networks.

3. Methods

In multi-class learning problems, we construct a classifier, $f: X \rightarrow Y_T$, mapping from an input space, X, to an output space, Y_T, consisting of n class values, based on a training set $Z_T \subset X \times Y_T$ sampled from the (target) domain of interest. In n-way k-shot classification, which is a special case that is commonly considered in few-shot learning, we have exactly k instances for each of the n classes. Cross-domain few-shot learning problems are commonly tackled by initially training a feature extraction model on an auxiliary set of data, $Z_S \subset X \times Y_S$, that has been obtained from a related (source) domain.[4] It is generally assumed that (i) the label sets of the source and target domains are different (i.e., $Y_T \neq Y_S$), and (ii) the input data are sampled from different distributions (i.e., $p_S(x) \neq p_T(x)$, where the random variable x takes values in X). Importantly, the size of Z_S, the auxiliary set, is not limited to any particular value. A common strategy in few-shot learning approaches is to use Z_S to train a feature extractor,

[4] We assume that only a single source domain is available because this is a common scenario in practice.

g, mapping from X to an intermediate feature space, I, of lower dimensionality (e.g., 512) than the input space. Then, Z_T, the few-shot training data for the target domain, is processed by the feature extractor and the feature vectors are used to obtain a robust classifier, h, that maps from I to Y_T. The final classification model, f is given by the composition of the feature extractor and the robust classifier, $f = h \circ g$. In this paper, we focus on image classification and assume that g is a convolutional neural network trained on the source domain data (i.e., the "auxiliary" data). The process is shown in Figure 1.

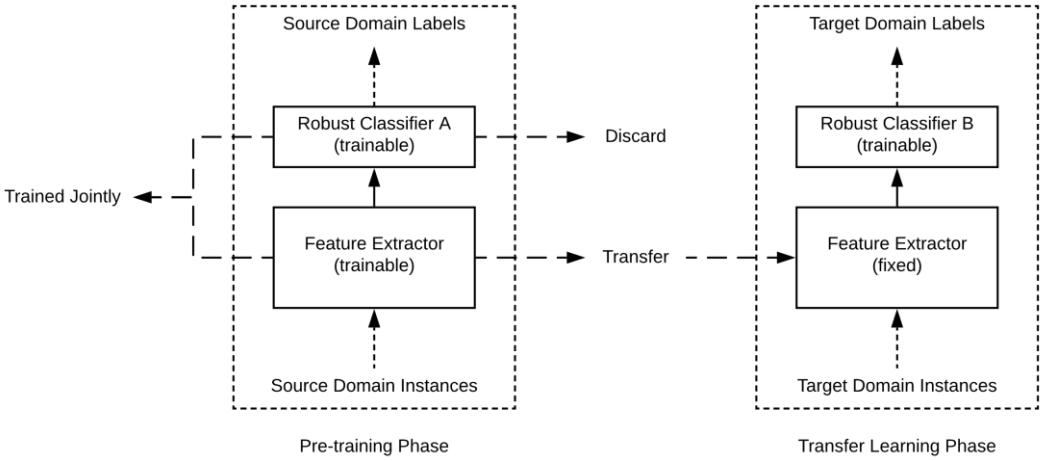

Figure 1. Cross-domain few-shot learning flow chart

3.1. Datasets

In the benchmark proposed by Guo et al. (2020), a small subset of ImageNet (Deng et al. 2009)—a large collection of images containing over 20,000 classes and, on average, over 500 instances per class—is utilised as the source domain, and there are four target domains with varying degrees of domain shift from the source domain. We adopt this benchmark for our experiments, and further add another dataset, Food-101 (Bossard et al. 2014), a highly specialised classification task, as a target domain. Additionally, we use different subsets of ImageNet as source domains. The feature extractor employed in Guo et al. (2020) utilises only a small subset of ImageNet, named miniImageNet, proposed by Vinyals et al. (2016), which is commonly used in few-shot learning research. MiniImageNet contains 100 classes in total, including 64 classes for training, 16 classes for validation, and 20 classes for testing; each class contains 600 instances. In Guo et al. (2020), the 64 training classes are used to train a ResNet-10 feature extractor. To extend the evaluation, we also investigate the performance of a set of larger ResNet models (more detail is provided in Section 3.4) pretrained on the much larger subset of ImageNet used in the 2012 Large Scale Visual Recognition Challenge (Russakovsky et al. 2015). This dataset contains over 1.2 million images spread across 1,000 classes.

We use all the target domains proposed by Guo et al. (2020) in our experiments:

- CropDisease (Mohanty et al. 2016) contains 54,306 instances composed of images of healthy and diseased plants. There are 38 classes of 14 plant species; each specimen is either healthy or infected with one of 26 diseases.

- EuroSAT (Helber et al. 2019) is a 10-class dataset of satellite images with 27,000 instances.
- ISIC (Tschandl et al. 2018, Codella et al. 2019) is a skin disease dataset with 10,015 instances in seven different classes.
- ChestX (Wang et al. 2017) is a frontal-view X-ray dataset containing 108,948 instances and eight different disease labels; each instance can have multiple labels. After the filtering process implemented by Guo et al. (2020), the dataset consists of 25,848 instances, each belonging to one of seven mutually exclusive classes.

The motivation for including these datasets in the CD-FSL benchmark in the above order is to introduce target domains that are increasingly different from the ImageNet source domain (Guo et al. 2020). CropDisease has both perspective and colour, and consists of natural objects, all of which are attributes of ImageNet. EuroSAT has no perspective, in contrast to CropDisease. ISIC does not contain natural objects, as opposed to EuroSAT and CropDisease. Lastly, ChestX does not exhibit perspective, colour, or natural objects.

To extend the benchmark, we include another dataset, Food-101 (Bossard et al. 2014), which is a food image dataset containing 101 classes and 1,000 images per class. For each class, there are 250 instances that are sanitised; the remainder may be heavily distorted or wrongly labelled. We only include the 250 sanitised instances of each class in our experiments to evaluate the performance of the algorithms and the effects of the normalisation methods, but utilise all of the 101 classes. We consider Food-101 to be at the same level as CropDisease in terms of similarity to the source domain—Food-101 also has perspective and colour, and images of (albeit man-made) objects. However, Food-101 is arguably more specialised and complex than CropDisease due to its greater number of classes and the subtle differences between food classes.

Note that there are a few similar classes between Food-101 and ImageNet. For example, Food-101 contains class “hamburger” and ImageNet contains class “cheeseburger”. Therefore, the pretrained feature extractor obtained from subsets of ImageNet may not be completely blind to certain classes in this target domain, which is not the case for the target domains EuroSAT, ISIC, or ChestX. However, ImageNet and Food-101 are assembled in different ways from different sources, and the similarity is limited to a very small number of classes. Also, ImageNet contains a large number of natural object categories, making it hard for other datasets of natural objects to exclude the ImageNet classes completely. For example, “corn” is a class in both ImageNet and CropDisease.

3.2. Robust Learning Algorithms

Few-shot learning using a pretrained feature extractor involves training a classifier on a small number of labelled instances (often between 10 and 100) complicated by the fact that the feature vectors can exhibit hundreds or thousands of dimensions. There are other machine learning applications, such as classification tasks in genomics and natural language processing tasks involving bags of words, that exhibit high-dimensional feature vectors. Learning algorithms that are suitable for these applications are thus also worth considering in the context of this paper. The full list of algorithms we evaluate in our experiments to train the “shallow” classifier h is the following:

- Logistic regression: a technique that uses linear combinations of feature values along with a logistic function to model class probabilities (le Cessie and van Houwelingen

1992). The model is obtained by optimising parameters for log likelihood on the training data, and can be regularised in various ways. We consider penalising the ℓ_1 or ℓ_2 norms of the parameters. Multi-class variants can be constructed via a problem transformation (we use 1-vs-rest), or using a softmax function instead of the logistic function to obtain a multinomial distribution over classes.

- Linear discriminant analysis: a method that also finds linear combinations of features that separate different classes in a dataset (McLachlan 1992). LDA assumes that the instances from each class are normally distributed, and the distributions of all the classes have identical covariance matrices. It can be regularised by adding a small value to the diagonal entries of the covariance matrix.
- Quadratic discriminant analysis: a method closely related to LDA; unlike LDA, QDA assumes that each class can have a different covariance matrix. This method can also be regularised by adding a small quantity to the diagonal of the covariance matrices.
- Random projection LDA ensemble: a committee of LDA models, where each member of the ensemble first reduces the dimensionality by multiplying the feature vector with a fixed random matrix. Each element in the projection matrices is randomly sampled from $N(0,1)$.
- Linear SVM: a method closely related to logistic regression (LR), but parameters are found by minimising the hinge loss instead of optimising the likelihood. In the standard formulation, which we apply, regularisation is performed by adding an ℓ_2 penalty. The 1-vs-rest method or pairwise classification is normally applied to enable multi-class classification using SVMs. We investigate both options.
- Naïve Bayes: a generative model that (naïvely) assumes conditional independence between all features (John and Langley 1995). When dealing with numeric attributes, it is common to assume class conditional normal densities, but we utilise supervised discretisation to transform continuous attribute values into discrete values (Yang and Webb 2009) to obtain higher accuracy.
- *k*-nearest neighbours: a method that classifies an unlabelled instance by referring to the class labels of its neighbours and assigning the most prominent label to the instance (Aha et al. 1991). *k*-NN gives the *k*-nearest neighbours uniform importance by default; alternatively, more weighting can be given to neighbours that are closer to the instance to be classified, e.g., by giving each neighbour a weighting of either $1 / distance$ or $1 - distance$. Our implementation utilises the $1 - distance$ weighting scheme, and the number of nearest neighbours is selected using internal leave-one-out cross-validation.
- Random forests: highly randomised ensembles of decision tree classifiers (Breiman 2001). A random forest maintains a number of different decision trees, each trained with a dataset that is sampled with replacement from the entire training set of instances. Randomness is also injected directly into the process of tree construction. The output of a random forest is the mode prediction of its decision trees, or the average of the per-class probability estimates obtained from the trees (we use the latter method); using an ensemble of trees has the effect of avoiding mistakes made by single trees due to over-fitting.
- Nearest shrunken centroid: a variant of the basic centroid classifier that has been developed for the gene expression domain (Tibshirani et al. 2003). The classifier maintains a centroid for each class of training instances, and assigns—to an unlabelled

instance—the class of its nearest centroid in scaled Euclidean space. For regularisation, the centroids are "shrunken" towards the global centroid of the data, potentially rendering all the centroids equal to the global centroid along some of the dimensions. Those dimensions become irrelevant for classification. The amount of shrinkage is determined by a hyperparameter that is optimised using internal 10-fold cross-validation.

- Cosine similarity: a method proposed for few-shot learning by Chen et al. (2019). This approach is similar to multinomial logistic regression but replaces the inner product between features and parameters with cosine similarity. It also removes the intercepts from the model.

3.3. Normalisation Methods

Feature vector normalisation is a common pre-processing step in machine learning, e.g., in document classification using bags of words, and we also consider its effect in our experiments. For a vector, $\vec{x}$, containing values x_1, x_2,…, x_n, normalisation can be defined as

$$\mathrm{norm}_p(\vec{x}) = \frac{\vec{x}}{(\sum_{k=1}^{n}|x_k|^p)^{\frac{1}{p}}} \quad (1)$$

where p denotes the ℓ_p norm that is applied. Using norm_1 means dividing each element in the vector by the sum of the absolute values contained in it, norm_2 divides by the Euclidean norm, and norm_∞ divides by the maximum absolute value. Once normalisation has been applied, the vector will have length one in the norm that is used, e.g., ℓ_1 norm yields unit rectilinear length, and ℓ_2 norm yields unit Euclidean length. Note that normalising instances in this manner is different from column/feature-wise normalisation: all feature values of an instance are scaled by the same value, and this value is instance dependent.

Table 1. Architectures of the five ResNet models

stage name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
con1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				

Table 2. PyTorch ResNet validation accuracy on ImageNet

Model structure	Top-1 error	Top-5 error
resnet18	30.24	10.92
resnet34	26.70	8.58
resnet50	23.85	7.13
resnet101	22.63	6.44
resnet152	21.69	5.94

3.4. Feature Extractors and Stage Activation

In our experiments, we consider five ResNet architectures—ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152—where the number in the name of each model denotes the number of convolutional layers in the architecture, as shown in Table 1.

Note that our experiments in Section 5.1 compare the performance of robust base learners using features from a PyTorch (Paszke et al. 2019) ResNet-10 model pretrained on miniImageNet for the CD-FSL benchmark (Guo et al. 2020), and a Keras (Chollet et al. 2015) ResNet-152 model pretrained on the ILSVRC2012 subset of ImageNet. In contrast, the experiments in Section 5.5 compare feature extractors using the five networks detailed in Table 2 as available in PyTorch (Paszke et al. 2019) in the form of networks pretrained on ILSVRC2012; their accuracy on the ImageNet validation data is shown in Table 2 (Pytorch Team 2019-2021).

The standard approach to extracting features with a ResNet Model is to use its output from the global pooling layer as feature vectors. However, global pooling can be applied to any convolutional layer to produce a feature vector of moderate length. Hence, to experiment with feature extraction at different stages of the networks, we apply global pooling to the end of each ResNet stage, i.e., conv2_x, conv3_x, and conv4_x in Table 1, in addition to the original conv5_x. At these points in the networks, the feature maps are downsized, and the number of channels is doubled. The lengths of the resulting feature vectors for conv2_x to conv5_x are 64, 128, 256, 512 for ResNet-18 and ResNet-34; they are 256, 512, 1024, 2048 for ResNet-50, ResNet-101, and ResNet-152. We evaluate each stage's feature vectors as well as the concatenation of all stages' feature vectors for each ResNet model.

3.5. Multi-instance Learning

Multi-instance learning concerns the use of machine learning when a bag of instances is used to represent an example associated with a single label, e.g., multiple images (instances) of one animal (example) taken in quick succession. Multi-instance learning can be utilised to alleviate data scarcity in FSL by converting one single example into a bag of differently augmented instances using standard augmentation techniques.

Standard machine learning without bags of examples will be referred to in this section as "mono-instance learning"[5] in contrast to "multi-instance learning". A comparison between mono- and multi-instance methods during the transfer learning phase is shown in Figure 2.

[5] Also known as "single-instance learning"

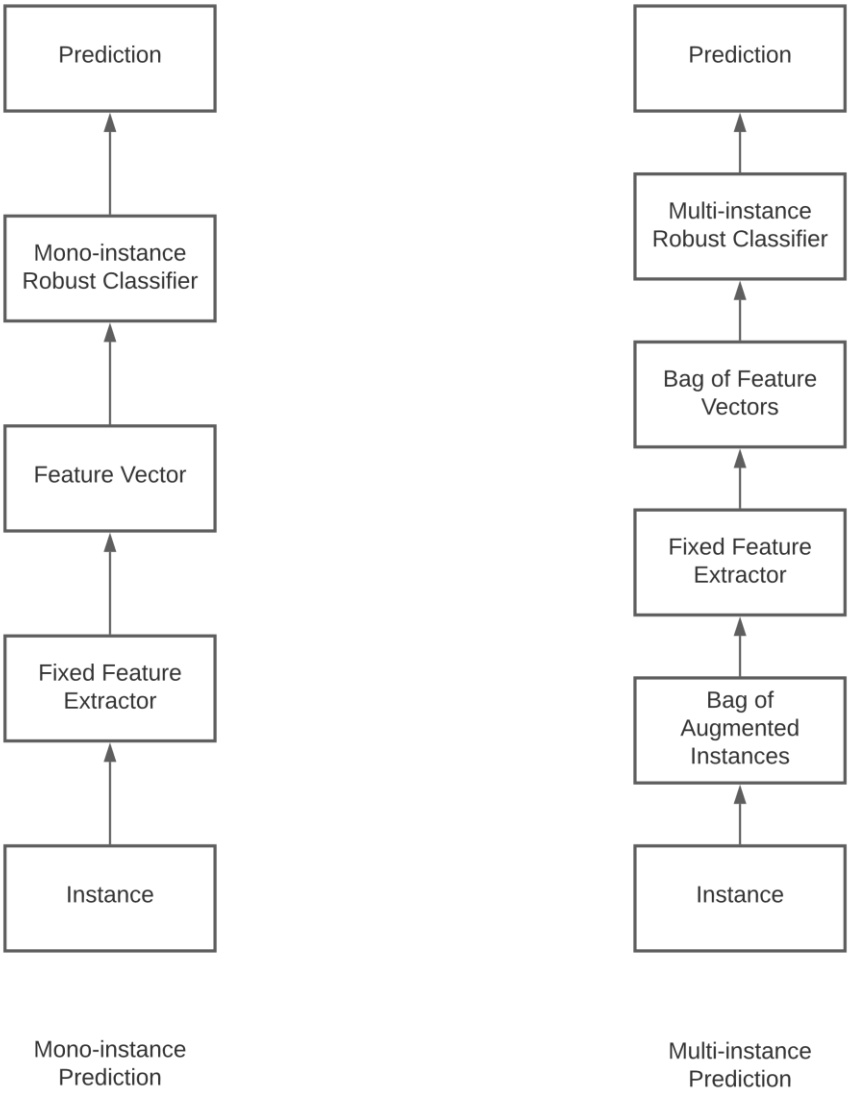

Figure 2. Mono-instance learning vs. Multi-instance learning

To transform the original mono-instance CD-FSL task into its multi-instance version, we turn each image into a 10-crop bag: the image is resized to 256×256 pixels, a 224×224 crop is taken from each of its corners as well as its centre, producing five crops, and the same process is applied to the horizontally flipped version of the image. A feature extractor is applied to each of the crops independently to create a bag of 10 feature vectors. The following multi-instance learning algorithms available in the WEKA software (Hall et al. 2009, Frank et al. 2016) are applied to each multi-instance CD-FSL task:

- SimpleMI: a bag of instances is reduced to a mono-instance by averaging across feature values; we use the cosine similarity classifier on this transformed data.
- MIWrapper: during training, a bag of instances is split up into its individual component instances that receive their bag's label; predictions are made by averaging the probability estimates obtained for individual instances in a bag—we again use the cosine similarity classifier on the transformed data.

- MILR$_{standard}$ – logistic regression is adapted to maximise loglikelihood at the bag level by combining per-instance probability estimates for a bag using the noisy-or operator.
- MILR$_{collective}$– logistic regression is adapted to maximise loglikelihood at the bag level by combining per-instance probability estimates for a bag using simple averaging.
- Miles$_{\sigma=\sqrt{80000}}$– the method presented in Chen et al. (2006) and evaluated as a generic transformation tool in Foulds and Frank (2008) is used to transform each bag into a mono-instance; the cosine similarity classifier is used on the transformed data.
- RELAGGS – the RELAGGS method for relation aggregation (Krogel and Wrobel 2003) is used to convert each bag into a mono-instance; again, the cosine similarity classifier is applied on the transformed data.

4. Results and Discussion

As discussed in Section 3.4, our first set of results is based on two pretrained feature extractors: a ResNet-10 network trained on miniImageNet based on the code made available by Guo et al. (2020), and the ResNet-152 model trained on the ILSVRC2012 subset of ImageNet available in Keras (Chollet et al. 2015). Both models receive as input square RGB images, with a side length of 84 pixels for ResNet-10 and a side length of 224 pixels for ResNet-152. The smaller network (ResNet-10) produces 512-dimensional feature vectors; the larger one yields vectors with 2,048 components.

For each target domain dataset, 600 different 5-way k-shot learning tasks are created for each value of $k \in \{5,20,50\}$. Each of the 600 tasks is set up to contain 15 test instances per class, regardless of the number of training instances. After feature extraction, shallow learners are trained on the data using the WEKA software (Version 3.9.5) (Frank et al. 2016).

In further experiments, the two pretrained networks are replaced with the five pretrained PyTorch ResNet models discussed in Section 3.4, and features are extracted from different stages of each ResNet model to provide a systematic comparison of different feature extractors and layer stages.

Lastly, each of the 600×5 CD-FSL tasks is converted to a multi-instance learning task by using 10-crop bags of instances in place of the original images and applying a multi-instance learning algorithm. The accuracy of multi-instance learning is compared with a centre-crop mono-instance baseline.

4.1. Performance of Classifiers for Few-Shot Learning

The first set of experiments is designed to determine which of the robust base classifiers yield the most accurate classifications in few-shot learning problems. To this end, all the above classifiers are evaluated on each of the target domains, using un-normalised feature vectors extracted by both networks, but considering only 5-shot problems. The results for ResNet-10 features are given in Table 3 and visualised in Figure 3; the results corresponding to ResNet-152 features are provided in Table 4 and visualised in Figure 5. Figures 4 and 6 provide some information on the statistical significance of the performance differences between classifiers, aggregated across all datasets, as determined by the Wilcoxon-Holm test (Wilcoxon 1945), where classifiers with no statistically significant difference in accuracy are grouped by a thick horizontal line. The figures, named critical difference diagrams, are

generated using code provided by Ismail Fawaz et al. (2019)[6]. The critical difference diagram was originally proposed by Demšar (2006).

The results show that ℓ_1-regularised 1-vs-rest logistic regression, naïve Bayes, *k*-NN, and random forests are outperformed by the other learning algorithms. Additional experiments (not included here) indicate that the poor performance of ℓ_1-regularised logistic regression, relative to other linear models, is primarily due to the difficulty of determining a value for the regularisation parameter that provides consistent performance. It is worth noting here that hyperparameter optimisation using internal cross-validation was not used for the results shown in this paper because it produced lower accuracy than the default hyperparameter settings in WEKA. This is most probably because the training folds available for internal cross-validation in few-shot episodes are very small, yielding high variance in the cross-validation-based estimates of performance.

The results also show that variants of LDA are competitive with other linear classifiers. This suggests that developing new forms of LDA that are designed specifically for few-shot learning may be a promising avenue for research. We can also see that ℓ_2-regularised logistic regression and the cosine similarity classifier, which has been shown to be competitive with the state-of-the-art in (Chen et al. 2019), perform similarly. Considering ℓ_2-regularised logistic regression is one of the most established ways to implement transfer learning with deep neural networks, one may ask whether actual progress has been achieved by developing variants of this old method.

Our results also show that predictive performance exhibits a consistent decrease as the domain shift increases, confirming the observations of Guo et al. (2020). Among the four datasets of Guo et al.'s original benchmark, CropDisease yields the highest estimated accuracy for all algorithms, followed, in order, by EuroSAT, ISIC and ChestX. No method improves in a meaningful manner on random guessing on the ChestX data (i.e., 20%). We can also see that performance on Food-101 is consistently between that on EuroSAT and ISIC. Thus, we can speculate that the domain shift between ImageNet and Food-101 is greater than that between ImageNet and EuroSAT, and smaller than that between ImageNet and ISIC.

[6] https://github.com/hfawaz/cd-diagram

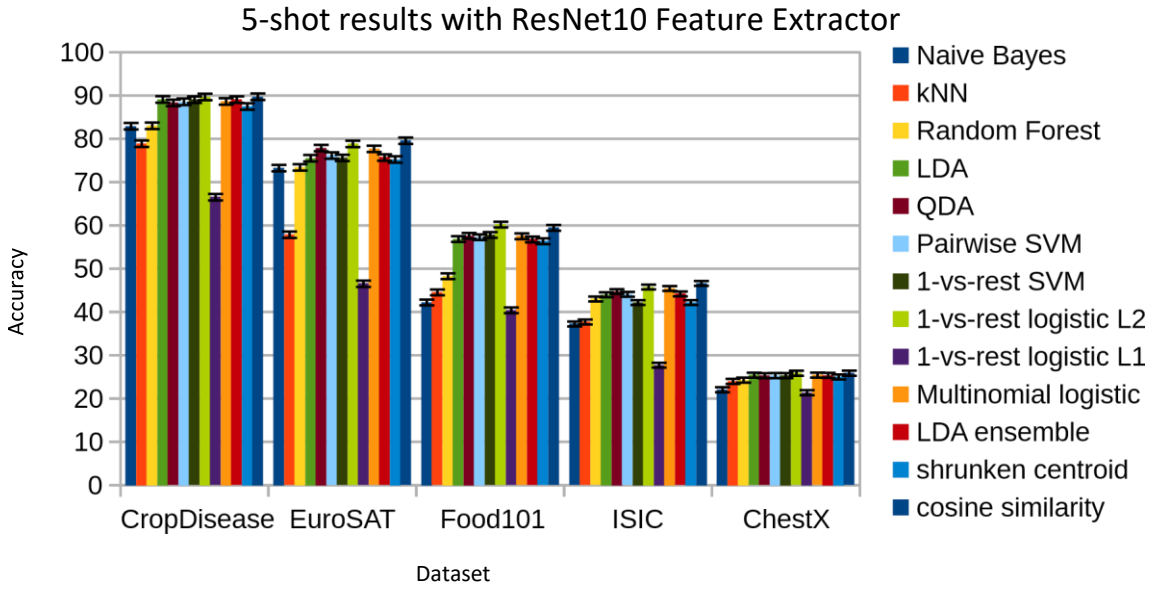

Figure 3. Visualisation of the ResNet-10 feature extractor experimental results

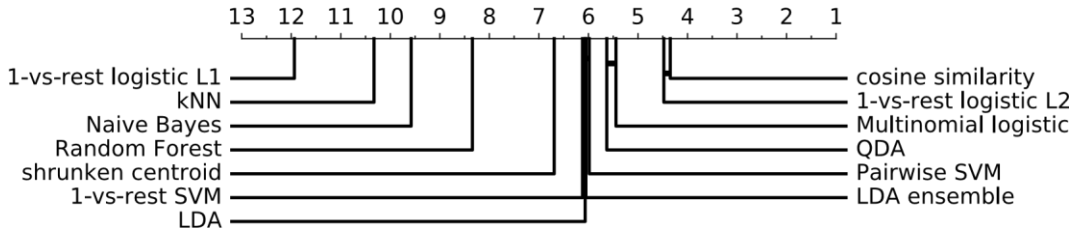

Figure 4. Critical difference diagram of the algorithms with the ResNet-10 feature extractor

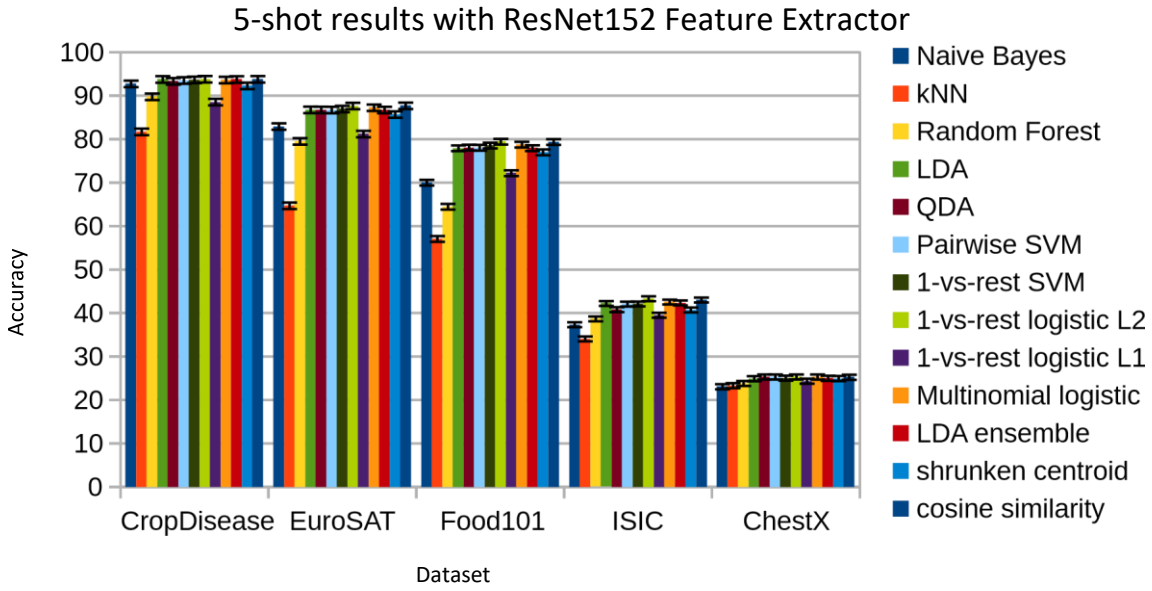

Figure 5. Visualisation of the ResNet-152 feature extractor experimental results

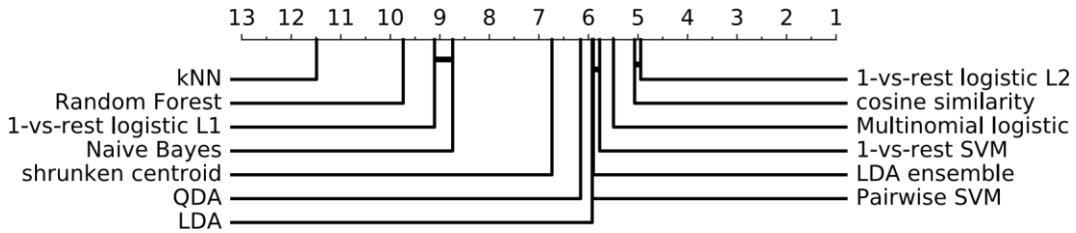

Figure 6. Critical difference diagram of the algorithms with the ResNet-152 feature extractor

Table 3. 5-shot experimental results with the ResNet-10 feature extractor

	CropDisease	EuroSAT	Food-101	ISIC	ChestX
Naïve Bayes	82.92±0.75	73.24±0.75	42.19±0.75	37.23±0.6	22.04±0.38
k-NN	78.86±0.74	57.85±0.78	44.51±0.73	37.68±0.6	23.97±0.39
Random Forest	83.00±0.66	73.40±0.68	48.27±0.70	43.00±0.54	24.26±0.43
LDA	89.09±0.56	75.52±0.65	56.85±0.77	43.97±0.59	25.40±0.41
QDA	88.30±0.58	77.86±0.61	57.58±0.77	44.70±0.57	25.27±0.41
Pairwise SVM	88.55±0.59	76.14±0.67	57.28±0.76	44.08±0.58	25.30±0.41
1-vs-rest SVM	89.03±0.58	75.60±0.69	57.80±0.77	42.17±0.59	25.29±0.41
1-vs-rest LR ℓ_2	89.66±0.56	78.82±0.61	**60.19±0.78**	45.76±0.58	25.85±0.42
1-vs-rest LR ℓ_1	66.51±0.88	46.52±0.81	40.38±0.68	27.70±0.48	21.35±0.37
Multinomial LR	88.64±0.58	77.68±0.63	57.50±0.77	45.42±0.59	25.41±0.42
LDA ensemble	89.05±0.56	75.68±0.64	56.76±0.77	44.17±0.59	25.36±0.43
Shrunken centroid	87.46±0.62	75.23±0.67	56.36±0.77	42.17±0.59	25.00±0.44
Cosine similarity	**89.71±0.55**	**79.56±0.60**	59.44±0.78	**46.60±0.59**	**25.86±0.42**

Table 4. 5-shot experimental results with the ResNet-152 feature extractor

	CropDisease	EuroSAT	Food-101	ISIC	ChestX
Naïve Bayes	92.71±0.51	82.90±0.51	69.99±0.87	37.31±0.52	23.04±0.39
k-NN	81.67±0.72	64.68±0.72	57.05±0.88	34.04±0.53	23.29±0.37
Random Forest	89.75±0.57	79.48±0.56	64.45±0.78	38.64±0.53	23.80±0.40
LDA	93.74±0.46	86.74±0.49	77.90±0.71	42.18±0.56	24.89±0.41
QDA	93.29±0.48	86.72±0.50	78.05±0.72	40.77±0.55	25.29±0.42
Pairwise SVM	93.50±0.47	86.70±0.50	78.02±0.72	42.04±0.57	25.29±0.43
1-vs-rest SVM	93.63±0.47	86.93±0.49	78.53±0.70	42.08±0.57	25.02±0.43
1-vs-rest logistic ℓ_2	**93.79±0.46**	87.62±0.47	**79.42±0.69**	**43.28±0.56**	**25.30±0.42**
1-vs-rest logistic ℓ_1	88.53±0.60	81.18±0.56	72.18±0.77	39.49±0.57	24.32±0.41
Multinomial logistic	93.59±0.47	87.22±0.48	78.74±0.70	42.52±0.56	25.27±0.44
LDA ensemble	93.72±0.47	86.70±0.49	77.93±0.71	42.32±0.56	24.97±0.41
Shrunken centroid	92.26±0.52	85.65±0.50	76.97±0.74	40.70±0.56	24.92±0.43
Cosine similarity	93.77±0.47	**87.67±0.47**	79.34±0.69	42.98±0.56	25.22±0.42

On the three datasets that are relatively similar to ImageNet, i.e., CropDisease, EuroSAT, and Food-101, the more complex ResNet-152 feature extractor, obtained by training on the full ImageNet dataset, yields a large increase in classification accuracy for all classifiers when compared to the results obtained using the ResNet-10 model However, its use decreases accuracy on the two datasets that are more different from ImageNet, i.e., ISIC and ChestX. We provide further discussion of the relationship between feature extractors and task domains in Section 5.4.

Table 5. Accuracy of the algorithms on ℓ_p-normalised Food-101-ResNet-152 feature vectors

	None	ℓ_1	ℓ_2	ℓ_3	ℓ_∞
Naïve Bayes	69.99±0.87	70.54±0.87	**70.81±0.85**	70.32±0.86	67.91±0.87
k-NN	57.05±0.88	**64.06±0.85**	62.80±0.85	59.10±0.88	51.98±0.85
Random Forest	64.45±0.78	**65.19±0.78**	64.74±0.80	64.27±0.81	63.36±0.77
LDA	77.90±0.71	77.76±0.70	**79.06±0.69**	78.74±0.71	74.57±0.76
QDA	78.05±0.72	77.70±0.71	**78.95±0.71**	78.53±0.71	75.27±0.75
Pairwise SVM	78.02±0.72	74.00±0.77	74.76±0.79	**78.84±0.72**	74.63±0.77
1-vs-rest SVM	78.53±0.70	70.33±0.87	77.00±0.74	**79.00±0.71**	76.16±0.74
1-vs-rest LR ℓ_2	79.42±0.69	77.68±0.71	79.36±0.69	**79.89±0.68**	79.57±0.69
1-vs-rest LR ℓ_1	**72.18±0.77**	20.00±0.00	20.17±0.14	34.94±0.86	57.83±0.91
Multinomial LR	**78.74±0.70**	72.91±0.70	78.68±0.71	**78.74±0.71**	75.83±0.74
LDA ensemble	77.93±0.71	77.58±0.69	**78.85±0.69**	78.54±0.70	74.75±0.75
Shrunken centroid	76.97±0.74	77.46±0.75	**78.32±0.73**	77.83±0.74	73.59±0.81
Cosine similarity	79.34±0.69	79.35±0.69	79.34±0.69	79.34±0.69	79.35±0.69

4.2. Feature Vector Normalisation

Next, we look at how normalising feature vectors affects accuracy in 5-shot learning. Table 5 presents accuracy on Food-101 using ResNet-152 feature vectors that have been normalised. The best result for each learning algorithm is shown in bold.

The results show that 1-vs-rest logistic regression with ℓ_1 regularisation, using the default hyperparameter setting in WEKA (i.e., $cost = 1$ in LIBLINEAR), yields low accuracy with ℓ_p-normalisation, obtaining scores between 20% and 57.83%. We note that using a larger value (e.g., $cost = 10^{10}$) improves accuracy due to a closer fit to the training data, yielding 77.4% with ℓ_1-normalised data. However, this decreases performance to 66.57% when not normalising the feature vectors.

The cosine similarity classifier is not affected at all by normalisation because ℓ_2-normalisation is performed as part of the algorithm anyway: classification is based on the direction of the feature vector. Hence, this classifier obtains the same accuracy regardless of whether normalisation is applied, and regardless of the value of p.

The main finding from Table 5 is that LDA with ℓ_2 normalisation yields highly competitive accuracy on Food-101 when using the ResNet-152 feature vectors. Further experimental results, shown in Table 6, demonstrate that LDA gains a performance increase from ℓ_2 normalisation on all the datasets. Hence, it appears that normalisation methods can provide algorithms such as LDA with a consistent increase in CD-FSL performance.

Table 6. Comparison between no normalisation and ℓ_2 normalisation for LDA on all the datasets

	CropDisease	EuroSAT	Food-101	ISIC	ChestX
None	93.74±0.46	86.74±0.49	77.90±0.71	42.18±0.56	24.89±0.41
$l_2$normalisation	**93.95±0.45**	**87.30±0.48**	**79.06±0.69**	**42.48±0.56**	**25.11±0.40**

Table 7. Comparison of different number of shots for 1-vs-rest logistic regression with ℓ_2 regularisation.

	CropDisease	EuroSAT	Food-101	ISIC	ChestX
1-shot	77.62±0.85	68.87±0.81	57.74±0.91	29.24±0.51	22.67±0.40
2-shot	87.06±0.66	79.39±0.64	68.68±0.84	34.37±0.57	23.72±0.42
3-shot	90.85±0.55	83.50±0.59	74.16±0.76	37.98±0.56	24.12±0.41
5-shot	93.79±0.46	87.62±0.47	79.42±0.69	43.28±0.56	25.30±0.42
20-shot	98.13±0.21	93.88±0.29	87.76±0.48	56.83±0.58	29.58±0.43
50-shot	98.95±0.15	95.84±0.23	90.68±0.41	64.16±0.54	33.12±0.47

4.3. Sample Efficiency

The performance of 1-vs-rest logistic regression with ℓ_2 regularisation in a 5-way classification problem given different numbers of training instances is shown in Table 7.

Unsurprisingly, for all the datasets, the accuracy increases as the number of shots increases. For the three datasets most similar to ImageNet, i.e., CropDisease, EuroSAT, and Food-101, the variance in the accuracy estimates across the 600 train-test experiments per sample size also decreases. However, for ISIC and ChestX, no such trend is observed as the number of shots increases. This, along with the results obtained when changing the feature extractor, perhaps reflects the fact that the images of the five target domains are not just increasingly different in their superficial similarity to the images of the source domain, i.e., ImageNet, but that the domains can be divided into two groups that are conceptually different with respect to the source domain. Indeed, classification problems in CropDisease, EuroSAT, and Food-101 are those of objects, while classification problems in ISIC and ChestX are those of diseases, which are conceptually more abstract, and ImageNet is a dataset of objects.

4.4. Few-Shot Fine-Grained Classification

Inspection of the results obtained so far show high accuracy on the CropDisease data. It is worth noting that this dataset contains *both* plant species and plant diseases. To illustrate the effect of this, Table 8 shows the relationship between classification performance and the number of plant species that occur in the training and test data across the 600 runs for the 5-way 5-shot experiment. These results are obtained using 1-vs-rest logistic regression with ℓ_2 regularisation on ResNet-152 feature vectors that have not been normalised. When interpreting these results, it is important to consider that less disease classification is involved when more plant species are present. For example, a run with five plant species effectively becomes a pure plant classification task; intuitively, this is a problem that is easier to solve. The results in Table 8 clearly demonstrate that accuracy is indeed positively correlated with the number of plant species that are present in a run.

This observation provides motivation to consider a more challenging fine-grained classification task involving plant diseases only and evaluate the two feature extractors on this task. Hence, we propose the TomatoDisease dataset, obtained from CropDisease by extracting all instances exhibiting tomato diseases. This subset does not involve plant species, comprising tomatoes only, yielding a harder classification problem. Indeed, Table 9 shows that accuracy is lower when evaluating on TomatoDisease instead of CropDisease.

Table 8. Classification accuracy increases as the number of plant classes increases in the 5-way 5-shot CropDisease iterations. Summarised from the 600 iterations of 1-vs-rest logistic regression with ℓ_2 regularisation on the un-normalised ResNet-152 feature vectors.

	mean	min	median	max
2 species and 3 in-species diseases (8 iter)	83.50	76.00	81.33	96.00
3 species and 2 in-species diseases (89 iter)	88.13	60.00	89.33	**100.0**
4 species and 1 in-species disease (274 iter)	93.41	76.00	94.67	**100.0**
5 species and 0 in-species disease (229 iter)	**96.43**	**85.33**	**97.33**	**100.0**

Table 9. TomatoDisease leads to lower classification accuracy than CropDisease for all the algorithms.

	ResNet-10		ResNet-152	
	CropDisease	TomatoDisease	CropDisease	TomatoDisease
Naïve Bayes	82.92±0.75	57.87±0.79	92.71±0.51	68.35±0.66
k-NN	78.86±0.74	59.04±0.66	81.67±0.72	56.12±0.70
Random Forest	83.00±0.66	62.10±0.65	89.75±0.57	64.63±0.62
LDA	89.09±0.56	**71.54±0.63**	93.74±0.46	**74.69±0.63**
QDA	88.30±0.58	70.95±0.62	93.29±0.48	73.32±0.62
Pairwise SVM	88.55±0.59	71.10±0.64	93.50±0.47	73.95±0.62
1-vs-rest SVM	89.03±0.58	71.07±0.61	93.63±0.47	74.32±0.63
1-vs-rest LR l^2	89.66±0.56	71.31±0.63	**93.79±0.46**	74.60±0.63
1-vs-rest LR l^1	66.51±0.88	47.89±0.70	88.53±0.60	65.96±0.65
Multinomial LR	88.64±0.58	70.61±0.62	93.59±0.47	73.70±0.62
LDA ensemble	89.05±0.56	71.51±0.62	93.72±0.47	74.58±0.63
Shrunken centroid	87.46±0.62	68.31±0.68	92.26±0.52	71.36±0.65
Cosine similarity	**89.71±0.55**	71.36±0.62	93.77±0.47	74.17±0.61

All the data in TomatoDisease is from the CropDisease dataset, but significantly worse performance is obtained when evaluating classifiers on the former, at least when using a feature extractor trained on ImageNet. Consequently, one may perhaps argue that domain shift between datasets is not restricted to superficial properties of the instances (e.g., colour, perspective, and objects present): the nature of the task also matters. When two tasks differ in nature, adequate classification performance can be hard to obtain even if superficial properties of their datasets are similar—for state-of-the-art image classification systems that rely on the transfer of empirically derived information.

4.5. Feature Extractors and Stage Activation

Table 10 shows the CD-FSL results obtained using feature vectors extracted from different stages of each PyTorch ResNet model, which are used to train a cosine similarity classifier. In the table, the first four major rows correspond to feature vectors extracted from individual stages, and the last major row corresponds to feature vectors produced by concatenating the feature vectors extracted from the four stages. The length of a concatenated feature vector is 960 for ResNet-18 and ResNet 34, and 3840 for ResNet-50, ResNet-101, and ResNet-152. (Please refer to Section 3.4 for the length of each component feature vector.)

Table 10. CD-FSL accuracy with different ResNet models and stages

Stage	ResNet	ChestX	CropDisease	EuroSAT	Food-101	ISIC
Conv2	18	22.60±0.41	83.43±0.64	76.11±0.73	38.21±0.61	37.30±0.57
	34	22.35±0.38	81.04±0.68	73.82±0.77	36.49±0.60	35.81±0.55
	50	23.28±0.40	82.36±0.66	75.63±0.73	38.30±0.62	37.46±0.56
	101	22.85±0.40	80.22±0.69	73.61±0.73	36.95±0.61	36.54±0.55
	152	22.82±0.38	80.11±0.69	73.44±0.76	36.80±0.61	36.32±0.55
Conv3	18	23.84±0.39	89.57±0.52	80.94±0.70	48.09±0.70	37.59±0.56
	34	23.91±0.42	91.41±0.47	82.80±0.66	49.95±0.72	38.93±0.55
	50	23.79±0.41	89.69±0.52	81.82±0.67	46.12±0.70	40.63±0.57
	101	23.37±0.40	88.02±0.56	80.97±0.68	42.89±0.65	39.11±0.56
	152	23.98±0.40	89.54±0.52	82.02±0.67	44.74±0.67	39.41±0.55
Conv4	18	25.20±0.41	94.07±0.42	87.86±0.52	61.32±0.79	41.24±0.52
	34	25.21±0.41	95.45±0.37	89.79±0.43	66.10±0.76	42.93±0.53
	50	25.78±0.43	95.48±0.37	90.00±0.44	65.88±0.78	43.86±0.56
	101	**25.98±0.42**	**95.55±0.36**	**90.69±0.41**	69.93±0.73	**45.33±0.56**
	152	25.51±0.42	95.47±0.38	90.24±0.41	70.92±0.73	45.15±0.56
Conv5	18	25.13±0.41	93.34±0.45	86.17±0.48	73.54±0.74	43.53±0.56
	34	25.08±0.44	95.45±0.36	90.05±0.45	66.22±0.77	42.58±0.51
	50	25.05±0.43	92.99±0.49	86.83±0.50	77.39±0.72	42.56±0.57
	101	24.63±0.40	92.13±0.49	86.57±0.50	78.57±0.71	42.70±0.56
	152	25.22±0.43	92.51±0.48	85.54±0.54	**80.01±0.70**	42.34±0.57
Stage All	18	25.28±0.41	93.41±0.44	86.20±0.49	73.74±0.74	43.64±0.54
	34	24.58±0.42	92.07±0.50	86.45±0.50	76.22±0.72	41.51±0.56
	50	25.06±0.43	93.09±0.48	86.91±0.50	77.41±0.72	42.70±0.57
	101	24.78±0.42	92.08±0.49	86.86±0.48	79.20±0.69	42.85±0.59
	152	24.86±0.43	92.34±0.48	86.11±0.50	79.99±0.67	42.14±0.55

For four of the five target domains—CropDisease, EuroSAT, ISIC, and ChestX— feature vectors extracted from the end of the Conv4 stage of the ResNet-101 model with global average pooling lead to superior performance compared with the other feature extractors and stage activations. For Food-101, the final stage of the ResNet152 feature extractor works best. Generally, in a convolutional neural network, the earlier stages detect lower-level graphical features, while the later stages detect higher-level ones. Hence, we can speculate that the optimal layer for transfer from the source domain to the target domain indicates the level of graphical detail shared by the two domains: a lower-level optimal transfer implies that the source and target domains only have simple and low-level graphical features in common, while a higher-level optimal transfer implies that they share many sophisticated and high-level features, and may even resemble each other to a degree. In the same vein, the smaller variants of a model architecture can be considered to facilitate a lower-level transfer, while the bigger variants facilitate a higher-level transfer.

A noteworthy observation is that, compared with the other four target domains, we achieve optimal transfer with a bigger model and higher-level feature vectors on Food-101. The original benchmark (Guo et al. 2020) used a basic ResNet-10 feature extractor pretrained on miniImageNet, and resized all images to 84 pixels by 84 pixels prior to feature extraction. We consider bigger ResNet models that take as input images of size 224 by 224 pixels. It is possible that Food-101's level of resemblance to ImageNet is different from those of the other four datasets, with the original four datasets more suited for a smaller feature extractor and lower image resolutions. However, it is difficult to pin down which factors lead to a certain level of resemblance between two domains, as it can be determined by a mixture of image size, photographic perspective, object semantics, etc., and systematically gauging the contribution of each factor may require extensive research.

Table 11. Multi-instance learning on bags of feature vectors from 10 crops (Resnet152)

	ChestX	CropDisease	EuroSAT	Food-101	ISIC
Centre-Crop CS	25.29±0.44	92.32±0.47	86.00±0.50	**81.26±0.64**	42.98±0.57
SimpleMI	25.81±0.42	93.94±0.41	87.26±0.51	80.56±0.66	**44.07±0.50**
MIWrapper	**26.27±0.45**	**94.16±0.38**	84.42±0.55	79.87±0.62	43.87±0.49
MILR$_{standard}$	22.95±0.39	77.68±0.77	69.86±0.71	59.66±0.78	34.13±0.50
MILR$_{collective}$	24.80±0.42	86.91±0.62	79.12±0.66	70.46±0.77	40.60±0.62
Miles$_{\sigma=\sqrt{80000}}$	24.64±0.43	88.31±0.62	80.55±0.64	78.26±0.69	38.93±0.56
RELAGGS	25.82±0.43	94.00±0.40	**87.28±0.51**	80.61±0.66	43.80±0.52

4.6. Multi-instance Learning

The results of multi-instance learning are shown in Table 11, with Resnet152 and a cosine similarity classifier. In comparison to a mono-instance centre-crop baseline, MIWrapper achieves the best result for ChestX and CropDisease, RELAGGS achieves the best result for EuroSAT, and SimpleMI achieves the best results for ISIC. The baseline achieves the best result for Food-101.

Similar to the stage activation experiment, Food-101 appears to be the outlier of the five datasets in the multi-instance learning experiment as well, as it is the only dataset not benefiting from any of the multi-instance learning methods. Moreover, the other four datasets, when converted to 10-crop bags, benefit the most from relatively simple multi-instance learning methods that apply basic forms of aggregation, i.e., SimpleMI, MIWrapper, and RELAGGS. It remains to be seen whether the more sophisticated methods, such as MILR and Miles, benefit from bigger bags of instances obtained with stronger image augmentation.

4.7. Future Research Directions

Our findings pose several research questions:

(1) Can we use the top-performing robust classifiers in CD-FSL to improve semi-supervised learning approaches, where the amount of labelled data is also limited?
(2) How can we methodically structure and train a feature extractor to achieve the best possible transfer between source and target domains?
(3) Is there a way to systematically quantify domain shift and gauge different factors' contribution to domain shift?
(4) How should one assemble a dataset from available data that exhibits minimal domain shift with respect to a given real-world learning task?
(5) How to better utilise multi-instance learning and data augmentation to alleviate the effect of data scarcity in CD-FSL?

5. Conclusion

We study cross-domain few-shot learning by evaluating and comparing robust learning algorithms, normalisation methods, and pretrained feature extractors. Our results may provide guidance on which combinations of robust classifier, feature extractor, and normalisation method are worth considering in practical CD-FSL problems. In particular, the cosine similarity classifier and 1-vs-rest logistic regression with ℓ_2 regularisation consistently perform well in our results, indicating standard ℓ_2-regularised logistic regression is a viable alternative to the cosine similarity classifier in CD-FSL. Our results also show that algorithms used in other classification problems involving high-dimensional data, such as gene expression data, namely random projection ensembles of LDA classifiers and nearest shrunken centroid classifiers, are applicable in CD-FSL scenarios.

The results also show that feature vector normalisation can yield consistent increases in classification accuracy in CD-FSL; this is the case with ℓ_2-normalised feature vectors and LDA in our experiments. Additionally, our results indicate that more complex feature extractors trained on larger datasets can increase classification accuracy noticeably when the target domain is similar to the domain used for training the feature extractor. This effect is diminished and largely disappears when target and source domain differ substantially. Our study also provides a detailed comparison of different models and stages for feature extraction. It indicates that the most appropriate model complexity and stage may correlate with the similarity of source and target domain. Finally, simple multi-instance learning methods applied with weakly-augmented bags of instances are shown to improve accuracy for the majority of the tested target domains.

References

Aha DW, Kibler D, Albert MK. 1991. Instance-based learning algorithms. Machine Learning. 6:37–66.

Bossard L, Guillaumin M, Van Gool L. 2014. Food-101 – mining discriminative components with random forests. In: ECCV. p. 446–461.

Breiman L. 2001. Random forests. Machine Learning. 45(1):5–32.

Chen WY, Liu YC, Kira Z, Wang YCF, Huang JB. 2019. A closer look at few-shot classification. In: ICLR.

Chen Y, Bi J, Wang J. 2006. Miles: Multiple-instance learning via embedded instance selection. IEEE PAMI. 28(12):1931–1947.

662 Chollet F, et al. 2015. Keras; [https://keras.io].
663 Codella NCF, Rotemberg V, Tschandl P, Celebi ME, Dusza SW, Gutman D, Helba B, Kalloo A,
664 Liopyris K, Marchetti MA, et al. 2019. Skin lesion analysis toward melanoma detection
665 2018: A challenge hosted by the international skin imaging collaboration (ISIC). ArXiv.
666 abs/1902.03368.
667 Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. 2009. ImageNet: A Large-Scale Hierarchical
668 Image Database. In: CVPR. p. 248–255.
669 Durrant RJ, Kabán A. 2015. Random projections as regularizers: learning a linear discriminant
670 from fewer observations than dimensions. Machine Learning. 99(2):257–286.
671 Dvornik N, Schmid C, Mairal J. 2020. Selecting relevant features from a multi-domain
672 representation for few-shot classification. In: ECCV. Springer. p. 769–786.
673 Finn C, Abbeel P, Levine S. 2017. Model-agnostic meta-learning for fast adaptation of deep
674 networks. In: ICML. p. 1126–1135.
675 Foulds J, Frank E. 2008. Revisiting multiple-instance learning via embedded instance
676 selection. In: AJCAI. Springer. p. 300–310.
677 Frank E, Hall MA, Witten IH. 2016. The WEKA workbench; [Online appendix for "Data mining:
678 Practical machine learning tools and techniques", Morgan Kaufmann, 4th edition].
679 Guo Y, Codella NC, Karlinsky L, Codella JV, Smith JR, Saenko K, Rosing T, Feris R. 2020. A
680 broader study of cross-domain few-shot learning. In: ECCV.
681 Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. 2009. The WEKA data
682 mining software: an update. SIGKDD Explorations. 11(1):10–18.
683 Hastie T, Tibshirani R, Friedman J. 2009. The elements of statistical learning: data mining,
684 inference, and prediction. Springer Science & Business Media.
685 He K, Zhang X, Ren S, Sun J. 2015. Deep residual learning for image recognition. CVPR:770–
686 778.
687 Helber P, Bischke B, Dengel A, Borth D. 2019. Eurosat: A novel dataset and deep learning
688 benchmark for land use and land cover classification. IEEE Journal of Selected Topics in
689 Applied Earth Observations and Remote Sensing. 12(7):2217–2226.
690 Hospedales TM, Antoniou A, Micaelli P, Storkey AJ. 2021. Meta-learning in neural networks:
691 A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence.
692 Ismail Fawaz H, Forestier G, Weber J, Idoumghar L, Muller PA. 2019. Deep learning for time
693 series classification: a review. Data Mining and Knowledge Discovery. 33(4):917–963.
694 John GH, Langley P. 1995. Estimating continuous distributions in Bayesian classifiers. In: UAI;
695 San Mateo. Morgan Kaufmann. p. 338–345.
696 Krizhevsky A. 2012. Learning multiple layers of features from tiny images. University of
697 Toronto.
698 Krogel MA, Wrobel S. 2003. Facets of aggregation approaches to propositionalization. In:
699 Work-in-Progress Track at the Thirteenth International Conference on Inductive Logic
700 Programming (ILP).
701 le Cessie S, van Houwelingen JC. 1992. Ridge estimators in logistic regression. Applied
702 Statistics. 41(1):191–201.
703 Lee K, Maji S, Ravichandran A, Soatto S. 2019. Meta-learning with differentiable convex
704 optimization. In: CVPR. p. 10657–10665.
705 Liu L, Hamilton WL, Long G, Jiang J, Larochelle H. 2021. A universal representation
706 transformer layer for few-shot image classification. In: ICLR. OpenReview.net.
707 McLachlan GJ. 1992. Discriminant analysis and statistical pattern recognition. Wiley.

Mohanty SP, Hughes DP, Salathé M. 2016. Using deep learning for image-based plant disease detection. Frontiers in Plant Science. 7:1419.
Pan SJ, Yang Q. 2010. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering. 22(10):1345–1359.
Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS. p. 8024–8035.
Pytorch Team. 2019-2021. ResNet | PyTorch; [https://pytorch.org/hub/pytorch_vision_resnet/]. Accessed: 2021-05-16.
Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, et al. 2015. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision. 115(3):211–252.
Snell J, Swersky K, Zemel RS. 2017. Prototypical networks for few-shot learning. In: NIPS. p. 4077–4087.
Sung F, Yang Y, Zhang L, Xiang T, Torr PH, Hospedales TM. 2018. Learning to compare: Relation network for few-shot learning. In: CVPR. p. 1199–1208.
Tibshirani R, Hastie T, Narasimhan B, Chu G. 2003. Class prediction by nearest shrunken centroids, with applications to DNA microarrays. Statistical Science:104–117.
Triantafillou E, Larochelle H, Zemel RS, Dumoulin V. 2021. Learning a universal template for few-shot dataset generalization. In: ICML; vol. 139. PMLR. p. 10424–10433.
Triantafillou E, Zhu T, Dumoulin V, Lamblin P, Evci U, Xu K, Goroshin R, Gelada C, Swersky K, Manzagol P, et al. 2020. Meta-dataset: A dataset of datasets for learning to learn from few examples. In: ICLR. OpenReview.net.
Tschandl P, Rosendahl C, Kittler H. 2018. The HAM10000 Dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Scientific Data. 5:180161.
Vinyals O, Blundell C, Lillicrap TP, Kavukcuoglu K, Wierstra D. 2016. Matching networks for one shot learning. In: NIPS. p. 3630–3638.
Wang H, Gouk H, Frank E, Pfahringer B, Mayo M. 2020. A comparison of machine learning methods for cross-domain few-shot learning. In: AJCAI. Springer. p. 445–457.
Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM. 2017. ChestX-Ray8: Hospital-scale chest x-ray database and benchmarks on weakly supervised classification and localization of common thorax diseases. CVPR:3462– 3471.
Yang Y, Webb GI. 2009. Discretization for naive-Bayes learning: Managing discretization bias and variance. Machine Learning. 74(1):39–74.
Wilcoxon F. 1945. Individual comparisons by ranking methods. Biometrics Bulletin 1(6):80–83.
Demšar J. 2006. Statistical comparisons of classifiers over multiple data sets. Machine Learning Research 7:1–30.

Figure Captions

Figure 1. Cross-domain few-shot learning flow chart
Figure 2. Mono-instance learning vs. Multi-instance learning
Figure 3. Visualisation of the ResNet-10 feature extractor experimental results