Creating Classifier Ensembles through Meta-heuristic Algorithms for Aerial Scene Classification

Álvaro R. Ferreira Jr.*, Gustavo H. de Rosa‡, João P. Papa‡, Gustavo Carneiro†, Fabio A. Faria*†
*Institute of Science and Technology - Universidade Federal de São Paulo, São José dos Campos, Brazil
aalvin10@gmail.com, ffaria@unifesp.br
†Australian Institute for Machine Learning - The University of Adelaide, Adelaide, Australia
gustavo.carneiro@adelaide.edu.au
‡ Department of Computing - São Paulo State University, Bauru, Brazil
{gustavo.rosa,joao.papa}@unesp.br

***Abstract*—Convolutional Neural Networks (CNN) have been being widely employed to solve the challenging remote sensing task of aerial scene classification. Nevertheless, it is not straightforward to find single CNN models that can solve all aerial scene classification tasks, allowing the development of a better alternative, which is to fuse CNN-based classifiers into an ensemble. However, an appropriate choice of the classifiers that will belong to the ensemble is a critical factor, as it is unfeasible to employ all the possible classifiers in the literature. Therefore, this work proposes a novel framework based on meta-heuristic optimization for creating optimized ensembles in the context of aerial scene classification. The experimental results were performed across nine meta-heuristic algorithms and three aerial scene literature datasets, being compared in terms of effectiveness (accuracy), efficiency (execution time), and behavioral performance in different scenarios. Our results suggest that the Univariate Marginal Distribution Algorithm shows more effective and efficient results than other commonly used meta-heuristic algorithms, such as Genetic Programming and Particle Swarm Optimization.**

I. Introduction

Remote sensing advancements increased the abstraction level from pixels into objects and scenes [1], becoming classification tasks more difficult to be learned by a classification model. Traditional pixel-based approaches commonly use spectral responses, such as RGB and NDVI channels, to define their input information. Object-based methods, such as the Geographic Object-based Image Analysis (GEOBIA), aim at identifying specific objects of interest, e.g., streets, lakes, and buildings, instead of assigning labels to every pixel in the image. Furthermore, scene-based approaches use the semantics of the entire image, where every pixel of the image is classified into the same label [2]. A particularly interesting remote sensing problem is aerial scene classification, which is challenging due to the high intra-class variability, and different scales and orientations of objects [1]. Aerial scene classification has crucial applications in military and civil fields, such as natural disaster monitoring, traffic supervision, and weapon guidance [3].

Methods that solve aerial scene classification can be divided into three categories: (i) low-level, (ii) mid-level, and (iii) deep-level [3]. Low-level methods are local descriptors-based approaches, such as Scale Invariant Feature Transform (SIFT) [4], Color Histogram (CH) [5], and Local Binary Patterns (LBP) [6]. Mid-level methods use local descriptor encodings in an attempt to create more semantic representations, such as Bag-of-Visual-Words (BoVW) [7]. Finally, deep-level approaches use deep learning architectures, e.g., VGG16 [8] and GoogLeNet [9]), to extract more discriminative visual-features and high-level semantic information from the images.

Convolutional Neural Networks (CNNs) have been applied in computer vision and machine learning tasks throughout the last years, primarily due to their capacity to extract proper high-level semantic information. One can observe that they have been successfully applied in distinct applications, such as action [10] and biometric recognition [11], as well as medical image analysis [12], to cite a few. Additionally, several research competitions (e.g., IARPA and GRSS) fostered the development of such techniques as they are capable of robustly classifying image datasets [13], [14]. Despite the recent success reached by CNN architectures, there are some real-world applications, such as biometrics, spoofing, noisy and adversarial scenarios [15], in which they still do not perform well. In such tasks, one might observe that the use of CNN ensembles might create more effective models that combine complementary pieces of information.

In the FMOW2018 (Functional Map of the World 2018) challenge[1] organized by IARPA, all of the top three methods used CNN ensembles as solutions [13]. The first-place proposed a composition of twelve CNN classifiers based on generic Dual-Path Networks (DPNs) [16], which were pre-trained with a variety of hyperparameters, scaling methods, and augmentations. The second-place created an ensemble of ResNet [17] and ResNeXt [18] models, along with a weights' initialization derived from fine-tuning previous chal-

[1]https://www.iarpa.gov/challenges/fmow.html

lenge data. Finally, the third-place proposed the Hydra framework, which creates CNN ensembles based on ResNet and DenseNet [19] architectures. Essentially, Hydra is initialized with naïve optimized CNNs, serving as its body. Moreover, its weights are fine-tuned with augmentation strategies multiple times, building a CNN ensemble that represents the Hydra's heads [20]. Similar to the FMOW2018 challenge, the IEEE GRSS Data Fusion Contest [14] also had two of its best works as CNN-based ensembles. The best work proposed a data fusion methodology based on multiple fully-convoluted networks and a post-classification procedure [21]. Regarding the second-place, the authors combined deep and shallow neural networks with ad-hoc spectral detectors [22].

Notwithstanding, it is possible to observe that the literature lacks strategies to select the best classifiers to compose the ensemble. Usually, those solutions adopt ad-hoc classifier selection based on their performance in the target application. In this context, this work proposes a novel framework based on meta-heuristic optimization for creating optimized ensembles in the context of aerial scene classification. The experimental results were performed across ten meta-heuristic algorithms and three aerial scene literature datasets, being compared in terms of effectiveness (accuracy), efficiency (execution time), and behavioral performance in an adverse scenario.

The remainder of this paper is organized as follows. Sections II and III present the theoretical background related to meta-heuristic optimization and ensemble learning, respectively. Section IV discusses the methodology adopted in this work, while Section V presents the experimental results. Finally, Section VI states conclusions and future works[2].

II. Meta-Heuristic Optimization

Optimization aims at selecting a solution among a set of possible candidates that best fits an objective function. It is possible to find in the literature several optimization methods, e.g., grid-search and gradient-based methods. Nonetheless, these methods are burdened with massive computational loads as they exhaustively search throughout the space or use derivative-based properties, not being feasible in more complex problems, such as exponential and NP-complete ones.

A recent attempt to overcome such problems is to employ a meta-heuristic-based optimization. Meta-heuristic algorithms are biologically-inspired techniques that attempt to mimic an intelligence behavior, often observed in nature or groups of animals and humans. Their main goal is to combine exploration and exploitation mechanisms to achieve sub-optimal solutions with a low computational burden.

In this work, we employ a vast number of state-of-the-art meta-heuristic techniques, ranging from evolutionary- to swarm-based algorithms, such as Artificial Bee Colony (ABC) [23], Bat Algorithm (BA) [24], Black Hole Algorithm (BHA) [25], Cuckoo Search (CS) [26], Firefly Algorithm (FA) [27], Flower Pollination Algorithm (FPA) [28], Genetic Programming (GP) [29], Particle Swarm Optimization (PSO) [30], and Univariate Marginal Distribution Algorithm (UDMA) [31]. Additionally, for the sake of space, we only describe one algorithm per taxonomy in the next subsections, i.e., GP (evolutionary), PSO (swarm), and UDMA (evolutionary, but with a slightly distinct approach).

[2]The source code is available online at https://github.com/gugarosa/evolutionary_ensembles

A. Genetic Programming

Genetic Programming is an evolutionary-based algorithm introduced by Koza [29] and based upon Darwin's Theory of Evolution, where its main idea is to use bio-inspired operators to create promising solutions and achieve an objective. Even though GP resembles the standard Genetic Algorithm (GA), there are fundamental differences between them. A typical solution of GP uses a tree-based structure composed of terminal and function nodes, such as the one illustrated in Figure 1. Necessarily, the terminal nodes represent constants or decision variables, while the function nodes are mathematical operators applied over the terminal nodes. During the evolution process, several procedures, such as selection, reproduction, mutation, and crossover, are employed in an attempt to produce better-fitted individuals.

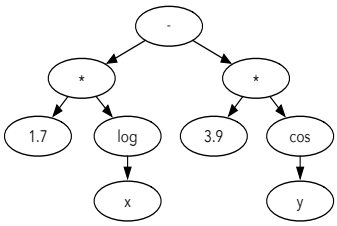

Fig. 1: A typical GP solution that represents the mathematical expression $1.7log(x) - 3.9cos(y)$.

During each iteration, the current best individuals are selected and reproduced, creating a possible better individual and keeping the most-fitted ones along with the future generations. Afterward, mutation and crossover operators are applied over the population to provide a variability factor. In other words, the mutation operator randomly modifies an allele (node or branch) from a specific tree, while the crossover operator swaps nodes or branches between two trees. Finally, the procedure is conducted until a convergence criterion is satisfied or the number of maximum iterations is reached, and the best individual (best decision variables) is harvested from the trees.

Let $\mathcal{T} = (t_1, t_2, \ldots, t_M)$ be a set of trees that compose the population. Additionally, let $t_i = h(e_i)$ be an individual tree, where $h : E \rightarrow \Re^N$ is the function that evaluates the mathematical expression e_i that belongs to the expression space E and returns an N-positional vector which contains the value of the decision variables from individual i. Furthermore, e_i is a mathematical expression obtained after traversing the tree i in a post-order course. During the evolution process, the population is evaluated and sorted in an ascending way, where the best individuals are the ones with the lowest fitness function value. As mentioned before, the first operator selects and reproduces individuals with a probability of p_r, while the probability p_m and p_c stands for the rate of mutation and

crossover, respectively. After employing the genetic operators, the population is again evaluated and sorted.

B. Particle Swarm Optimization

Particle Swarm Optimization is a swarm-based optimization algorithm inspired by social behavior stimulus [30]. Social behavior-based learning allows particles (solutions) to combine details from previous and current positions with other particles' positions, yielding in a better search space exploration. Analogously, one can observe this process as the social interaction of birds looking for food or humans looking for a common objective.

Let $\mathcal{S} = (s_1, s_2, \ldots, s_M)$ be the particles that constitute the swarm, such that $s_i = (\boldsymbol{\psi}_i, \boldsymbol{\rho}_i)$, where $\boldsymbol{\psi}_i \in \Re^N$ and $\boldsymbol{\rho}_i \in \Re^N$ are the position and velocity of particle i, respectively. Additionally, let $\hat{\boldsymbol{\psi}}_i$ be particle's i best local position, and $\boldsymbol{g}$ be the best overall solution (global). The algorithm starts by initializing the position and velocity of each particle with random values. During each iteration t, every particle has its position evaluated concerning a fitness function. After evaluating all particles, the global minimum is updated with the particle that achieved the best position (minimum fitness value) in the swarm. This process is conducted until a convergence criterion is satisfied or the number of maximum iterations is reached. Finally, one can observe Equations 1 and 2 as the velocity and position update of particle i at time step t, respectively:

$$\boldsymbol{\rho}_i^{t+1} = w\boldsymbol{\rho}_i^t + c_1 r_1(\hat{\boldsymbol{\psi}}_i - \boldsymbol{\psi}_i^t) + c_2 r_2(\boldsymbol{g} - \boldsymbol{\psi}_i^t) \tag{1}$$

and

$$\boldsymbol{\psi}_i^{t+1} = \boldsymbol{\psi}_i^t + \boldsymbol{\rho}_i^{t+1}, \tag{2}$$

where w is the inertia weight that handles interaction between particles, while $r_1, r_2 \in [0, 1]$ give stochastic traits to the PSO algorithm. Furthermore, variables c_1 and c_2 are cognitive and social constants that conduct the particles onto the search space.

C. Univariate Marginal Distribution Algorithm

Univariate Marginal Distribution Algorithm has been introduced by Mühlenbein et al. [31] and is considered one of the most straightforward Estimation of Distribution Algorithms [32] (EDAs) that is capable of assuming independence between decision variables. Let $f : \{0, 1\}^n \rightarrow \mathbb{R}$ be a pseudo-boolean function, where an individual is a bit-string (0 or 1). In order to optimize f, the algorithm undergoes an iterative process, as follows:

- Independently and identically sampling a population of λ individuals (solutions) from the current probabilistic model;
- Evaluating solutions;
- Updating the model from the best-fitted solutions μ.

Each sampling and update cycle is called a generation (iteration). In each iteration, the current probabilistic model is represented as a vector $\boldsymbol{p}_t = (p_t(1), ..., p_t(n)) \in [0, 1]^n$, where each component (marginal) is depicted by $p_t(i) \in [0, 1]$. Additionally, $p_t \in \mathbb{N}$ is the probability of sampling the number one (1) in the $i-th$ bit position of an individual at iteration t. Finally, each individual $\boldsymbol{x} = (x_1, ..., x_n) \in \{0, 1\}^n$ is sampled from the joint probability, as follows:

$$Pr(\boldsymbol{x}|\boldsymbol{p}^t) = \prod_{i=1}^{n} p_t(i)^{x_i}(1 - p_t(i))^{(1-x_i)}. \tag{3}$$

One must observe that extreme probabilities, i.e., zero and one, should be avoided for each marginal point, as this impacts in preventing the $i - th$ bit from having its value changed and, consequently, ignoring some regions of the search space. To overcome such problem, all $p_{t+1}(i)$ margins are restricted within the closed range $[1/n, 1-1/n]$, whereas the values $1/n$ and $1 - 1/n$ are called lower and upper bounds, respectively. Such an approach is denoted as margin-based UMDA (M-UMDA).

Furthermore, one can observe that the search procedure is conducted by sampling models of the variables and not by randomly crossing and mutating them. This approach leads to a better optimization process as it reduces the generation of poor-quality solutions, thus improving the possibility of generating a better-quality solution.

III. Ensemble Learning

Ensemble learning consists of a combination of classifiers which focus on solving a unique problem. Their primary difference from single classifiers is the use of several combined classifiers, allowing them to accomplish more effective learning [33]. An ensemble of classifiers is composed of several base classifiers, such as decision trees, support vector machines, and neural networks, among others. Furthermore, when base classifiers are combined, they create a unique and stronger model. It is known that the generalization ability of an ensemble is usually higher than base classifiers, due to the increase in the diversity of features extracted and decisions made [34].

A critical distinction between ensembles concerns their classification, which is divided into two categories: (i) homogeneous, if the same base classifiers compose the ensemble, and (ii) heterogeneous, if different base classifiers compose the ensemble. In this work, we will use a homogeneous ensemble composed of several descriptors and CNNs. Additionally, we will use a boolean-based strategy, as described by Section III-A.

A. Boolean Voting-based Ensemble

Despite our present work focusing on CNNs, it is important to highlight that such a procedure applies to any neural network-based classifier, for instance, traditional Multilayer Perceptron (MLP) or even Recurrent Neural Networks (RNN).

Broadly speaking, given a collection of K classifiers, let $T_i \in \mathbb{N}^{\mathcal{C} \times K}$ denote the target of a given sample $\boldsymbol{x}_i$ belonging to each of the $\mathcal{C}$ possible classes according to each of the k models. More specifically, this matrix is the concatenation of

the outputs of each classifier. Finally, the boolean voting-based ensemble combines all classifiers as follows:

$$F(T_i, \boldsymbol{b}) = \sum_{k=1}^{K} b_k \cdot T_i, \tag{4}$$

where $\boldsymbol{b}$, a binary vector of K positions ($\boldsymbol{b} \in [0,1]^K$), contains the importance degree (weight) of each base classifier in the ensemble. Further, let $\boldsymbol{q}_i = F(T_i, \boldsymbol{b}) \in \mathbb{R}^{\mathcal{C}}$ be the unnormalized score of $\boldsymbol{x}_i$ belonging to each of the possible classes ($\mathcal{C}$) according to the ensemble, then the the predicted label $\hat{y}_i$ is computed as:

$$\hat{y}_i = \underset{\mathcal{C}}{\operatorname{argmax}}\, \boldsymbol{q}_i. \tag{5}$$

IV. Experimental Methodology

This section presents the methodology employed in this work. Essentially, it describes how to construct the base classifiers, how to define the evolutionary framework for the classifier-based ensembles, as well as the employed datasets and the experimental setup.

A. Constructing Base Classifiers

Initially, in order to better describe the employed datasets, we opted to use two feature extraction procedures: (i) global image descriptors and (ii) deep learning architectures. Regarding the global image descriptors, we used color- and text-based visual properties representations[3], such as Border/Interior Pixel Classification (BIC) [37], Color Coherence Vector (CCV) [38], Global Color Histogram (GCH) [5], Quantized Compound Change Histogram (QCCH) [39], and Local Activity Spectrum (LAS) [40]. Regarding the deep learning architectures, we used five Convolutional Neural Networks architectures with an ImageNet [41] pre-trained transfer learning approach, such as VGG [8] (VGG16 and VGG19), GoogLeNet (Inception-V3 [42], Xception [43]), and ResNet (ResNet-50 [17]).

Finally, we have used seven learning techniques with their default hyper-parameters to create the base classifiers[4]: Decision Tree (DT) [44], Naïve Bayes (NB) [45], k-Nearest Neighbors (kNN) [46] using $k = \{1, 3, 5, 7\}$, and Support Vector Machine [47] with a polynomial kernel. It is important to recall that a base classifier is composed of a tuple (feature extraction and learning techniques). Thus, 70 base classifiers (7×10 different tuples) are available for further experimentation.

B. Evolutionary Framework for Classifier Ensembles

Figure 2 illustrates the experimental pipeline used in this work. The idea is to select the most suitable base classifiers using a meta-heuristic optimization algorithm and further compose a boolean-based ensemble.

Initially, given a specific dataset, the feature extraction techniques are applied to encode visual properties into (e.g., color, texture, shape) feature vectors. Furthermore, the dataset is split into three sets (training, validation, and testing) using a 5-fold cross-validation procedure. The training set is used as input by the machine learning techniques to create base classifiers, while the validation sets are used to conduct the meta-heuristic optimization. Finally, the proposed approach is validated with the testing sets.

Concerning the meta-heuristic optimization, an initial population of classifier ensembles (individuals) is randomly built. During each iteration, the population of individuals are evaluated through a fitness function (accuracy over the validation set) and then, each particular meta-heuristic strategy is used to update the individuals for the next generation. The evolution process is conducted until a stopping criterion is satisfied or the maximum number of iterations is reached. Finally, the best individual, which represents the best ensemble, is selected and evaluated on the testing set.

C. Datasets

The proposed approach is validated on three literature remote sensing scene classification datasets: WHU-RS19 [48], RSSCN7 [49], and UC Merced Land Use [50]. Table I describes with more details each dataset.

TABLE I: Description of each dataset used in this work.

Dataset	Classes	Images per Class (Total)	Dimension	Spatial Resolution
RSSCN7	7	400 (2,800)	400×400	–
UCMerced	21	100 (2,100)	256×256	0.3m
WHU-RS19	19	50–61 (1,005)	600×600	up to 0.5m

All datasets have been evaluated within a 5-fold cross-validation procedure, being split into three distinct sets (training, validation, and testing). Figure 3 depicts some aerial scenes' examples extracted from the WHU-RS19 dataset.

D. Experimental Setup

Table II presents the meta-heuristic algorithms hyper-parameters used in this paper, except for the BHA, which is a parameterless approach. Regarding the UMDA algorithm, stagnation means the stopping criteria in case of non-evolution, i.e., the amount of generation with no progress of the best individual in the population.

V. Experiments and Discussion

This section presents three sets of experiments to validate the proposed framework on three well-known aerial scene datasets (WHU-RS19, RSSCN7, and UCMerced). First, a classification analysis in three different scenarios has been conducted with three different experiments: (1) ensembles built with only classifiers based on global image descriptors (Global); (2) ensembles built with only classifiers based on deep learning features (CNN); and (3) ensembles built with any type of classifiers (Global+CNN = ALL). The last one is called adverse scenario since the worst classifiers (Global) are brought together with the most accurate classifiers (CNN) in order to fool the optimization algorithms. Second, we conducted an analysis of the number of base classifiers present in

[3]The image descriptors used in this work are based on extensive experiments performed in [35], [36].

[4]The implementations are available on WEKA: http://www.cs.waikato.ac.nz/~ml/weka.

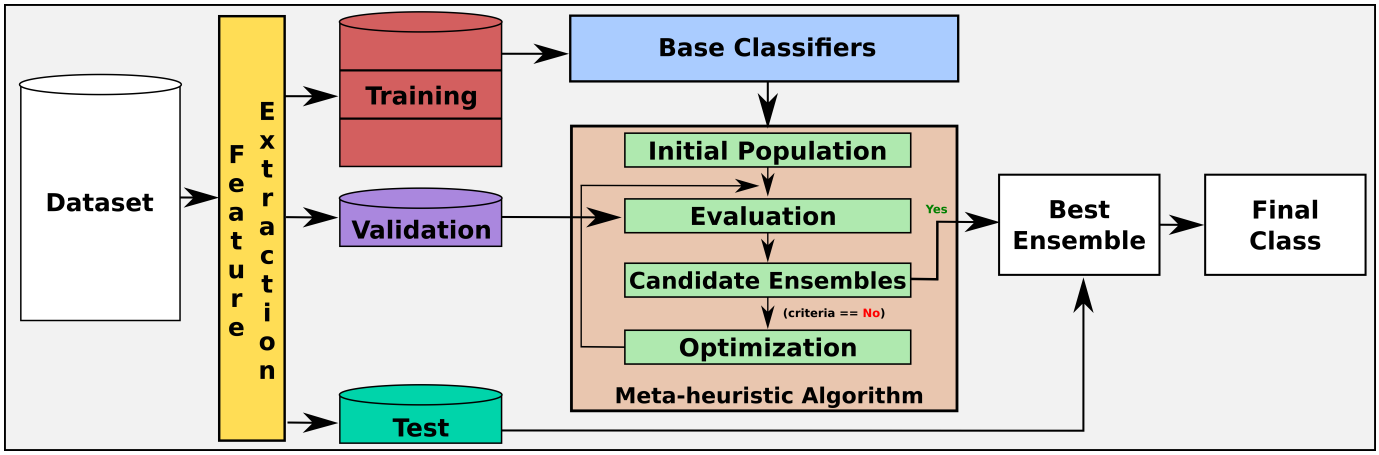

Fig. 2: The proposed framework based on meta-heuristic algorithms for creating classifier ensembles.

Fig. 3: Examples of aerial scenes from the WHU-RS19 dataset, which are similar to the RSSCN7 and the UCMerced datasets.

each final ensemble built by optimization algorithms. Finally, the third analysis, an efficiency experiment has been performed to verify which algorithm is faster throughout optimization process in the proposed framework.

A. Classification Analysis

Table III shows the classification performance for nine optimization algorithms and the well-known baseline majority voting (MV), which does not use any optimization algorithm to build the final ensemble. In gray are the best optimization algorithm results per experiment. In blue are average of the accuracies to all optimization algorithms. In red are the gain of the best algorithms against the baseline (MV).

In general, all optimization algorithms have achieved better classification results than the baseline (MV) with an exception to RSSCN7 dataset using only Global classifiers that MV achieved 81.8% of mean accuracy against 81.6 from average of algorithms. Furthermore, the number of classifiers used in ensembles built by optimization algorithms is at least 49% (17 out of 35 available classifiers) and up to 65% (25 out of 70 available classifiers) lower than MV approach.

Considering the best algorithms for each performed experiment using Global classifiers, it is possible to observe that BHA algorithm achieved good results on the RSSCN7 dataset with mean accuracy of 81.9% and CS algorithm has achieved better classification results on two datasets (WHU-RS19 and UC Merced) with mean accuracy of 78.2% and 79.9%, respectively.

The minimum gain of the best optimization algorithms in relation to the baseline (MV) is 0.1% of mean accuracy (BHA using Global classifiers in RSSCN7 dataset) and 17 less classifiers in the ensemble (ABC using Global classifiers

TABLE II: Parameter values for each meta-heuristic algorithm.

Algorithm	Parameter	Value
ABC	Number of trials	10
BA	Frequency range (f_{min}, f_{max})	[0, 2]
	Loudness (A)	0.5
	Pulse rate (r)	0.5
CS	Step size (α)	1.0
	Lévy flight controller (β)	1.5
	Nests replacement probability (p)	0.2
FA	Randomization (α)	0.5
	Attractiveness (β)	0.2
	Light absorption coefficient (γ)	1.0
FPA	Lévy flight controller (β)	1.5
	Lévy flight scaler (η)	0.2
	Local pollination probability (p)	0.8
GP	Reproduction probability (p_r)	0.25
	Mutation probability (p_m)	0.1
	Crossover probability (p_c)	0.1
PSO	Inertia weight (w)	0.7
	Cognitive constant (c_1)	1.7
	Social constant (c_2)	1.7
UMDA	Population size (μ)	500
	Number of generations (t)	250
	Generation stagnation	0.2
	Initial probabilities (p_t)	0.5
	Upper-bound margin ($1 - \frac{1}{n}$)	0.95
	Lower-bound margin ($\frac{1}{n}$)	0.05

in UCMerced dataset) than baseline (MV). The maximum gain of the optimization algorithms, considering the mean accuracy, is 2.8% (CS using Global classifiers in WHU-RS19 dataset) and 45 less classifiers in final ensemble (UDMA using ALL classifiers in WHU-RS19 dataset).

In adverse scenarios, it is possible to observe that the optimization algorithms are rarely affected. Notice that the best optimization algorithm (UDMA) with only CNN classifiers in the WHU-RS19 dataset still achieved better results with ALL classifiers, but it has dropped the mean accuracy from 96.9 to 96.0 when exposed to such adverse scenario. The same behavior has been observed by other algorithms in the UCMerced dataset as well. Furthermore, in experiments using RSSCN7 dataset, some optimization algorithms have maintained and even increased their mean accuracy, such as ABC, CS, FPA, GP, PSO, and UDMA.

B. Efficiency Analysis

Table IV shows an efficiency analysis among the nine algorithms to build their final ensembles considering the optimization process applied to ALL classifiers, i.e., 70 available classifiers on the validation set. As the baseline (MV) does not have optimization process, it has not been included in this experiment. In this experiment it is possible to observe that UDMA algorithm was the fastest approach among the nine algorithms in all datasets. UDMA is $1.4\times$, $3.17\times$ and $3.18\times$ faster than the most efficient optimization algorithm using WHU-RS19, RSSCN7, and UCMerced datasets, respectively.

TABLE III: Classification results and confidence interval (±) computed over a 5-fold cross validation protocol. The number of classifiers present in each final ensemble approach is represented by #.

Algs.	WHU-RS19					
	Global (35)	#	CNN (35)	#	ALL (70)	#
ABC	76.5 (±3.0)	16	95.8 (±1.0)	17	94.5 (±1.8)	32
BA	76.8 (±3.0)	18	95.6 (±1.0)	17	93.9 (±1.5)	30
BHA	76.5 (±3.0)	16	95.8 (±1.0)	15	94.4 (±1.3)	33
CS	**78.2** (±3.0)	17	95.7 (±1.0)	16	95.6 (±1.4)	**25**
FA	76.5 (±3.0)	18	95.2 (±2.0)	17	94.6 (±1.7)	35
FPA	77.6 (±3.0)	**15**	96.6 (±1.0)	14	95.3 (±1.0)	33
GP	77.2 (±3.0)	26	95.1 (±2.0)	**12**	93.6 (±1.6)	**25**
PSO	75.9 (±3.0)	18	95.7 (±1.0)	14	94.1 (±2.0)	31
UDMA	76.8 (±2.8)	19	**96.9** (±1.0)	14	**96.0** (±1.5)	30
Average	**76.7**	**20**	**95.7**	**17**	**94.5**	**34**
Baseline (MV)	75.4 (±3.0)	35	94.2 (±1.0)	35	93.4 (±1.5)	70
Gain (Best × MV)	**2.8**	**-20**	**2.7**	**-23**	**2.6**	**-45**

Algs.	RSSCN7					
	Global (35)	#	CNN (35)	#	ALL (70)	#
ABC	81.8 (±1.0)	**18**	90.3 (±0.0)	17	90.7 (±0.6)	35
BA	81.5 (±1.0)	20	90.4 (±1.0)	18	90.3 (±0.7)	36
BHA	**81.9** (±1.0)	19	90.4 (±1.0)	16	90.2 (±0.6)	**32**
CS	81.5 (±2.0)	21	90.6 (±1.0)	**14**	90.9 (±0.9)	**32**
FA	81.3 (±2.0)	19	90.1 (±1.0)	18	90.0 (±0.8)	38
FPA	81.3 (±2.0)	**18**	90.7 (±1.0)	16	90.7 (±1.2)	35
GP	81.0 (±2.0)	27	89.6 (±1.0)	17	90.3 (±0.9)	44
PSO	81.5 (±1.0)	**18**	90.4 (±1.0)	18	90.4 (±0.8)	35
UDMA	**81.9** (±1.1)	21	**90.9** (±1.0)	16	**91.1** (±0.4)	**32**
Average	**81.6**	**22**	**90.3**	**19**	**90.4**	**39**
Baseline (MV)	81.8 (±1.0)	35	89.5 (±1.0)	35	89.8 (±0.9)	70
Gain (Best × MV)	**0.1**	**-17**	**1.4**	**-21**	**1.3**	**-38**

Algs.	UCMerced					
	Global (35)	#	CNN (35)	#	ALL (70)	#
ABC	79.0 (±1.0)	**18**	94.2 (±1.0)	16	94.1 (±1.5)	34
BA	78.9 (±1.0)	20	94.0 (±1.0)	17	93.6 (±0.9)	36
BHA	78.9 (±1.0)	19	94.9 (±0.0)	16	93.6 (±1.3)	38
CS	**79.9** (±1.0)	20	94.8 (±1.0)	**13**	94.1 (±1.2)	30
FA	79.0 (±1.0)	21	93.9 (±1.0)	16	92.9 (±1.5)	37
FPA	79.7 (±2.0)	19	94.3 (±1.0)	16	94.1 (±1.1)	31
GP	78.9 (±1.0)	25	94.0 (±2.0)	15	93.6 (±1.6)	41
PSO	79.6 (±1.0)	20	94.0 (±1.0)	16	93.3 (±0.8)	36
UDMA	**79.9** (±0.7)	20	**94.7** (±1.0)	15	**94.4** (±1.4)	**29**
Average	**79.3**	**22**	**94.2**	**18**	**93.6**	**38**
Baseline (MV)	78.9 (±1.0)	35	93.5 (±1.0)	35	92.6 (±1.7)	70
Gain (Best × MV)	**1.0**	**-17**	**1.2**	**-22**	**1.8**	**-41**

TABLE IV: Efficiency results computed in seconds (s) for the optimization process on the validation set in a 5-fold cross validation protocol. @R means the ranking position of each algorithm in the three different datasets used in this paper.

Algorithms	Time(s)					
	WHU-RS19	@R	RSSCN7	@R	UCMerced	@R
ABC	532 (±16)	9°	2946 (±95)	9°	2150 ±70)	9°
BA	371 (±5)	7°	1944 (±102)	7°	1580 ±107)	7°
BHA	366 (±11)	6°	1908 (±88)	5°	1494 ±53)	5°
CS	520 (±12)	8°	2865 (±84)	8°	2264 ±87)	8°
FA	181 (±10)	2°	924 (±51)	2°	655 ±53)	2°
FPA	347 (±12)	5°	1924 (±65)	6°	1525 ±22)	6°
GP	182 (±9)	3°	1041 (±52)	4°	660 ±47)	3°
PSO	182 (±9)	4°	939 (±73)	3°	661 ±54)	4°
UDMA	**132** (±29)	1°	**292** (±31)	1°	**208** ±51)	1°

VI. Conclusion

Deep learning architectures, in particular the convolutional neural networks (CNNs), have been applied to several knowledge areas (e.g. medicine, biology, agriculture, security, and remote sensing) due to the excellent classification accuracy results. Among the challenging tasks in remote sensing area is classification task, which can be performed in three levels of abstraction (pixel, object and scenes).

Currently, aerial scene classification task has been intensively studied in the literature due to the availability of datasets and its applicability in military and civilians affairs. However,

the high intra-class variability of the objects with different scales and orientations in images make the challenge for learning techniques even greater.

Even though the CNN architectures achieve surprisingly accurate results in the literature, there are still real applications that a single architecture is not enough to solve them. Thus, an alternative to overcome that problem might be the ensemble of CNN architectures. Many papers in the literature have adopted ensemble of CNN-based classifiers to improve the effectiveness of the results of their proposed approaches. However, most of the developed solutions adopt an ad-hoc strategy or without criteria for choosing the classifiers that compose the ensemble of classifiers. In this sense, we propose a novel framework for creating ensembles based on optimization algorithms in the context of aerial scene classification.

In this work, we have performed three different experiments: (1) is a comparative study among nine different optimization algorithm using three different scenarios (Global, CNN, and ALL) and three well-known aerial scene datasets (WHU-RS19, RSSCN7, and UCMerced); (2) is a comparative analysis of the number of base classifiers present in the final ensembles built by optimization algorithms; and (3) an efficiency analysis among the nine algorithms used in the optimization process.

In the first experiment, we observed that all optimization algorithms achieved better classification results than the baseline (MV) in almost all experiments with an exception to RSSCN7 dataset using only Global classifiers. Also, it was possible to note that the UDMA algorithm achieved the best results in the CNN and ALL scenarios. CS and BHA achieved the best results in the Global scenario.

In the second experiment, we showed that the best optimization algorithms reduced the number of classifiers in the final ensembles by at least 49% (ABC using 18 of 35 Global classifiers in the UCMerced dataset) and might reach a reduction of 65% (25 of 70 classifiers used by UDMA in the WHU-RS19 dataset). Finally, in the last experiment, we could show that UDMA algorithm is at least $1.4\times$ faster than the most efficient optimization algorithm compared in this paper.

As a future work, we intend to perform experiments with other real applications (e.g., pest identification and splicing detection), level of abstraction (e.g., pixel and region), deep neural networks (e.g., Restricted Boltzmann Machines and Long Short-term Memory Networks), and learning tasks (e.g., adaptation domain and one-class). Furthermore, we plan to combine diversity measures and accuracy to improve the classification results of the optimization algorithms.

Acknowledgements

The authors thanks São Paulo Research Foundation (FAPESP) for support through grants #2013/07375-0, #2014/12236-1, #2017/25908-6, #2018/23908-1, #2019/07665-4 and #2019/02205-5, the Brazilian scientific funding agency CNPq through the Universal Projects (grants #408919/2016-7 and #427968/2018-6) and Research Fellowship (grant #307066/2017-7), as well as NVIDIA Corporation for the donation of the GPUs used for this research.

References

[1] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," *CoRR*, vol. abs/1508.00092, 2015. [Online]. Available: http://arxiv.org/abs/1508.00092

[2] R. F. Chew, S. Amer, K. Jones, J. Unangst, J. Cajka, J. Allpress, and M. Bruhn, "Residential scene classification for gridded population sampling in developing countries using deep convolutional neural networks on satellite imagery," *International Journal of Health Geographics*, vol. 17, no. 1, p. 12, May 2018.

[3] Y. Yu and F. Liu, "A two-stream deep fusion framework for high-resolution aerial scene classification," *Computational Intelligence and Neuroscience*, 2018.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[5] M. Swain and D. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[6] D. chen He and L. Wang, "Texture unit, texture spectrum, and texture analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 28, no. 4, pp. 509–512, 1990.

[7] S. Avila, N. Thome, M. Cord, E. Valle, and A. de A. Araújo, "BOSSA: Extended BoW formalism for image classification," in *IEEE ICIP*, 2011, pp. 2909–2912.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 00, 2015, pp. 1–9.

[10] G. Yao, T. Lei, and J. Zhong, "A review of convolutional-neural-network-based action recognition," *Pattern Recognition Letters*, 2018.

[11] K. Sundararajan and D. L. Woodard, "Deep learning for biometrics: A survey," *ACM Computing Survey*, vol. 51, no. 3, pp. 65:1–65:34, May 2018.

[12] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375–9389, 2018.

[13] G. Christie, N. Fendley, J. Wilson, and R. Mukherjee, "Functional map of the world," in *IEEE CVPR*, 2018.

[14] B. L. Saux, N. Yokoya, R. Hansch, and S. Prasad, "2018 ieee grss data fusion contest: Multimodal land use classification," *IEEE Geoscience and Remote Sensing Magazine*, vol. 6, no. 1, pp. 52–54, March 2018.

[15] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," *arXiv preprint arXiv:1702.02284*, 2017.

[16] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, "Dual path networks," in *NIPS*, 2017.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE CVPR*, 2016, pp. 770–778.

[18] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *IEEE CVPR*, 2017, pp. 5987–5995.

[19] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016. [Online]. Available: http://arxiv.org/abs/1608.06993

[20] R. Minetto, M. Pamplona Segundo, and S. Sarkar, "Hydra: An ensemble of convolutional neural networks for geospatial land classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6530–6541, 2019.

[21] Y. Xu, B. Du, and L. Zhang, "Multi-source remote sensing data classification via fully convolutional networks and post-classification processing," in *IGARSS*, 2018, pp. 3852–3855.

[22] D. Cerra, M. Pato, E. Carmona, S. M. Azimi, J. Tian, R. Bahmanyar, F. Kurz, E. Vig, K. Bittner, C. Henry, P. d'Angelo, R. Müller, K. Alonso, P. Fischer, and P. Reinartz, "Combining deep and shallow neural networks with ad hoc detectors for the classification of complex multi-modal urban scenes," in *IGARSS*, 2018, pp. 3856–3859.

[23] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[24] X.-S. Yang and A. H. Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations*, vol. 29, no. 5, pp. 464–483, 2012.

[25] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Information sciences*, vol. 222, pp. 175–184, 2013.

[26] X.-S. Yang and S. Deb, "Engineering optimisation by cuckoo search," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 1, pp. 330–343, 2010.

[27] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal Bio-Inspired Computing*, vol. 2, no. 2, pp. 78–84, 2010.

[28] S.-S. Yang, M. Karamanoglu, and X. He, "Flower pollination algorithm: A novel approach for multiobjective optimization," *Engineering Optimization*, vol. 46, no. 9, pp. 1222–1237, 2014.

[29] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.

[30] J. Kennedy and R. Eberhart, *Swarm Intelligence*. M. Kaufman, 2001.

[31] H. Mühlenbein and G. Paass, "From recombination of genes to the estimation of distributions i. binary parameters," in *International conference on parallel problem solving from nature*. Springer, 1996, pp. 178–187.

[32] M. Hauschild and M. Pelikan, "An introduction and survey of estimation of distribution algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 3, pp. 111–128, 2011.

[33] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, Oct 1990.

[34] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, Jul. 1990.

[35] F. A. Faria, J. A. dos Santos, S. Sarkar, A. Rocha, and R. da S. Torres, "Classifier selection based on the correlation of diversity measures: When fewer is more," in *SIBGRAPI*, Arequipa, Peru, august 2013, pp. 16–23.

[36] F. A. Faria, J. A. dos Santos, A. Rocha, and R. da S. Torres, "A framework for selection and fusion of pattern classifiers in multimedia recognition," *PRL*, vol. 39, no. 0, pp. 52 – 64, 2014.

[37] R. Stehling, M. Nascimento, and A. Falcao, "A compact and efficient image retrieval approach based on border/interior pixel classification," in *ACM Conference on Information and Knowledge Management (CIKM)*, 2002, pp. 102–109.

[38] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *ACM Intl. Conf. on Multimedia (MM)*, 1996, pp. 65–73.

[39] C. Huang and Q. Liu, "An orientation independent texture descriptor for image retrieval," in *Intl. Conf. on Communications, Circuits and Systems*, 2007, pp. 772–776.

[40] B. Tao and B. Dickinson, "Texture recognition and image retrieval using gradient indexing," *Journal of Visual Communication and Image Representation (JVCI)*, vol. 11, no. 3, pp. 327–342, 2000.

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE CVPR*, june 2009, pp. 248–255.

[42] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE CVPR*, 2016, pp. 2818–2826.

[43] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *arXiv preprint arXiv:1610.02357*, 2016.

[44] S. Lomax and S. Vadera, "A survey of cost-sensitive decision tree induction algorithms," *ACM Computing Surveys (CSUR)*, vol. 45, no. 2, pp. 16:1–16:35, Mar. 2013.

[45] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000.

[46] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*, 1st ed. Springer, 2001.

[47] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Workshop on Computational Learning Theory*, ser. COLT '92, 1992, pp. 144–152.

[48] G.-S. Xia, W. Yang, J. Delon, Y. Gousseau, H. Sun, and H. Maître, "Structural High-resolution Satellite Image Indexing," in *ISPRS TC VII Symposium - 100 Years ISPRS*, Vienna, Austria, Jul. 2010, pp. 298–303.

[49] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 11, pp. 2321–2325, Nov 2015.

[50] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *SIGSPATIAL*, ser. GIS '10. New York, NY, USA: ACM, 2010, pp. 270–279.