2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Online Feature Extraction for the Incremental Learning of Gestures in Human-Swarm Interaction

Jawad Nagi, Alessandro Giusti, Farrukh Nagi, Luca M. Gambardella, Gianni A. Di Caro

***Abstract*— We present a novel approach for the online learning of hand gestures in swarm robotic (multi-robot) systems. We address the problem of online feature learning by proposing *Convolutional Max-Pooling* (CMP), a simple feed-forward two-layer network derived from the deep hierarchical Max-Pooling Convolutional Neural Network (MPCNN). To learn and classify gestures in an online and incremental fashion, we employ a 2nd order online learning method, namely the Soft-Confidence Weighted (SCW) learning scheme. In order for all robots to collectively take part in the learning and recognition task and obtain a swarm-level classification, we build a distributed consensus by fusing the individual decision opinions of robots together with the individual weights generated from multiple classifiers. Accuracy, robustness, and scalability of obtained solutions have been verified through emulation experiments performed on a large data set of real data acquired by a networked swarm of robots.**

I. Introduction

Human-Swarm Interaction (HSI) is a young line of research that investigates *interfacing mechanisms* and *communication strategies* for interaction and cooperation between humans and robot swarms. If swarm robotic systems are to be used in human environments, efficient *online incremental learning mechanisms* are desirable, in order for human operators to reliably make a swarm of robots learn different commands (e.g., to perform a joint mission) on the spot and based on a few given samples. Existing mechanisms for facilitating interaction between humans and swarm robotic (multi-robot) systems include the use of face and gaze detection [1] and human body (skeleton) postures [2]. In our previous work [3], [4], [5], [6] we focused on the use of *hand gestures*, as a way to provide commands to individual or groups of robots situated in the physical proximity of the human operator. Compared to other interfaces such as faces, facial expressions, gazing, and body postures, hand gestures are easily recognizable [7] and have the advantage of being easy, natural, and intuitive for use. Because of these nice properties they have been extensively employed in human-robot interaction [8], [9].

In order to allow the robots in a swarm to effectively learn and recognize hand gestures presented by human operators, in our previous work [3] we made use of a Max-Pooling Convolutional Neural Network (MPCNN) [10], a big and deep variant of the Convolutional Neural Network [11] (CNN). However as the MPCNN is a batch-learning algorithm (derived from the CNN), it does not provide the capability to learn in an online fashion, and requires collecting a large number of labelled samples for training. This is time consuming, computationally expensive, and might not be really feasible in practice.

To overcome the limitations of batch-learning algorithms such as CNNs, in other works [5] we defined a prior set of *geometrical and shape properties* and computed numerical hand-crafted features in an online and incremental fashion using these properties. However, we cannot guarantee that such hand-crafted features can provide an optimal representation of a gesture (i.e., hand silhouette/contour). Additional geometrical and shape properties could be derived to make the feature representation more robust. This would proportionally increase the computational burden, and still would not provide any guarantees on how good (or redundant) feature descriptors are.

In [4] we went a step further, by proposing a system which allows effective online incremental learning. The human operator can provide only a binary (yes/no) feedaback, and not anymore full labeled samples. However, also in this case we are using the hand-crafted geometrical and shape properties to extract the features for learning. Therefore, in this paper we want to build a fully online and incremental system, focusing on the problem of *online feature learning* from segmented image masks. At this aim we introduce *Convolutional Max-Pooling* (CMP), a biologically inspired approach derived from the MPCNN. The idea is that the relevant hand features are extracted automatically from the images, online, and passed to the machine learning component. This approach is expected to be intrinsically more robust than the hand-crafted one, and allows to automate online the entire gesture learning and recognition process.

This paper is organized as follows. Section II discusses the related work in different domains. Section III presents the contributions of our work, i.e., the algorithms and methods for solving the aforementioned issues. Section IV presents the findings of the experiments evaluated on real data, whereas Section V presents concluding remarks.

II. Related Work

A. Human-Swarm Interaction

Being a relatively new area of research, human-swarm interaction (HSI) aims on investigating approaches for interaction and cooperation between humans and swarms. The majority of work for interacting with multiple robots [12] using distributed forms of sensing has been recently proposed

J. Nagi, A. Giusti, L. Gambardella and G. A. Di Caro are with the Dalle Molle Institute for Artificial Intelligence (IDSIA), Lugano, Switzerland. email: jawad,alessandrog,luca,gianni@idsia.ch
F. Nagi is with the Department of Mechanical Engineering, University Tenaga Nasional (UNITEN), Putrajaya, Malaysia. email: farrukh@uniten.edu.my

978-1-4799-3684-7/14/$31.00 ©2014 IEEE

by the research group of Vaughan [13], [14], [2], [1], [15]. In [13], [14] they developed an eye contact approach based on gaze detection (using face detection) for interacting with the robots in a multi-robot system. Many other recent works of the same group in this domain [2], [1], [15] have adopted gaze detection as a means of face engagement for initiating interaction between multiple robots and a human. Our approach is different, as we consider the use of *hand gestures* naturally supplemented by fine controls through tangible input gadgets (i.e., colored gloves).

B. Hand Gestures

Recognition of hand gestures is an active area of research, with motivating applications such as human-robot interaction (HRI) [16], sign language interpretation, gaming controllers, virtual reality, and assistive environments. During the last decade numerous research efforts have promoted the use of hand gestures as effective interfacing mechanisms for the control of robots by human operators [8], [9]. However, Hand shape recognition [17], [7] using vision is a challenging problem in the pattern recognition, due to the ambiguities associated with different hand poses and angles, finger flexibility, and computer vision problems associated with different lightning conditions and partial occlusions. However, with the use of a distributed sensing system (i.e., a networked swarm robots of with onboard cameras as in our case), these limitations can be partially overcome by leveraging on the presence of multiple input that can be fused together. This makes robot swarms as potentially quite robust to learn and identify in real-time hand gestures presented by human operators within physical proximity.

C. Online Feature Learning and CNNs

Efficient feature learning is one major factors involved in the success of image learning and recognition systems. It requires that features have the most discriminative characteristics among different classes, while retaining *rotational*, *translational* and *scale* invariance characteristics within the same class. However, currently existing hand-crafted feature learning schemes such as connected-component analysis, image moments, and geometrical shape properties, do not guarantee the optimal representation of an object in an image.

During the past three decades, many research works have focused on automated feature learning approaches, based on raw pixel intensities in the images. Many of these works are based on the observation that the human visual system efficiently recognizes and localizes objects within cluttered scenes. As the mammalian visual cortex is the most powerful vision system in nature, many authors have proposed biologically inspired approaches which emulate its behavior. The NeoCognitron receptive fields discovered from Hubel & Wiesel's [18] classic work on the *cat's primary visual cortex* is considered as the first extensive use of receptive fields in hierarchical neural systems, that have inspired many of the more recent variants such as Convolutional Neural Networks (CNNs) [11], [19].

CNNs based on *Multi-Stage Hubel-Wiesel Architectures* have been used in many real-world applications such as generic object recognition, optical character recognition (OCR), face recognition [20]. Hubel and Wiesel identified that, there exists a complex arrangement of *simple* and *complex* cells within the visual cortex, which are sensitive to small sub-regions of the input space, i.e., the *receptive field*. The filters present in this local in input space are best suited to exploit the strong spatially local correlation present in natural images [21]. Since CNNs belong to a wide class of trainable multi-stage feature learners, their feature representations are known to be hierarchical [22]. This suggests that feature learning architectures should have multiple layers stacked on top of each other, one for each level in the feature hierarchy.

Current state-of-the-art hierarchical feature learning systems compute all localized features from input images, convolving image patches with filters. Filter responses are repeatedly pooled and re-filtered, resulting in features that are computed using deep feed-forward network architectures. The most recent and successful variant of the CNN following this scheme, is the *Max-Pooling CNN* (MPCNN) [10], a trainable deep hierarchical neural network with alternating convolutional and max-pooling layers, that replicates orientation-selective *simple cells* with overlapping local receptive fields, and *complex cells* performing feature selection (pooling) operations with non-overlapping receptive fields.

While CNNs offer the advantages of robust feature learning from images, they are batch-learners and do not provide capabilities of online learning for real-world applications, and require large number of labeled samples for training prior to the testing (validation) phase. To overcome these limitations of CNNs, we introduce *Convolutional Max-Pooling* (see Section III-C) an online feature learning scheme inspired from the MPCNN, that does rely upon a training mechanism and is suitable for online learning with robot swarms.

III. Methods

By exploiting the properties of distributed mobile sensing in swarm robotic systems (as presented in our previous work [5]), we allow an entire robot swarm to act as a single powerful augmented sensor (i.e., to concurrently gather perceptual information), while offering robustness to individual robot failures. With the use of local mobility rules [5] our robot swarm can effectively position itself surrounding a human operator. This enables a swarm to effectively sense and learn mission commands (signals) from human operators. In this work, the sensing, learning and recognition of gestures initiates after robots in a swarm have detected a human in the environment, which is achieved by detecting a tangible input device (i.e., an orange colored glove in our case) using the on-board cameras of the robots (see Figure 1). The online incremental learning process is described as follows.

After a glove has been detected, the human operator presents a gesture to the swarm. Each robot in the swarm processes an acquired image (gesture observation) and computes relevant and meaningful information (i.e., statistical features)

regarding the presented gesture, using our CMP network (see Section III-C). Using the features computed from an observation, each robot incrementally learns multiple classes of hand gestures (see Section III-D) from a predefined set of known gesture classes (see Section III-A) using a multi-class supervised machine learning approach. The classification output for predicting a single gesture produces a *posterior probability vector* (a decision of the recognized gesture) for each robot, that enables a swarm of robots as a whole to reach a consensus (i.e., an overall swarm decision; mutual agreement) about the gesture issued by the human. For every new gesture presented by the human to the swarm, this entire process iterates in an online incremental fashion.

A. Gesture Vocabulary

In order to refer to a practical set of gestures for learning in a purely online and incremental setting, we defined a simple vocabulary of $K = 6$ hand gestures (represented by *finger counts* from 0 to 5) as illustrated on the top left of Figure 1. This vocabulary serves as a simple interaction and communication protocol between humans and robot swarms, and could provide a basic set of control commands for tasks that can be executed by a swarm.

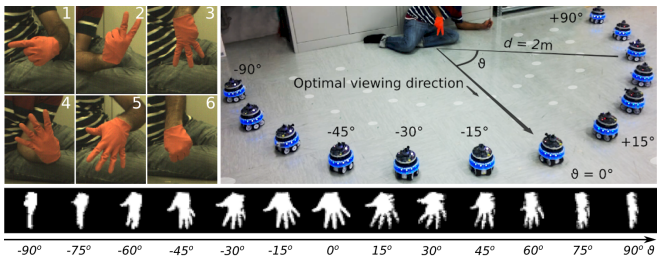

Fig. 1: Top left: Vocabulary of $K = 6$ hand gestures, representing *finger counts* from 0 to 5. Top right: Swarm formation used for dataset acquisition ($d = 2$m). Bottom: Segmented hand gesture masks (contours) after color segmentation, corresponding to the top right Figure.

B. Image Preprocessing

After a glove has been detected, the next step is to separate the hand gesture from the image background. As we adopt a tangible input (i.e., a glove) with a known characteristic color, this facilitates fast and efficient retrieval of the hand contour (silhouette) corresponding to the presented gestures. At this aim, we employ a standard *color-based segmentation* approach in the HSV (Hue, Saturation, Value) color space to separate hand gestures from the entire image background.

As the hand projection normally covers a very small fraction of area (number of pixels) in an image, we convert the acquired RGB images into HSV color space so that lighting variations are confined to the V (Brightness) channel, whereas the H (Hue) and S (Saturation) channels identify unique colors (i.e., the area of the H,S plane correspond to the glove color). In our case, a simple rectangular area identified by the parameters $\{H_{\min}, H_{\max}, S_{\min}, S_{\max}\}$ provides satisfactory segmentation when supplemented by an additional constraint of $V_{\min}$ (useful to discard dark areas of the image which provide very unreliable Hue pixels). These five parameters can be easily estimated from a single image, and are fixed given the type of illumination conditions in the environment (i.e., fluorescent or natural light).

Feature learning by replicating the receptive fields of the visual cortex, requires all binary images (segmented images with $[0,1]$ pixels) to be of equal size and in a square region of interest. At this aim, we perform an inspection of the size distribution of all segmented images from our dataset, and determine a dimension of 24 pixels (similar to the MNIST database [11] of characters) to be an appropriate size for representing the segmented hand contour masks. Next, all images are rescaled to 24×24 pixels while keeping the aspect ratio into consideration. Finally, the resized images are padded with 4 background pixels (pixels with intensity value 0) on each side, resulting in a square ROI of 28×28 pixels (i.e., a dimension size $N = 28$) with the segmented hand gesture mask being centered in the square ROI, as illustrated in the bottom corner of Figure 1.

C. Convolutional Max-Pooling

The living mammalian retina represents the visual world in a set of about a dozen different *feature detecting* parallel representations. Inspired by Hubel & Wiesel's [18] work on the mammalian virtual cortex and by Convolutional Neural Networks [11]), we consider an MPCNN [10], that alternates convolutional with max-pooling layers, as a basic building block. Emulating the receptive fields in the mammalian visual cortex, we propose an online incremental feature learning scheme, *Convolutional Max-Pooling* (CMP) as illustrated in Figure 2, that comprises of a simple two-layer feed-forward network derived from the deep hierarchical MPCNN architecture.

We make use of a two-layers CMP network, with one convolutional layer and a sub-sampling layer (that performs the max-pooling operation), as shown in Figure 2. The inputs to the CMP network are the segmented, rescaled, and padded binary images of 28×28 pixels. As simple-cell receptive fields in the mammalian cortex can be described by 2D Gabor functions (i.e., filters that are selective for orientation), in the first layer of the CMP we employ an overlapping *Convolutional kernel* of size (C_x, C_y) (i.e., a bank of Gabor filters) and shift over all the valid sub-regions in the image (using convolution with local filters to compute higher-order features), such that the kernel is completely inside the image. This is achieved by centering the 2D Gabor filters with a full range of orientations at each possible position and scale. In our implementation, we use 8 orientations and 16 different scales, which correspond to a bank of 128 Gabor filters.

The second layer of our CMP network performs sub-sampling, where input images are tiled in non-overlapping subregions from which only one output value is extracted. This results in two common choices applicable for pooling: *maxima* and *average*, usually referred to as Max-Pooling and Avg-Pooling respectively. We adopt Max-Pooling as it introduces small invariance to translation and distortion, and

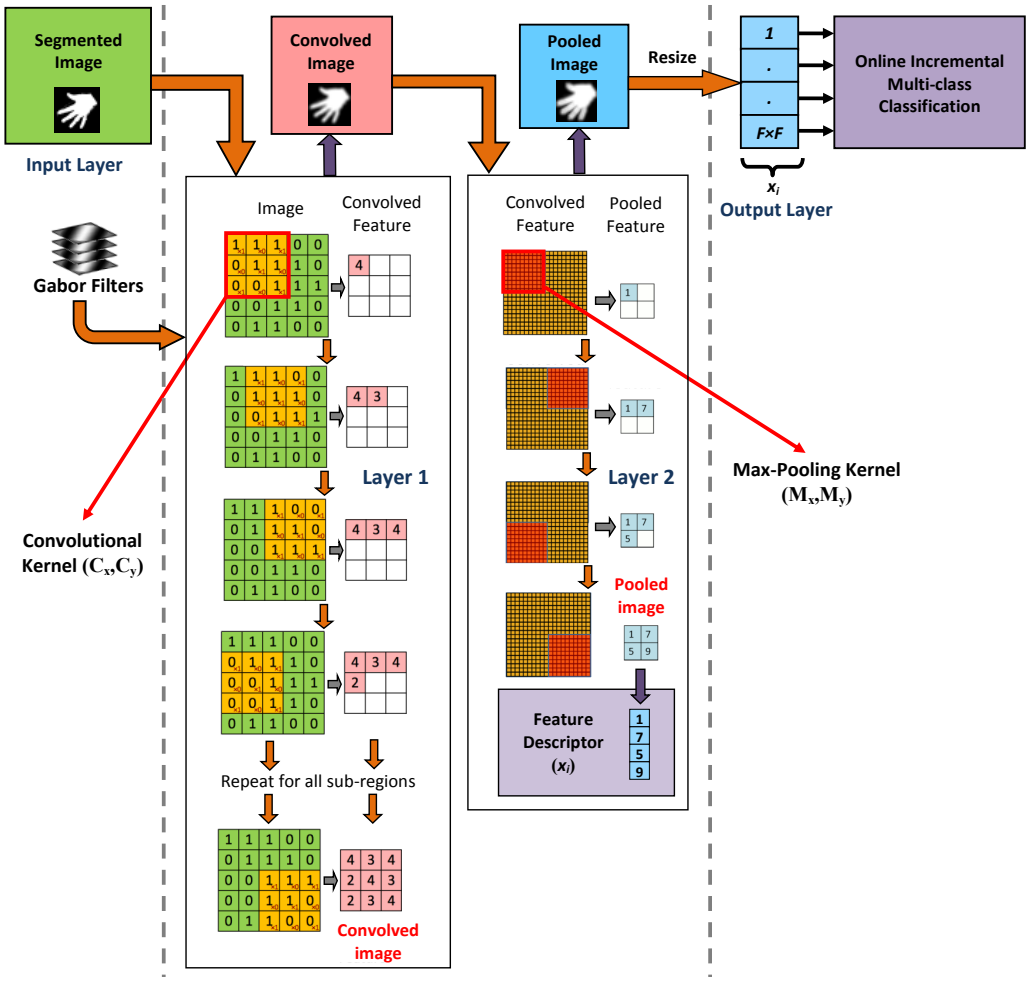

Fig. 2: Feature computation using a two-layer *Convolutional Max-Pooling* (CMP) network.

leads to faster convergence and better generalization [23]. In the second layer we adopt a non-overlapping *Max-Pooling kernel* of size (M_x, M_y) pixels and shift it over the regions of the convolved image, where the output of each region is computed using the maximum activation over non-overlapping rectangular regions of the kernel size. In practice, the 128 feature maps generated for an input image after convolution with the filterbank, are used for the max-pooling operation. The resultant of the second layer is computed by performing a sum over all the convoluted and pooled input feature maps. As max-pooling non-linearly downsamples an image by a factor of M_x and M_y along each direction, the scope of the max-pooling layer in our approach not only reduces the computational burden, but also provides sparser representations as well as a form of translation invariance and, also performs feature selection which is considered a smart way to reduce the dimensionality of intermediate representations.

The output from our two-layer feed-forward network is a downsampled grayscaled image (with $[0,255]$ intensity pixels), which after reshaping as a 1-dimensional vector corresponds to a *feature descriptor* $\mathbf{x}_i$ with $F \times F$ elements, where $F = N - ((C_x, C_y) + 1) - ((M_x, M_y) + 1)$, as illustrated by the pseudocode in Algorithm 1. As our CMP network computes $\mathbf{x}_i$ independently from every acquired gesture observation, these feature descriptors can be directly used with supervised machine learning algorithms for classification and regression tasks. In order to reduce the dimensionality of the large feature space without significantly deteriorating the discriminative quality of the learned features, smaller image dimension sizes (e.g. $N < 20$) can be used.

Compared to trainable feature learning architectures such as CNNs (batch-learning algorithms), our CMP network does not require a training phase and provides the capability to incrementally (instantaneously) learn features as new observations are sensed and acquired. Moreover, using our approach features are learned independently, irrespective of the class/category, which is not the case with CNNs, as they require observations from all classes for training.

D. Online Incremental Multi-Class Classification

In distributed learning environments where different predictors are trained on different portions of the data in parallel, model fusion is a useful approach to combine multiple learned predictors to produce a better single predictor and to reduce duplicated learning efforts. At this aim, we adopt the recently proposed 2nd-order *Soft-Confidence Weighted*

(SCW) online learning scheme [24], as it offers an informed and effective way to fuse different classifiers after distributed training, while at the same time provides an efficient approach for online multi-class learning.

Algorithm 1 Two-layer Feed-forward CMP Network.

```
I ← (segmented image of dimension N)
for i = 1:(# overlapping subregions of size (C_x, C_y) in I) do
    for j = 1:(# orientation-selective Gabor filters) do
        Convolution of every subregion i using (C_x^j, C_y^j)
        Generate a feature map I_i^j for every Gabor filter j
    end for
end for
for k = 1:(# feature maps (I_i^k ∈ I)) do
    for m = 1:(# non-overlapping subregions (M_x, M_y) in I_i^k) do
        Max-pooling of every subregion m using (M_x, M_y)
    end for
    Sum every feature map I_i^k into I^cp
end for
x_i ← resize(I^cp); //Feature descriptor
```

As each robot in our swarm is equipped with its own classifier, and multiple robots in the swarm learn the same classification task in parallel from different points of view, fusing multiple classifier models is of fundamental interest in a swarm. Considering a set of robots in a swarm $\{r_k\}_{k=1}^{N_{robots}}$ where r_k corresponds to the kth robot, the combined model of all robots, also a Gaussian, can be computed as the one that is closest to all other k distributions in the sense of a chosen divergence (e.g. distance measurement). For instance, in the case of KL divergence the combined model parameters $(\bar{\mu}_i, \bar{\Sigma}_i)$ for each ith binary classifier are given by: $\bar{\Sigma}_i = \left(\sum_{k=1}^{K} (\Sigma_k^i)^{-1}\right)^{-1}$ and $\bar{\mu}_i = \bar{\Sigma}_i \sum_{k=1}^{K} (\Sigma_k^i)^{-1} \mu_k^i$ where (μ_k^i, Σ_k^i) denotes the ith binary classifier of the kth robot M_k.

To improve the efficacy of 1st-order learning methods and to better explore the underlying structure between features, the SCW learning scheme [24] addresses the problems of the confidence-weighted (CW) learning strategy [25], [26], by applying the soft-margin idea in Support Vector Machines (SVMs) [27] and variants of Passive-Aggressive (PA) algorithms [28] to the CW learning method. The SCW learning scheme suggests that low-confidence feature weights should be updated more aggressively than high-confidence ones.

In 2nd-second order learning methods, the weights of a linear classifier are associated to confidence information via a multivariate diagonal Gaussian distribution, with mean vector $\mu \in \mathbb{R}^d$, and covariance matrix $\sum \in \mathbb{R}^{d \times d}$. On receiving a length-$d$ feature vector $\mathbf{x}_i \in \mathbb{R}^d$ at iteration i, the SCW classifier draws a weight vector $w_t \sim \mathcal{N}(\mu_i, \Sigma_i)$ and predicts the corresponding label as $\mathrm{sign}(\mathbf{w}_i \cdot \mathbf{x}_i)$. The absolute value of the prediction margin $|\mathbf{w}_i \cdot \mathbf{x}_i|$ is interpreted as proportional to the confidence level in predicting the label.

When the true label $y_i \in \{-1, +1\}$ is revealed, the SCW learner updates the weight distribution $\mathcal{N}(\mu, \Sigma)$ by minimizing the Kullback-Leibler divergence between the current (new) weight distribution $\mathcal{N}(\mu_i, \Sigma_i)$ and the old one, while ensuring that the probability of correct classification for the current observation is no smaller than the confidence parameter $\eta \in (0.5, 1)$, where η is the probability required to update the distribution on the current observation. The SCW employs parameter $C > 0$ to control the trade-off between conservativeness and aggressiveness (i.e., the trade-off between keeping the previous information and minimizing the current loss), and recasts the CW constraint as an adaptive regularizer in an unconstrained minimization problem on each learning iteration.

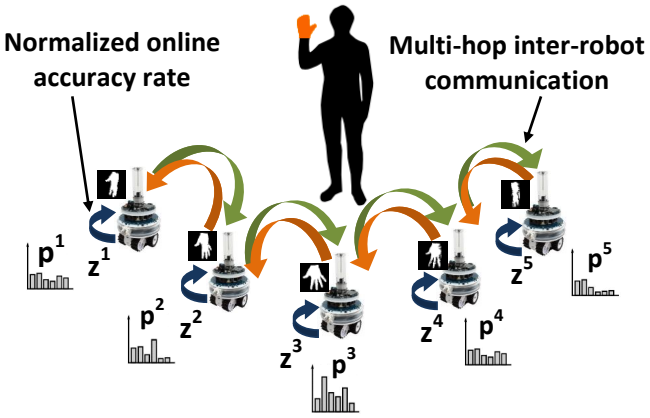

Fig. 3: A robot swarm engaging in a distributed consensus.

The outcome for classifying a feature descriptor $\mathbf{x}_i$ using a learned SCW model results in a *posterior probability vector* $\mathbf{p}$, generated by each robot (i.e., each robot's *opinion* regarding the decision of the recognized gesture). For a classified feature descriptor, its posterior probability can be represented by $\mathbf{p_j} \in \{p_1, p_2, ..., p_K\}$, where $j = \{1, ..., K\}$ and $K = 6$ represents the number of gestures in the vocabulary. To make things simpler, we conveniently normalize posterior probability vectors, so that the sum of each probability vector equals to 1 over all K classes, such that $\sum_{j=1}^{K} \mathbf{p}_j = 1$. The largest element in p_j, namely $\arg\max\{\mathbf{p}\}$ for each robot, represents the predicted gesture class.

E. Multi-robot Implementation

In our multi-robot implementation, we consider a swarm of $\{r_i\}_{i=1}^{N_{robots}}$, where each robot r_i is equipped with an individual *feature learning* and *classification* module. Combining the opinions generated by each individual robot $\mathbf{p}^{\mathbf{r}_i}$ for a classified gesture, provides a strategy to boost the overall recognition performance individual robots and obtain an overall swarm-level decision (or mutual agreement) regarding the gesture presented by the human. In order to obtain a swarm-level prediction of a hand gesture which accounts for all the different viewing positions of robots in the swarm, we implement a *distributed consensus* protocol. Figure 3 depicts a scenario where a swarm of 5 robots collectively and cooperatively decide the gesture presented by a human operator by building a consensus.

We estimate the online learning confidence of each robot in the swarm by introducing an online accuracy rate, $oa_t^{\mathbf{r}_i} = (\sharp correct predictions / \sharp samples)$ computed at time step t. To obtain a weight that provides a ratio of every robot's learning

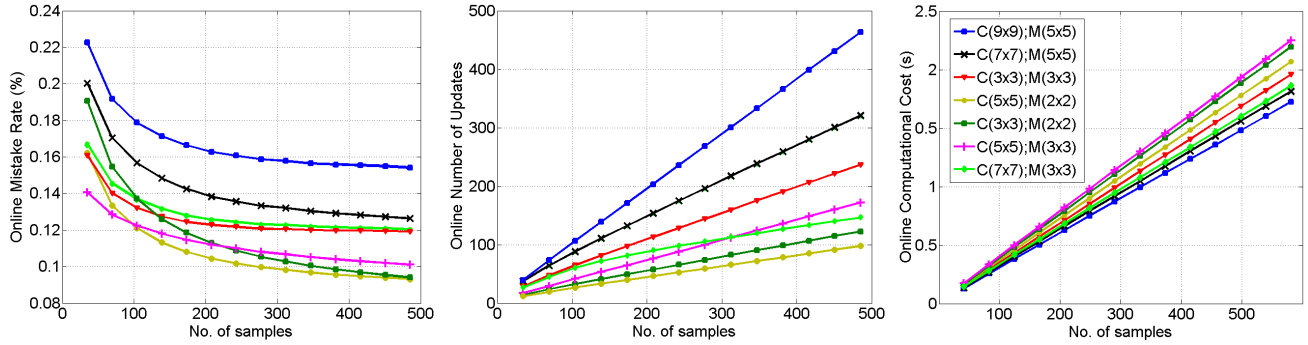

Fig. 4: Effect of the convolutional and max-pooling kernel sizes ((C_x, C_y) and (M_x, M_y) respectively) on the performance of the SCW learning scheme. Left: The online mistake rate of the SCW learning scheme vs. the number of samples learned. Center: The number of online updates (samples) used for training by the SCW vs. the number of samples observed. Right: The online computation cost (in seconds) to perform online updates.

confidence with respect to all other robots in the swarm, we compute a *normalized online accuracy weight* (NOAW), $\mathbf{z}_t^{\mathbf{r}_i} = \left(oa_t^{\mathbf{r}_i} / \sum_{j=1}^{N_{robots}} oa_t^{\mathbf{r}_j} \right)$ (a scaled value between a closed interval of $[0,1]$). To build a distributed consensus we weightage the classifiers (robots) with respect to their learned expertise by computing:

$$\lambda_t = \underset{\mathbf{p}}{arg\,max} \left(\frac{\sum_{i=1}^{N_{robots}} \left(\mathbf{z}_t^{\mathbf{r}_i} \cdot \mathbf{p}^{\mathbf{r}_i} \right)}{\sum_{j=1}^{N_{robots}} \mathbf{p}^{\mathbf{r}_j}} \right) \tag{1}$$

where $\mathbf{z}_t^{\mathbf{r}_i}$ represents the individual weight of each robot at time step t, $\mathbf{p}^{\mathbf{r}_i}$ represents the posterior probability vector of a robot, and λ_t represents the swarm-level decision (i.e., the outcome of distributed consensus) for the presented gesture.

IV. Results and Discussion

To demonstrate and quantify the capabilities of the developed system, we performed experiments investigating performance, robustness, and efficiency of the solutions in Section III. For multi-class classification, we adopt LIBOL [29], a library for online learning algorithms that provides a C++ implementation of the SCW learning scheme.

A. Data Acquisition

To allow flexibility in making different experiments, we acquired a large dataset of gesture images and used them to perform emulation experiments. In particular, we built a dataset of images using a swarm of 13 foot-bot robots [30] equipped with front-mounted cameras that acquired RGB images in a native resolution of 512×384 pixels (0.2 megapixels) with an aspect ratio of 1.33:1 (4:3). Using 13 robots we could acquire relatively large amounts of images of our gesture vocabulary from multiple points of view. The images, once labelled with their known ground truth, are used to learn qualitative and meaningful features, and then are used for online incremental learning (training) and recognition (testing). To acquire the dataset, robots were positioned at evenly-spaced angles of 15° forming a semi-circle centered around on the human operator showing the gesture, as illustrated on the top right of Figure 1. Using this configuration, each robot acquired and stored approximately 180 unprocessed images while a human operator for a short time presented gestures directed towards the robot precisely in front (at 0°; optimal viewing angle). With the use of a vocabulary of $K = 6$ hand gestures, in total the swarm roughly acquired $13 \times 180 \times 6 = 14,000$ images. This process was repeated 5 times, once for a different distance $d = \{1,2,3,4,5\}$m between the robots and the human showing the gesture. This resulted in a dataset of approximately $70,000$ images acquired by the robot swarm from a total of $13 \times 5 = 65$ different viewpoints.

As the dataset comprises of 6 different classes of gestures with measurements taken at 13 different points of view and 5 different distances, and as hand gestures are presented with *rotational*, *translational* and *scale* invariances, online incremental learning of such a dataset is considered a challenging problem in the pattern recognition community.

The acquired dataset is used for running quantitative emulation experiments: gesture observations are sampled from this dataset of real images both for learning and classification. Using ground truth (pre-labelled information) from a gesture observation, a simulated robot positioned at $(d;\ \theta)$ in the polar plane centered on the human hand 'sees' an observation that is randomly selected from the subset of the gesture observations that were acquired (during dataset collection) from the viewpoints closest to $(d;\ \theta)$. We conduct experiments by first determining the best parameters of the SCW learning scheme using a cross validation (CV) approach on our entire dataset. All results are averaged over 1000 trials, where on each trial randomly permuted sequences from the entire dataset are used for subsampling.

B. Effect of Kernel Sizes on Learning Performance

We first study the effect of using different convolution (C_x, C_y) and max-pooling (M_x, M_y) kernel sizes on the performance of the SCW learning scheme, as shown in Figure 4. For instance, $C(5,5)$ represents a convolution kernel of size (5×5), and similarly $M(3,3)$ indicates a max-pooling kernel of size (3×3). The online mistake rate, as shown on the left of Figure 4, is highly influenced by the size of the kernels

used to learn and represent the features. It results that the size of the max-pooling kernel should always be small (e.g. 3×3 or 2×2), so that during pooling operations there is not a heavy loss of information in the quality of the features. Convolution kernels with sizes greater than (5×5) do not provide good representations of the features, due to the many invariances (rotation, scale and translation) in our dataset. The most optimal combination of the kernel parameters that produce lowest mistake are convolution kernel of size (5×5) and max-pooling kernel of size (2×2).

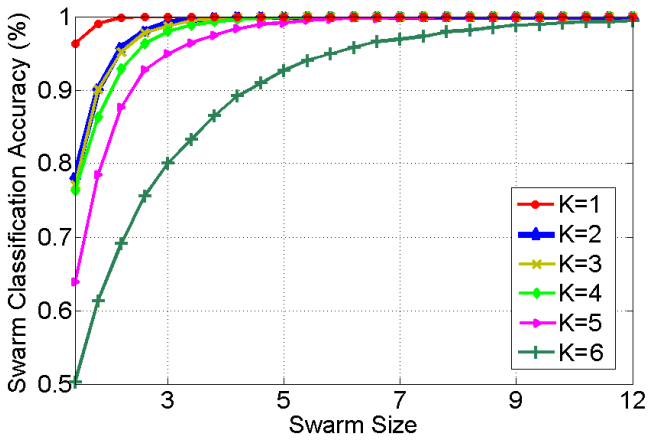

Fig. 5: Effect of the number of robots in the swarm on the classification accuracy of the of $K=6$ gesture classes.

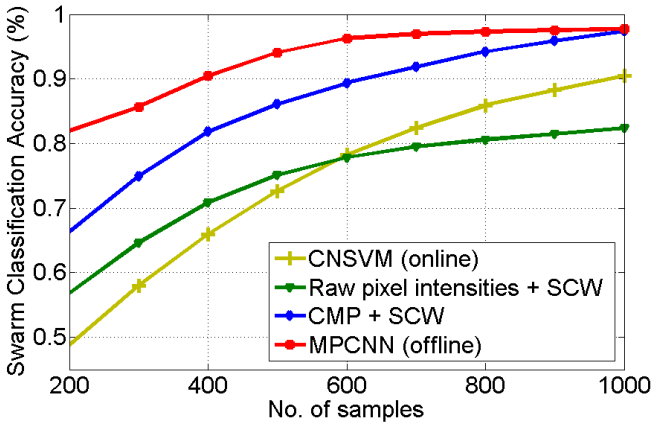

Fig. 6: Swarm classification accuracies for different learning schemes.

The online update rates are shown in the center subplot of Figure 4. The runtime characteristics are similar to those of the online mistake rate. The online update results indicate that, the lesser the mistakes (or higher learning confidence) of an online learning algorithm, the lesser the number of online updates it requires, which holds true for online learning algorithms. Similarly, the results on the right of Figure 4 represent the online computational cost (time taken; in seconds). Using larger kernel sizes decreases the computational time: this is because large size kernels reduce the number of sub-regions in the image, thereby reducing the overall computational time. Overall, it is suggested that, smaller kernel sizes are more advantageous, as they offer lower mistake rates and require lesser online updates.

C. Impact of Swarm Size on Classification Accuracy

We study the impact of a swarm's classification accuracy on its size, as shown in Figure 5. The value of K represents the 6 hand gestures corresponding to Figure 1. It can be observed that simple gestures, such as the closed hand (fist), or gestures with one or two fingers, are more easy classifiable than gestures with three or more finger counts. There are two main reasons for this. The first is that, due to color-based segmentation errors, some fingers of the hand maybe may not be detected, and the second is that, as our dataset comprises of images from multiple points of view, finger counts cannot be easily estimated (even by humans) from bad points of view (angles). The more number of fingers there are in a gesture, the difficulty in correctly classifying that gesture increases. Increasing the swarm size has positive results on the overall classification performance of the swarm, and the more number of robots are used, the better will be the overall swarm classification performance.

D. Comparison with Existing Approaches

We perform a comparative analysis of the performance of our CMP network against different feature learning approaches, as shown in Table I. Using all the available samples in our gesture dataset, we compute features using different approaches and adopt the SCW learning scheme for computing the 10-fold CV accuracy as a measure for estimating the discriminative power of the features. Our two-layer CMP network (with convolution and max-pooling kernel sizes of 5×5 and 2×2 pixels respectively) results in well-behaved feature descriptors and outperforms existing approaches in literature that have been adopted for similar learning and recognition tasks (from segmented image masks). Compared to MPCNNs, the quality of the learned features using CMP is lower, since it lacks a trainable architecture.

TABLE I: Evaluation of different feature learning approaches for gesture recognition

Feature Learning Approach	Training	CV Accuracy
Fast Fourier Transform [31]	Online	65.88%
Raw Pixel Intensities	Online	70.56%
Skeletonization [32]	Online	72.04%
Hu Invariant Moments [33]	Online	77.46%
Geometric Shape Properties [34]	Online	79.15%
Convolutional Max-Pooling (CMP)	Online	89.37%
MPCNN [3]	Offline	96.52%

We also performed an analysis considering our previous work, by comparing the performance of our proposed system using a swarm of 10 robots with that of a batch learning system, as illustrated in Figure 6. As the performance of batch-learning (offline) algorithms (offline; MPCNN) is remarkably better both in terms of feature learning and multi-class classification, our CMP integrated with the SCW learning scheme performs exceptionally well, in spite lacking

a deep hierarchical architecture. Although our CMP network requires a larger amount of training samples to achieve the same generalization performance as the batch-learner, the CMP provides online feature learning capabilities at a much lower computational cost.

V. Conclusions

In this paper, we presented an online incremental approach towards the problem of online automatic learning and recognition of hands gestures by robot swarms. At this aim, we introduced Convolutional Max-Pooling (CMP), a biologically inspired approach derived from the alternating convolutional and max-pooling layers of the MPCNN, for the online extraction of features from gesture images. To experimentally evaluate the performance of our proposed architecture a large amount of real data was gathered using a swarm of robots, and emulation tests were performed. The good performance reported in the experiments justifies our use for addressing robust and scalable solutions for the problem of online learning and recognition of gestures for human-swarm interaction.

One of the limitations of our approach is the large feature space generated after passing a segmented image mask through a small two-layered network. This is computationally heavy for the classifier, especially for datasets with large number of classes. At this aim, in future we work will consider reducing the dimensionality of the feature space by adopting a hierarchical configuration of spatial pyramid max-pooling, that is known to reduce the curse of dimensionality.

Acknowledgments

This research was supported by the Swiss National Science Foundation (SNSF) through the National Centre of Competence in Research (NCCR) Robotics (`www.nccr-robotics.ch`). The authors participate in Project 4: *Distributed Robotics*, sub-project 4.5, *Symbiotic human-swarm cooperation*.

References

[1] S. Pourmehr, V. M. Monajjemi, R. T. Vaughan, and G. Mori, "You two! take off!: Creating, modifying and commanding groups of robots using face engagement and indirect speech in voice commands," in *Proc. of the IEEE Intl. Conf. on IROS*, 2013.

[2] S. Pourmehr, M. Monajjemi, J. Wawerla, R. T. Vaughan, and G. Mori, "A robust integrated system for selecting and commanding multiple mobile robots," in *Proc. of the IEEE ICRA*, 2013.

[3] J. Nagi, F. Ducatelle, G. A. D. Caro, D. Cireşan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. M. Gambardella, "Max-pooling convolutional neural networks for vision-based hand gesture recognition," in *Proc. of the 2nd IEEE ICSIPA*, 2011, pp. 342–347.

[4] J. Nagi, H. Ngo, A. Giusti, L. M. Gambardella, J. Schmidhuber, and G. A. D. Caro, "Incremental learning using partial feedback for gesture-based human-swarm interaction," in *Proc. of the 21st IEEE Intl. Symp. on RO-MAN*, 2012, pp. 898–905.

[5] A. Giusti, J. Nagi, L. M. Gambardella, and G. A. D. Caro, "Cooperative sensing and recognition by a swarm of mobile robots," in *Proc. of the 25th IEEE/RSJ Intl. Conf. on IROS*, 2012, pp. 551–558.

[6] J. Nagi, G. A. D. Caro, A. Giusti, F. Nagi, and L. M. Gambardella, "Convolutional neural support vector machines: Hybrid visual classifiers for multi-robot systems," in *Proc. of ICMLA*, 2012, pp. 27–32.

[7] R.-X. Hu, W. Jia, D. Zhang, J. Gui, *et al.*, "Hand shape recognition based on coherent distance shape contexts," *Pattern Recognition*, vol. 45, no. 9, pp. 3348–3359, 2012.

[8] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Trans. on SMC*, vol. 43, no. 3, pp. 311–324, 2007.

[9] J. P. Wachs, M. Kölsch, H. Stern, *et al.*, "Vision-based hand-gesture applications," *Comm. of the ACM*, vol. 54, no. 2, pp. 60–71, 2011.

[10] D. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Proc. of IJCAI*, 2011, pp. 1237–1242.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[12] A. Rule and J. Forlizzi, "Designing interfaces for multi-user, multi-robot systems," in *Proc. of Intl. Conf. on HRI*, 2012, pp. 97–104.

[13] A. Couture-Beil, R. Vaughan, and G. Mori, "Selecting and commanding individual robots in a multi-robot system," in *Proc. of the Canadian Conf. on CRV*, 2010, pp. 159–166.

[14] B. Milligan, G. Mori, and R. Vaughan, "Selecting and commanding groups in a multi-robot vision based system," in *Proc. of the 6th ACM/IEEE Intl. Conf. on HRI*, 2011, pp. 415–415.

[15] V. M. Monajjemi, J. Wawerla, R. T. Vaughan, and G. Mori, "Hri in the sky: Creating and commanding teams of uavs with a vision-mediated gestural interface," in *Proc. of IEEE Intl. Conf. on IROS*, 2013.

[16] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: a survey," *Found. and Tren. in HCI*, vol. 1, no. 3, pp. 203–275, 2007.

[17] N. Duta, "A survey of biometric technology based on hand shape," *Pattern Recognition*, vol. 42, no. 11, pp. 2797–2806, 2009.

[18] D. H. Wiesel and T. N. Hubel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. of Physio.*, vol. 160, pp. 106–154, 1962.

[19] Y. LeCun and Y. Bengio, *Convolutional Networks for Images, Speech, and Time-Series*, M. A. Arbib, Ed. MIT Press, 1995.

[20] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proc. of the IEEE International Symposium on Circuits and Systems*, 2010, pp. 253–226.

[21] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, "A committee of neural networks for traffic sign classification," in *Proc. of the Int. Joint Conference on Neural Networks (IJCNN)*, San Francisco, 2011.

[22] K. Sohn, D. Y. Jung, H. Lee, and A. O. Hero, "Efficient learning of sparse, distributed, convolutional feature representations for object recognition," in *Proc. of ICCV*, Nov. 2011, pp. 2643–2650.

[23] D. Scherer, A. Muller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Proc. of ICANN*, 2010, pp. 92–101.

[24] J. Wang, P. Zhao, and S. C. H. Hoi, "Exact soft confidence-weighted learning," in *Proc. of the 29th ICML*, 2012.

[25] K. Crammer, M. D. Fern, and O. Pereira, "Exact convex confidence-weighted learning," in *Advances in NIPS*, vol. 21, 2008.

[26] M. Dredze, K. Crammer, and F. Pereira, "Confidence-weighted linear classification," in *Proc. of the 25th ICML*, 2008, pp. 264–271.

[27] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[28] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *Journal of Machine Learning Research (JMLR)*, vol. 7, pp. 551–585, 2006.

[29] S. C. Hoi, J. Wang, and P. Zhao, *LIBOL: A Library for Online Learning Algorithms*, Nanyang Technological University, 2012. [Online]. Available: http://LIBOL.stevenhoi.org

[30] M. Bonani, V. Longchamp, S. Magnenat, *et al.*, "The marxbot, a miniature mobile robot opening new perspectives for the collective-robotic research," in *Proc. of IROS*, Taipei, 2010, pp. 4187–4193.

[31] Y. Ren and F. Zhang, "Hand gesture recognition based on meb-svm," in *Proc. of Intl. Conf. on Embed. Softw. & Syst.*, 2009, pp. 344–349.

[32] M. van Eede, D. Macrini, A. Telea, C. Sminchisescu, and S. Dickinson, "Canonical skeletons for shape matching," in *Proc. of the 18th Intl Conf. on Pattern Recognition (ICPR)*, 2006, pp. 64–69.

[33] Z. G. Y. Liu and Y. Sun, "Static hand gesture recognition and its application based on support vector machines," in *Proc. of Intl. Conf. on Softw. Eng., Artif. Intell. & Distrib. Comp.*, 2008, pp. 517–521.

[34] T. R. Trigo and S. R. M. Pellegrino, "An analisys of features for hand-gesture classification," in *Proc. of Intl. Conf. on Syst., Signals and Image Proc.*, Jun. 2010, pp. 412–415.