Using Novelty Search to Explicitly Create Diversity in Ensembles of Classifiers

Rui P. Cardoso
rui.cardoso@imperial.ac.uk
Electrical and Electronic Engineering
Imperial College London
London, UK

Emma Hart
e.hart@napier.ac.uk
School of Computing
Edinburgh Napier University
Edinburgh, UK

David Burth Kurka
Jeremy V. Pitt
{d.kurka,j.pitt}@imperial.ac.uk
Electrical and Electronic Engineering
Imperial College London
London, UK

ABSTRACT

The diversity between individual learners in an ensemble is known to influence its performance. However, there is no standard agreement on how diversity should be defined, and thus how to exploit it to construct a high-performing classifier. We propose two new behavioural diversity metrics based on the divergence of *errors* between models. Following a neuroevolution approach, these metrics are then used to guide a novelty search algorithm to search a space of neural architectures and discover behaviourally diverse classifiers, iteratively adding the models with high diversity score to an ensemble. The parameters of each ANN are tuned individually with a standard gradient descent procedure. We test our approach on three benchmark datasets from Computer Vision — CIFAR-10, CIFAR-100, and SVHN — and find that the ensembles generated significantly outperform ensembles created without explicitly searching for diversity and that the error diversity metrics we propose lead to better results than others in the literature. We conclude that our empirical results signpost an improved approach to promoting diversity in ensemble learning, identifying what sort of diversity is most relevant and proposing an algorithm that explicitly searches for it without selecting for accuracy.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; *Supervised learning by classification*; **Ensemble methods**; **Bio-inspired approaches**; **Distributed algorithms**; *Computer vision*;

KEYWORDS

Diversity, novelty search, machine learning, ensemble

ACM Reference Format:
Rui P. Cardoso, Emma Hart, David Burth Kurka, and Jeremy V. Pitt. 2021. Using Novelty Search to Explicitly Create Diversity in Ensembles of Classifiers. In *2021 Genetic and Evolutionary Computation Conference (GECCO '21), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3449639.3459308

GECCO '21, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8350-9/21/07…$15.00
https://doi.org/10.1145/3449639.3459308

1 INTRODUCTION

Ensemble models are capable of outperforming their individual learners w.r.t. predictive accuracy by taking an average of the individual predictions, thereby decreasing the variance of the final prediction. Dietterich [11] explains how, in order to ensure such a performance gain, the members of an ensemble must be *diverse* in terms of their *behaviour*, i.e. in terms of the *errors* they make in their predictions. However, there is no standard agreement on how diversity should be defined, and thus how to exploit it to construct a high-performing classifier.

In this paper, we first consider metrics which define diversity in terms of the *divergence of errors* made by different models and/or their level of disagreement, characterising the *behaviour* of models. Novelty search [22] is used to search over a space of neural network architectures, maximising a novelty score defined by the behavioural metric w.r.t to the other individuals in the population. Each individual network discovered is optimised using standard gradient descent on a training dataset before its novelty score is calculated. Networks with high-scoring novelty are iteratively added to an archive which forms the final ensemble. The method therefore explicitly *searches* for diversity amongst learners. Diversity drives not only this search, but also the construction of the final ensemble. We test our approach on three benchmark datasets from Computer Vision — CIFAR-10, CIFAR-100 [20], and SVHN [32] — and find that the *error diversity* metrics we propose, used in conjunction with novelty search, lead to higher-performing ensembles than other metrics commonly used in the literature and that the ensembles generated by explicitly searching for diversity significantly outperform those that use either only implicit measures to encourage diversity or random search approaches that simply reward it. The main contributions of this paper are twofold: (1) it describes a systematic novelty search method to evolve an ensemble of individual classifiers which are *behaviourally diverse* by explicitly searching for this diversity and (2) provides new insights into how diversity impacts ensemble performance and which diversity metrics are most appropriate to defining behavioural diversity. We conclude that our empirical results signpost an improved approach to promoting diversity in ensemble learning, identifying what sort of diversity is most relevant and proposing an algorithm that explicitly searches for it without selecting for accuracy.

2 BACKGROUND

Diversity, in particular *behavioural* diversity, defined in terms of the *error diversity*, is key to the performance of ensemble models

[11]. However necessary a condition, Dietterich [11] points out that diversity itself is not sufficient to ensure that an ensemble outperforms its base learners and that these must be simultaneously diverse and accurate. The notion of diversity of interest here may be understood as the level of *disagreement* among the learners in an ensemble with regard to each data point. Whereas for regression ensembles it is possible to derive an exact mathematical formulation for this diversity by calculating the *covariance matrix* of predictions [5], which can be incorporated in the loss function, its definition for classification ensembles is less obvious, thus posing challenges. The literature proposes different metrics for measuring diversity (e.g. [21, 43]), which attest to the vagueness of this concept when tackling ensembles of classifiers. Some of these metrics are pairwise, i.e. measured locally between each pair of learners, or non-pairwise, i.e. a single measure which expresses the global diversity of the ensemble. Bian and Chen [4] propose a diversity metric for classification ensembles based on *error decomposition*, which is typically applied to regression ensembles. They conduct a theoretical investigation into the relationship between this metric and the generalisation of ensembles which informs the design of an ensemble pruning method.

Generation of diversity is mostly done *implicitly*: setting different initial weights in each model, using different training data, different architectures, or different learning algorithms [15]. While it is hoped that these approaches will generate enough diversity, this process is not directly controlled. Several multiobjective methods have been suggested which explicitly maintain a certain level of diversity in the Pareto fronts. For example, Pasti et al. [33] present an immune-inspired multiobjective optimisation algorithm which optimises two objectives, the diversity of the ensemble and the output error, comparing it with another single-objective algorithm which only minimises errors and relies on implicit diversity. They use the *disagreement* metric that we mention in section 3.2. They find that implicit diversity leads to a higher performance gain than their multiobjective approach and discuss the trade-off between diversity and individual model performance.

The use of evolutionary methods is commonplace in tackling the problem of ensemble learning. Zhang and Bhattacharyya [46] construct an ensemble of GP classifiers trained on different small subsets of the training data. They ascribe the higher classification accuracy and less overfitting observed with their approach, in comparison with alternative methods, to the increased diversity provided by the GP search. Baldominos et al. [1] propose an evolutionary algorithm to construct a committee of neural networks and hybridise it with *transfer learning* in an attempt to reduce computational time. They define diversity in terms of architectural diversity and incorporate it in the fitness function. Bhowan et al. [2] employ a multiobjective GP approach to evolve classifier ensembles that are both accurate and diverse in order to tackle the problem of unbalanced data, which calls for increased diversity amongst learners. They encourage diversity w.r.t. to the class output of learners by adding a correlation term in the fitness function, as well as a population-level penalty term. They refine their approach in [3]. Nag and Pal [30] present yet another multiobjective approach for simultaneous feature selection and design of an ensemble of classifiers, but diversity is not defined as an explicit objective. García-Pedrajas et al. [13] evolve ensembles of neural network models using a cooperative coevolution algorithm for ensuring that the models in each ensemble perform well together. In a similar capacity to some of the other papers just mentioned, they use a multiobjective optimisation approach to incorporate four objectives of diversity: correlation, a metric of functional diversity, mutual information, and the Q statistic ([21]). [6, 7] propose an algorithm that searches for diversity using MAP-Elites [29], maximising both diversity and performance of the ensemble members. They observe that promoting diversity leads to significantly higher ensemble performance, even when comparing against ensembles exclusively encompassing the best architecture found during the search. However, they only focus on architectural diversity and, much like some of the approaches just mentioned, their method does not search for diversity explicitly, but rather only rewards it. Of particular interest to this paper is *neuroevolution* [12] and neural architecture search (NAS) [25, 35, 37, 39], which refers to techniques evolving and searching for diverse architectures and sets of hyperparameters to construct neural network models. Optimising hyperparameters is an open problem which is often tackled in an *ad hoc* fashion; evolutionary methods can provide a solution to this problem by harnessing parallelisation to enhance the exploration of vast search spaces [38].

One approach to evolving diverse solutions is to apply novelty search (NS) [22]; this is an approach to evolutionary computation which instead of rewarding objective fitness, rewards the novelty of a solution compared to those in the current population and an archive of previously discovered solutions. Novelty is domain-specific and can be determined w.r.t. the behaviour of a solution or its genotype. Variants of the method include an element of local competition (NSLC [23]), which forces solutions which are close in the novelty space to compete with their neighbours based on objective fitness. This approach has been found to deal well with the problems posed by function plateaus and local optima, outperforming objective-based methods in some applications. Szerlip et al. [41] use NS to accumulate divergent discriminative features in an unsupervised fashion. We use NS to evolve an ensemble of classifiers that are diverse w.r.t. their behaviour, specifically the prediction errors on a validation dataset, searching in the space of hyperparameters. The approach does not explicitly select for accuracy, unlike the approaches mentioned before. Classification accuracy is obtained by optimising the parameters of each network with a standard gradient descent procedure, but does not influence the novelty score. In contrast to previous multiobjective approaches which trade diversity against accuracy, our approach explicitly focuses on the creation of a diverse ensemble. It uses behavioural diversity metrics to guide a search over the space of neural network architectures, iteratively constructing an ensemble made up of the most diverse models.

3 MATERIALS AND METHODS

We use NS to evolve individual neural network models that are behaviourally diverse. The NS searches a space of architectures, whereas the parameters of the neural networks are optimised with a standard gradient descent procedure. The most diverse models are added to an ensemble, which is used for prediction on a test set. Our method is evaluated against some baselines: random search,

ensembles using only implicit diversity, and single classifiers. These steps are described in the following subsections.

3.1 Individual Neural Networks

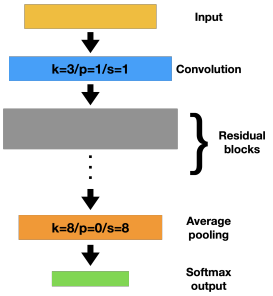
The individual models evolved by our procedure are *residual neural networks* [16] based on the wide architectures proposed by Zagoruyko and Komodakis [45]. Figure 1a shows a generic neural network such as those evolved by our NS algorithm. They are made up of a *convolutional layer* with kernel size 3, padding 1, and stride 1; a variable-length sequence of *residual blocks*; an *average pooling* layer with kernel size 8, padding 0, and stride 8; and a final linear output layer with a *softmax* activation function. The output size, i.e. number of channels, of the convolutional layer and each residual block is variable. Figure 1b illustrates a generic residual block. It is a block of kernel size 3 such as those used by Zagoruyko and Komodakis [45], meaning it is made up of two sequential convolutional layers with kernel size 3, padding 1, and the same, though variable, output size. The output of a block is the sum of its input with the output of the second of the two sequential convolutional layers; note that if their size and shape do not coincide, an extra convolutional layer must first be applied to the input. The stride of the second convolution in the block is always 1; the stride of the first convolution is 2 for the *last two* residual blocks in the network and 1 for all other blocks. If r is the number of residual blocks in the network, the effect of this is that the first $r-2$ blocks *do not* reduce the *dimensionality* of the input data, while the last two halve both the width and height of the input feature planes. Batch normalisation is applied before the average pooling layer and each convolution bar the very first one in the network. Each convolutional layer is followed by a ReLU activation function [31]. Furthermore, the residual blocks apply a *dropout* layer between each convolutional layer.

The *hyperparameters* of each network are evolved by NS. The relevant degrees of freedom are the output size of the first convolution (Figure 1a); the number of residual blocks in the network; the output size of each residual block (i.e. the output size common to all its convolutional layers); and the *probability* of dropout at each residual block. Each individual in the population is then defined by a variable-length vector, depending on the number of blocks r: $[C, O_1, ..., O_r, P_1, ..., P_r]$, where C is the output size of the first convolution, O_i is the output size of block i, and P_i its dropout probability. Each individual is mapped to a Pytorch module [34] for implementation purposes. The *parameters* of each network are randomly initialised and then optimised by a standard gradient descent procedure.

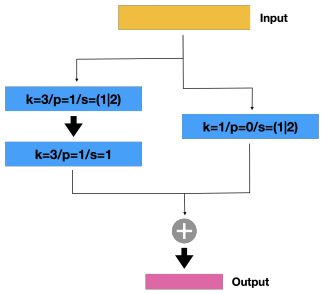
3.2 Diversity Metrics

The key element of NS is the definition of the metric to calculate novelty. We consider three different behavioural metrics, two of which we define ourselves, calculated between pairs of individuals, that are used by the NS procedure to calculate novelty scores. We also consider two selection metrics for adding members to the final ensemble, one of which is also defined by us. Both are calculated in a non-pairwise fashion.

3.2.1 Behavioural metrics for guiding the NS. Let $\boldsymbol{y}_i$ be the vector of predictions for model i with each prediction y_i^n for data point



(a) Generic residual neural network as those evolved by our procedure (k = kernel size; p = padding; s = stride)



(b) Generic residual block. Note that the convolution on the right-hand side is only necessary when the number of channels and/or dimensions of the input are not the same as the output

Figure 1: Generic topology of individual neural networks

$\boldsymbol{x}^n$ being a class label in $\{1..C\}$. Let $\boldsymbol{p}_i$ be a binary vector where $p_i^n = 1$ if the prediction y_i^n is correct and $p_i^n = 0$ otherwise. Let N^{11}, N^{00}, N^{01}, and N^{10}, respectively, be the total number of test instances where two models are both correct, both incorrect, and when one is correct and the other is not. The first diversity metric we consider is the *proportion of different errors* between two models when at least one of them is incorrect. We propose this metric since it provides insight into the divergence between the errors made by two models. We have defined it as:

$$\text{prop}_{i,j} = \frac{N^{01} + N^{10}}{N^{00} + N^{01} + N^{10}} \tag{1}$$

Consider now the *two's complement* of the binary vector of correct predictions $\boldsymbol{p}_i, \boldsymbol{w}_i$ (i.e. the binary vector of wrong predictions). The next metric we propose is the *cosine distance* between the binary vectors of wrong predictions made by two models i and j. Like $\text{prop}_{i,j}$, we consider this metric because it is a measure of the distance between the errors made by two models. We have defined it as:

$$\text{cos\_dist}_{i,j} = 1 - \frac{\boldsymbol{w}_i \cdot \boldsymbol{w}_j}{\|\boldsymbol{w}_i\| \|\boldsymbol{w}_j\|} \tag{2}$$

Finally, we consider a widely used metric (e.g. [21, 33, 43]) defined as the *disagreement* between two models, i.e. the proportion of test instances where one of them is correct and the other is not. We take this metric into account since it enables the judgement of how commonly two models disagree on any test instance. It is defined as:

$$\text{dis}_{i,j} = \frac{N^{01} + N^{10}}{N^{00} + N^{01} + N^{10} + N^{11}} \tag{3}$$

We note here that the two metrics we propose focus more closely on the instances where there was at least one error, i.e. which at least one of the two models has misclassified, and are calculated with respect to those instances. On the other hand, the last metric is simply a proportion of the instances where the two models disagree, calculated w.r.t. the entire test set; it is thus less informative with regard to errors.

3.2.2 Ensemble selection metrics. The first selection metric for adding members to the final ensemble which we propose is simply the mean behavioural diversity metric measured between a candidate ensemble member i and each of the current ensemble members. Let S be the set of ensemble members. This metric is thus defined as:

$$\text{Mean b. d.} = \overline{\text{div\_metric}}_i = \frac{1}{||S||} \sum_{j \in S} \text{div\_metric}_{i,j} \tag{4}$$

Where $\text{div\_metric}_{i,j}$ is one of $\text{prop}_{i,j}$, $\text{cos\_dist}_{i,j}$, and $\text{dis}_{i,j}$. The higher this mean, the more diversity there is among the candidate model and the current ensemble members.

The other selection diversity metric we consider is the *entropy* of predictions among the ensemble members, as defined in [43]. Let S_i be the candidate ensemble created by adding i to S. The entropy is then:

$$E = \frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} -\frac{N_c^n}{||S_i||} \log_C \left(\frac{N_c^n}{||S_i||} \right) \tag{5}$$

Where N is the number of test instances, C is the number of classes and N_c^n is the number of models which assign class c to instance $\boldsymbol{x}^n$. The higher this entropy value, the higher the diversity amongst the members of the candidate ensemble. We note that this entropy metric does not focus on instances where there were errors in the classification, but rather on the entire test set. On the other hand, the mean behavioural metric might focus on those instances if the behavioural metric is one of $\text{prop}_{i,j}$ or $\text{cos\_dist}_{i,j}$, as discussed before.

The previous definitions assume S to be a non-empty set. When S is empty, i.e. when no models have been added to the ensemble yet, the first model to be added is the one with the best *novelty score*, as explained further down in the description of the algorithm.

3.3 Novelty Search Algorithm

Our algorithm for building an ensemble implements NS as described by Lehman and Stanley [22], applying it to our problem domain. Algorithm 1 presents the pseudocode for this procedure. The original training data is split into two sets, one for training and one for validation. The parameters of the models are optimised on the training set with standard gradient descent and the validation set is used to get predictions by each model which will enable the calculation of diversity metrics.

Selection in NS is driven by the novelty score, which computes the sparseness at any point in the behaviour space, defined by the behavioural metric. Areas with denser clusters of visited points are considered less novel and therefore rewarded less. This is defined as the average distance to the K-nearest neighbours of a point, calculated with respect to the other individuals in the current generation and to a stored *archive* of previously sampled solutions. Hence the novelty score is calculated as:

$$NS_i = \frac{1}{k} \sum_{k=0}^{K} \text{div\_metric}(x_i, \mu_k) \tag{6}$$

where μ_k is the kth-nearest neighbour of x_i with respect to the *behavioural* diversity metric $\text{div\_metric}_{i,j}$, selected from the metrics defined in section 3.2.

Individuals are selected for reproduction on the basis of their novelty scores using a tournament selection procedure. In the interests of promoting divergence and avoiding convergence, reproduction only uses mutation. Mutation either adds or removes a randomly chosen residual block from an individual, modifying input/output sizes at the mutation point as necessary; changes the output size and dropout probability of a random block; or swaps two consecutive blocks chosen at random. After a new individual is produced, its parameters are optimised with gradient descent.

After evaluation of the entire population, n_A randomly chosen individuals are added to the *archive*, following the method suggested in [14]. In addition, the individual from the population that scores the highest ensemble selection metric (section 3.2) is added to the ensemble; the size of the final ensemble is therefore the number of iterations of the NS. Ensemble selection metrics are calculated w.r.t. current ensemble members; if the ensemble is the empty set (i.e. in the first iteration), then the individual with the highest novelty score is selected as its first member.

3.4 Evaluation of the Evolved Ensemble

In order to evaluate the performance of the evolved ensemble, we use the stacking technique [44], which trains a linear model to weight the predictions of each individual learner. This linear model is trained for a configurable number of iterations on the validation set mentioned in section 3.3. This is to avoid overfitting the test set.

3.5 Baseline Methods

We consider two baseline methods for comparing the results produced by our approach. The first one is a simple random search algorithm. This algorithm is identical to algorithm 1, except that each new generation is initialised at random, rather than being the result of reproduction based on the novelty scores of the individual members of the previous generation. The point of this baseline is to determine the performance gain added by the NS. Models are added to the final ensemble based on the ensemble selection metric (section 3.2), as previously. Note that in this case we only use behavioural metrics when calculating a mean behavioural metric between a candidate model and current ensemble members, which is used as an ensemble selection metric. There is therefore no need to calculate behavioural metrics amongst the individuals of the population, neither to keep an archive of past individuals, as these metrics are not used to guide the NS as before.

The second baseline method is an ensemble generated with mechanisms that only *implicitly* promote diversity, as described in section 2, namely an ensemble of *architecturally diverse* neural networks, initialised with different random weights. This ensemble is created in the same fashion as the random initialisation that is used for both the first population in the NS and throughout the random search.

Algorithm 1 Ensemble evolution through NS

randomly initialise population pop
$archive \leftarrow \emptyset$
$ensemble \leftarrow \emptyset$
draw $train\_split$ and val_split from training set $\mathcal{D}$
set evolution iterations $epochs$
set training iterations $iters$
select behavioural div_metric$_{i,j}$ from section 3.2
ns_i is the fitness value (novelty score) of model m_i
for $epochs$ **do**
  for model $m_i \in pop$ **do**
    train$(m_i, train_split, iters)$ ▷ standard gradient descent optimisation
  end for
  $all \leftarrow pop \cup archive$
  for pair $m_i, m_j \in all \times all$ where $m_i \neq m_j$ **do**
    $dist_{i,j} \leftarrow$ div_metric(m_i, m_j) ▷ calculated on val_split
  end for
  for model $m_i \in pop$ **do**
    $ns_i \leftarrow \frac{1}{k} \sum_{m_n \in \mathcal{N}_i} dist_{i,n}$ ▷ where $\mathcal{N}_i$ is the set of k-nearest neighbours of m_i in all
  end for
  $archive \leftarrow archive \cup$ sample$(pop, sample_size)$
  adds new member to $ensemble$ ▷ the one with the highest ensemble selection metric
  $s \leftarrow$ select(pop) ▷ tournament selection w.r.t. novelty score
  $pop' \leftarrow \emptyset$
  for model $m_i \in s$ **do**
    $m'_i \leftarrow$ mutate(m_i) ▷ as described in the text
    add m'_i to pop'
  end for
  $pop \leftarrow pop'$
end for

The members of this ensemble are trained on different subsets of the data.

We do not compare our approach to any baseline that simply searches for accuracy as the importance of diversity is already established in the literature (e.g. [11]). We are instead interested in comparing methods which realise this notion of diversity in different ways.

4 EXPERIMENTS

Experiments have been conducted on three datasets — CIFAR-10, CIFAR-100, and SVHN. Note again that the goal of this paper is not to produce models competitive with state-of-the-art results. Our methods construct ensembles made of small and shallow neural networks trained for a limited number of epochs; in contrast, the methods in the literature which achieve the best results on these datasets require considerably greater computational effort and/or extensive fine-tuning of hyperparameters (e.g. [9, 10, 17, 19, 24, 42]), with some requiring at least dozens of GPU days (e.g. [40]). We are instead interested in studying the effects of diversity upon ensemble performance. Table 1 lists the common parameters whose values remain fixed throughout these experiments and Table 2 details the variables (hyperparameters) that are changed to generate different

Table 1: Common parameters fixed throughout the experiments per method(s)

Parameter	Method(s)	Value
Evolution iterations	NS and random	10
Training epochs	NS and random	40
Training epochs	Implicit ensemble	80
Stacking epochs	All	10
Data split (CIFAR)	All	40000 train / 10000 validation
Data split (SVHN)	All	43257 / 30000
Size of subsets	Implicit ensemble	20000
Batch size	All	128
Population size	NS and random	30
K (nearest neighbours)	NS	3
Size of tournament	NS	10
Archive sample size n_A	NS	5

Table 2: Range of hyperparameters

Parameter	Value[1]
Number of blocks	2:6
Size of the first convolution	4:16:4
Size of residual blocks	24:32:4
Dropout prob. in blocks	0.1:0.4:0.1

[1] Notation is $start$:end or $start$:end:$step$

neural network models and the ranges of their respective values. These hyperparameters have been chosen in order to produce small architectures that are easy to train in parallel. The NS algorithm and the random search baseline have been tested with different combinations of parameter settings, namely by instantiating the behavioural diversity metric used for calculating the novelty score of each individual and the ensemble selection diversity metric used for selecting a member of each generation to be added to the final ensemble. This is shown in Table 3. Note that, when running the baseline with entropy as the ensemble selection metric, the behavioural metric becomes irrelevant, as novelty scores are only ever calculated in the very first iteration in order to select the first ensemble member. For this reason, we consider only the combination with prop$_{i,j}$ when entropy is used with the random search baseline. All experiments have been carried out 10 times in order to assert statistical significance when comparing results. We next describe the hypotheses that this experimental work has put to the test.

Hypothesis 1 (Ensemble vs Single Best Individual). *Ensembles constructed from a set of individual classifiers chosen to maximise behavioural diversity outperform their single best member.*

This is tested by comparing the ensembles generated by NS and the random search baseline to their single best member. Recall that our procedures only search for diversity; accuracy is obtained by training each neural network with standard gradient descent. We aim to understand whether searching for diversity alone still

Table 3: Combinations of parameter settings

Selection metric	Mean behavioural metric		Entropy	
Behavioural metric	NS	Random	NS	Random
$\mathbf{prop}_{i,j}$	✓	✓	✓	✓
$\mathbf{cos\_dist}_{i,j}$	✓	✓	✓	✗
$\mathbf{dis}_{i,j}$	✓	✓	✓	✗

ensures that the selected models are accurate or if that might have a negative impact on ensemble performance.

Hypothesis 2 (Novelty Search vs Random Search). *Ensembles evolved by the NS algorithm (section 3.3) lead to higher test set accuracy than that of those found by the random search (section 3.5).*

We expect the NS algorithm to lead to higher-performing ensembles since, unlike the random search, it not only rewards diversity, but also actively searches for it.

Hypothesis 3 (Novelty Search vs Implicit Diversity Ensembles). *Ensembles evolved by the NS algorithm have higher test set accuracy than ensembles generated with standard methods that only implicitly promote diversity.*

In order to test this, we generate ensembles of *architecturally diverse* neural networks which are trained on *different subsets* of the data and are initialised with different random weights (section 3.5). We expect the experimental work to confirm that explicitly searching for diversity leads to better ensemble accuracy than relying on implicit mechanisms for generating it.

Hypothesis 4 (Error Diversity Metrics vs Generic Diversity Metrics). *$prop_{i,j}$ and $cos\_dist_{i,j}$ lead to better-performing ensembles than $dis_{i,j}$. The mean behavioural metric, when used in conjunction with the first two of these, leads to better results than entropy.*

We formulate this hypothesis because error diversity has been argued to be the most important type of diversity in ensemble learning [11]. Both the disagreement metric and entropy are more generic metrics of diversity than those we propose, which focus more closely on error instances.

5 RESULTS AND DISCUSSION

This section presents the results for the three datasets considered — CIFAR-10, CIFAR-100, and SVHN — and discusses whether or not they reject the hypotheses of section 4. Table 4 shows the accuracy results on all datasets for the three approaches we have considered: the NS algorithm, the random search, and the ensemble built with implicit diversity mechanisms. With minimal hyperparameter fine-tuning, these results are in line with figures reported in the literature for more specialised models of similar complexity (e.g. [8, 18, 26–28, 36]).

5.1 Hypothesis 1

Table 5 shows the results for all datasets of paired Mann-Whitney significance tests comparing the accuracy of the ensembles with that of their respective highest-performing individual. We observe that the ensembles typically outperform their single best individual in a statistically significant way. The only exceptions to this rule have been observed with the $dis_{i,j}$ behavioural metric from the literature, for which statistical significance is not observed on the CIFAR-10 dataset, and with the random search baseline when entropy is the selection metric, the only case where the single best individual outperforms the ensemble. These observations meet the expectations and justify the claim of Hypothesis 1 that the neural networks in the evolved ensembles are both diverse and accurate and that explicitly searching for diversity alone without rewarding accuracy, at least with the NS algorithm, does not impact negatively upon ensemble performance. On the contrary, given how the ensembles evolved with NS outperform those evolved with random search, as discussed below, we argue that it is precisely this explicit search for diversity that could lead to better ensemble accuracy.

5.2 Hypothesis 2

Table 6 shows the results of Mann-Whitney significance tests between the NS and the random search baseline. The ensembles evolved by the NS significantly outperform those evolved by the random search for all cases on CIFAR-100, as well as on CIFAR-10 except when the baseline $dis_{i,j}$ metric is the behavioural metric. The NS does not do better than the random search on SVHN. The accuracy results between the NS and the random search are very similar for the error diversity metrics $prop_{i,j}$ and $cos_{i,j}$, as well as when entropy is the selection metric, as per table 4. Interestingly, the random search considerably outperforms the NS when the disagreement metric $dis_{i,j}$ is used. As discussed later in this section, it is clear that the error diversity metrics that we propose lead to better results than the disagreement metric. The reason why the NS with these error diversity metrics does not outperform random search on SVHN requires further investigation but is likely a characteristic of the problem domain. SVHN is a simpler dataset and therefore finding the best-performing networks within the ranges defined in Table 2 is easier and a random search could prove the better strategy over NS. It is possible that the error diversity metrics lead the NS to trade accuracy for diversity on this dataset. In other words, if most neural network models are accurate but with similar behaviour, forcing the search to keep finding more behaviourally diverse models could result in less accurate learners, impacting ensemble performance negatively; this appears to be the case on SVHN, especially with the disagreement metric commonly used in the literature. On the hardest dataset of the three, CIFAR-100, the NS always outperforms the random search and the difference in accuracy that results from each is largest, as per Table 4; on the second hardest, CIFAR-10, the NS is only outperformed by the random search in combination with the underperforming disagreement metric. It is thus possible that the NS is more useful on harder problems, upon which the search must be guided by some non-random criterion. These observations justify the claim we make in Hypothesis 2 that, at least in some cases, explicitly searching for diversity, which the NS algorithm does, leads to better accuracy than simply rewarding it, as in the random search. However, there remain open questions regarding the conditions under which this can be observed, namely those pertaining to the problem domain and to the diversity metrics which define the search space of the NS. Further investigation into the trade-off between diversity and accuracy is required as well.

Table 4: Accuracy results for the NS algorithm and the two baseline methods. Median values over 10 runs

Dataset		Novelty Search				Random Search				Implicit
		Mean b. d.		Entropy		Mean b. d.		Entropy		
		Ensemble	Single best	Ensemble	Single best	Ensemble	Single best	Ensemble	Single best	
CIFAR-10	**prop**$_{i,j}$	83.51%	79.155%	81.46%	77.965%	80.54%	77.285%	73.885%	75.495%	76.315%
	cos_dist$_{i,j}$	83.29%	78.845%	81.31%	77.87%	80.61%	77.325%	✗	✗	
	dis$_{i,j}$	70.30%	71.135%	67.48%	64.845%	77.13%	77.205%	✗	✗	
CIFAR-100	**prop**$_{i,j}$	45.42%	41.405%	43.68%	40.855%	40.895%	38.855%	35.245%	37.54%	35.99%
	cos_dist$_{i,j}$	44.695%	41.065%	43.045%	40.34%	41.305%	39.155%	✗	✗	
	dis$_{i,j}$	43.53%	40.03%	41.875%	39.655%	39.875%	38.33%	✗	✗	
SVHN	**prop**$_{i,j}$	91.435%	88.805%	87.48%	85.035%	91.77%	89.72%	85.555%	84.165%	90.65%
	cos_dist$_{i,j}$	91.285%	88.555%	87.955%	84.82%	91.96%	89.395%	✗	✗	
	dis$_{i,j}$	86.19%	84.345%	81.505%	74.675%	91.01%	89.035%	✗	✗	

Table 5: Ensemble vs single best individual. Best case shown when statistical significance at the 1% level is observed

Method	Metrics		Best CIFAR-10	Best CIFAR-100	Best SVHN
Novelty search	prop$_{i,j}$	Mean b. d.	Ensemble	Ensemble	Ensemble
		Entropy	Ensemble	Ensemble	Ensemble
	cos$_{i,j}$	Mean b. d.	Ensemble	Ensemble	Ensemble
		Entropy	Ensemble	Ensemble	Ensemble
	dis$_{i,j}$	Mean b. d.	N/a	Ensemble	Ensemble
		Entropy	N/a	Ensemble	Ensemble
Random search	prop$_{i,j}$	Mean b. d.	Ensemble	Ensemble	Ensemble
		Entropy	Single best	Single best	N/a
	cos$_{i,j}$	Mean b. d.	Ensemble	Ensemble	Ensemble
	dis$_{i,j}$	Mean b. d.	N/a	Ensemble	Ensemble

Table 6: NS vs random search. Best case shown when statistical significance at the 1% level is observed

Metrics		Best CIFAR-10	Best CIFAR-100	Best SVHN
prop$_{i,j}$	Mean b. d.	Novelty search	Novelty search	Random search
	Entropy	Novelty search	Novelty search	N/a
cos$_{i,j}$	Mean b. d.	Novelty search	Novelty search	Random search
dis$_{i,j}$	Mean b. d.	Random search	Novelty search	Random search

5.3 Hypothesis 3

Regarding the ensembles which are generated with only an implicit definition of diversity (last column of Table 4), significance tests show that, on CIFAR-100, both the NS and the random search significantly outperform this baseline in all cases except when random search is used with entropy; on CIFAR-10, no statistical significance is observed when the random search uses the disagreement metric dis$_{i,j}$, but the baseline significantly outperforms both search methods in all other cases where this metric is used and also when compared to the random search with entropy; on SVHN, the baseline significantly outperforms the search methods in more cases, namely all those using entropy and when the NS uses the disagreement metric, with no statistically significant difference observed when the random search uses this same metric. We can therefore attest the claim of Hypothesis 3 by observing that the methods tend to outperform the implicit diversity baseline when the error diversity metrics that we propose in section 3.2 are used; using the disagreement and entropy metrics often leads to worse performance than this baseline, likely because, as discussed before, they lead the search to excessively trade the accuracy of its individual learners for diversity of behaviours. We also note that the performance of this baseline on SVHN is close to that of the best cases produced by the NS; given its smaller complexity w.r.t. the other search methods, this means that it could be advantageous on easier datasets such as SVHN.

5.4 Hypothesis 4

Table 7 shows the results of significance tests between the accuracy results produced by different diversity metrics. We can observe that the behavioural metrics prop$_{i,j}$ and cos$_{i,j}$ perform similarly well, with no statistically significant difference found between the results produced by these two metrics in any parameter setting. They both significantly outperform the dis$_{i,j}$ metric in the NS with the mean behavioural diversity metric as the ensemble selection metric; however, both when entropy is the selection metric and in the random search, this is not observed on all datasets. Settings that

Table 7: Different diversity metrics. Best case shown when statistical significance at the 1% level is observed

Method	Fixed metric	Varied metrics		Best CIFAR-10	Best CIFAR-100	Best SVHN
Novelty search	Mean behavioural diversity	$\text{prop}_{i,j}$	$\text{cos}_{i,j}$	N/a	N/a	N/a
		$\text{prop}_{i,j}$	$\text{dis}_{i,j}$	$\text{prop}_{i,j}$	$\text{prop}_{i,j}$	$\text{prop}_{i,j}$
		$\text{cos}_{i,j}$	$\text{dis}_{i,j}$	$\text{cos}_{i,j}$	$\text{cos}_{i,j}$	$\text{cos}_{i,j}$
	Entropy	$\text{prop}_{i,j}$	$\text{cos}_{i,j}$	N/a	N/a	N/a
		$\text{prop}_{i,j}$	$\text{dis}_{i,j}$	$\text{prop}_{i,j}$	N/a	$\text{prop}_{i,j}$
		$\text{cos}_{i,j}$	$\text{dis}_{i,j}$	$\text{cos}_{i,j}$	N/a	$\text{cos}_{i,j}$
	$\text{prop}_{i,j}$	Mean b. d.	Entropy	Mean b. d.	Mean b. d.	Mean b. d.
	$\text{cos}_{i,j}$	Mean b. d.	Entropy	Mean b. d.	Mean b. d.	Mean b. d.
	$\text{dis}_{i,j}$	Mean b. d.	Entropy	Mean b. d.	N/a	Mean b. d.
Random search	Mean behavioural diversity	$\text{prop}_{i,j}$	$\text{cos}_{i,j}$	N/a	N/a	N/a
		$\text{prop}_{i,j}$	$\text{dis}_{i,j}$	$\text{prop}_{i,j}$	N/a	$\text{prop}_{i,j}$
		$\text{cos}_{i,j}$	$\text{dis}_{i,j}$	$\text{cos}_{i,j}$	N/a	$\text{cos}_{i,j}$
	$\text{prop}_{i,j}$	Mean b. d.	Entropy	Mean b. d.	Mean b. d.	Mean b. d.

use the mean behavioural metric as an ensemble selection metric tend to statistically outperform those using entropy, for both the NS and the random search; the only case where this is not observed with statistical significance is on the CIFAR-100 dataset when the NS uses the $\text{dis}_{i,j}$ metric.

We note that the metrics we propose, $\text{prop}_{i,j}$ and $\text{cos}_{i,j}$, tend to lead to better results than the disagreement metric commonly found in the literature, $\text{dis}_{i,j}$. In addition, calculating the mean behavioural metric w.r.t. to the current ensemble members and using it as a selection metric tends to work better than using entropy, which as mentioned before might lead to worse ensemble performance than that of the single best individual. As we have mentioned before, the behavioural metrics we propose focus more closely on error instances, i.e. instances that at least one of the models has misclassified, unlike the common disagreement metric, which is calculated w.r.t. to all test instances. Additionally, entropy is also calculated over all instances in the test set and broadly measures the general divergence amongst the predictions made by the ensemble members. This justifies the claim we make in Hypothesis 4 about the diversity metrics which lead to better-performing ensembles: the error diversity metrics we propose seem to be correlated with better ensemble accuracy and are thus better suited to the task of searching for diversity. Further research into the implications of this finding is required to understand how it can be fully leveraged to evolve high-performing ensembles.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we propose an innovative NS algorithm, augmented with two novel behavioural metrics, which evolves ensembles by explicitly searching for behavioural diversity, unlike other methods found in the literature, which typically either rely on *implicit* mechanisms for promoting diversity or only reward it by including it as an objective while searching for models which represent trade-offs between diversity and accuracy. Our procedure not only rewards diversity, but rather actively searches for it by guiding the NS with a definition of novelty score which is based on how diverse each individual neural network is w.r.t. to the other individuals in the population. The accuracy of the individual learners is ensured by a standard gradient descent procedure, but it is not taken into account in the NS. We investigate three *behavioural diversity* metrics, two of which we propose ourselves, and two metrics for selecting individuals to be added to the final ensemble, one of which is also defined by us as the mean of behavioural diversity metrics calculated w.r.t. current ensemble members and the other is the entropy of candidate ensembles.

The results show that our approach succeeds at evolving an ensemble by explicitly searching for behavioural diversity, significantly outperforming, particularly on harder datasets, a random search baseline which merely rewards diversity and a baseline ensemble generated with implicit diversity. They also show that the ensembles almost always outperform their best individual learner, meaning that the method is able to generate and select diverse enough learners while maintaining accuracy. Of particular relevance is the observation that, amongst the diversity metrics we have considered, the error diversity metrics we propose lead to better results, i.e. they push the NS towards better areas of the search space. All these observations provide valuable insights into the problem of promoting diversity in ensembles of classifiers, suggesting not only that *explicit* methods such as the one we present here should be adopted on harder problems that implicit methods struggle to solve, but also that diversity metrics should focus directly and closely on the *errors* made by individual learners. As mentioned before, our implementation uses only small and shallow neural network models. Therefore, an open question that remains is whether the methods we present here can be scaled with more complex models so that the accuracy of the ensembles becomes competitive with the state of the art.

There are other open questions arising directly from the results here presented. Future research will further investigate the trade-off between diversity and ensemble accuracy and new ways to guide the search for high-performing ensembles with error diversity. It will also seek to improve the NS algorithm, both reducing the computational effort required to run it by using surrogate models such as the proposed by Siems et al. [37] — thus eliminating the need to train each individual model with gradient descent — and considering more sophisticated methods to generate new architectures, such as NEAT [39].

REFERENCES

[1] Alejandro Baldominos, Yago Saez, and Pedro Isasi. 2019. Hybridizing evolutionary computation and deep neural networks: an approach to handwriting recognition using committees and transfer learning. *Complexity* 2019 (2019).

[2] Urvesh Bhowan, Mark Johnston, Mengjie Zhang, and Xin Yao. 2012. Evolving diverse ensembles using genetic programming for classification with unbalanced data. *IEEE Transactions on Evolutionary Computation* 17, 3 (2012), 368–386.

[3] Urvesh Bhowan, Mark Johnston, Mengjie Zhang, and Xin Yao. 2013. Reusing genetic programming for ensemble selection in classification of unbalanced data. *IEEE Transactions on Evolutionary Computation* 18, 6 (2013), 893–908.

[4] Yijun Bian and Huanhuan Chen. 2019. When does Diversity Help Generalization in Classification Ensembles? (2019). arXiv:1910.13631

[5] Gavin Brown, Jeremy L. Wyatt, and Peter Tiňo. 2005. Managing diversity in regression ensembles. (2005).

[6] Rui P. Cardoso, Emma Hart, David Burth Kurka, and Jeremy Pitt. 2021. WILDA: Wide Learning of Diverse Architectures for Classification of Large Datasets. In *Applications of Evolutionary Computation - 24th International Conference, EvoApplications 2021, Held as Part of EvoStar 2021, Virtual Event, April 7-9, 2021, Proceedings (Lecture Notes in Computer Science)*, Pedro A. Castillo and Juan Luis Jiménez Laredo (Eds.), Vol. 12694. Springer, 649–664. https://doi.org/10.1007/978-3-030-72699-7_41

[7] Rui P Cardoso, Emma Hart, and Jeremy V Pitt. 2020. Diversity-Driven Wide Learning for Training Distributed Classification Models. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion (GECCO '20)*. Association for Computing Machinery, New York, NY, USA, 119–120. https://doi.org/10.1145/3377929.3390012

[8] Tsung-Han Chan, Kui Jia, Shenghua Gao, Jiwen Lu, Zinan Zeng, and Yi Ma. 2015. PCANet: A Simple Deep Learning Baseline for Image Classification? *IEEE Transactions on Image Processing* 24, 12 (Dec 2015), 5017–5032. https://doi.org/10.1109/tip.2015.2475625

[9] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. 2019. AutoAugment: Learning Augmentation Policies from Data. (2019). arXiv:cs.CV/1805.09501

[10] Terrance DeVries and Graham W. Taylor. 2017. Improved Regularization of Convolutional Neural Networks with Cutout. (2017). arXiv:cs.CV/1708.04552

[11] Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*. Springer, 1–15.

[12] Dario Floreano, Peter Dürr, and Claudio Mattiussi. 2008. Neuroevolution: From architectures to learning. (2008). https://doi.org/10.1007/s12065-007-0002-4

[13] Nicolás García-Pedrajas, César Hervás-Martínez, and Domingo Ortiz-Boyer. 2005. Cooperative coevolution of artificial neural network ensembles for pattern classification. *IEEE transactions on evolutionary computation* 9, 3 (2005), 271–302.

[14] Jorge Gomes, Pedro Mariano, and Anders Lyhne Christensen. 2015. Devising effective novelty search algorithms: A comprehensive empirical study. In *GECCO 2015 - Proceedings of the 2015 Genetic and Evolutionary Computation Conference*. https://doi.org/10.1145/2739480.2754736

[15] S Gu, R Cheng, and Y Jin. 2015. Multi-objective ensemble generation. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5, 5 (2015), 234–245. https://doi.org/10.1002/widm.1158

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[17] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Mia Xu Chen, Dehao Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V. Le, Yonghui Wu, and Zhifeng Chen. 2019. GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism. (2019). arXiv:cs.CV/1811.06965

[18] Yangqing Jia, Chang Huang, and Trevor Darrell. 2012. Beyond spatial pyramids: Receptive field learning for pooled image features. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2012.6248076

[19] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. 2020. Big Transfer (BiT): General Visual Representation Learning. (2020). arXiv:cs.CV/1912.11370

[20] Alex Krizhevsky. 2009. Learning Multiple Layers of Features from Tiny Images. *... Science Department, University of Toronto, Tech. ...* (2009). https://doi.org/10.1.1.222.9220 arXiv:arXiv:1011.1669v3

[21] Ludmila I. Kuncheva and Christopher J. Whitaker. 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning* (2003). https://doi.org/10.1023/A:1022859003006

[22] Joel Lehman and Kenneth O. Stanley. 2011. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation* (2011). https://doi.org/10.1162/EVCO_a_00025

[23] Joel Lehman and Kenneth O Stanley. 2011. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 211–218.

[24] Senwei Liang, Yuehaw Khoo, and Haizhao Yang. 2020. Drop-Activation: Implicit Parameter Reduction and Harmonic Regularization. (2020). arXiv:cs.LG/1811.05850

[25] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan Yuille, and Li Fei-Fei. 2019. Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation. (2019). arXiv:cs.CV/1901.02985

[26] Julien Mairal, Piotr Koniusz, Zaid Harchaoui, and Cordelia Schmid. 2014. Convolutional Kernel Networks. (2014). arXiv:cs.CV/1406.3332

[27] Mateusz Malinowski and Mario Fritz. 2014. Learning Smooth Pooling Regions for Visual Recognition. https://doi.org/10.5244/c.27.118

[28] Mark D. McDonnell and Tony Vladusich. 2015. Enhanced Image Classification With a Fast-Learning Shallow Convolutional Neural Network. (2015). arXiv:cs.NE/1503.04596

[29] Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. (2015). arXiv:cs.AI/1504.04909

[30] Kaustuv Nag and Nikhil R Pal. 2015. A multiobjective genetic programming-based ensemble for simultaneous feature selection and classification. *IEEE transactions on cybernetics* 46, 2 (2015), 499–510.

[31] Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve Restricted Boltzmann machines. In *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*.

[32] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. 2011. Reading Digits in Natural Images with Unsupervised Feature Learning. (2011).

[33] R Pasti, L N De Castro, G P Coelho, and F J Von Zuben. 2010. Neural network ensembles: Immune-inspired approaches to the diversity of components. *Natural Computing* 9, 3 (2010), 625–653. https://doi.org/10.1007/s11047-009-9124-1

[34] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary Devito Facebook, A I Research, Zeming Lin, Alban Desmaison, Luca Antiga, Orobix Srl, and Adam Lerer. 2019. Automatic differentiation in PyTorch. In *Advances in Neural Information Processing Systems 32*.

[35] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. 2019. Regularized Evolution for Image Classifier Architecture Search. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (Jul. 2019), 4780–4789. https://doi.org/10.1609/aaai.v33i01.33014780

[36] Pierre Sermanet, Soumith Chintala, and Yann LeCun. 2012. Convolutional Neural Networks Applied to House Numbers Digit Classification. (2012). arXiv:cs.CV/1204.3968

[37] Julien Siems, Lucas Zimmer, Arber Zela, Jovita Lukasik, Margret Keuper, and Frank Hutter. 2020. NAS-Bench-301 and the case for surrogate benchmarks for neural architecture search. (2020). arXiv:2008.09777

[38] Kenneth O. Stanley, Jeff Clune, Joel Lehman, and Risto Miikkulainen. 2019. Designing neural networks through neuroevolution. (2019). https://doi.org/10.1038/s42256-018-0006-z

[39] Kenneth O. Stanley and Risto Miikkulainen. 2002. Evolving Neural Networks Through Augmenting Topologies. *Evolutionary Computation* 10, 2 (2002), 99–127. http://nn.cs.utexas.edu/?stanley:ec02

[40] Y. Sun, B. Xue, M. Zhang, G. G. Yen, and J. Lv. 2020. Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification. *IEEE Transactions on Cybernetics* 50, 9 (2020), 3840–3854. https://doi.org/10.1109/TCYB.2020.2983860

[41] Paul A. Szerlip, Gregory Morse, Justin K. Pugh, and Kenneth O. Stanley. 2014. Unsupervised Feature Learning through Divergent Discriminative Feature Accumulation. (2014). arXiv:cs.NE/1406.1833

[42] Mingxing Tan and Quoc V. Le. 2020. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. (2020). arXiv:cs.LG/1905.11946

[43] Rick Van Krevelen. 2005. *Error Diversity in Classification Ensembles*. Ph.D. Dissertation. https://doi.org/10.13140/RG.2.1.3809.8964

[44] David H. Wolpert. 1992. Stacked generalization. *Neural Networks* (1992). https://doi.org/10.1016/S0893-6080(05)80023-1

[45] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide Residual Networks. (2016). arXiv:cs.CV/1605.07146

[46] Yifeng Zhang and Siddhartha Bhattacharyya. 2004. Genetic programming in classifying large-scale data: an ensemble method. *Information Sciences* 163, 1-3 (2004), 85–101.