

Classifier Ensembles for Image Identification Using Multi-objective Pareto Features

Wissam A. Albukhanajer^{a,*}, Yaochu Jin^b, Johann A. Briffa^c

^a*Surrey Space Centre, Faculty of Engineering and Physical Sciences, University of Surrey, Guildford GU2 7XH UK*

^b*Department of Computer Science, Faculty of Engineering and Physical Sciences, University of Surrey, Guildford GU2 7XH UK*

^c*Department of Communications & Computer Engineering, Faculty of Information & Communication Technology, University of Malta, Msida MSD 2080, Malta*

Abstract

In this paper we propose classifier ensembles that use multiple Pareto image features for invariant image identification. Different from traditional ensembles that focus on enhancing diversity by generating diverse base classifiers, the proposed method takes advantage of the diversity inherent in the Pareto features extracted using a multi-objective evolutionary Trace Transform algorithm. Two variants of the proposed approach have been implemented, one using multilayer perceptron neural networks as base classifiers and the other k -Nearest Neighbor. Empirical results on a large number of images from the Fish-94 and COIL-20 datasets show that on average, ensembles using Pareto features perform much better than traditional classifier ensembles using the same features and data randomization. The better classification performance of the proposed ensemble is further supported by diversity analysis using a number of measures, indicating that the proposed ensemble consistently produces a higher degree of diversity than traditional ones. Our experimental results demonstrate that the proposed classifier ensembles are robust to various geometric transformations in images such as rotation, scale and translation, and to additive noise.

Keywords: Pareto front, classifier ensembles, majority voting, image identification, Trace transform, evolutionary multi-objective optimization

1. Introduction

The idea of combining the outputs of several individual classifiers to produce an ensemble has gained a lot of interest in many machine learning communities and pattern recognition applications [1, 2, 3, 4]. In machine learning, classifier ensembles have been very popular recently because they often improve classification performance significantly. A conventional classifier ensemble combines the outputs of several individual classifiers, each of which is referred to as an *ensemble member*, or an *individual learner*, or a *base classifier*. Typically, individual outputs of an ensemble are combined using a decision-making mechanism such as majority voting, averaging, bagging or stacking to produce a final decision [5]. Depending on whether the ensemble members are of the same type or of different types of models, the ensemble can be *homogeneous* and *heterogeneous*. In this work, we focus on homogeneous ensembles that use the same type of models as base classifiers.

Diversity of homogeneous ensembles can be achieved via data randomization or using different learning algorithms in training the base classifiers [6]. Bian and Wang [7] suggested to enhance ensemble diversity by training base classifiers with different learning algorithms. The authors showed that among the different models used, the

radial basis function (RBF) neural network has the largest diversity in both homogeneous and heterogeneous ensembles. On the other hand, multilayer perceptron (MLP) neural networks have the smallest diversity [7]. A different idea is to include a penalty term for diversity in the cost function in training base classifiers [8]. In case the ensemble members are trained using population based search methods, individuals in the last generation can be made use of to create an ensemble [9]. In addition, Pareto-based multi-objective learning [10] has widely been adopted for creating ensembles, where, apart from minimizing the error on a training data set, a second objective will be optimized, e.g., minimizing the error on another dataset [11], maximizing a diversity measure [12], or minimizing complexity [13, 14].

Although a high degree of diversity is desired in general [15], it has also been found that diversity sometimes negatively impacts the overall ensemble accuracy, in particular when all base classifiers are weak [16, 17]. In other words, to construct accurate base classifiers is as important as to create highly diverse base classifiers. The key challenge in designing ensembles becomes how to gain a higher degree of diversity without degrading the accuracy of the base classifiers.

There is a few research on how to obtain diversity of ensembles through the use of different features as inputs to the base classifiers. In [18], a subset of features is selected

*Corresponding author

Email address: w.a.albukhanajer@surrey.ac.uk
(Wissam A. Albukhanajer)

using a Bayesian technique as the input for base classifiers and a multi-objective optimization algorithm was employed to minimize the error on training data and minimize the number of selected features. In [19], complementary features, such as textures and edge-based local features, are combined at the descriptor extraction level to improve classification performance in a single classifier. Su *et al.* [20] proposed to use both global and a set of local facial features to create a hierarchical ensemble classifier, which produced impressively better results than the state-of-the-art on two large-scale face databases. Subsamples of images are used as features to be fed into randomized trees for classification of images [21]. However, none of the above methods has taken explicit measures to optimize the extracted features according to performances related to classification.

Several feature extraction techniques have been proposed for image analysis, such as Wavelet transform [22], Hidden Markov Model [23], Image Moments [24] and Scale Invariant Feature Transform (SIFT) [25]. Recent research highlights the importance of combining different features in an ensemble to produce efficient and accurate results. In [26], the authors use four features such as color/edge moments to train an ensemble of decision trees to accurately predict the illumination in images. In [27] the authors proposed an approach that extract a compact set of hierarchical space-time tree structures of human actions from training videos and build an ensemble of the discovered trees in combination with simpler action words and pairwise structures for action recognition. Moreover, in [28] a discriminative ferns ensemble is proposed using simple binary features for hand pose recognition. The method uses histogram of bit vectors to achieve translation invariance and compared to the Random forest ensemble. The Trace transform [29] is one of the powerful tools for invariant image analysis. It is an effective method for extracting image features that are robust to rotation, scale and translation (RST).

In this paper, a multi-objective evolutionary Trace transform algorithm is used to extract a set of Pareto image features. We aim to achieve diversity in classifier ensembles by exploiting these image features. We show that the Pareto features near the knee point of the Pareto front can provide sufficient diversity for classifier ensembles for RST invariant image identification. We demonstrate that the performance of the proposed ensemble is better than the average performance of conventional homogeneous ensembles generated by data manipulation that use a single set of features in addition to the traditional ensembles, such as Random Forests [30], that use the same multiple features.

The rest of this paper is organized as follows. Section 2 briefly review the Pareto feature selection method together with a short introduction to the Trace transform.

In Section 3, we describe in detail the construction of homogeneous classifier ensembles, including the methodology for combining individual outputs of the base classifiers. Experimental results on two image datasets are given in

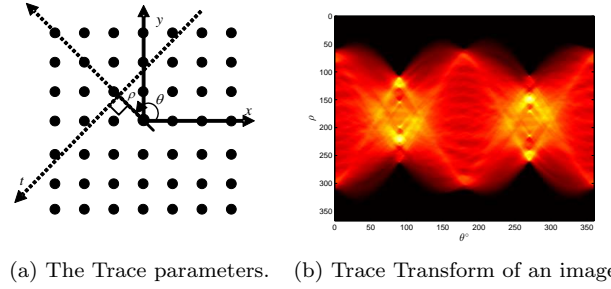


Figure 1: The Trace Transform [29].

Section 4, followed by an analysis of the diversity of the compared ensembles. Conclusions and future work are presented in Section 5.

2. Multi-objective Pareto Image Features

2.1. Trace Transform for Image Feature Extraction

We start with a brief overview of the Trace transform and the theory of Triple features. Kadyrov and Petrou introduced the Trace transform (TT) in [29], using some functionals to produce Triple features that are RST invariant. Given a two-dimensional image $I(x, y)$ of size $M \times N$ pixels, a new representation of I is obtained by applying the *Trace* functional T along lines crossing the image I along different orientations θ and distances ρ from the image origin, as shown in Fig. 1(a). The new representation is a function of two variables, ρ and θ , and can generally be represented as a matrix. An example is given in Fig. 1(b). The process of converting the image from the spatial domain (x, y) into the transform domain (ρ, θ) is called the *Trace transform*. In fact, the generation of a Triple feature requires two further steps. First, a *Diametric* functional D is applied to the (transform domain) output of the Trace transform along the ρ dimension, resulting in a vector of length n_θ equal to the number of orientations considered in computing the Trace transform. Next, a *Circus* functional C is applied to this sequence, producing a scalar value called the *Triple feature*.

Refer to Fig. 1(a), to work out the value of each point along the tracing line t , an index I_B is defined at a fine grid over the original image I such that $0 \leq I_B \leq (t_{end} - t_{begin})$. Each pixel tile in the original image is replaced by fine tile with size $B_N \times B_N$. Therefore, the size of the fine grid is $MB_N \times NB_N$. The t_{I_B} values of the points along the line and their corresponding (i, j) pixel coordinates are given by [31]:

$$\begin{aligned} t_{I_B} &= (t_{begin} + I_B)\Delta t \\ x &= \lfloor (x_B + \lfloor I_B x_{inc} \rfloor) / B_N \rfloor \\ y &= \lfloor (y_B + \lfloor I_B y_{inc} \rfloor) / B_N \rfloor \end{aligned}$$

where t_{begin} and t_{end} are the start and the end point on the tracing line, respectively. The (x_B, y_B) are the fine grid coordinates. And x_{inc} and y_{inc} are the x and y increments in the fine grid domain, respectively. The (x, y)

pixel coordinates are used to obtain the grey value of the image at point with $t = t_{IB}$. Readers are referred to [31] for details about the definition of the beginning and the end points of each tracing line.

If appropriate functionals are used, the extracted features do not change much when the image is subjected to geometric transformations [29]. Refer to Appendix 5 for definitions of possible Trace transform functionals.

Trace transform is a general case of the well-known Radon transform [32]; the first step in Trace transform becomes similar to Radon transform when the Trace functional is the integral. The Radon transform is known for its strength in capturing the directional features of an image and robustness to zero mean white noise. Therefore, Trace transform can be seen as a more general case of Radon transform. The Trace transform has been successfully applied to several machine vision problems [33, 34, 35]. It is evident, however, that the optimal combination of Trace, Diametric and Circus functionals is problem dependent and little effort has been dedicated to addressing this issue with few exceptions. For example, Liu and Wang [36] used a number of Trace functionals in a voting ensemble for face recognition, where however, eight Trace functionals were arbitrarily chosen from 22 functionals defined in [37].

2.2. Evolutionary Multi-Objective Trace Transform

From the above description, we can see that it is important to find the optimal combination of the functionals for extracting Triple features capable of discriminating one image from another, which can be measured by between-class variance, while recognizing geometrically transformed versions as the same image, often known as within-class variance. However, maximizing between-class variance and minimizing within-class variance can be conflicting, and such optimization problems having multiple conflicting objectives are called multi-objective optimization (MOO) [38].

Different from single objective optimization, MOO often has a set of solutions rather than one single optimal solution. Let S denote a set of all feasible solutions of size N_s . Consider two solutions $s_1, s_2 \in S$, each having N_f objective functions f_1, f_2, \dots, f_{N_f} to be minimized. Then, solution s_1 is said to dominate s_2 if $f_i(s_1) \leq f_i(s_2)$ for all $i \in \{1, 2, \dots, N_f\}$, and $f_j(s_1) < f_j(s_2)$ for at least one index $j \in \{1, 2, \dots, N_f\}$. In the set of feasible solutions there always exist some solutions that are not dominated by any other solution. These solutions are called *Pareto optimal* or *nondominated* solutions, and the curve or surface formed by all nondominated solutions in the objective space is called the *Pareto front* [38].

Evolutionary algorithms (EAs) are a class of population-based global optimization techniques inspired from natural evolution [39, 40]. Over the past two decades, EAs have been shown very powerful in solving optimization problems, including MOO problems [41]. In addition, multi-objective evolutionary algorithms (MOEA) have widely been used to solve multi-objective learning problems [42].

In this work, we use the Evolutionary Trace Transform in the presence of Noise (ETTN) [43], which adopts the elitist nondominated sorting genetic algorithm (NSGA-II) [44], one popular and powerful MOEA that does not involve explicit user preferences in comparison to MOEA/D (see [45] and [46]), to search for the optimal combinations of functionals in Trace transform to produce RST invariant Triple features, where the within-class variance and between-class variance are used as two objectives in the evolutionary algorithm. In ETTN, the following two objective functions are minimized:

$$\begin{aligned} \min\{f_1, f_2\}, \\ f_1 = S_w, \\ f_2 = \frac{1}{(S_b + \epsilon)}, \end{aligned} \quad (1)$$

where $\epsilon > 0$ is a small constant to avoid division by zero. S_w and S_b are within-class variance and between-class variance, respectively, which are defined as follows:

$$\begin{aligned} S_w &= \sum_{k=1}^K \sum_{j=1}^{C_k} (\Xi_{jk} - \mu_k^\Xi)^2, \\ S_b &= \sum_{k=1}^K (\mu_k^\Xi - \mu^\Xi)^2, \end{aligned} \quad (2)$$

where

$$\mu_k^\Xi = \frac{1}{C_k} \sum_{j=1}^{C_k} \Xi_{jk}, \quad \mu^\Xi = \frac{1}{K} \sum_{k=1}^K \mu_k^\Xi$$

and K is the number of classes and C_k the number of samples in class k . μ_k^Ξ is the mean of class k of Ξ Triple features, Ξ_{jk} is the j^{th} Triple features of class k , and μ^Ξ is the mean of all classes of Ξ Triple features.

To find the optimal combinations for Trace functions for extracting robust and RST invariant features using ETTN, a few selected images from the Fish-94 database with low resolutions (Appendix 5) have been used. These sample images include three different classes, each containing five different types of changes, as follows:

- *Sample 1*: A low-resolution image (64×64) generated from a randomly chosen original image (256×256);
- *Sample 2*: Random rotation, scale and translation of Sample 1 with Gaussian noise (standard deviation=4);
- *Sample 3*: Random rotation, scale and translation of Sample 1 with Gaussian noise (standard deviation=6);
- *Sample 4*: Random rotation of Sample 1;
- *Sample 5*: Random scale of Sample 1.

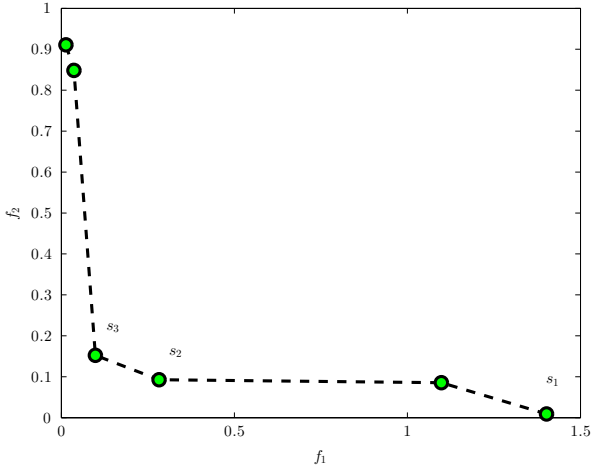


Figure 2: An illustrative example of a set of solutions obtained by ETTN. Each point on the Pareto front represents a particular combination of functionals in the Trace transform, with which an image feature can be extracted.

In the evolutionary algorithm, 150 population size is chosen and the probability of mutation and cross-over are 0.125 and 0.9, respectively. After 200 generations of evolution, a set of non-dominated solutions are obtained, as presented Fig. 2. Details about each solution are given in the next section.

It should be pointed out that although the Pareto functional combinations are obtained using the Fish-95 database, we will also use them for extracting features for images from other databases. This is encouraging because it indicates that Pareto functional combinations for one image database can be generalized to others.

2.3. Selection of Pareto Features

Each of the six solutions shown in Fig. 2 represents a combination of functionals in the Trace transform that can be used to extract a Pareto image feature. These Pareto image features trade off between within-class variance and between-class variance. The question now is which solution should be employed for image classification. In multi-objective optimization, Pareto solutions for which an improvement in one objective will result in a severe degradation in at least another one objective are known as knee solutions, e.g., solutions s_2 and s_3 in Fig. 2. It is known in MOO that, without a predefined user preference, knee solutions are naturally most preferred because they often represent the best trade-offs [42, 47].

Due to the above reasons, knee points such as s_2 and s_3 in Fig. 2 are preferred solutions on the Pareto front and can be chosen as inputs to classifier ensembles. In addition, we can also choose solutions with a large between-class variance, such as s_1 in Fig. 2. The rationale lies in the fact that a solution with a large between-class variance will make it easier to distinguish one image from another if its within-class variance is sufficiently small.

Table 1: Pareto Solutions as Pairs of Features

Pair ID	Triple features pair
X_1	s_1, s_2
X_2	s_1, s_3
X_3	s_2, s_3

Table 2: PARETO SOLUTIONS. REFER TO APPENDIX 5 FOR DESCRIPTION OF EACH FUNCTIONAL.

Label	Triple Features
s_1	$T_0 D_5 C_5$
s_2	$T_0 D_3 C_2$
s_3	$T_0 D_1 C_2$

Features extracted by Trace transform are one dimensional (1-D), though in practice, 2-D features are usually used for image classification by combining two 1-D features. Table 1 lists three pairs of Pareto solutions that can be used as 2-D input features to the base classifiers in an ensemble. Table 2 depicts details about the functionals making up each solution, also refer to Appendix 5 for description of each functional. It is worth mentioning that compared to the traditional Trace transform [29] which uses thousands of features, the ETTN has proven to be able to accurately classify noisy RST deformed images with a much lower computational cost. Readers are referred to [43] and [48] for more details.

3. Construction of Classifier Ensembles

As discussed in Section I, ensemble members should be diverse yet accurate. Diversity of ensembles can be achieved by either promoting diversity in the output of the ensemble members given the same input feature, or by using different input features for different members. The hypothesis of this work is that the Pareto image features extracted by ETTN naturally offer a set of diverse features that can be taken advantage by a classifier ensemble, thereby removing the burdensome task to construct diverse base classifiers.

In this section, we start with a brief overview of the construction of classical ensembles using single features, where diversity of the ensemble members is achieved by data randomization. This is followed by our proposal to construct ensembles using Pareto features. In each case, two types of base classifiers, namely, multilayer Perceptron (MLP) neural networks and k -Nearest Neighbor (k -NN) are adopted for their simplicity and variant structure such that the choice of a strong classifier is not biased.

3.1. Classical ensembles with single features

In homogeneous ensembles, all base classifiers use the same type of learning models and the same input features, where diversity of the ensemble is introduced in training

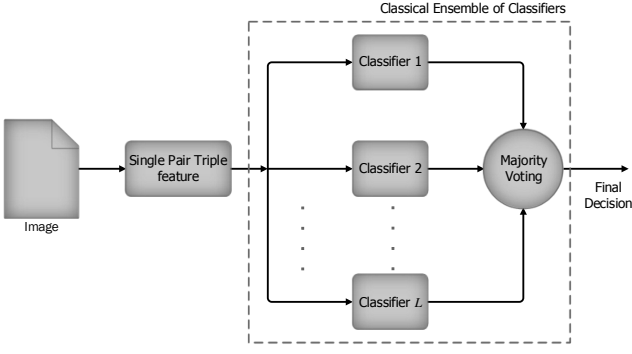


Figure 3: A classifier ensemble with a single feature set as input.

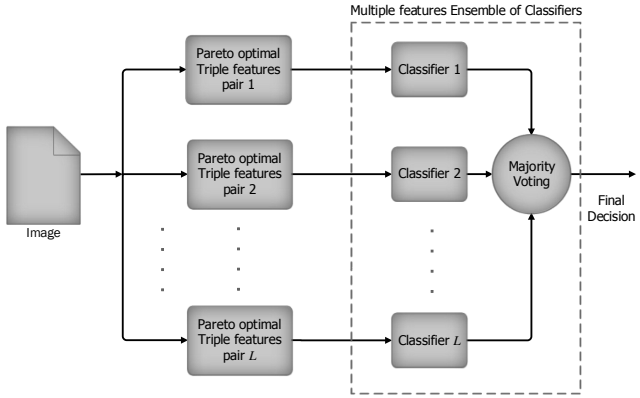


Figure 4: A classifier ensemble with different Pareto features as input.

the base classifiers. Typical approaches for achieving diversity include the use of different starting conditions, different learning parameters or even different learning algorithms. Alternatively, different base classifiers can be trained using randomly chosen subsets of the training data [49], which, for short, is termed data randomization. For these ensembles, a single feature vector is used for all ensemble members. In this work, any Pareto solution, which is a single pair of Triple features from the ETTN will be used. An example of such ensemble of size L is shown in Fig. 3.

Within an ensemble, each base classifier is of the same model type, i.e. either MLP or k -NN. However, they are trained using different subsets of the training data to achieve diversity.

3.2. Classifier ensembles with multiple features

To make use of the natural diversity offered by the Pareto features extracted by ETTN, we construct classifier ensembles using the same type of base classifiers trained using the same training data. Note, however, that different ensemble members use different Triple features as the input. This arrangement is illustrated in Fig. 4.

3.3. Majority voting

There are several methods to combine the output of base classifiers to obtain a final decision [50], including majority voting, weighted majority voting and naive Bayes combination. In this work, majority voting [51] is adopted to combine the output of the base classifiers for its simplicity and it does not require extra efforts in learning or applying weights. In majority voting, the final decision is the one predicted by a majority of the ensemble members, or the null class if such a majority cannot be reached. Consider L base classifiers M_k , $k \in \{1, 2, \dots, L\}$, where each classifier decides between c classes or the null output. The output of each classifier is therefore a class label $y_k = j \in \{0, 1, 2, \dots, c\}$, where 0 represents the null output. The majority vote (MV) can be described as follows:

$$MV = \begin{cases} j & \text{if } |\{y_k : y_k = j\}| \geq \lfloor L/2 \rfloor + 1, \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where $j \in \{1, 2, \dots, c\}$ and $k \in \{1, 2, \dots, L\}$.

4. Experimental Results

4.1. Compared algorithms

We investigate the performance of two different types of ensembles, one based on single features and the other based on multiple Pareto features. For convenience, we name the compared ensembles as follows. The first part of the name indicates the number of feature pairs used, i.e. SF, MF stands for single and multiple features, respectively. The second part identifies the base classifier model used, i.e. MLP, kNN for multilayer perceptron and k -nearest neighbor, respectively.

There are two ensembles using multiple features. MF-MLP consists of three base classifiers using MLP, each having feature pairs X_1, X_2, X_3 , respectively, as the input, refer to Table 1. MF-kNN is similar to MF-MLP, except that all base classifiers use kNN. The base classifiers in both ensembles are trained using the same training data and the same learning algorithm.

There are six classical ensembles using single features, an MLP and a kNN variant for each of X_1, X_2, X_3 . For example, SF1-MLP consists of three base classifiers using MLP, each member using feature pair X_1 as the input. Similarly, SF2-MLP and SF3-MLP denote the ensembles with the MLP as base classifiers whose input feature is X_2, X_3 , respectively. SF1-kNN, SF2-kNN, SF3-kNN are similar, except that the base classifiers use kNN models. Among several classical ensembles (such as Random Subspace [49] and Rotational Forest [52]), Random Forests [30] is a commonly used ensemble algorithm. In Random Forests, the accuracy of classification has increased significantly using a number of trees to generate an ensemble, which allows the trees to vote for the suitable class. In this work, we also compare the performance with the classical Random Forests ensemble.

Table 3: Parameter settings for the ensembles using single features

Parameter	Value/Description ^a
Learning rate(s) (MLP)	0.3, 0.31, 0.32
Momentum(s) (MLP)	0.2, 0.21, 0.22
Epoch (MLP)	5000
Training samples/Class	44
Validation criteria	Cross-validation
Number of folds	2, 3, 5
Sampling method ^b	Linear, Shuffled, Stratified
k (k -NN)	1
Measure type (k -NN)	Euclidean Distance

^a Multiple values correspond to ensembles SF1–SF3, respectively.

^b *Linear sampling*: partitions the set without changing the order of elements (i.e. subsets with consecutive examples are created); *Shuffled sampling*: builds random subsets of the set (seed = 1992); *Stratified sampling*: builds random subsets and ensures that the class distribution in the subsets is the same as in the whole set [54] (seed = 1000).

Table 4: Parameter settings for the ensemble using Pareto features

Parameter	Value/Description
Learning rate (MLP)	0.3
Epoch (MLP)	5000
Training samples/Class	44
Validation criteria	Cross-validation
Number of folds	5-fold
Sampling method	Stratified
k (k -NN)	1
Measure type (k -NN)	Euclidean Distance

4.2. Experimental setups

To empirically compare the performance of the different classifier ensembles, two image databases, Fish-94 database [29] and the Columbia COIL-20 database [53], are used. Details about the two databases are provided in Appendix 5. Parameter settings for the base classifiers using single input features are given in Table 3, while those for the proposed ensembles using Pareto features are given in Table 4. For the Random Forests, the number of trees are chosen such that there is no further improvement in accuracy on the training dataset when increase the number of trees. The sufficient number of trees in Random Forest ensemble is ten trees.

For each database, we used 44 samples per class for training the base classifiers, which include the original image, 24 rotated images, 9 scaled images and 10 translated images. In other words, each training sample may have only one type of geometrical transformation, with no noise being added. By contrast, the test images are created by performing combinations of RST transformations of the original images and adding Gaussian noise or salt and pepper noise. This noise is added to the whole image (i.e. on both the object and background). Example test images

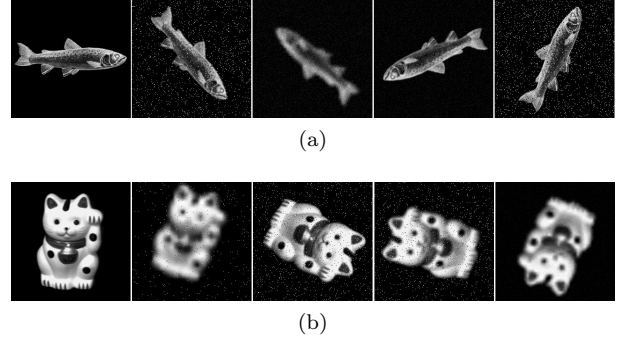


Figure 5: Sample test images from (a) Fish-94 database and (b) COIL-20 database. From the left, each row contains the original image and four modified versions.

are given in Fig. 5.

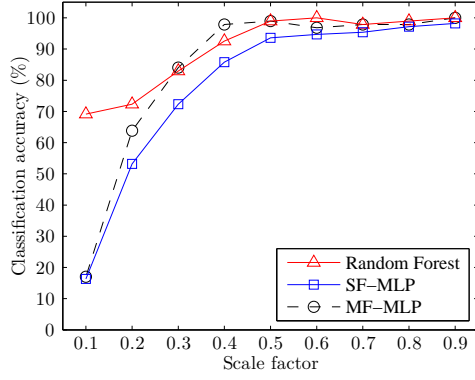
4.3. Performance Analysis

To measure the performance of each ensemble, we calculate the percentage of the correctly classified samples on the test set as the classifier accuracy.

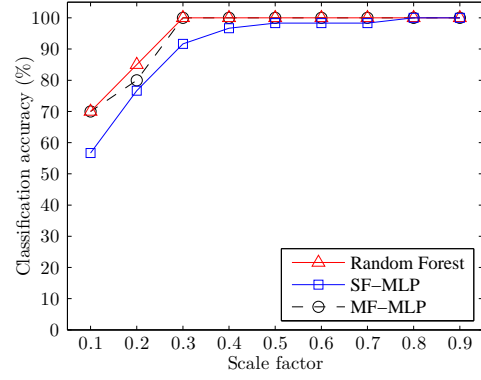
We start by evaluating the performance of MF-MLP on images with different scale factors and no additive noise. This is compared with the average performance of three ensembles using single features, namely, SF1-MLP, SF2-MLP, and SF3-MLP, as well as the classical Random Forests ensembles.

For the given experimental settings, the classification accuracy of MF-MLP, the average accuracy of the three classical ensembles (denoted by SF-MLP) and the Random Forests ensemble are shown in Fig. 6 for the Fish-94 and COIL-20 databases. It can be seen that the overall performance of MF-MLP is consistently better than the mean of the corresponding classical ensembles on both databases. For images from the Fish-94 database, the performance of MF-MLP is significantly better than SF-MLP especially when the scale factor is below 0.5, and it shows a comparative performance in comparison with the Random Forests. Similarly, for images from the COIL-20 databases, MF-MLP also performs impressively better than the mean of ensembles with single input features when the scale factor is between 0.1 and 0.5. Note that the Random Forests shows comparative results in terms of accuracy comparing with the MF-MLP. However, this will not be the case in the presence of rotation and translation transformation and in the presence of noise.

Next, we investigate the invariance performance of the ensembles under more realistic conditions, where each test image has rotation, translation, scaling, and additive noise. Fixed sets of test images were generated, with random rotation and translation, for a range of scaling factors and noise variance. The same test images were used with the compared ensembles. The classification performance of MF-MLP is compared with the average performance of the three classical SF-MLPs, as shown in Fig. 7. Similarly

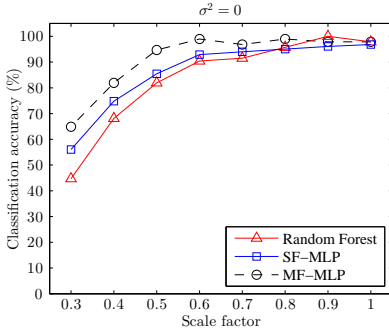


(a) Fish database.

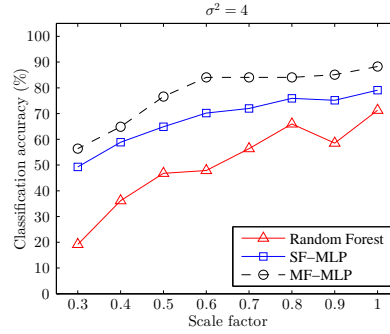


(b) COIL-20 database.

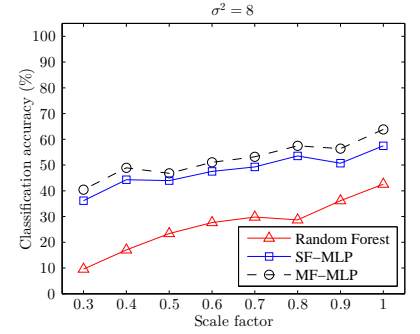
Figure 6: Robustness to different scale factors. MF-MLP, average performance of three classical ensembles SF-MLP and Random Forests ensembles.



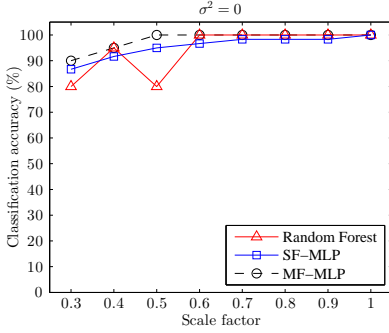
(a) Fish-94.



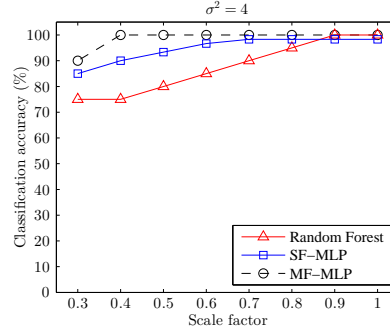
(b) Fish-94.



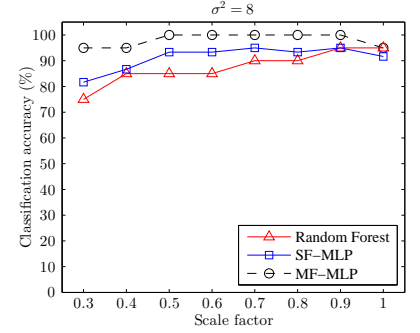
(c) Fish-94.



(d) COIL-20.



(e) COIL-20.



(f) COIL-20.

Figure 7: Robustness to random rotation, translation, and fixed scaling and additive Gaussian noise of zero mean and variance σ^2 for MF-MLP, the average performance of three classical ensembles SF-MLP and Random Forests ensemble, for Fish-94 and COIL-20 database.

to what we observed in Fig. 6, MF-MLP performs consistently better than the mean performance of the three classical classifier ensembles SF-MLP for all noise levels on both databases. Note, however, that the performance of ensembles on the Fish-94 database seriously degrades in the presence of higher noise. This is probably because in this database the objects occupy only a small part of the whole image.

We repeat the experiments using salt and pepper noise instead of Gaussian noise. This noise model randomly re-

places pixels with pure white or black, with a given probability. Results for RST transformation and salt & pepper noise are shown in Fig. 8. Similar to the case with Gaussian noise, MF-MLP performs better than the average performance of the three classical SF-MLP ensembles and Random Forests ensembles. We again see that all ensembles are very sensitive to higher noise for the Fish-94 database.

Finally, we repeat the experiments using k -NN base classifiers. Results for additive Gaussian noise are given

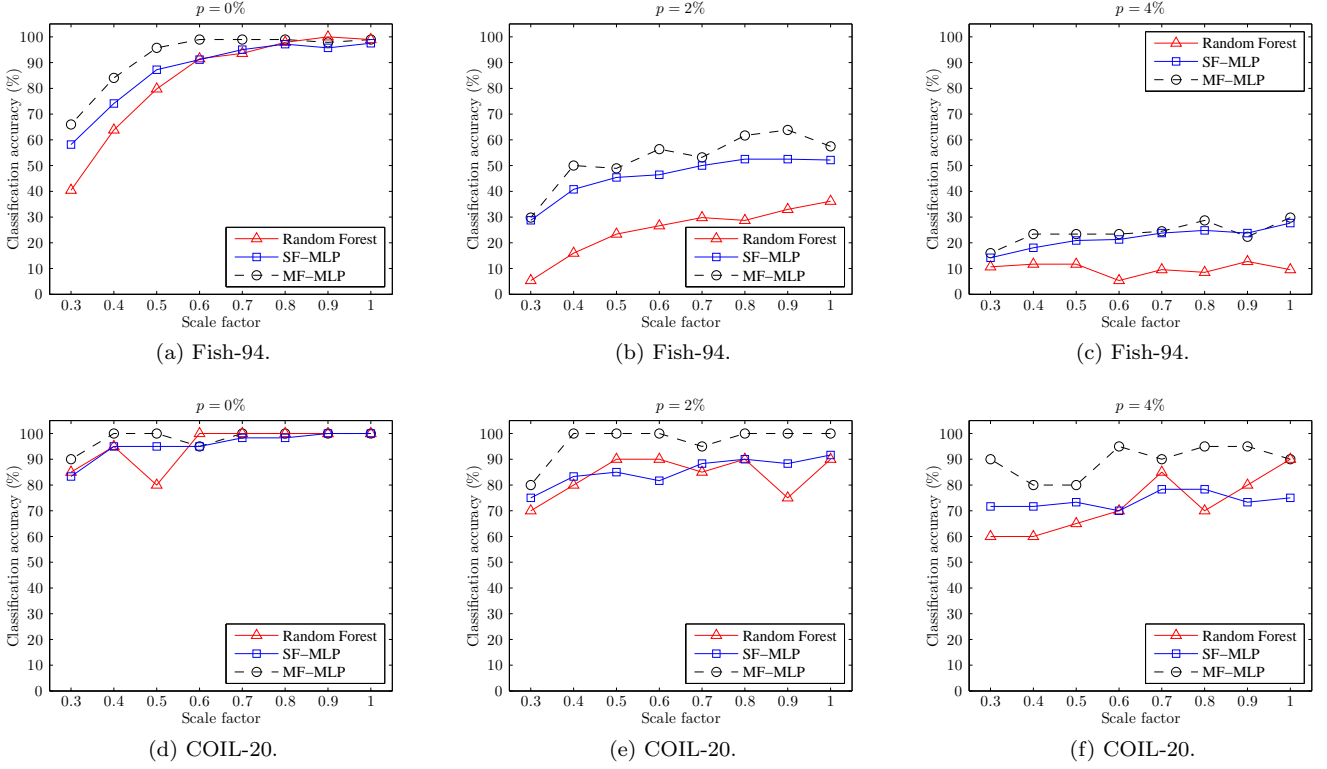


Figure 8: Robustness to random rotation, translation, and fixed scaling and salt & pepper noise with proportion p of affected pixels, for MF-MLP, the average performance of three classical ensembles SF-MLP and Random Forests ensembles, for Fish-94 and COIL-20 datasets.

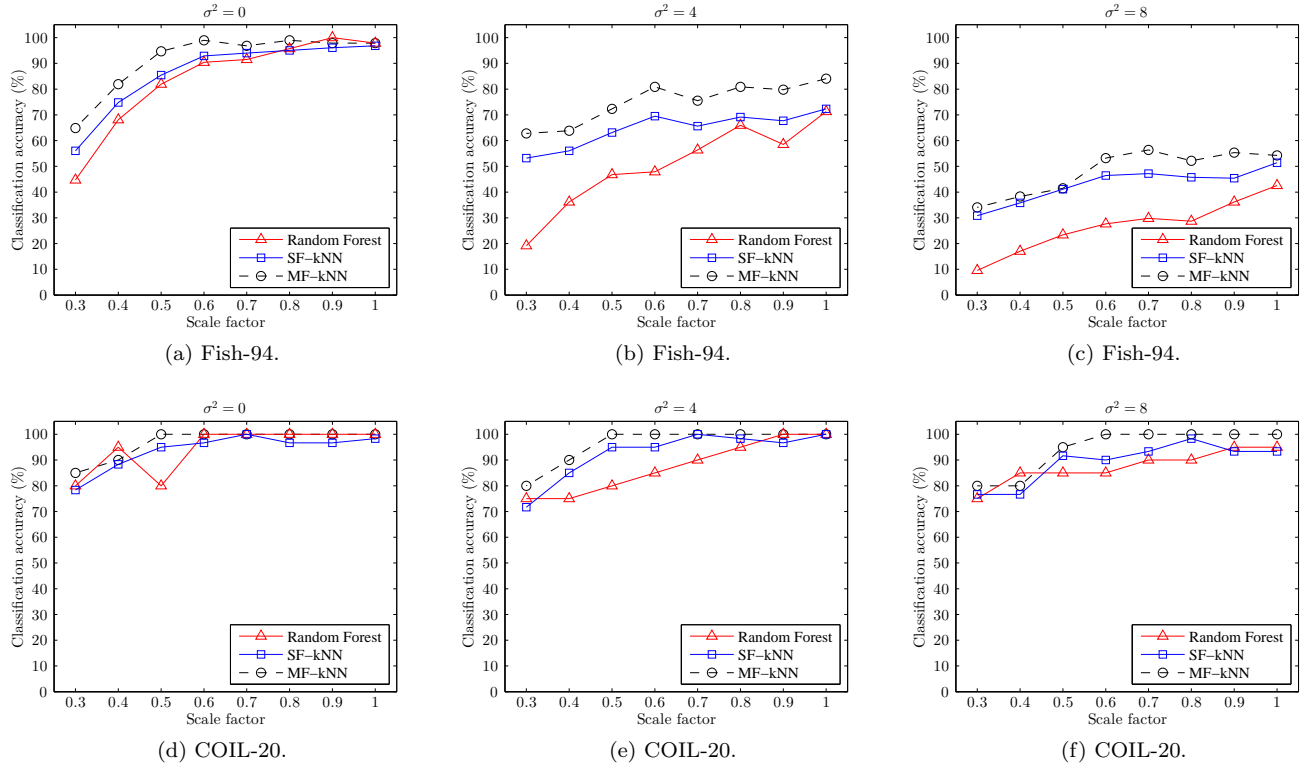


Figure 9: Robustness to random rotation, translation, and fixed scaling and additive Gaussian noise of zero mean and variance σ^2 for MF-kNN, the average performance of three classical ensembles SF-kNN and Random Forests ensembles, for Fish-94 and COIL-20 database.

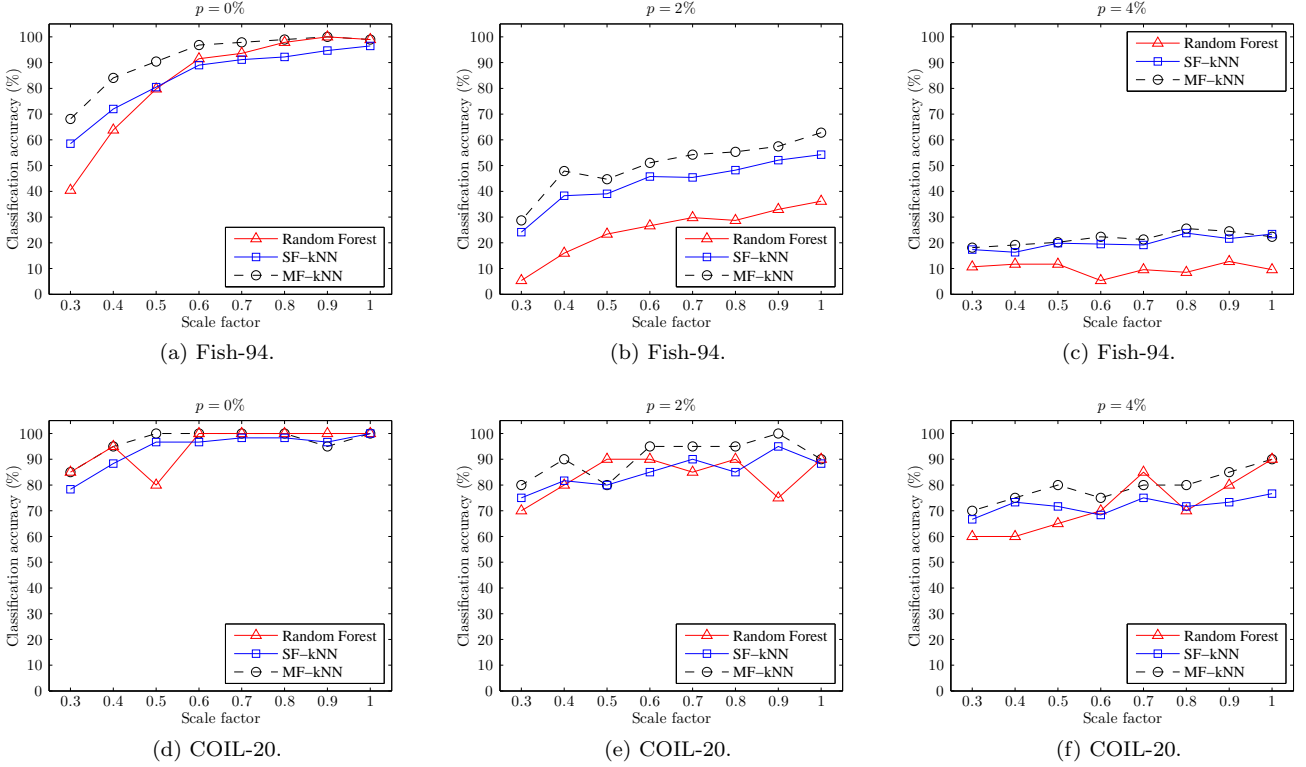


Figure 10: Robustness to random rotation, translation, and fixed scaling and salt & pepper noise with proportion p of affected pixels, for MF-kNN, the average performance of three classical ensembles SF-kNN and Random Forests ensembles, for Fish-94 and COIL-20 datasets.

in Fig. 9, while results for salt and pepper noise are given in Fig. 10. Very similar results to those using MLP as the base classifiers have been obtained. Overall, MF-kNN outperforms the classical ensembles and SF-kNN in all cases considered in this work. This implies that the performance improvement achieved by the ensembles with Pareto features does not rely on the base classifier used. The results confirm our hypothesis that the performance of the ensembles with different Pareto features as input is better than the classical ensembles and the average performance of classical ensembles generated using the same features.

In the following, we quantitatively analyzed the diversity of the ensembles using various diversity measures, hoping to gain some more insight into the performance difference between the two types of ensembles.

4.4. Diversity Analysis

Much work has been done to investigate the influence of diversity and accuracy of member classifiers on the accuracy of the overall ensemble [55, 7, 56]. In general, diversity measures can be divided into two categories: pairwise and non-pairwise. In the previous section, we have shown that the ensembles using multiple Pareto features as input outperform those using the counterpart single features. To examine if such performance improvement can be attributed to a higher degree of diversity offered by the Pareto features, we quantitatively compare the diversity

of ensemble under study using six diversity measures addressed in [16]. In the following, we briefly introduce the six diversity measures, starting with three pairwise ones. In all cases, the diversity is calculated over a set of N_t test samples.

Q statistics: To assess the similarity of classifier outputs, Yule's Q statistics [57] for two classifiers M_1 and M_2 can be formulated. Let N^{ab} represent the number of samples for which M_1 is correct if $a = 1$ or incorrect if $a = 0$ and M_2 is correct if $b = 1$ or incorrect if $b = 0$. Note that by definition $N^{11} + N^{10} + N^{01} + N^{00} = N_t$. The Q measure is then given by:

$$Q_{M_1, M_2} = \begin{cases} \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}} & \text{if } N^{11}N^{00} + N^{01}N^{10} > 0, \\ 1 & \text{otherwise.} \end{cases} \quad (4)$$

Disagreement measure: This measure calculates how often two classifiers have different opinions. It is defined as the ratio between the number of observations when M_1 is correct and M_2 is incorrect (or vice versa) to the total number of observations:

$$\text{Dis}_{M_1, M_2} = \frac{N^{01} + N^{10}}{N_t}. \quad (5)$$

The double-fault measure: As the name suggests, this measure considers the proportion of the number of incorrect classifications made by both classifiers. It can be writ-

ten as:

$$DF_{M_1, M_2} = \frac{N^{00}}{N_t}. \quad (6)$$

By their very nature, pairwise measures determine the diversity between each pair of classifiers. As we are concerned with the overall diversity of an ensemble, for an ensemble with L classifiers and a diversity measure D we compute the average diversity between all distinct pairs of base classifiers as follows:

$$\bar{D} = \frac{2}{L(L-1)} \sum_{\substack{i, j \in \{1, 2, \dots, L\}, \\ i < j}} D_{M_i, M_j} \quad (7)$$

Kohavi-Wolpert variance: This measure accounts for the variability of the predicted class label for a given dataset using base classifiers with two possible outputs: correct output and incorrect output. This measure is defined by

$$\sigma_{KW} = \frac{1}{N_t L^2} \sum_{i=1}^{N_t} N_c(i)(L - N_c(i)) \quad (8)$$

where $N_c(i)$ is the number of base classifiers that correctly classify input sample i .

Entropy: This is an information-theoretic measure for the uncertainty in the ensemble output, which is considered as a random variable. It is computed by

$$H = \frac{1}{N_t(L - \lceil L/2 \rceil)} \sum_{i=1}^{N_t} \min\{N_c(i), L - N_c(i)\} \quad (9)$$

where maximum diversity occurs when $H = 1$.

Generalized diversity: Minimum diversity occurs when the failure of one classifier is accompanied by failure of other classifiers. In contrast, maximum diversity occurs when the failure of one classifier is accompanied by correct classification of other classifiers. The generalized diversity measure relates the probability at which two classifiers fail with the probability at which one classifier fails. Let p_k be the probability that exactly k of L classifiers are incorrect, which can be estimated by

$$p_k = \frac{|\{i : N_c(i) = L - k\}|}{N_t} \quad (10)$$

where $k \in \{0, 1, 2, \dots, L\}$ and $i \in \{1, 2, \dots, N_t\}$. The generalized diversity measure is given by

$$GD = 1 - \frac{\sum_{k=1}^L \frac{k(k-1)p_k}{L(L-1)}}{\sum_{k=1}^L \frac{k p_k}{L}}. \quad (11)$$

Maximum diversity occurs when $GD = 1$.

To assess the diversity of each ensemble we perform experiments using 20 objects from each of the Fish-94 and COIL-20 databases. For each diversity measure, we calculate the average value over three different runs using different test objects. Results are given in Tables 5–8.

From these calculations, we can see that MF-MLP and

Table 5: Diversity measures and average performance for MLP based ensembles (Fish-94)

Diversity ^a	MF-MLP	SF1-MLP	SF2-MLP	SF3-MLP
\bar{Q} (↓)	-0.106	0.969	0.997	1.000
\bar{D}_{is} (↑)	0.400	0.089	0.056	0.033
$\bar{D}F$ (↓)	0.189	0.256	0.461	0.367
σ_{KW} (↑)	0.133	0.030	0.019	0.011
H (↑)	0.600	0.133	0.083	0.050
GD (↑)	0.595	0.159	0.086	0.051
P_I	0.611	0.700	0.511	0.617
P_{MV}	0.667	0.733	0.517	0.633

^a Diversity is greater if the measure is lower (↓) or greater (↑).

Table 6: Diversity measures and average performance for MLP based ensembles (COIL-20)

Diversity ^a	MF-MLP	SF1-MLP	SF2-MLP	SF3-MLP
\bar{Q} (↓)	0.494	1.000	1.000	1.000
\bar{D}_{is} (↑)	0.156	0.022	0.022	0.000
$\bar{D}F$ (↓)	0.011	0.256	0.150	0.033
σ_{KW} (↑)	0.052	0.007	0.007	0.000
H (↑)	0.233	0.033	0.033	0.000
GD (↑)	0.926	0.074	0.076	0.000
P_I	0.911	0.883	0.839	0.967
P_{MV}	0.967	0.883	0.850	0.967

^a Diversity is greater if the measure is lower (↓) or greater (↑).

Table 7: Diversity measures and average performance for k -NN based ensembles (Fish-94)

Diversity ^a	MF-kNN	SF1-kNN	SF2-kNN	SF3-kNN
\bar{Q} (↓)	0.193	0.973	1.000	1.000
\bar{D}_{is} (↑)	0.378	0.133	0.000	0.000
$\bar{D}F$ (↓)	0.244	0.250	0.400	0.517
σ_{KW} (↑)	0.126	0.044	0.000	0.000
H (↑)	0.567	0.200	0.000	0.000
GD (↑)	0.530	0.247	0.000	0.000
P_I	0.567	0.683	0.600	0.483
P_{MV}	0.633	0.717	0.600	0.483

^a Diversity is greater if the measure is lower (↓) or greater (↑).

MF-kNN exhibit much greater diversity over their counterparts using single features in terms of all the six considered diversity measures. This confirms our hypothesis that diversity in ensembles using multiple Pareto features is much greater than ensembles using the same input feature whose diversity is mainly achieved using data randomization. Note that in training the base classifiers of the ensembles using multiple Pareto features, no explicit measures have been taken to promote diversity. Finally, we calculate the average performance of individual clas-

Table 8: Diversity measures and average performance for k -NN based ensembles (COIL-20)

Diversity ^a	MF-kNN	SF1-kNN	SF2-kNN	SF3-kNN
\bar{Q} (\downarrow)	0.020	1.000	1.000	1.000
\bar{D}_{is} (\uparrow)	0.211	0.000	0.000	0.000
$\bar{D}F$ (\downarrow)	0.022	0.150	0.250	0.033
σ_{KW} (\uparrow)	0.070	0.000	0.000	0.000
H (\uparrow)	0.317	0.000	0.000	0.000
GD (\uparrow)	0.886	0.000	0.000	0.000
P_I	0.872	0.850	0.750	0.967
P_{MV}	0.933	0.850	0.750	0.967

^a Diversity is greater if the measure is lower (\downarrow) or greater (\uparrow).

sifiers (P_I) and the average performance of the majority voting (P_{MV}) over the three different runs.

These results show how the increased diversity also increases the reliability and improves the performance of the ensembles. It is worth pointing out that in all cases, the performance of the majority voting ensemble is better than or as good as the performance of the individual classifiers, as indicated in the last two rows of Tables 5–8.

5. Conclusions

In this paper we proposed to construct classifier ensembles using multiple Pareto features as inputs extracted by a multi-objective evolutionary Trace Transform algorithm. Two types of models, multilayer perceptron and k -Nearest Neighbor base classifiers are used as the base classifiers for the ensembles. To verify that multiple Pareto features can provide sufficient diversity, no particular measures are taken in generating ensemble members. By contrast, for traditional classifier ensembles using single input features, data randomization techniques have been employed in training the ensemble members to explicit promote the diversity of ensembles.

The proposed ensembles using multiple Pareto features are empirically compared with the classical ensembles using single features and the classical Random Forest for image identification on two widely used image databases. Experimental results on both databases demonstrate that ensembles using multiple Pareto features consistently outperform the ensembles using single features and Random Forests, when the images are subject to geometrical transformations and additive noise. To further confirm the hypothesis that the better performance of the ensembles using multiple Pareto features can be attributed to the diversity in the multiple Pareto features, we quantitatively compared the diversity of the two types of ensembles using six diversity measures. Our results confirm that the ensembles using Pareto features are more diverse than traditional ones using single features.

Future work will focus on the multi-objective evolutionary optimization of the Trace transform by include di-

versity as an additional objective using trainable and non-trainable ensemble methods rather than simple majority voting. One potential advantage of promoting diversity in features over promoting diversity in the ensemble members is that diversity in features may be less conflicting with the accuracy of the base classifiers. Meanwhile, it might also be of interest to combine the idea of achieving diversity in features with that of promoting diversity in the base models via data randomization.

Acknowledgements

This work was supported in part by the EPSRC Industrial CASE Studentship under Grant 10001560, in part by the Joint Research Fund for Overseas Chinese, Hong Kong, and in part by the Macao Scholars of the National Natural Science Foundation of China under Grant 61428302.

Appendix

Trace transform Functionals

We list here the functionals used to produce the Pareto features of Table 2, shown in Fig. 2. The fourteen Trace functionals are given in Table 9, six Diametric functionals are given in Table 10, and six Circus functionals are in Table 11.

Datasets

The first database used in this work is the Fish-94 database, containing 94 grayscale images of fish, originally presented in [29]. These images are originally of size 200×400 pixels, which we convert to a standard size of 256×256 pixels for convenience. This conversion is done by scaling the images down and padding with the background pixel value. For reference, these are shown in Fig. 11.

The second database is the Columbia Object Image Library of 20 grayscale objects (COIL-20) [53]. This database consists of 20 different objects at 256×256 pixels. These are shown in Fig. 12.

References

- [1] S. Yan, X. Xu, D. Xu, S. Lin, X. Li, Image classification with densely sampled image windows and generalized adaptive multiple kernel learning, *IEEE Transactions on Cybernetics* 45 (3) (2015) 395–404. doi:10.1109/TCYB.2014.2326596.
- [2] F. Yang, G.-S. Xia, G. Liu, L. Zhang, X. Huang, Dynamic texture recognition by aggregating spatial and temporal features via ensemble {SVMs}, *Neurocomputing* 173, Part 3 (2016) 1310 – 1321.
- [3] C. J. Tan, C. P. Lim, Y.-N. Cheah, A multi-objective evolutionary algorithm-based ensemble optimizer for feature selection and classification with neural network models, *Neurocomputing* 125 (2014) 217 – 228.
- [4] H. Lam, U. Ekong, H. Liu, B. Xiao, H. Araujo, S. H. Ling, K. Y. Chan, A study of neural-network-based classifiers for material classification, *Neurocomputing* 144 (2014) 367 – 377. doi:10.1016/j.neucom.2014.05.019.

Table 9: List of Trace Functionals T

Label	Functional definition ^a
T_0	$\sum_{i=1}^{n_t} \tau_i$
T_1	$\max_{i=1}^{n_t} \tau_i - \min_{i=1}^{n_t} \tau_i$
T_2	$\left(\sum_{i=1}^{n_t} \tau_i ^{\frac{1}{2}}\right)^2$
T_3	$\left(\sum_{i=1}^{n_t} \tau_i ^4\right)^{\frac{1}{4}}$
T_4	$\sum_{i=1}^{n_t} \tau_i' $
T_5	$\sqrt{\frac{1}{n_t} \sum_{i=1}^{n_t} (\tau_i - M)^2}, M = \frac{1}{n_t} \sum_{i=1}^{n_t} \tau_i$
T_6	$\sum_{i=1}^{n_t} \sqrt{ \tau_i }$
T_7	$\max_{i=1}^{n_t} \tau_i $
T_8	$\sum_{i=c}^{n_t} (i-c)^2 \tau_i, c = \frac{\sum_{i=1}^{n_t} i \tau_i }{\sum_{i=1}^{n_t} \tau_i }$
T_9	$\sum_{i=c}^{n_t} (i)^2 \tau_i, c = \frac{\sum_{i=1}^{n_t} i \tau_i }{\sum_{i=1}^{n_t} \tau_i }$
T_{10}	$\sum_{i=c}^{n_t} (i)^3 \tau_i, c = \frac{\sum_{i=1}^{n_t} i \tau_i }{\sum_{i=1}^{n_t} \tau_i }$
T_{11}	$\sum_{i=c}^{n_t} (r)^{0.5} \tau_i, c = \frac{\sum_{i=1}^{n_t} l \tau_i }{\sum_{i=1}^{n_t} \tau_i }, r = l-c ,$ $l = 1, 2, \dots, n_t$
T_{12}	$\sum_{i=c}^{n_t} (r) \tau_i, c = \frac{\sum_{i=1}^{n_t} l \tau_i }{\sum_{i=1}^{n_t} \tau_i }, r = l-c ,$ $l = 1, 2, \dots, n_t$
T_{13}	$\sum_{i=c}^{n_t} (r)^2 \tau_i, c = \frac{\sum_{i=1}^{n_t} l \tau_i }{\sum_{i=1}^{n_t} \tau_i }, r = l-c ,$ $l = 1, 2, \dots, n_t$

^aIn the definitions, n_t is the total number of image pixels along the tracing line (as defined by ρ and θ) and τ_i is the value of the i^{th} pixel along this line.

Table 10: List of Diametric Functionals D

Label	Functional definition ^a
D_0	$\sum_{i=1}^{n_\rho} \delta_i$
D_1	$\max_{i=1}^{n_\rho} \delta_i$
D_2	$\left(\sum_{i=1}^{n_\rho} \delta_i ^{\frac{1}{2}}\right)^2$
D_3	$\left(\sum_{i=1}^{n_\rho} \delta_i ^4\right)^{\frac{1}{4}}$
D_4	$\sqrt{\sum_{i=1}^{n_\rho} \delta_i^2}$
D_5	$\max_{i=1}^{n_\rho} \delta_i - \min_{i=1}^{n_\rho} \delta_i$
D_6	$\sum_{i=1}^{n_\rho} \delta_i' $
D_7	$\sum_{i=c}^{n_\rho} (i-c)^2 \delta_i, c = \frac{\sum_{i=1}^{n_\rho} i \delta_i }{\sum_{i=1}^{n_\rho} \delta_i }$

^aIn the definitions, n_ρ is the number of columns in the Trace matrix (i.e. along parameter ρ) and δ_i is the value in the i^{th} column of the Trace matrix for the given θ .

Table 11: List of Circus Functionals C

Label	Functional definition ^a
C_0	$\sum_{i=1}^{n_\theta} \xi_i$
C_1	$\text{median}_{i=1}^{n_\theta} \xi_i$
C_2	$\sqrt{\frac{1}{n_\theta} \sum_{x=1}^{n_\theta} (\xi_i - M)^2}, M = \frac{1}{n_\theta} \sum_{i=1}^{n_\theta} \xi_i$
C_3	$\sum_{i=1}^{n_\theta} \xi_i' $
C_4	$\max_{i=1}^{n_\theta} \xi_i$
C_5	$\max_{i=1}^{n_\theta} \xi_i - \min_{i=1}^{n_\theta} \xi_i$

^aIn the definitions, n_θ is the total number of elements in the row direction of trace matrix (i.e. along parameter θ) and ξ_i is the value of the i^{th} sample in the Diametric vector.

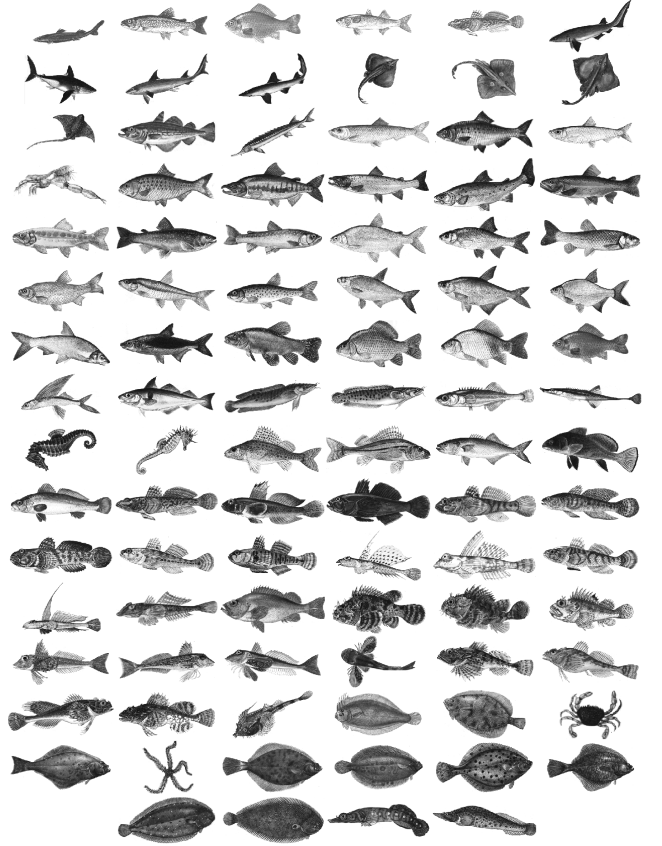


Figure 11: Fish-94 database.

- [5] A. Jain, R. P. W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (1) (2000) 4–37. doi:10.1109/34.824819.
- [6] G. Brown, J. Wyatt, H. R., X. Yao, Diversity creation methods: A survey and categorisation, Journal of Information Fusion 6 (1) (2005) 5–20.
- [7] S. Bian, W. Wang, Investigation on diversity in homogeneous

- and heterogeneous ensembles, in: IEEE International Joint Conference on Neural Networks, 2006. IJCNN '06, 2006, pp. 3078–3085. doi:10.1109/IJCNN.2006.247268.
- [8] Y. Liu, X. Yao, T. Higuchi, Evolutionary ensembles with negative correlation learning, IEEE Transactions on Evolutionary Computation 4 (4) (2000) 380–387. doi:10.1109/4235.887237.
- [9] X. Yao, Y. Liu, Making use of population information in evolutionary artificial neural networks, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 28 (3) (1998) 417–425. doi:10.1109/3477.678637.
- [10] Y. Jin, Multi-Objective Machine Learning, Springer, Berlin, 2006.
- [11] H. Abbass, Pareto neuro-evolution: constructing ensemble of neural networks using multi-objective optimization, in: The 2003 Congress on Evolutionary Computation, 2003. CEC '03,



Figure 12: COIL-20 database.

- [23] M. Hu, Y. Wang, Z. Zhang, D. Zhang, J. Little, Incremental learning for video-based gait recognition with lbp flow, *IEEE Transactions on Cybernetics* 43 (1) (2013) 77–89. doi:10.1109/TSMCB.2012.2199310.
- [24] L. Li, W. Lin, X. Wang, G. Yang, K. Bahrami, A. Kot, No-reference image blur assessment based on discrete orthogonal moments, *IEEE Transactions on Cybernetics PP* (99) (2015) 1–1. doi:10.1109/TCYB.2015.2392129.
- [25] D. G. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision* 60 (2) (2004) 91–110. URL <http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>
- [26] D. Cheng, B. Price, S. Cohen, M. S. Brown, Effective learning-based illuminant estimation using simple features, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1000–1008. doi:10.1109/CVPR.2015.7298702.
- [27] S. Ma, L. Sigal, S. Sclaroff, Space-time tree ensemble for action recognition, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5024–5032. doi:10.1109/CVPR.2015.7299137.
- [28] E. Krupka, A. Vinnikov, B. Klein, A. B. Hillel, D. Freedman, S. Stachniak, Discriminative ferns ensemble for hand pose recognition, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3670–3677. doi:10.1109/CVPR.2014.469.
- [29] A. Kadyrov, M. Petrou, The trace transform and its applications, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (8) (2001) 811–828.
- [30] L. Breiman, Random forests, *Machine Learning* 45 (1) (2001) 5–32. doi:10.1023/A:1010933404324.
- [31] M. Petrou, F. Wang, A Tutorial on the Practical Implementation of the Trace Transform, *Handbook of Texture Analysis*, Imperial College Press, 2008, Ch. 11, pp. 313–346. doi:10.1142/9781848161160_0011.
- [32] S. R. Deans, *The Radon Transform and Some of Its Applications*, Krieger Publishing Company, 1983.
- [33] M. Petrou, A. Kadyrov, Affine invariant features from the trace transform, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (1) (2004) 30–44.
- [34] N. Liu, H. Wang, Modeling images with multiple trace transforms for pattern analysis, *IEEE Signal Processing Letters* 16 (5) (2009) 394–397.
- [35] G. Goudelis, K. Karpouzis, S. Kollias, Exploring trace transform for robust human action recognition, *Pattern Recognition* 46 (12) (2013) 3238 – 3248. doi:<http://dx.doi.org/10.1016/j.patcog.2013.06.006>. URL <http://www.sciencedirect.com/science/article/pii/S0031320313002586>
- [36] N. Liu, H. Wang, Classification of transformed face images with majority voting, in: IEEE International Conference on Systems, Man and Cybernetics, 2007. ISIC, 2007, pp. 2860–2864. doi:10.1109/ICSMC.2007.4413907.
- [37] S. Srisuk, M. Petrou, W. Kurutach, A. Kadyrov, A face authentication system using the trace transform, *Pattern Analysis and Applications* 8 (1) (2005) 50–61.
- [38] K. Miettinen, *Nonlinear Multiobjective Optimization*, Kluwer, Boston, 1999.
- [39] T. Bäck, D. Fogel, M. Zbigniew (Eds.), *Handbook of Evolutionary Computation*, Oxford University, Oxford, 1997.
- [40] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University Press, 1996.
- [41] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, 1st Edition, John Wiley & Sons. Ltd, England, 2002.
- [42] Y. Jin, B. Sendhoff, Pareto-based multiobjective machine learning: An overview and case studies, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 38 (3) (2008) 397–415. doi:10.1109/TSMCC.2008.919172.
- [43] W. A. Albukhanajer, J. A. Briffa, Y. Jin, Evolutionary Multi-objective Feature Extraction in the Presence of Noise, *IEEE*

Vol. 3, 2003, pp. 2074–2080 Vol.3. doi:10.1109/CEC.2003.1299928.

- [12] H. Chen, X. Yao, Multiobjective neural network ensembles based on regularized negative correlation learning, *IEEE Transactions on Knowledge and Data Engineering* 22 (12) (2010) 1738–1751. doi:10.1109/TKDE.2010.26.
- [13] Y. Jin, T. Okabe, B. Sendhoff, Neural network regularization and ensembling using multi-objective evolutionary algorithms, in: *Congress on Evolutionary Computation*, 2004. CEC’04, Vol. 1, 2004, pp. 1–8 Vol.1. doi:10.1109/CEC.2004.1330830.
- [14] C. Smith, Y. Jin, Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction, *Neurocomputing* 143 (0) (2014) 302 – 311. doi:<http://dx.doi.org/10.1016/j.neucom.2014.05.062>.
- [15] S. Wang, X. Yao, Relationships between diversity of classification ensembles and single-class performance measures, *IEEE Transactions on Knowledge and Data Engineering* 25 (1) (2013) 206–219. doi:10.1109/TKDE.2011.207.
- [16] L. I. Kuncheva, C. J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, *Machine learning* 51 (2) (2003) 181–207. URL <http://pages.bangor.ac.uk/~mas00a/papers/lkml.pdf>
- [17] D. de Oliveira, A. Canuto, M. C. P. De Souto, The diversity/accuracy dilemma: An empirical analysis in the context of heterogeneous ensembles, in: *IEEE Congress on Evolutionary Computation*, 2009. CEC ’09, 2009, pp. 939–946. doi:10.1109/CEC.2009.4983046.
- [18] H. Chen, X. Yao, Evolutionary multiobjective ensemble learning based on bayesian feature selection, in: *IEEE Congress on Evolutionary Computation*, 2006. CEC’06, 2006, pp. 267–274. doi:10.1109/CEC.2006.1688318.
- [19] L. Xie, Q. Tian, M. Wang, B. Zhang, Spatial pooling of heterogeneous features for image classification, *IEEE Transactions on Image Processing* 23 (5) (2014) 1994–2008. doi:10.1109/TIP.2014.2310117.
- [20] Y. Su, S. Shan, X. Chen, W. Gao, Hierarchical ensemble of global and local classifiers for face recognition, *IEEE Transactions on Image Processing* 18 (8) (2009) 1885–1896. doi:10.1109/TIP.2009.2021737.
- [21] A. Descampe, C. De Vleeschouwer, P. Vanderghenst, B. Macq, Scalable feature extraction for coarse-to-fine JPEG 2000 image classification, *IEEE Transactions on Image Processing* 20 (9) (2011) 2636–2649. doi:10.1109/TIP.2011.2126584.
- [22] Y. Dong, D. Tao, X. Li, J. Ma, J. Pu, Texture classification and retrieval using shearlets and linear regression, *IEEE Transactions on Cybernetics* 45 (3) (2015) 358–369. doi:10.1109/TCYB.2014.2326059.

Transactions on Cybernetics (45). doi:10.1109/TCYB.2014.2360074.

- [44] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective Genetic Algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.
- [45] X. Ma, F. Liu, Y. Qi, L. Li, L. Jiao, X. Deng, X. Wang, B. Dong, Z. Hou, Y. Zhang, J. Wu, Moea/d with biased weight adjustment inspired by user preference and its application on multi-objective reservoir flood control problem, Soft Computing 20 (12) (2016) 4999–5023. doi:10.1007/s00500-015-1789-z.
- [46] S. Lin, F. Lin, H. Chen, W. Zeng, A moea/d-based multi-objective optimization algorithm for remote medical, Neurocomputing (2016) –doi:http://dx.doi.org/10.1016/j.neucom.2016.01.124.
URL <http://www.sciencedirect.com/science/article/pii/S0925231216309195>
- [47] L. Rachmawati, D. Srinivasan, Multiobjective evolutionary algorithm with controllable focus on the knees of the pareto front, IEEE Transactions on Evolutionary Computation 13 (4) (2009) 810–824. doi:10.1109/TEVC.2009.2017515.
- [48] W. A. Albukhanajer, Y. Jin, J. A. Briffa, Neural Network Ensembles for Image Identification Using Pareto-optimal Features, in: 2014 IEEE Congress on Evolutionary Computation, 2014. CEC’14, Beijing, China, 2014.
- [49] B. Zhang, T. D. Pham, Phenotype Recognition with Combined Features and Random Subspace Classifier Ensemble, BMC Bioinformatics 20 (128). doi:doi:10.1186/1471-2105-12-128.
- [50] L. Kuncheva, A theoretical study on six classifier fusion strategies, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2) (2002) 281–286. doi:10.1109/34.982906.
- [51] L. I. Kuncheva, Combining Pattern Classifiers Methods and Algorithms, John Wiley & Sons. Inc., Hoboken, New Jersey., 2004.
- [52] J. Rodriguez, L. Kuncheva, C. Alonso, Rotation forest: A new classifier ensemble method, Pattern Analysis and Machine Intelligence, IEEE Transactions on 28 (10) (2006) 1619–1630. doi:10.1109/TPAMI.2006.211.
- [53] S. A. Nene, S. K. Nayar, H. Murase, Columbia Object Image Library (COIL-20), Tech. Rep. CUCS-005-96, Columbia University (February 1996).
URL <http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php>
- [54] RapidMiner, Online (Apr. 2012).
URL <http://rapid-i.com/content/view/181/190/>
- [55] L. Hansen, P. Salamon, Neural network ensembles, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (10) (1990) 993–1001. doi:10.1109/34.58871.
- [56] L. Kuncheva, C. Whitaker, C. Shipp, R. Duin, Limits on the majority vote accuracy in classifier fusion, Pattern Analysis & Applications 6 (1) (2003) 22–31. doi:10.1007/s10044-002-0173-7.
URL <http://dx.doi.org/10.1007/s10044-002-0173-7>
- [57] G. U. Yule, On the association of attributes in statistics: With illustrations from the material of the childhood society, &c, Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character 194 (252-261) (1900) 257–319. arXiv:<http://rsta.royalsocietypublishing.org/content/194/252-261/257.full.pdf+html>, doi:10.1098/rsta.1900.0019.
URL <http://rsta.royalsocietypublishing.org/content/194/252-261/257.short>