A Comparison of Machine Learning Methods for Cross-Domain Few-Shot Learning

A Comparison of Machine Learning Methods for Cross-Domain Few-Shot Learning

Hongyu Wang[1], Henry Gouk[2], Eibe Frank[1], Bernhard Pfahringer[1], and Michael Mayo[1]

[1] Department of Computer Science, University of Waikato, Hamilton, New Zealand
{hw168@students., eibe@, bernhard@, mmayo@}waikato.ac.nz
[2] School of Informatics, University of Edinburgh, Edinburgh, United Kingdom
henry.gouk@ed.ac.uk

Abstract. We present an empirical evaluation of machine learning algorithms in cross-domain few-shot learning based on a fixed pre-trained feature extractor. Experiments were performed in five target domains (CropDisease, EuroSAT, Food101, ISIC and ChestX) and using two feature extractors: a ResNet10 model trained on a subset of ImageNet known as miniImageNet and a ResNet152 model trained on the ILSVRC 2012 subset of ImageNet. Commonly used machine learning algorithms including logistic regression, support vector machines, random forests, nearest neighbour classification, naïve Bayes, and linear and quadratic discriminant analysis were evaluated on the extracted feature vectors. We also evaluated classification accuracy when subjecting the feature vectors to normalisation using p-norms. Algorithms originally developed for the classification of gene expression data—the nearest shrunken centroid algorithm and LDA ensembles obtained with random projections—were also included in the experiments, in addition to a cosine similarity classifier that has recently proved popular in few-shot learning. The results enable us to identify algorithms, normalisation methods and pre-trained feature extractors that perform well in cross-domain few-shot learning. We show that the cosine similarity classifier and ℓ^2-regularised 1-vs-rest logistic regression are generally the best-performing algorithms. We also show that algorithms such as LDA yield consistently higher accuracy when applied to ℓ^2-normalised feature vectors. In addition, all classifiers generally perform better when extracting feature vectors using the ResNet152 model instead of the ResNet10 model.

Keywords: Cross-Domain Few-Shot Learning, Pre-trained Feature Extractors, Normalisation, Transfer Learning

1 Introduction

Convolutional neural networks have greatly changed the way in which supervised learning is used to solve image classification problems. However, they come with one major drawback: a very large volume of annotated images is generally required to train a network with good accuracy. In many situations it is not feasible

to gather datasets of sufficient size to train such models, be it due to cost or other resource limitations. In these cases, so-called few-shot learning methods can be applied. Few-shot learning (FSL) refers to the task of learning in a target domain from a very limited number of annotated instances [23]. A key component in practical few-shot learning is prior knowledge gained from a source domain that is in some way related to the target domain. The knowledge gleaned from the source domain can be utilised to compensate for the scarcity of available instances in the target domain, enabling the algorithm to construct a model that can make more accurate predictions than a model trained solely on the target domain. In this sense, few-shot learning generally can be seen as an instance of transfer learning [17].

Modern studies on FSL primarily tackle the case where there is little shift between the source and target domains [20, 23]. A common experimental protocol is to take a single dataset and allocate some classes as the source domain, and the remaining classes as the target domain. In contrast, cross-domain few-shot learning (CDFSL) refers to few-shot learning problems where the instances of the source domain and those of the target domain are obtained from strictly different origins (and are not, e.g., instances from the same dataset that belong to different sets of classes) [10]. CDFSL is important because it aims to achieve efficient learning in one field with knowledge from another. This is in line with one of the original pursuits of few-shot learning—giving machine learning human-like sample efficiency in the sense that humans can learn to perform new tasks reasonably well with only a few examples. Perhaps even more importantly, real-world problems that require CDFSL are far more common than those that require learning of new classes in the same domain.

Many existing approaches to FSL involve training a "shallow" (e.g., linear) classifier on features extracted by a convolutional network. These methods typically make use of meta-learning techniques based on episodic training [23] to obtain a feature extractor that can produce domain-general features. A common strategy is to propose new classification rules, such as the nearest centroid classifiers used by prototypical networks [20] or the linear support vector machines (SVMs) used by MetaOptNet [14], and plug them into an episodic training framework. However, interestingly, Guo et al. [10] show that in the CDFSL setting, simple transfer learning performed by pre-training a feature extractor in the source domain and building a linear classifier on the extracted features in the target domain significantly outperforms meta-learning approaches designed for the standard FSL setting. This finding is based on a new benchmark for CDFSL that makes use of miniImageNet (a subset of the ILSVRC 2012 dataset [7]) as the source domain and various other problems as the target domains.

Our paper aims to provide a comprehensive analysis of how different "shallow" classifiers perform when applied to features extracted using a pre-trained network in a CDFSL problem.[3] We make use of the experimental framework developed by Guo et al. [10] to facilitate a further comparison with transfer

[3] Our code and data are available at `https://zenodo.org/record/4047034/files/CDFSL_reproducibility.zip?download=1`

learning approaches. We also investigate the impact of various normalisation procedures and extend the experimental setting to a larger feature extraction network trained on the full ILSVRC 2012 dataset. Because the number of training instances is very small in relation to the size of the feature vectors, classifiers that were originally designed for analysing data produced in genomics experiments are also included in our comparison.

The main findings of our experiments are that (i) generalisations of linear discriminant analysis are effective for few-shot learning problems, (ii) feature vector normalisation is a useful pre-processing tool, and (iii) logistic regression performs as well as the cosine similarity classifier proposed by Chen et al. [5]—a method competitive with state-of-the-art few-shot learning approaches.

2 Learning Methods and Normalisation Schemes

Multi-class classification is the task of constructing a classifier, $f : \mathcal{X} \to \mathcal{Y}_T$, that maps from an input space, $\mathcal{X}$, to an output space, $\mathcal{Y}_T$, consisting of n class values, using a training set $Z_T \subset \mathcal{X} \times \mathcal{Y}_T$ sampled from the target domain of interest. An n-way k-shot classification problem, the standard setting in few-shot learning, has exactly k instances for each of the n classes. Cross-domain few-shot learning problems are most commonly solved by first training a feature extractor on an auxiliary set of data, $Z_S \subset \mathcal{X} \times \mathcal{Y}_S$, from a related source domain. Two defining characteristics are (i) the source and target label sets are different (i.e., $\mathcal{Y}_T \neq \mathcal{Y}_S$); and (ii) input data for the source and target domains are sampled according to different distributions (i.e., $p_S(X) \neq p_T(X)$, where X is a random variable taking values in $\mathcal{X}$). Crucially, there are no restrictions on the size of Z_S, the auxiliary set of training data. The framework employed by all recent few-shot learning approaches is to use Z_S to train a feature extractor, g, that maps from $\mathcal{X}$ to some intermediate representation in another vector space, $\mathcal{I}$. Then Z_T, the smaller training set in the target domain, is used to construct a classifier, h, that maps from $\mathcal{I}$ to $\mathcal{Y}_T$. Finally, f is obtained via their composition, $f = h \circ g$. In our paper, g is assumed to be a convolutional neural network that has been pre-trained on a source domain classification problem.

In the benchmark proposed by Guo et al. [10], ImageNet is utilised as the source domain, and the target domains are CropDisease [16], EuroSAT [11], ISIC [22] and ChestX [24]. We adopt this benchmark for our experiments, and include another dataset, Food101 [3], which acts as an additional highly-specialised classification task that can be used as a target domain.

2.1 Robust Learning Algorithms

One of the defining characteristics of few-shot learning is that one must train a classifier on a small number of instances (typically in the region of 10–100) but each instance may have hundreds or thousands of dimensions. Other machine learning application domains, such as genomics and natural language processing problems dealing with bag-of-words representations, also exhibit similar issues.

As such, we include a number of learning algorithms specifically designed for this scenario, but motivated by different applications. The full list of methods we consider for training the classifier h comprises the following algorithms:

- Logistic regression (LR) [13], using multinomial or 1-vs-rest classification
- Linear discriminant analysis (LDA) [15]
- Random projection LDA ensemble [8]
- Linear SVM [18], using pairwise or 1-vs-rest classification
- Naïve Bayes [25]
- k-nearest neighbours (kNN) [1]
- Random forests [4]
- Nearest shrunken centroid [21]
- Cosine similarity [5]

2.2 Normalisation Methods

We also consider the impact of normalising feature vectors for machine learning. For a vector, $\vec{x}$, containing values $x_1, x_2, \ldots, x_n$, normalisation is defined as

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

where p indicates the ℓ^p norm used in the normalisation process. Applying norm_1 corresponds to dividing each element in the vector by the sum of its absolute values, norm_2 divides by the Euclidean norm, and norm_∞ divides by the maximum absolute value in the vector. After normalisation, the resulting vector will have length one in whatever norm was applied in this process. Crucially, this form of scaling differs significantly from multiplying all instances in the training dataset by a single number, as is done when dividing by the variance during standardisation: the value used to scale the features is instance dependent.

3 Theoretical Analysis of Normalised Linear Classifiers

In this section, we consider the impact on Rademacher-based generalisation bounds [2] from normalising feature vectors when using a linear model, and provide a theoretical explanation on the benefit of normalisation in CDFSL. A standard Rademacher-based generalisation bound takes the form of

$$R(h) \leq \frac{1}{m} \sum_{i=1}^{m} \ell(h(\vec{x}_i), y_i) + 2\hat{\mathcal{R}}(H) + O\left(\frac{1}{\sqrt{m}}\right), \tag{2}$$

where ℓ is a 1-Lipschitz loss function, R is the risk (i.e., expected loss), H is a hypothesis class, $h \in H$ is a single hypothesis, and m is the size of the training set. The complexity of the hypothesis class can be characterised via empirical Rademacher complexity [2],

$$\hat{\mathcal{R}}(H) = \mathbb{E}_{\vec{\sigma}}\left[\sup_{h \in H} \sum_{i=1}^{m} \sigma_i h(\vec{x}_i)\right],$$

where $\vec{\sigma}$ is a vector of Rademacher distributed random variables. A commonly known [19] upper bound for the empirical Rademacher complexity of a linear function class,

$$H = \{\vec{x} \mapsto \vec{x} \cdot \vec{w} : \|\vec{w}\| \leq B\},$$

is given by

$$\hat{\mathcal{R}}(H) = \frac{XB}{\sqrt{m}},$$

where B is the upper bound for the Euclidean norm of the weight vector $\vec{w}$ and X is the smallest number for which $\|\vec{x}_i\|_2 \leq X, \forall i \in \{1, \ldots, m\}$. One can see immediately from this bound that any preprocessing step that reduces the Euclidean norm of feature vectors reduces the capacity for a fixed linear hypothesis class to overfit the training data. However, this does not take into account how a preprocessing step might affect the loss on the training dataset—the other quantity that is used to bound the expected loss in Equation 2.

Denote by S the training set $\{(\vec{x}_i, y_i)\}_{i=1}^m$. We shall assume that $X = 1$. Note that this is not a very limiting assumption, as all feature vectors in the training set can be scaled by some value, $c = \frac{1}{\max_i \|\vec{x}_i\|_2}$, to make this true, and then B can be scaled by $\frac{1}{c}$ to ensure $\hat{\mathcal{R}}(H)$ remains the same. Now we let $Z = \{(\vec{z}_i, y_i) \,|\, \vec{z}_i = \text{norm}_2(\vec{x}_i)\}_{i=1}^m$ be a normalised version of the training set. Trivially, we have that $\|\vec{x}_i\|_2 \leq \|\vec{z}_i\|_2$, with equality occurring only if $\|\vec{x}_i\|_2 = 1$.

For simplicity, let us consider the case where ℓ is the hinge loss,

$$\ell(\vec{x}, y) = [1 - y\vec{x} \cdot \vec{w}]_+.$$

We can see that for a single instance

$$\begin{aligned} \ell(\vec{x}_i, y_i) &\geq [1 - |y_i \vec{x}_i \cdot \vec{w}|]_+ \\ &\geq [1 - \|\vec{x}_i\|_2 \cdot \|\vec{w}\|_2]_+ \\ &\geq [1 - \|\vec{z}_i\|_2 \cdot \|\vec{w}\|_2]_+, \end{aligned}$$

where the first inequality is due to decreasing monotonicity, the second is from the Cauchy-Schwarz inequality, and the last is because $\|\vec{x}_i\|_2 \leq \|\vec{z}_i\|_2$. This demonstrates that the potential for $\vec{z}_i$ to have a larger magnitude than $\vec{x}_i$ can lead to a lower loss when normalisation is used. Because the hinge loss is an upper bound for the zero–one loss, this may also improve classification accuracy.

4 Empirical Comparison

Two pre-trained networks are used for extracting features from images: a ResNet10 model trained on miniImageNet using the code provided by Guo et al. [10], and the ResNet152 model trained on the 2012 ILSVRC subset of ImageNet that is available through Keras [6]. Both models receive as input 224×224 pixel RGB images. The smaller network (ResNet10) produces 512-dimensional feature vectors, whereas the larger network produces feature vectors with 2,048 components.

Table 1. 5-shot experimental results with the ResNet10 feature extractor

	CropDisease	EuroSAT	Food101	ISIC	ChestX
Naïve Bayes	82.92±0.75	73.24±0.75	42.19±0.75	37.23±0.6	22.04±0.38
kNN	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±0.6**	59.44±0.78	**46.6±0.59**	**25.86±0.42**

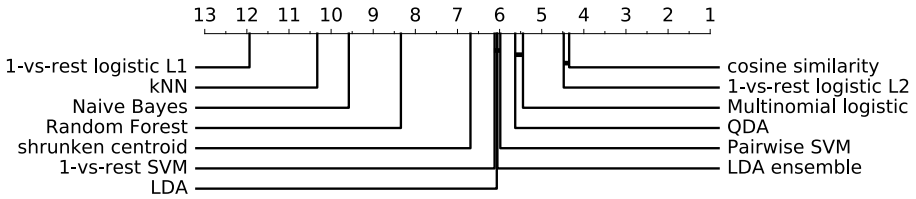

Fig. 1. Critical difference diagram of the algorithms with the ResNet10 feature extractor

For each target domain dataset, a large number of 5-way k-shot learning problems are generated: 600 different few-shot learning problems per target domain dataset for each value of $k \in \{5, 20, 50\}$. Each of these problems contains 15 test instances per class, regardless of the number of training instances. After features are extracted for each image using either of the feature extractors, the shallow learners are trained using the WEKA software (Version 3.9.5) [9].

4.1 Performance of Classifiers for Few-Shot Learning

The first experiments aim to determine which of the classifiers we consider are most useful for few-shot learning problems. For these experiments, all classifiers are applied to each of the target domains, using un-normalised feature vectors extracted by both networks. In this subsection, only 5-shot problems are considered. The results for ResNet10 features are given in Table 1, and the results corresponding to ResNet152 features are provided in Table 2. Figures 1 and 2 demonstrate statistical significance of the performance differences between clas-

Table 2. 5-shot experimental results with the ResNet152 feature extractor

	CropDisease	EuroSAT	Food101	ISIC	ChestX
Naïve Bayes	92.71±0.51	82.90±0.51	69.99±0.87	37.31±0.52	23.04±0.39
kNN	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	86.7±0.49	77.93±0.71	42.32±0.56	24.97±0.41
Shrunken centroid	92.26±0.52	85.65±0.5	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

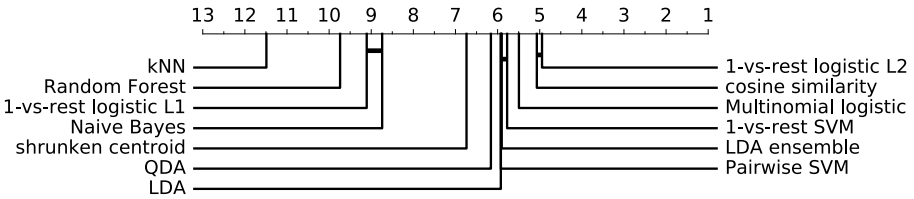

Fig. 2. Critical difference diagram of the algorithms with the ResNet152 feature extractor

sifiers, aggregated across all datasets, as determined by the Wilcoxon-Holm test (procedure described in [12], on page 22).

The first observation is that ℓ^1-regularised 1-vs-rest LR, naïve Bayes, kNN, and random forests are significantly outperformed by the other approaches. The poor performance of ℓ^1-regularised LR, compared to other linear models, is mostly due to the difficulty in finding a suitable regularisation value that leads to consistently good performance. Note that grid search was not used to tune the hyperparameters because it yielded lower accuracy than using the algorithms' default hyperparameters in WEKA. The most likely reason is that the training and test folds in this set-up are very small, inducing high variance in the accuracy estimates obtained using cross-validation.

The second observation is that variants of LDA perform competitively with other linear models, which suggests that developing new generalisations of LDA specifically for few-shot learning could be a fruitful direction for future research. Another finding is that ℓ^2-regularised LR and the cosine similarity classifier—a method that is known to be competitive with the state-of-the-art [5]—perform similarly. Considering ℓ^2-regularised LR is perhaps the oldest and most estab-

lished way to perform transfer learning in a deep learning context, this raises the question of whether any significant progress has been made by introducing variants of this basic approach to formulating linear classifiers.

The accuracy of all of the algorithms decreases as the domain shift increases, consistent with the findings of Guo et al. [10]. In particular, among the four datasets of the original CDFSL benchmark, CropDisease results in the highest accuracy for all of the algorithms, followed by EuroSAT, ISIC and ChestX in this exact order, with all of the algorithms performing only slightly better than random guessing on ChestX (i.e., 20%). For all of the algorithms, the accuracy on Food101 is shown to be consistently between EuroSAT and ISIC. It can therefore be speculated that the domain shift from ImageNet to Food101 is greater than the domain shift from ImageNet to EuroSAT and smaller than that from ImageNet to ISIC.

The more sophisticated ResNet152 feature extractor, trained on the more comprehensive version of ImageNet, grants a very substantial performance boost to all of the classifiers over the ResNet10 model on the three datasets that are relatively similar to ImageNet, i.e., CropDisease, EuroSAT and Food101. However, the ResNet152 model yields a performance decrease on the two datasets that are relatively different from ImageNet, i.e., ISIC and ChestX. More discussion on the relation between feature extractors and task domains will be provided in Section 4.3.

4.2 Feature Normalisation

We now consider the effect of normalising the feature vectors in 5-shot learning. Table 3 shows the accuracy of the algorithms on Food101-ResNet152 feature vectors normalised in various ways, with the best result for each learning algorithm shown in bold.

1-vs-rest logistic regression with ℓ^1 regularisation, with its default regularisation hyperparameter value (i.e., $cost = 1$ in the LIBLINEAR implementation used in WEKA), is observed to achieve low accuracy with ℓ^p normalised data, ranging from 20% to 57.83%. When given a large value of the regularisation parameter (e.g., $cost = 10^{10}$), thus enabling closer fit to the training data, its accuracy improves, achieving 77.4% with ℓ^1 normalised data. Unfortunately, such a large value decreases the classifier's performance with un-normalised feature vectors to 66.57%.

It should be noted that the normalisation method is irrelevant to the cosine similarity classifier, because the direction of each feature vector is used for classification and all of the feature vectors are ℓ^2-normalised in the classifier. This means that the cosine similarity classifier produces the same result regardless of whether the feature vectors are normalised, or the value of p in ℓ^p normalisation.

Table 3 shows that LDA with ℓ^2 normalisation achieves very competitive performance on the Food101-ResNet152 feature vectors. Further experiments show that LDA gains a performance increase from ℓ^2 normalisation as opposed to no normalisation on all of the datasets. This is shown in Table 4. It can be argued that certain normalisation methods can give certain algorithms a

Table 3. Accuracy of the algorithms on ℓ^p-normalised Food101-ResNet152 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
kNN	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

Table 4. Comparison between no normalisation and ℓ^2 normalisation for LDA on all of the datasets

	CropDisease	EuroSAT	Food101	ISIC	ChestX
None	93.74±0.46	86.74±0.49	77.9±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**

consistent increase in CDFSL performance, and thus a competitive edge. One such example is LDA with ℓ^2 normalisation.

4.3 Few-Shot Fine-Grained Classification

Particularly accurate results are obtained on the CropDisease data in the experiments presented above, but it is noteworthy that in fact, this is a dataset of *both* plant species and different plant diseases. Table 5 shows the relation between the classification accuracy and the number of plant species present in the training and test data in each of the 600 runs performed for the 5-way 5-shot experiment, using 1-vs-rest logistic regression with ℓ^2 regularisation on the unnormalised CropDisease-ResNet152 feature vectors. In each of the 600 iterations of the experiment, i.e., a classification task with 5 classes, the presence of more plant species indicates that less disease classification is involved: an iteration with five plant species virtually becomes a pure plant classification task. The table shows that the accuracy is positively correlated with the number of plant species in an iteration.

Thus, to investigate the effect of the two feature extractors on purely the task of fine-grained classification of *plant diseases*, we propose use of the TomatoDisease dataset, a subset of CropDisease that contains all of its instances pertaining to tomato diseases. TomatoDisease does not involve different plant species and

Table 5. Classification accuracy increases as the number of selected plant classes increases in the 5-way 5-shot CropDisease data. Summarised from the 600 iterations of 1-vs-rest logistic regression with ℓ^2 regularisation on the un-normalised CropDisease-ResNet152 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 6. TomatoDisease leads to lower classification accuracy than CropDisease for all of the algorithms.

Accuracy	ResNet10		ResNet152	
	CropDisease	TomatoDisease	CropDisease	TomatoDisease
Naïve Bayes	82.92±0.75	57.87±0.79	92.71±0.51	68.35±0.66
kNN	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

focuses solely on tomato diseases, making this a much more challenging problem. Table 6 shows the accuracy of the classifiers is lower on TomatoDisease when compared with the original CropDisease.

All of the instances and classes in TomatoDisease are from the original CropDisease dataset, and yet the robust classifiers exhibit significantly worse performance on TomatoDisease than CropDisease when using a feature extractor trained on ImageNet. Thus, it can be argued that domain shift between datasets is not only related to the superficial differences in instance properties (such as image colours, perspectives and objects in them) but also the nature of the tasks represented by the datasets in question. When two tasks are similar in format but different in nature, it can still be hard, even for a sophisticated and well-trained AI system, to perform adequately in one task by relying mainly on its empirical knowledge of the other task.

5 Conclusion

In this paper, we evaluated and compared different robust learning algorithms, normalisation methods and pre-trained feature extractors in the context of cross-domain few-shot learning. We demonstrated that the combination of a good robust classifier, an effective feature extractor, and a suitable normalisation method, can provide a significant performance increase in CDFSL problems. We showed that, in the various CDFSL experiments we performed, the cosine similarity classifier and 1-vs-rest logistic regression with ℓ^2 regularisation are consistent top-performers amongst the algorithms we evaluated, which indicates that the old and established ℓ^2-regularised logistic regression is a viable alternative to the competitive cosine similarity classifier in CDFSL. It was also shown that algorithms used in the gene expression domain, namely the random projection ensemble of LDA classifier and the nearest shrunken centroid classifier, are applicable in CDFSL scenarios. We additionally demonstrated that certain combinations of classifiers and normalisation methods perform consistently better in CDFSL tasks than their counterparts without normalisation; one such example is LDA with ℓ^2-normalised feature vectors. Finally, more sophisticated and better trained feature extractors are shown to increase the classification accuracy considerably for target domains that are similar to the source domain in properties and concept, while this positive effect is weakened or virtually non-existent for target domains that are drastically different from the source domain.

Research questions that can be derived from our paper include:

1. Can the top-performing robust classifiers in CDFSL be utilised to improve semi-supervised learning approaches?
2. How to methodically structure and train feature extractors to achieve an optimal transfer between source domains and target domains?
3. How to systematically quantify domain shift?
4. How to assemble datasets from accessible data to have minimal domain shift to known real-world problems?

References

1. David W. Aha, Dennis Kibler, and Marc K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.
2. Peter L Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.
3. Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, pages 446–461, 2014.
4. Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
5. Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019.
6. François Chollet et al. Keras. `https://keras.io`, 2015.
7. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, pages 248–255, 2009.

8. Robert J Durrant and Ata Kabán. Random projections as regularizers: learning a linear discriminant from fewer observations than dimensions. *Machine Learning*, 99(2):257–286, 2015.
9. Eibe Frank, Mark A. Hall, and Ian H. Witten. The WEKA workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, 4th Edition, 2016.
10. Yunhui Guo, Noel C. Codella, Leonid Karlinsky, James V. Codella, John R. Smith, Kate Saenko, Tajana Rosing, and Rogerio Feris. A broader study of cross-domain few-shot learning. In *ECCV*, 2020.
11. 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.
12. Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019.
13. Saskia le Cessie and Johannes C. van Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201, 1992.
14. Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, pages 10657–10665, 2019.
15. Geoffrey J. McLachlan. *Discriminant analysis and statistical pattern recognition*. Wiley, 1992.
16. Sharada P. Mohanty, David P. Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7:1419, 2016.
17. Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.
18. John Platt. Fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods: Support Vector Learning*, pages 185–208, 02 1999.
19. Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
20. Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *NIPS*, pages 4077–4087, 2017.
21. Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu. Class prediction by nearest shrunken centroids, with applications to DNA microarrays. *Statistical Science*, pages 104–117, 2003.
22. Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The HAM10000 Dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5:180161, 2018.
23. Oriol Vinyals, Charles Blundell, Timothy P. Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NIPS*, pages 3630–3638, 2016.
24. Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. ChestX-Ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. *CVPR*, pages 3462–3471, 2017.
25. Ying Yang and Geoffrey I. Webb. Discretization for Naive-Bayes learning: Managing discretization bias and variance. *Machine Learning*, 74(1):39–74, 2009.