A Simple and Efficient Approach for Cooperative Incremental Learning in Robot Swarms

Gianni A. Di Caro, Alessandro Giusti, Jawad Nagi, Luca M. Gambardella
Dalle Molle Institute for Artificial Intelligence (IDSIA), Lugano, Switzerland
Email: {gianni,alessandrog,jawad,luca}@idsia.ch

***Abstract*—In the context of vision-based hand gesture recognition, we study how a robot swarm can incrementally and cooperatively learn to classify an unseen gesture vocabulary using a simple information sharing mechanism. Training examples and correction feedback are interactively provided by a human instructor. Each robot in the swarm is equipped with a statistical classifier, which is built and progressively updated using the input from the instructor. In order to learn collectively and speed-up the process, the robots share with each other a selection of the locally acquired gesture data.**

Extensive experiments on a real-world dataset show that the proposed cooperative learning approach is effective and robust, in spite of its simplicity. Accounting for bandwidth limitations in network communications, we study the impact of different strategies for the selection of the shared data, and we investigate the effect of swarm size and the amount of shared information on the learning speed.

I. Introduction

Robot swarms can be conveniently employed as mobile sensor-actuator networks. Their intrinsic spatial distributedness, redundancy of resources, and parallelism, are in fact key properties to perform distributed sensing and actuation tasks. In previous work [1], by leveraging on these characteristics, we used robot swarms for *distributed vision-based recognition of hand gestures*. A single robot observes the gesture image from its own point of view, segments the hand outline, and extracts a set of features. The robot then uses a statistical classifier to assign a probability to each class the gesture could belong to. These probabilities reflect how certain the robot is about the classification of the gesture. Since the typical robots used in swarms are not very powerful both in terms of processing and sensing, we implemented a distributed protocol to let the robots cooperatively fuse their probabilistic opinions by exploiting multi-hop wireless communications, and rapidly generate a single swarm-level decision about the observed gesture through of a form of distributed consensus.

In this paper we start from the same basic scenario as described above. However, while in the previous work robots were performing cooperative *classification* based on the same classifier trained *offline* and given as input to all robots, here we study how the swarm can cooperatively *learn* hand gestures. That is, how robots can cooperatively train their classifiers *online*. We consider the case of learning online *incrementally*, starting from scratch, through the examples and the feedback provided by a *human teacher*. In order to minimize human-swarm interaction time, and speed-up the learning curve, we propose a simple yet effective way to realize *cooperative learning at the swarm level* by exploiting communication-based information sharing. Moreover, we explicitly address the issue of dealing with a bandwidth-constrained and unreliable communication channel (as it is normally the case in ad hoc wireless networks). We use the same architecture as before, in which each robot has its own statistical classifier (an off-the-shelf *Support Vector Machine*). However, now, each robot *trains its own classifier*, using (i) the features extracted from a set of a locally available labeled examples, which correspond to the gesture images acquired from the robots' specific viewpoint (the robot's *experience*), and (ii) also exploiting the experience from other robots. The set is initially empty. In an *initial learning phase* the human instructor briefly shows to the swarm each gesture in the vocabulary, which "sets the problem" and provides each robot with an initial, yet minimal set of labeled samples which are used to train a first classifier. At *successive interaction rounds*, which might happen at any time and place, the human shows a selected gesture to the swarm (either for the purpose of training the swarm or to actually issue a command associated to the gesture): the robots in the swarm cooperatively sense and classify the gesture, and communicate the result to the instructor by synchronously blinking their on-board LEDs. The human either explicitly corrects or silently confirms the swarm's classification. Based on the received feedback, each robot can then add the new labeled sample to its training set, and possibly retrain its local classifier to incrementally improve its gesture classification.

In the process described so far, each robot relies on its own local experience for learning. The presence of the swarm is only exploited at classification time in terms of *distributed consensus*, which is quite limiting. Therefore, we propose a simple approach to to let the robots effectively *learn in a fully cooperative way*: through multi-hop wireless communications the robots locally share with each other their local experiences. In this way, each robot can incrementally make its local training database grow by exploiting the images sampled from other robots. Due to robots' spatial distribution, the information sharing allows a robot to learn the appearance of the same gesture as seen also from the points of view of other robots. This results in a better classification accuracy in subsequent *interaction rounds*, since robot positions will change with respect to each other and with respect to the hand due to human and robot mobility. As a consequence, the swarm can exploit the robots' multiplicity in two ways: (i) learning faster and more robustly through cooperative information sharing, (ii) producing more accurate classifications through consensus,

This research was supported by the Swiss National Science (SNSF) Foundation through the National Centre of Competence in Research (NCCR) Robotics (www.nccr-robotics.ch).

978-1-4799-2722-7/13/$31.00 ©2013 IEEE

which fuses the opinions of all robots.

In order to perform information sharing, in the most general case when no network infrastructure is present, the swarm needs to rely on the wireless ad hoc network among the robots. However, since bandwidth is usually a scarce resource in such networks, the complete sharing of the information of all robot experiences may not be feasible, in practice, within a time delay compatible with the constraints online operations (even when an infrastructure-based wireless network is used). This is particularly true when the swarm is large and/or a large number of images are locally acquired by robots. In the latter case also computation time and memory requirements become an issue, especially for embedded systems like typical robot swarms. Therefore, we designed and validated different strategies to let each robot *intelligently select the best training data to spread throughout the network*. The aim of the paper is to address the issue of limited communication bandwidth for information sharing by finding the right balance between *optimizing the training information available at the robots at the swarm-level*, and *minimizing bandwidth occupancy to guarantee scalability*. The issue of limited on-board computation resources is addressed indirectly: limiting the amount of shared information automatically reduces the needs for on-board computations. While we focus on robot swarms, our claim is that strategies for an intelligent selection of the training data for sharing can result in an advantage in many different practical real-world scenarios when communication / computation resources and/or rapid response time are an issue (e.g., in applications of wireless sensor networks).

As a final remark, it is important to point out that the presented ideas can be immediately applied to other supervised learning scenarios in swarms of robots or networked sensors, and be based on different sensing data (e.g., audio).

The paper is organized as follows. Related work is discussed in Section I-A. The techniques used for sensing gestures and for fusing robot opinions are briefly presented in Section II. The main contribution, i.e. the cooperative learning approach, is presented in Section III, and is experimentally validated in Section IV, by investigating the effects of different aspects such as swarm size, available bandwidth, feature vector size, and information sharing strategy.

A. Related Work

We consider an *incremental* (or *online*) learning [2] scenario: training examples come over time provided interactively, one at a time, by a human teacher. This differs from a typical batch learning scenario, in which training examples are make available altogether at the beginning. In the context of incremental leaning, research has mainly focused on the development of techniques for real-time, quick and cheap updating of the classifiers whenever new experiences become available [3], [4]. Examples are the online variants of techniques such as the Adaptive Resonance Theory [5], Multi-layer Perceptrons [6] and Support Vector Machines [7]. We share with this approaches the high-level goal of exploiting new experiences as they become available. However, we are mainly interested in how such experiences can be shared among the robots in a swarm, addressing cooperation and communication issues. Therefore, we do not focus on the development of novel complex techniques for the efficient updating of single classifiers. At this aim, we conveniently make use of a standard approach based on a *Support Vector Machine* (SVM) [8]: when new samples are added to the training data, the SVM classifier is re-trained from scratch. A similar approach has been used in [9] for incremental image classification learning. In our test scenario, we observed that even when the training set was growing up to hundreds of samples, the computation time for re-training the SVM on-board of the small robots that we used, the foot-bots (see Section IV), was of order of a few seconds (e.g., 2 seconds for 100 samples, see Section IV-A). This is a latency time which is fully compatible with the real-time specifications of the interactive teaching scenario we are considering, in which a human shows a gesture and can in principle wait a few seconds before showing the next one. Reaching 'true' real-time would require more performing hardware. The potential limitations of the current implementation have been addressed in another work [10], in which we proposed novel incremental learning algorithms based on both full and binary feedback.

The use of SVMs for performing *cooperative learning* and recognition tasks for distributed systems such as Wireless Sensor Networks, has been widely investigated [11]. In this context, most works, including ours, can be classified as *parallel learning* [12] approaches (e.g., as *Ensemble of Classifiers* [13]), in which each node/robot trains its own SVM independently based on partitioned inputs or different parameters. The outputs of the classifiers are then combined in some static, adaptive, or even topology-depending way [14], and the feedback is processed accordingly. Several works [15], [16] have also addressed the problem of *distributed training* (or interdependent training), in which all nodes participate to the training of a single classifier.

In this context, our approach can be configured as a distributed form of the popular *bagging* classifier: each robot individually trains an SVM based on a bootstrapped replica of the entire training dataset. The replica is implicitly derived from robot's own observations and from the selected training data it receives from other robots through cooperative information sharing. Bagging is at the same time simple, lightweight, and well-know to show excellent performance in case of limited training data [13]. These are precisely the conditions of the scenario we are considering: robots used in swarm systems are typically not extremely powerful (due to the need to balance the number with the cost), communication bandwidth is limited, and it is necessary to learn fast out of a few samples due to the interactive nature of the scenario, which requires the physical presence of a human instructor. The main contribution of the paper is precisely in defining, and showing in practice, the effectiveness of a simple cooperative learning system which is compliant both with the typical constraints of swarms and with the requirements of interactive scenarios. Instead, many works in the literature propose sophisticated cooperative learning schemes which are either too expensive for computation or communication (this is especially true for distributed training approaches), and/or are mostly appropriate for large datasets.

One important contribution of our approach is the use and the investigation of *information exchange* among robots to improve and speed-up the learning process. A similar approach has been followed in a number of works considering multi-agent or multi-robot systems [17], [18]. Our criteria for selecting the training data to be disseminated, which rely on an original distinction between personal and shared data, are derived from these previous works and adapted to be computationally light and optimize mutual information. Interestingly, the issue of selecting training information to disseminate, is related (but not equivalent) to the problem of selecting which data points should be labeled in an *active learning* [19] setting. Similarities with this problem are discussed in Section III-B.

II. Gesture sensing and swarm classification

In this section we briefly describe how the robots make use of their locally trained classifier for distributed hand gesture recognition based on the system presented in [1]. That is, we describe the actual usage scenario, when a *classifier has already been learned*. This part is described first, in order to illustrate how image recognition and fusion of information is performed, which is instrumental to understand the rationale behind the swarm learning strategy proposed in Section III, which is the main contribution of the paper.

We assume that the swarm is randomly scattered in the environment. Without losing generality, we assume that the human who intends to provide commands using hand gestures wears a glove with a known characteristic color.

A. Single robot gesture recognition

At each individual robot, the image recognition process starts with the segmentation of the hand in the acquired image. The robot interprets its shape, and compute a classification vector $\mathbf{C}$, representing the robot *opinion*, which contains a probability value for each known gesture. Throughout the paper, we consider a set of $K = 6$ gestures in terms of *finger counts*, from zero (closed hand) to five fingers, as illustrated in Figure 1. Gestures may be shown in any rotation and in any finger combination.

Fig. 1. An example of the six finger-count gestures.

The single robot hand recognition problem is intrinsically difficult because of the following reasons: the image may be taken from an arbitrarily bad point of view (angle and distance), which depends on the position of the robot when the recognition process started; the hand orientation is unknown (and it is not trivial to robustly estimate it from the image); camera resolution and processing power of the robots typically used in swarm robotics [20], such as the foot-bot robots (see Section IV) that we use in the experiments, do not allow to perform fast and accurate image processing on-board. For instance, Figure 3 shows that, when the gestures are viewed from non optimal (i.e., non perpendicular) angles, they can be hard to classify even for a human observer. We solve the visual recognition task following a basic computer vision approach based on *segmentation*, *feature extraction*, and *supervised classification*.

Color-based segmentation. Once an image is acquired, the first processing step consists in *segmenting* the hand, by exploiting the characteristic color of the glove using a standard color-based segmentation approach in the HSV color space. The largest *connected component* in the resulting binary image is identified as the hand, and used to compute a set of quantitative *features*.

Feature computation and selection. The used features are *shape descriptors* (area, perimeter, form factor, compactness, roundness, aspect ratio, area moments of inertia, Hu invariant moments, etc.) derived considering geometric properties frequently used in the the literature for segmented object recognition tasks [21]. We defined a total of 110 features which serve the purpose of representing shape descriptors that are invariant towards rotation, translation, and scale. The features were ranked according to their *information gain*, computed using a dataset of hand gesture images. This information measure was used to identify the features that provide higher discriminative power. Reducing the dimensionality of the feature space results in better classification performance at a lower computational cost. In most of the experiments reported in the following, the top $F = 40$ features are used for classification. In Section IV-D we study different values for F and discuss the resulting trade-off in performance.

Probabilistic classification. The resulting feature vector, which is composed by F elements, is classified by means of a non-linear SVM with an RBF-*Gaussian* kernel, using the implementation from the LIBSVM C++ library. The classifier is incrementally trained using the samples acquired by the robot itself, as well as by the rest of the swarm, according to the swarm learning mechanisms discussed in Section III. The SVM classifier returns the posterior probabilities of the 6 classes, i.e., a 6-dimensional opinion vector $\mathbf{C}$, whose elements sum to one.

B. Swarm consensus

Since several robots simultaneously engage in the recognition task, and possibly output different classification vectors, their opinions need to be fused in order to obtain a single outcome representing the *swarm decision*. Data are exchanged among the robots using *multi-hop* packet forwarding in the wireless ad hoc network formed by the swarm. As previously pointed out, in this paper we use a very simple *consensus* approach for reaching a swarm-level decision: robots exchange their opinion vectors through, and compute a fused swarm-level opinion vector as the element-wise average of all robot opinions. The class corresponding to the largest element of this opinion vector is chosen as swarm decision. This instance of *plurality voting* [13] is a simplified version of a more sophisticated consensus strategy that we developed taking into account time and communication constraints [1].

III. Learning as a swarm

The cooperative, communication-based learning approach that we propose aims to achieve two related goals. *a*) Learn a *new* set of gestures from scratch during an *initial learning* phase, during which the user presents the new set of gestures to the swarm. *b*) Incrementally improve the recognition ability as gestures are presented to the swarm either to improve the recognition performance over time or for the execution of the commands related to the gestures. In the following we refer to this phase as *interaction rounds*.

In our scenario, learning refers to the supervised training of the SVM classifier which is used by the robots for visual recognition. An effective training requires that a sufficient amount of labeled samples of gesture images taken from multiple different positions with respect to the gesture are made available to the robots. In fact, during the normal course of operations of the swarm, each robot is expected to see the hand from a different point of view, which changes over time due to robot movements.

The robots start the learning process from scratch, meaning that no gesture vocabulary is given a priori, and no classifier initially exists. Robots accumulate knowledge in terms of labeled training samples: each robot processes the image it has acquired and saves the resulting feature vector along with the corresponding ground truth label. In this way, sampled data accumulates in the local training set of the robot, which incrementally grows during the learning process. A robot also share its local experience by sending out some of its training samples, in the form of the associated labeled feature vectors. When a robot receives such data from the network, it includes it in its local training set. As a result, a subset of all training data gets shared across robots.

A. Phases for swarm learning

1) Initial learning phase: During the *initial learning* phase, a user explicitly trains the swarm by presenting the full vocabulary of gestures once (whose cardinality K is assumed known a priori). In particular, each robot acquires M images for each of the K different gestures in the vocabulary. Gestures are numbered sequentially and shown in order. Operationally, the robots in the swarm flash their LEDs in order to instruct the human teacher to show the next gesture in the sequence. As a result, all acquired images can be associated to the respective ground truth gesture.

Robots are randomly distributed in the environment and in parallel acquire visual data from multiple points of view for each gesture. Each of the $M \cdot K$ images is processed, yielding to an F-dimensional feature vector, which is stored together with the corresponding ground truth label.

At this point, however, each robot only knows about the appearance of the gestures from its own point of view. Before training their local classifiers, each robot *exchange* its training data with other robots (by wireless broadcast). This cooperative behavior result in the fact that each robot can augment its local training set with data acquired by other robots from different points of view. The strategies for selecting which and how many instances to exchange are discussed later. The end of the first interaction round is the only moment when all robots have the same classifier.

2) Interaction rounds: Immediately after the initial learning phase, robots have a classifier in place, and can therefore start performing classification tasks. However, the size of the classifier is minimal and is not expected to provide good recognition accuracies. It will incrementally grow when the human instructor will show new training samples.

The interaction rounds following the first begins with the user signaling the willingness to provide a gesture, that can represent a command to execute, in addition to be used as training sample. The human shows the gloved hand, that will be possibly detected by some of the robots in the swarm whose main task consists precisely in monitoring the environment for the presence of the glove. In principle, the human can attract the attention also by using other means (i.e., whistling). When a robot detects the glove, it locally broadcasts a message to the other robots scattered in the environment. On reception of the message, they turn in place in order to position themselves looking towards the hand.

After positioning, each robot acquires a single image of the gesture, computes the feature vector, and applies its local, most recently-trained classifier, in order to output the opinion vector C. The simple consensus protocol described in Section II-B is then executed by averaging the opinions of all robots, and eventually the swarm makes a selection regarding the gesture. Each robot signals the detected gesture by blinking lights (e.g., if the gesture was classified as finger count 3, all robots will flash their LEDs 3 times).

Here we consider a *full-feedback* scenario: if the answer of the swarm is incorrect, the human agent will *correct* the wrong decision, providing the correct classification label. This can be realized in a number of different ways. In our implementation we use a simple visual signal: a rapid waving of hands, which can be robustly detected by a single robot, indicates that swarm decision was wrong. If it is the case, the swarm proposes the second most-likely decision (i.e., the class corresponding to the second largest probability in the fused opinion vector), possibly iterating this procedure until the correct decision is proposed and the user implicitly accepts it by not reacting to it (i.e., standing still).

After instructor's feedback, the swarm knows the correct label associated to the original gesture, and can possibly start acting accordingly (e.g., an additional gesture can serve to distinguish between training and commanding). At the same time, each robot saves in its training set the feature vector that was generated from the acquired image, along with its ground truth label. Robots can now decide, using the strategies in Section III-B, whether to *share* or not the new experience with the rest of the swarm. At any time a robot can decide to *retrain* its local classifier, to improve its accuracy for the next interaction round with the user.

B. Strategies for sharing experiences

In this section we propose different ways to effectively share training data taking into account communication and computational constraints. The most basic strategy for sharing experiences (i.e., feature vectors) consists in spreading every single feature vector to all robots in the swarm. While providing the guarantee that all robots share the same classifiers and make use of the maximal amount of the available training information, this strategy has two main drawbacks: (i) the size of the classifiers can rapidly grow, especially in case of large swarms, which might have dramatic consequences on the processing time required for retraining; (ii) a large communication overhead is generated, due to all multi-hop message passing that are needed to uniformly spread the data throughout the swarm. Both (i) and (ii) may impair swarm's capability to react in real-time and to show a scalable behavior, robust to different networking environments.

As a consequence, sharing all experiences may not be feasible in practice, and a criterion to select the appropriate subset of training data to be shared with the rest of the swarm is necessary. Let R be the set of robots in the swarm, and let T_r denote the set of training instances currently available at robot $r \in R$. The set T_r is partitioned in three subsets:

- T_r^p, including the (personal) instances acquired by r and which have not been sent out to the rest of the swarm;
- T_r^s, including the (shared) instances acquired by r and which have been sent out to the rest of the swarm;
- T_r^e, including the (external) instances originally acquired by a robot $r' \neq r$, and which have been sent out by r' and received by r.

In case of full connectivity, r has no interest in broadcasting instances in T_r^s or T_r^e, because they are already known to the rest of the swarm. In fact, the set of instances $T_r^s \cup T_r^e$ represents the *shared knowledge* of the swarm.

We consider three possible criteria for a robot r to select the instance $t \in T_r^p$ which should be sent out first.

- *Random selection:* the instance is selected in a purely random way, without any intelligent criterion. This strategy provides a baseline for performance.
- *Novelty-driven selection:* robots prefer to share the instance which brings the most novel information compared to the instances already known by the rest of the swarm. Intuitively, this selection mimics the behavior of a teacher who shows examples that are as different as possible from those already known to the class. The selection is implemented as follows:

$$\arg\max_{t \in T_r^p} \|\mathbf{f}(t), \mathbf{f}(t')\|, \quad \forall t' \in (T_r^s \cup T_r^e),$$

  where $\mathbf{f}(t)$ is the feature vector for the training instance t, $c(t)$ is its ground truth class, and $\|\mathbf{f}(t), \mathbf{f}(t')\|$ is the Euclidean distance in the feature space.
- *Representativity-driven selection:* a robot shares the instance that best represents its current knowledge about a given class. This corresponds to a teacher who privileges examples which are the most typical ones. The selection is implemented by first sampling a class c' at random, and then choosing t such that:

$$\arg\min_{t \in T_r^p,\ c(t)=c'} \|\mathbf{f}(t), \mathbf{f}_{c'}\|,$$

  where $\mathbf{f}_{c'}$ represents the centroid, in the feature space, of all instances in T_r whose ground truth class is c'.

Figure 2 shows a an illustration of the different strategies.

At a robot r, once a training instance t is selected, its feature vector and label are sent out to the rest of the swarm by local broadcast. It will be received by the wireless neighbors of r, and then further relayed in multi-hop fashion to all other robots in the swarm. After the broadcast, the following updates are performed:

- in r, t is removed from T_r^p and moved to T_r^s;
- in all other robots $r' \neq r$ receiving the message, t is included in $T_{r'}^e$.

As a result, after this phase, t will never be broadcast again by any robot. If robot r has n training instances to disseminate, the mechanisms above are repeated n times.

It's interesting to note how the problem of selecting which instance to disseminate is related to *active learning* [19] strategies. In active learning, one has a large number of unlabeled instances and can "buy" labels for some of them. Many approaches have been proposed in order to optimize this choice. In the case of SVMs, in particular, the problem has been extensively studied: the feature-space distance between each instance and the separating hyperplane is often used as a criterion [22], by privileging instances either close or distant from the margin. In our context, the problem is different, because we are selecting which instances to provide to *other robots*, and we only have partial knowledge of which other instances they are already aware of. Moreover, in general, every robot knows about a different set of instances, which further complicates the task.

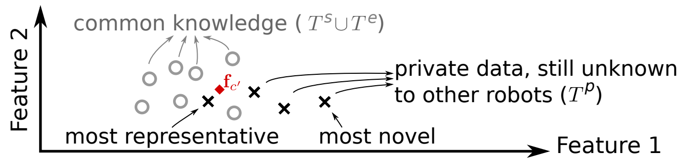

Fig. 2. Illustration of the criteria for the selection of samples to broadcast. For a given class c', all training samples known to a robot are represented as points in the feature space. Gray circles: samples already known to the swarm. Black crosses: samples only known to the robot itself. Red diamond: centroid of all personal samples ($\mathbf{f}_{c'}$).

IV. Implementation and Experimental Results

The described system has been implemented and validated on real robots using *foot-bots* [23] (Figure 3), small mobile platforms specifically designed for swarm robotics applications. We used a subset of the platform capabilities, namely: a *frontal camera*, which acquires 512×384 RGB images, and an IR-based *range-and-bearing* sensor and actuator system, which allows a robot to detect its *line-of-sight* neighbors up to a range of a few meters, and communicate with them through a low-bandwidth (100 Bits/s) channel. The foot-bots have quite a limited computational power, being equipped with an ARM 11 at 533MHz.

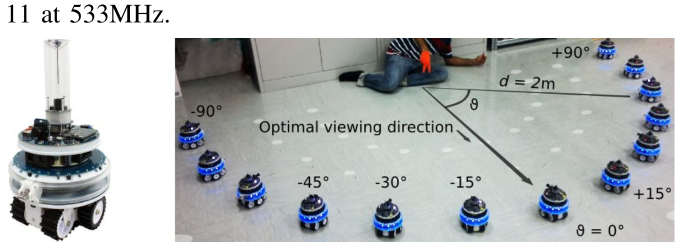

Fig. 3. The foot-bot robot and the dataset acquisition setup.

In order to be able to perform a large set of experiments, a swarm of 13 foot-bots was used to acquire a large dataset of 74,000 gesture images from different points of view with known ground truth (see Figure 3). The dataset has been used for running quantitative *emulation* experiments: robot observations are sampled from this dataset of real images, and realistic simulations are built using these data. When simulating an interaction in which the user is showing gesture c, each robot r is first assigned to a randomly defined position $(\vartheta, d)_r$ in the area in front of the gesture. Then, the image i "observed" by robot r is randomly selected from the subset of the dataset images whose ground truth class is c, and whose associated position $(\vartheta, d)_i$ is as close as possible to $(\vartheta, d)_r$. A control mechanism ensures that each dataset instance is used only once per simulation run.

A. Experimental protocol

Each simulation run starts at the beginning of the initial learning phase: each robot is initialized with an *empty training set* and placed at a random position. $M = 5$ samples are then acquired for each of the $K = 6$ classes by each robot. This corresponds to a very limited initial training effort since each robot only acquires 30 images, which require a very short time to process. More in general, it is important to remark that, using vectors of 40 features for the images, the time required for training the SVM classifier on a foot-bot is below 2 seconds for 100 training samples, and goes up to 20 seconds for 400 samples. These relatively short computation times support the practical feasibility of the approach.

After acquisition, each robot broadcasts a subset of the acquired samples. The value of the parameter $B \in [0, 1]$ defines the fraction of the newly acquired samples which are disseminated by a robot. For example, for $B = 0$, robots do not communicate; for $B = 1$, robots exchange all acquired data, and share the same training set at any moment of the simulation; for $B = 0.1$, each robot shares $0.1 \cdot 30 = 3$ training instances. In general, when $B \in (0, 1)$, the samples to be shared are selected according to one of the three strategies described in Section III-B.

After sharing takes place, the training set of each robot includes $KM + BKM(N-1)$ training samples. For example, for $B = 0.2$ a robot r in a swarm of $N = 10$ robots would have 84 instances in its training set T_r: 24 in T_r^p (still unknown to the rest of the swarm), 6 in T_r^s (already disseminated to the rest of the swarm), and 54 in T_r^e (received from other robots of the swarm). The 60 instances in $T_r^s \cup T_r^e$ are the current *common knowledge* of the swarm.

After each robot trains its first classifier, the system enters the *interaction rounds* phase, in which we simulate 150 random commands given by the user to the swarm through gestures. Before each command is given, robot positions with respect to the hand are randomized, to simulate a realistic scenario in which commands are given at different times to the swarm, and between commands the robots perform their own tasks, which causes them being randomly scattered in the environment. For each command, each robot acquires and classifies a single image. By means of the consensus protocol described in Section II-B, the swarm eventually converges to a decision for the gesture, which can be correct or wrong. In both cases each robot in the swarm eventually acquires the correct label for the image gesture and adds the related information to the subset T_r^p of its training set.

After each group of 10 commands, robots exchange $B \cdot 10$ training samples (selected within the whole T_r^p subset, which may also include samples acquired during previous observations but not samples which have already been disseminated), then retrain their classifiers. Such a synchronized behavior is not strictly required by our approach, since robots could autonomously decide when to disseminate training data.

The average classification accuracy of the swarm is computed after every round of 10 commands. In this way, for a full simulation run we obtain 15 accuracy values measured at different stages of the swarm learning process. The first of these values, corresponding to the first set of 10 commands of the *interaction rounds* phase, is obtained using the classifiers trained only with the *initial learning* data. Subsequent values correspond to incrementally larger training sets.

For each set of simulation parameters, 50 simulation runs are performed using different realizations of the random variables (i.e., robot positions, observations' sampling from the dataset, gesture sequence). Results in the plots are the averages of the multiple runs.

The experimental protocol was explicitly designed to challenge the swarm with a rather difficult recognition problem, which is functional to our goal of measuring the impact of different learning strategies. In particular, it is important to make the following remarks:

- The source dataset is challenging, because gestures are shown in many different rotations and some are represented in different variations. Most viewpoints do not allow to distinguish different gestures, even for an human observer. Many samples in the dataset present serious segmentation issues due to several factors related to practical problems typically arising in the real-world situations as the one we consider.
- Each robot only acquires a single observation per command, and the swarm implements a straightforward consensus protocol. Our previous work [1] shows that acquiring multiple observations before issuing a decision can dramatically improve swarm accuracy.
- Robots' initial learning is based on a very small dataset.

B. Swarm size and amount of shared information

Figure 4 shows the learning curves for different swarm sizes and for different amounts of exchanged information (e.g., 20% communication means that each robot only shares 20% of its personal instances). The left plot shows that larger swarms yield significantly better accuracy at all stages of the training process. Two factors contribute to this effect:

- when $B > 0$, a large swarm is trained much faster than a small swarm, because it collectively acquires and exchanges a proportionally much larger amount of training data. A single robot, or a swarm which does not exchange training data (Figure 4 right), learns in fact very slowly;

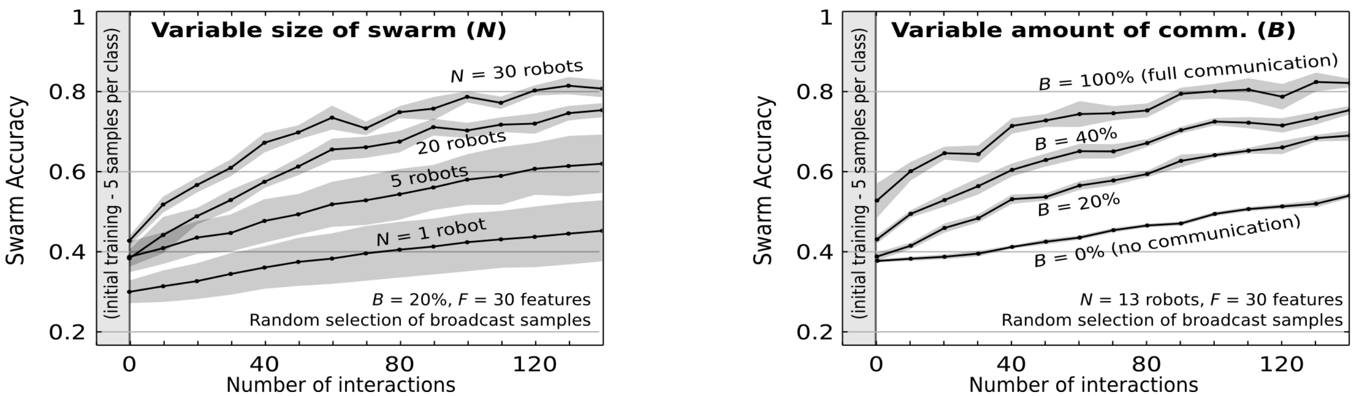

Fig. 4. Average accuracy vs. stage of training. Left: Curves for different swarm sizes, with robots sharing $B = 25\%$ of their observations using representativity-driven selection. Right: Curves for $N = 13$ robots, corresponding to different communication loads (different percentages of instances being shared among robots), $B = \{0\%, 20\%, 40\%, 100\%\}$ with random selection of instances. Grey bands correspond to confidence intervals.

- when recognizing a gesture, a large swarm enjoys a more powerful consensus ability, because more observations are accounted for.

The contribution of the former factor is explored in Figure 4 (right), which shows how communication improves the learning ability in a 13-robot swarm. The contribution of the latter factor is isolated when comparing the bottom lines of the left and right plots of Figure 4. In both cases, no communication is allowed, and each robot learns independently from the rest of the swarm. The only difference among the two scenarios is given by the size of the swarm, which affects accuracy because of the different amount of data acquired during the consensus procedure. As expected, the 13-robot swarm results to be more accurate than a single robot.

After a very large number of interaction rounds, training sets grow large enough that no further increase of accuracy is possible in practice. At this point, all scenarios reported in Figure 4 (right) are expected to yield the same accuracy.

C. Instance selection strategies

The effect of each of the three strategies for selecting the training instances to disseminate when $B \in (0, 1)$ (Section III-B) is reported in Figure 5 (right).

Surprisingly, giving priority to novel instances results in a performance which is comparable to that of purely random selection for almost the entire learning process. On the other hand, giving priority to the most representative instances leads to a significantly faster learning rate, especially during initial phases. This could be explained by considering that a representative sample somehow summarizes multiple instances, since it lies near to their centroid. Therefore, it can be more informative compared to the typical characteristics of a given class. Conversely, *novel* samples appear to be more useful later on in the learning process, probably due to their contribution in refining the decision boundaries.

D. Number of features in bandwidth-limited scenarios

Communication constraints are the main reason for considering limiting the amount of training data shared among the robots. Since training instances are exchanged as feature vectors with the respective ground truth classes, the dimensionality of these vectors is an important parameter: larger feature vectors lead to more powerful classifiers, but at the same time require more bandwidth for dissemination. In bandwidth-limited scenarios, a tradeoff emerges, because using more features implies disseminating less training instances, which has a negative impact on learning speed.

Figure 5 (left) explores such tradeoff, and reports learning curves when using different feature vector sizes, represented by the simulation parameter $F = \{2, 5, 10, 20, 30, 40, 60, 120\}$. We assume that each robot has the opportunity to disseminate a fixed amount of data, corresponding to a total of 120 features after each set of 12 commands is given (i.e., approximately 500 bytes assuming a single-precision floating point representation). Therefore, when small feature vectors are used ($F = \{2, 5, 10\}$), all acquired samples to be shared among the swarm. However, the individual classifiers are still not powerful enough, due to the poor information in the feature vectors, and yield relatively poor recognition accuracy. On the other extreme, if $F = 120$ each robot can only disseminate 1/12 of the acquired instances, such that the training sets grow at a much slower rate, leading again to slow learning. In general, relatively small feature vectors ($F = \{10, 20\}$) lead to better accuracy during the very first training stages, because they allow to quickly build moderately-sized training sets. However, in the later stages, the classifier is not powerful enough to exploit the growing size of the training set. Instead, relatively large feature vectors ($F = \{40, 60\}$) lead to suboptimal results at the beginning, but are well able to fully exploit the larger training sets accumulated in the later stages of training. Intermediate values ($F = \{30, 40\}$) lead to nearly-optimal accuracy at all learning stages.

V. Conclusions

We presented an approach for training, from scratch, a robot swarm to visually recognize hand gestures through a set of interaction rounds with a human instructor who incrementally presents training samples to the swarm. We exploited communication among the robots in order to implement swarm learning as a fully cooperative process and proposed different strategies for an intelligent, communication-aware selection of the information to be shared across the swarm. Through an extensive set of simulations based on real data, we studied the effects of swarm size and amount and quality of information used for classification. We showed that intelligent information

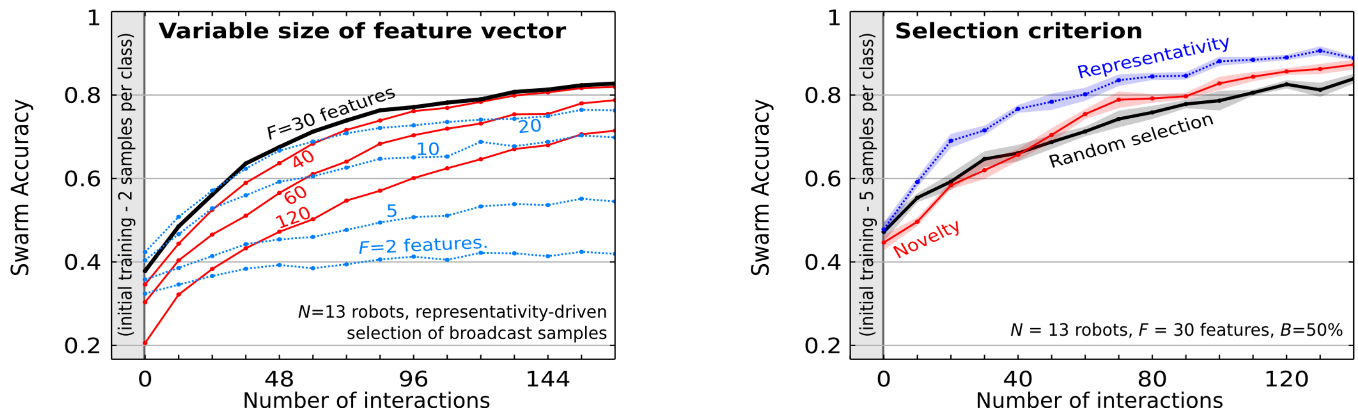

Fig. 5. Left: Experiments with different number of features. Right: Effect of different selection strategies for sharing instances.

sharing allows the swarm to learn rapidly, after a few interaction rounds. Larger swarms exhibit significantly faster learning rates and recognition accuracy than smaller ones, confirming the soundness of the approach.

In bandwidth-limited scenarios, which are a likely occurrence for swarms, prioritizing the sharing of *representative* (rather than *novel*) training instances leads to measurable performance improvements over random selection. Moreover, we observed that in such scenarios the size of the feature vectors used for learning determines a tradeoff between power of the classifier and efficiency of information sharing, with an optimal tradeoff corresponding to intermediate sizes.

Future directions will consider: novel strategies for selecting when and how robots should disseminate their data, how to let the instructor intelligently select the sequence of gesture samples based on the current swarm performance, and testing the approach on different visual and audio recognition tasks.

References

[1] A. Giusti, J. Nagi, L. Gambardella, and G. A. Di Caro, "Cooperative sensing and recognition by a swarm of mobile robots," in *Proceedings of the 25th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 551–558.

[2] C. Giraud-Carrier, "A note on the utility of incremental learning," *AI Commununications*, vol. 13, no. 4, pp. 215–223, 2000.

[3] W.-M. Shen, "Efficient incremental induction of decision lists - can incremental learning outperform non-incremental learning?" University of Southern California, Tech. Rep. USC-ISI-96-012, 1996.

[4] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[5] K. S. Yap, C. P. Lim, and I. Z. Abidi, "A hybrid ART-GRNN online learning neural network with a epsilon-insensitive loss function," *IEEE Transactions on Neural Networks*, vol. 19, no. 9, pp. 1641–1646, 2008.

[6] H.-J. Rong, G.-B. Huang, N. Sundararajan, and P. Saratchandran, "Online sequential fuzzy extreme learning machine for function approximation and classification problems," *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, vol. 39, no. 4, pp. 1067–1072, 2009.

[7] M. Karasuyama and I. Takeuchi, "Multiple incremental decremental learning of support vector machines," *IEEE Trans. on Neur. Netw.*, vol. 21, no. 7, pp. 1048–1059, 2010.

[8] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 1–27, 2011.

[9] A. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class batch-mode active learning for image classification," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 1873–1878.

[10] J. Nagi, H. Ngo, A. Giusti, L. Gambardella, J. Schmidhuber, and G. A. Di Caro, "Incremental learning using partial feedback for gesture-based human-swarm interaction," in *Proceedings of the 21st IEEE International Symposium on Robot and Human Interactive Communication (Ro-Man)*, 2012, pp. 898–905.

[11] H. Zheng, S. R. Kulkarni, and H. V. Poor, "Attribute-distributed learning: Models, limits, and algorithms," *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 386–398, 2011.

[12] H. Xiao, "Towards parallel and distributed computing in large-scale data mining: A survey," Technical Univ. of Munich, Tech. Rep., 2010.

[13] R. Polikar, "Ensemble based systems in decision making," *IEEE. Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.

[14] D. Impedovo and G. Pirlo, "Updating knowledge in feedback-based multi-classifier systems," in *Proc. of the Intl. Conf. on Document Analysis and Recognition (ICDAR)*, 2011, pp. 227–231.

[15] Y. Lu, V. Roychowdhury, and L. Vandenberghe, "Distributed parallel support vector machines in strongly connected networks," *IEEE Transactions on Neural Networks*, vol. 19, no. 7, pp. 1167–1178, 2008.

[16] T. Alpcan and C. Bauckhage, "A discrete-time parallel update algorithm for distributed learning," in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR)*, 2008, pp. 1–4.

[17] P. Tangamchit, J. M. Dolan, and P. Khosla, "Crucial factors affecting cooperative multirobot learning," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, 2003, pp. 2023—2028.

[18] A. Navia-Vazquez, D. Gutierrez-Gonzalez, E. Parrado-Hernandez, and J. J. Navarro-Abellan, "Distributed support vector machines," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 1091–1097, 2006.

[19] D. Cohn, Z. Ghahramani, and M. Jordan, "Active learning with statistical models," *arXiv preprint cs/9603104*, 1996.

[20] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, D. Burnier, A. Campo, A. L. Christensen, A. Decugnière, G. A. Di Caro, F. Ducatelle, E. Ferrante, A. Förster, J. Guzzi, V. Longchamp, S. Magnenat, J. Martinez Gonzales, N. Mathews, M. Montes de Oca, R. O'Grady, C. Pinciroli, G. Pini, P. Rétornaz, J. Roberts, V. Sperati, T. Stirling, A. Stranieri, T. Stützle, V. Trianni, E. Tuci, A. E. Turgut, and F. Vaussard, "Swarmanoid: A novel concept for the study of heterogeneous robotic swarms," *IEEE Robotics & Automation Magazine*, vol. 20, no. 4, pp. 60–71, 2013.

[21] I. Valavanis and D. Kosmopoulos, "Multiclass defect detection and classification in weld radiographic images using geometric and texture features," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7606–7614, 2010.

[22] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in *Proceedings of the 17th International Conference on Machine Learning (ICML)*, 2000, pp. 839–846.

[23] M. Bonani, V. Longchamp, S. Magnenat, P. Rétornaz, D. Burnier, G. Roulet, F. Vaussard, H. Bleuler, and F. Mondada, "The marXbot, a Miniature Mobile Robot Opening new Perspectives for the Collective-robotic Research," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 4187–4193.