Max-Pooling Convolutional Neural Networks for Vision-based Hand Gesture Recognition

Jawad Nagi*, Frederick Ducatelle*, Gianni A. Di Caro*, Dan Cireşan*, Ueli Meier*, Alessandro Giusti*, Farrukh Nagi#, Jürgen Schmidhuber*, Luca Maria Gambardella*

* *Dalle Molle Institute for Artificial Intelligence (IDSIA), University of Lugano & SUPSI*
Manno-Lugano, Switzerland
{jawad,frederick,gianni,dan,ueli,alessandrog,luca,juergen}@idsia.ch

Department of Mechanical Engineering, University Tenaga Nasional
Putrajaya, Selangor, Malaysia
farrukh@uniten.edu.my

***Abstract*—Automatic recognition of gestures using computer vision is important for many real-world applications such as sign language recognition and human-robot interaction (HRI). Our goal is a real-time hand gesture-based HRI interface for mobile robots. We use a state-of-the-art big and deep neural network (NN) combining convolution and max-pooling (MPCNN) for supervised feature learning and classification of hand gestures given by humans to mobile robots using colored gloves. The hand contour is retrieved by color segmentation, then smoothened by morphological image processing which eliminates noisy edges. Our big and deep MPCNN classifies 6 gesture classes with 96% accuracy, nearly three times better than the nearest competitor. Experiments with mobile robots using an ARM 11 533MHz processor achieve real-time gesture recognition performance.**

I. Introduction

Hand gesture recognition has been a very active computer vision research topic in recent years with motivating applications such as human-robot interaction (HRI), sign language interpretation, computer games control, virtual reality and assistive environments [1]. Research activities on HRI has increased dramatically in recent times due to the widespread availability of cost-effective digital cameras suitable for ubiquitous computing [2], [3].

Compared to many existing HRI interfaces, hand gestures have the advantage of being easy to use as well as being natural and intuitive [4]. In order for a gesture recognition system to be real-time, robust and deployable in uncontrolled environments, the system needs to be able to operate in complex scenes with different backgrounds under variable lightning conditions [5], while taking into consideration different gesture positions, orientation and occlusions [6]. The work we present in this paper is the first step towards a complete interaction system between humans and robotic swarms, where the humans provide commands to the swarm using gestures, and the robots in the swarm make use of distributed and coordinated sensing and classification to reach swarm-level consensus about the gesture and execute the command associated to it.

In general, vision-based hand gesture recognition approaches fall into two major categories: 3D model-based methods and 2D appearance model-based methods [7]. 3D hand models may exactly describe hand movement and finger flexibility however, these approaches are computationally expensive and not suitable for real-time implementation [8] on smaller domestic robots. Therefore, in this paper we focus on 2D appearance model-based methods.

Recently, different research efforts on 2D appearance model-based methods for gesture recognition have emerged [9], [10], [11], [12], [13], [14], [15], amongst which supervised and unsupervised learning techniques such as Neural Networks (NNs), Support Vector Machine (SVMs) and Nearest-Neighbor [16], [17], [18] classifiers have gained familiarity. However, feature learning is not a part of such classification schemes and needs to be performed separately to compute features such as edges, gradients, pixel intensities and object shape. For general objection recognition and image classification tasks, variants of Convolutional Neural Networks (CNNs) [19], [20], [21], [22] have emerged as robust supervised feature learning and classification tools, especially when combined with max-pooling [23], [22]. Therefore, this paper presents, for the first time, the use of big and deep *Max-Pooling CNNs* (MPCNNs) for hand gesture recognition in HRI applications.

II. The Camera Hardware

The mobile *foot-bot* robots, small ground robots developed within the Swarmanoid project [24] (http://www.swarmanoid.org), were used as platform for our vision-based gesture recognition system. The foot-bot robot, shown in Figure 2, is specifically designed for swarm robotics research. It is powered using an on-board ARM 11 533MHz processor with 128MB RAM and programmed in a Linux-based operating environment. It has various sensors and actuators, such as: range and bearing, Wi-Fi, 3-axis accelerometers and gyroscopes, infrared-based proximity, and distance scanner.

The foot-bot is equipped with two built-in cameras. One is omni-directional, with a hyperbolic mirror mounted at the bottom of the glass-tube in Figure 2. The other is a camera that can be placed looking either to the front or to the ceiling. Both cameras are capable of capturing video streams and images at a maximum resolution of 3.0 megapixels.

978-1-4577-0242-6/11/$26.00 ©2011 IEEE

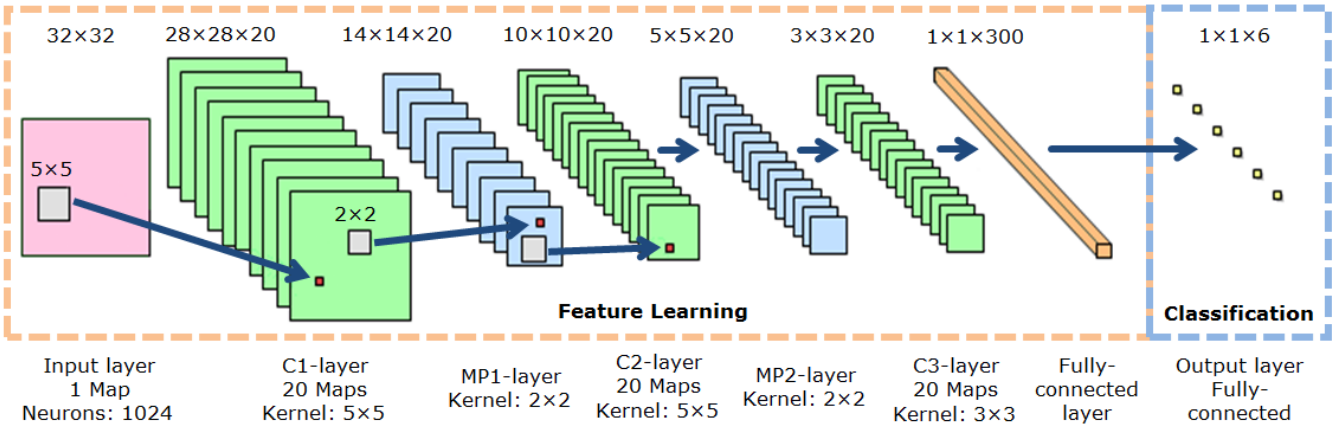


Fig. 1. MPCNN architecture using alternating convolutional and max-pooling layers.

III. Max-Pooling Convolutional Neural Networks

Convolutional NNs (CNNs) are multi-layered NNs specialized on recognizing visual patterns directly from image pixels [19], [25]. They are well-known for robustness to distortion and minimal pre-processing [21]. They were used for detection and recognition of objects including faces [21], hands, logos, text [25], with record accuracy and real-time performance. CNNs were also used for vision-based obstacle avoidance for mobile robots [17], image restoration, and segmentation of biological images [20].

Fig. 2. The *foot-bot* mobile robot used for the experiments.

We use the special max-pooling [23] CNN of [22], the MPCNN. MPCNNs have *convolutional* layers alternating with *subsampling* layers [22]. They belong to a wide class of models generally termed *Multi-Stage Hubel-Wiesel Architectures*, following Hubel and Wiesel's classic 1962 work on the cat's primary visual cortex [26], which identified orientation-selective *simple cells* with local receptive fields similar to those of convolutional layers, and *complex cells* performing subsampling-like operations [27]. MPCNNs vary in how convolutional and subsampling layers are realized and trained [22]. Figure 1 illustrates our MPCNN architecture used.

A. Convolutional layer

A convolutional layer is parametrized by: the number of maps, the size of the maps and kernel sizes. Each layer has M maps of equal size (M_x, M_y). A kernel of size (K_x, K_y) (as shown in Figure 1) is shifted over the valid region of the input image i.e. the kernel is completely inside the image [22]. Each map in layer L^n is connected to all maps in layer L^{n-1}. Neurons of a given map share their weights but have different input fields.

B. Max-pooling layer

The output of the max-pooling layer is given by the maximum activation over non-overlapping rectangular regions of size (K_x, K_y). Max-pooling creates position invariance over larger local regions and down-samples the input image by a factor of K_x and K_y along each direction [23]. Max-pooling leads to faster convergence rate by selecting superior invariant features which improves generalization performance.

C. Classification layer

After multiple convolutional and max-pooling layers, a shallow Multi-layer Perceptron (MLP) is used to complete the MPCNN. The output layer has one neuron per class in the classification task [22]. A *softmax* activation function is used, thus each neuron's output represents the posterior class probability.

IV. Data Acquisition and Modelling

A. Data acquisition

Gestures are presented to the foot-bots using colored gloves to facilitate the retrieval of the hand contour. Images are captured by the front (CMOS) camera of the robot using a Bayer color filter array. The Bayer pattern is transformed into its corresponding RGB image. All images are stored in an 8-bit unsigned Portable Network Graphics (PNG) format. In our vision-based gesture recognition system, we define a vocabulary of gestures based on the *count of fingers*. Therefore, in total, 6 *gestures* (classes) are defined, from *count* $= 0$ until *count* $= 6$, as shown in Figure 3.

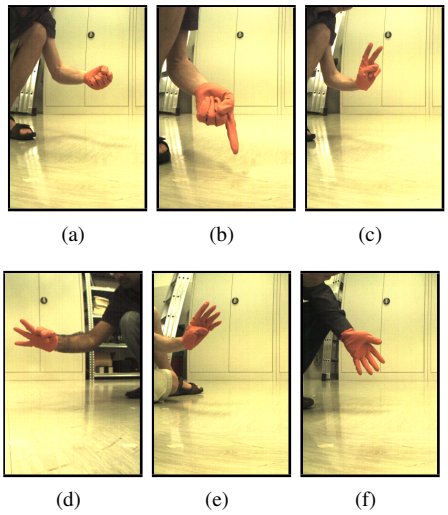


Fig. 3. Number of gesture classes defined using the *count of fingers*.

We introduce an *image-tuning routine* for optimal imaging results by automatically adjusting the camera's *gain* and *saturation* settings on each control step, based on the illumination conditions in the surrounding environment. During the data acquisition process, we capture images in a size of 512 x 384 pixels (0.2 megapixels) with an aspect ratio of 1.33:1 (4:3). The reason for choosing smaller resolution images is due to the necessity of real-time implementation.

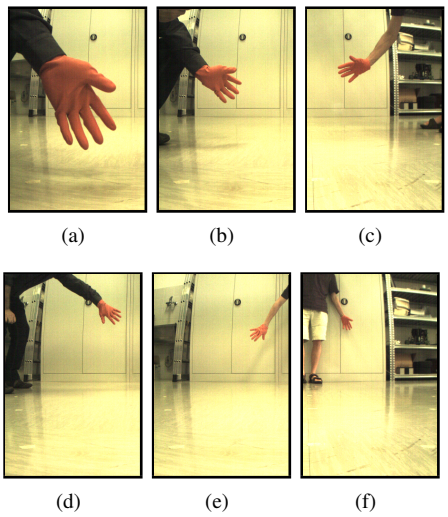


Fig. 4. Images taken at different distances from the robot. (a) 0.5m (b) 1.0m (c) 1.5m (d) 2.0m (e) 2.5m (f) 3.0m

During the data acquisition phase, a total of 6000 images were acquired at 6 different distances (in meters) from the robot, $d = [0.5\ 1.0\ 1.5\ 2.0\ 2.5\ 3.0]$, as illustrated in Figure 4. For each distance, a minimum of 1000 images were captured. The reason that 3 meters was chosen as the maximum distance is because, beyond that it became visually challenging for humans to distinguish the correct finger count by inspecting the images. As illustrated in Figure 5 during data acquisition, different finger combinations were used for each gesture class in order to make the system more robust for identifying the number of active fingers.

B. Color segmentation and preprocessing

The Red, Green, Blue (RGB) color space is the most common color space used to represent color images. However, RGB is an additive color space and it has a high correlation, non-uniformity and mixing of chrominance and luminance data [28]. Thus, RGB is not suitable for color analysis and color based recognition.

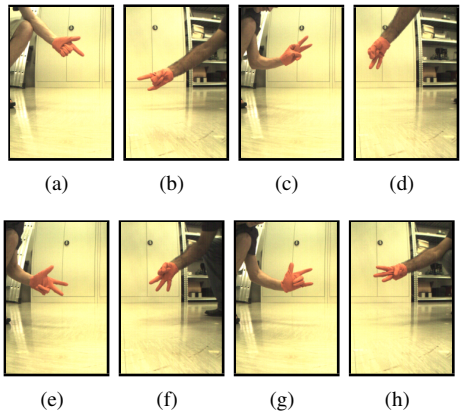


Fig. 5. Images taken using different finger combinations for each class.

In recent times, researchers have proposed the use of the YCbCr color space containing luminance (Y) and chrominance (CbCr) information, whereby the explicit separation of luminance and chrominance components reduces the effect of uneven illumination making it attractive for color based segmentation. In our approach to segment the colored glove, a parametric *Single Gaussian Model* (SGM) method is applied to model the orange glove color using the mean and the covariance of the chrominant color with a bivariate Gaussian distribution. Figure 6 illustrates the process modelled to segment the orange colored glove. The glove color distribution is modelled using an elliptical Gaussian joint Probability Density Function (PDF) using the following expression:

$$p[c/W_s] = (2\pi)^{-1}|\sum_s|^{-\frac{1}{2}} exp^{(c-\mu_s)T\sum_s^{-1}(c-\mu_s)} \quad (1)$$

where c is a color vector representing the random measured values of chrominance (x, y) of a pixel with coordinates (i, j) in an image, and W_s is the class describing the glove color.

$$c = [x(i,j)y(i,j)]^T \quad (2)$$

$$\mu_s = \frac{1}{n}\sum_{j=1}^{n} c_j \quad (3)$$

$$\sum_s = \frac{1}{n-1} \cdot \sum_{j=1}^{n}(c_j - \mu_s)(c_j - \mu_s)^T, \quad (4)$$

where μ_s represents the mean vector and $\sum_s$ represents the covariance matrix for the orange chrominance of the glove.

Using equations 2, 3 and 4, the Mahalanobis distance $\lambda(c)$, is calculated using equation 5, which measures the distance between the color matrix c_j and the mean vector μ_s.

$$\lambda(c) = (c_j - \mu_s)^T \sum_s^{-1} (c_j - \mu_s). \quad (5)$$

The Mahalanobis distance $\lambda(c)$ of a color vector c, is the luminance and chrominance threshold used to segment the orange colored glove from the image. During thresholding, the 3-channel image is transformed into a 1-channel *binary* ($[0, 1]$ pixel) image. Segmentation results of the orange colored glove are shown in Figures 7(a) through (d). Moreover, for real-world gesture recognition applications it is possible to replace our colored glove model with recent skin color models present in literature.

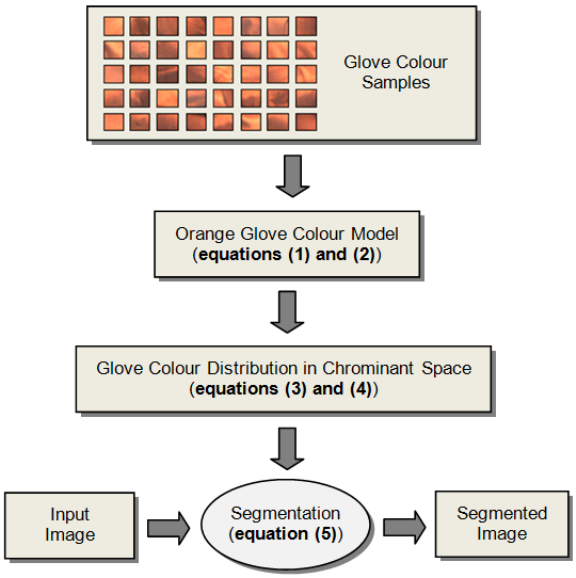


Fig. 6. Flowchart of framework modelled to segment orange glove.

The shape of the hand in the thresholded images indicates that the orange glove is well separated from the background, however, the hand contour contains a significant amount of noisy edges. To smooth the hand contour, *Morphological Opening* is performed, i.e. the images are *eroded* and then *dilated* using a structuring element of size 3 x 3, $se = [0\ 1\ 0;\ 1\ 1\ 1;\ 0\ 1\ 0]$. The smoothed images are shown in Figures 7(e) through (h). Our MPCNN implementation requires all images to be of equal size. After visual inspection of the image size distribution all images are resized to 28 x 28 pixels and padded with 4 black pixels on each side, resulting in an image size of 32 x 32 pixels as shown in Figure 8.

V. Experimental Results

The acquired 6000 images are split in ratios of 60% and 40% for the training and test sets respectively, where 3600 images are used for training and the remaining 2400 are used for testing. In order to evaluate our dataset and approach we compare the performance of our system with the state-of-the-art vision-based object and gesture recognition techniques.

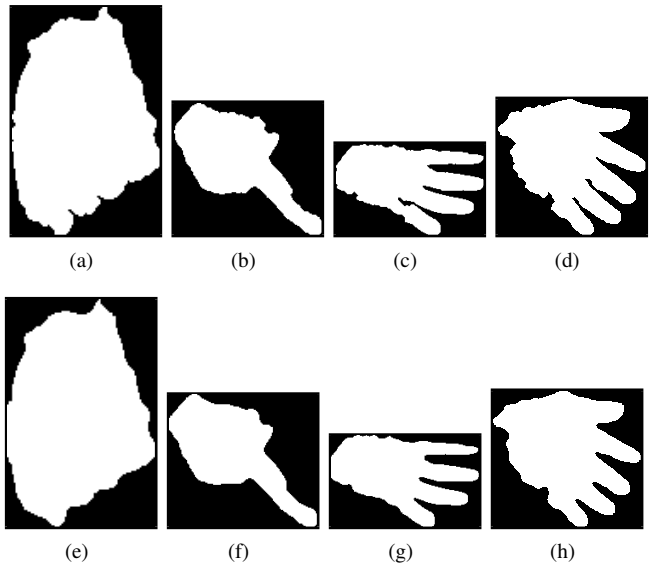


Fig. 7. Figures (a) through (d) represent the thresholded images. Figures (e) through (f) represent the smoothed images.

A. Existing approaches

Many existing vision-based object and gesture recognition approaches are present in literature, however we evaluate the most recent and familiar approaches, which compute image features of interest such as edges, gradients, pixel intensities and object shape. Since SVMs [29] have gained much attention in recent times due to their powerful generalization capabilities as gesture classifiers [16], [18] we evaluate different feature learning schemes using SVMs.

Fig. 8. Images used to construct training and test sets. Each image represents one of the 6 gesture classes (finger count).

The following approaches are evaluated in this paper using our dataset: (i) The authors in [30], [31], [32] use *Hu Invariant Moments* for feature learning from images of different objects and gestures; (ii) Unsupervised feature learning is applied by authors in [33] using the *Spatial Pyramid* (generally referred to as *Bag of Features* or *Bag of Words* (BoW)) a combination of SIFT and k-means; (iii) *Shape properties* of objects such as roundness, form factor, compactness, eccentricity, perimeter, solidity etc are used by the authors in [31], [34]; (iv) Skeletonization has been proposed by the authors in [35], [36] for gesture recognition tasks, such as the counting the number of fingers; (v) *Pyramid of Histogram Oriented Gradients* (PHOG) [37], a variant of the famous HOG descriptor [38], gained

popularity for its vectorized HOG feature learning approach; (vi) The Fast Fourier Transform (FFT) has been used by the authors in [39] to represent the shape of the hand contour in images using the spatial domain; (vii) CNNs called *Tiled CNNs* [40] are supervised feature learners and classifiers able to learn complex invariances such as scale and rotational invariance. The errors obtained on the test sets using these schemes are tabulated in Table 1.

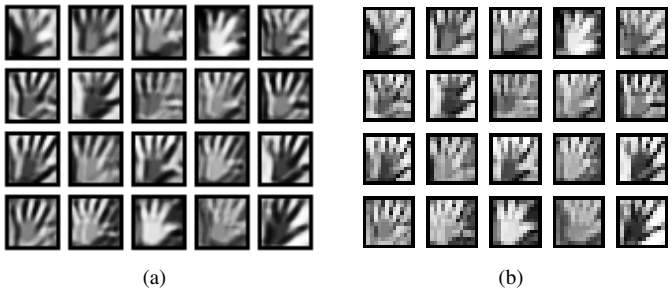
B. Big and deep MPCNNs

Our plain feed-forward MPCNN architecture is trained using on-line gradient descent. Images from the training set are rotated in order to learn rotational invariant features. All images from the training set are used for training and also for validation. Training ends once the validation error is zero (usually after 50 to 100 epochs). Initial weights are drawn from a uniform random distribution in the range $[-0.05,\ 0.05]$.

TABLE I
EVALUATION OF DIFFERENT GESTURE RECOGNITION APPROACHES

Feature Learner	Classifier	Reference	Error Rate
PHOG	SVM	[37]	27.04%
FFT	SVM	[39]	25.32%
Skeletonization	SVM	[35] [36]	21.55%
Hu Invariant Moments	SVM	[30] [31] [32]	20.34%
Shape Properties	SVM	[31] [34]	17.91%
Spatial Pyramid (BoW)	SVM	[33]	15.68%
Tiled CNN	NN	[40]	9.52%
Big and Deep MPCNN	NN	Proposed	3.23%

Our MPCNN architecture consists of 6 hidden layers as shown in Figure 2, where C-layer represents convolutional layers and MP-layer represents max-pooling layers. We use n = 20 maps in our implementation, the activations of the $C1$- and MP-1 layer for the input image shown in Figure 8(f) are shown in Figure 9(a) and 9(b), respectively. The free parameters used for training are indicated in Figure 2, where the output maps of the last convolutional layer ($C3$) are down-sampled to 1 pixel per map, resulting in a 1 x 300 feature vector for classification. All results reported are averaged by using 100 separate training and test sets, where each training set is constructed by selecting 60% random samples from each class, while the remaining 40% samples from each class are used in each test set.



(a) (b)

Fig. 9. Image activations using n = 20 maps. (a) represents the activations of Figure 8(f) for the C1-layer in Figure 2. (b) represents the activations of Figure 8(f) for the MP-1 layer in Figure 2.

We pick the trained MPCNN with the lowest validation error and evaluate it on the test set (i.e. the test for best validation). The best test error as shown in Figure 10 is 3.23%, where the training and validation errors are 0.002% and 0.0012% respectively. As seen from Figure 10, 80 epochs are sufficient to reach the lowest test error. Using a system with a Core i5-650 (3.20 GHz) processor with 4GB DDR3 RAM the computation time per training epoch is 426.12 s, while for evaluating the validation and test sets it takes 189.12 and 48.58 s respectively. Performing offline training and online testing using the foot-bot with an ARM 11 533MHz processor with 128MB RAM, it takes 0.82 s to capture, process and classify a single image, which indicates real-time performance using the C++ implementation of the MPCNN from [22]. The results in Table 1 indicate that our approach outperforms current object recognition techniques by far, making it the best choice for real-time gesture recognition in HRI applications.

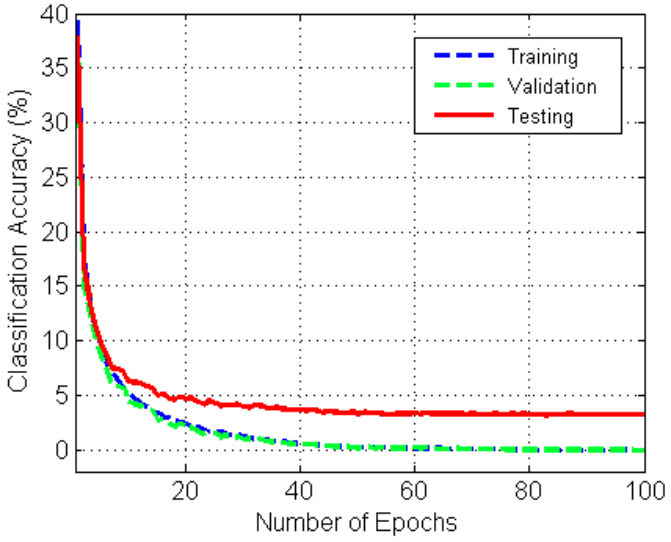


Fig. 10. Classification accuracies for training, validation and test sets using the big and deep MPCNN architecture shown in Figure 2. Results are averaged using 100 training and testing sets, where each set constructed by selecting random samples.

VI. CONCLUSION

This paper presents a state-of-the-art big and deep MPCNN for recognition of hand gestures in HRI applications. Our MPCNN combines convolution and max-pooling for supervised feature learning and classification of hand gestures from images. Experiments with mobile robots using an ARM 11 533MHz processor achieve real-time gesture recognition performance with a classification rate of 96%.

Our current implementation uses a vocabulary of 6 hand gestures, however, the vocabulary can be extended to 11 classes using two hands, i.e. from *finger count* 0 to 10. The obtained results show that our vision-based gesture recognition system can be effectively employed for HRI, even for small and relatively not powerful robots, such as domestic mobile robots (e.g., Roomba and Scooba), and swarm robotic systems.

Future work will precisely include the application of this work in the context of human-swarm interaction.

ACKNOWLEDGMENTS

This research was supported by the Swiss National Science Foundation (SNSF) through the National Centre of Competence in Research (NCCR) Robotics. This work was also supported by a FP7-ICT-2009-6 EU Grant under Project Code 270247: A Neuro-dynamic Framework for Cognitive Robotics: Scene Representations, Behavioural Sequences and Learning.

REFERENCES

[1] A. Stefan, V. Athitsos, J. Alon, and S. Sclaroff, "Translation and scale invariant gesture recognition in complex scenes," in *Proc. of the 1st ACM Intl. Conf. on Pervasive Technologies Related to Assistive Environments*, 2008, pp. 1–8.

[2] A. Malima, E. Ozgur, and M. Cetin, "A fast algorithm for vision-based hand gesture recognition for robot control," in *Proc. of 14th IEEE Conf. on Signal Processing and Communications Applications*, Antalya, Apr. 2006, pp. 1–4.

[3] X. Yin and M. Xie, "Finger identification and hand posture recognition for human-robot interaction," *Image and Vision Computing*, vol. 25, no. 8, pp. 1291–1300, Aug. 2007.

[4] Y. Fang, K. Wang, J. Cheng, and H. Lu, "A real-time hand gesture recognition method," in *Proc. of IEEE Intl. Conf. on Multimedia and Expo*, Beijing, Jul. 2007, pp. 995–998.

[5] X. Yin and X. Zhu, "Hand posture recognition in gesture-based human-robot interaction," in *Proc. of 1st IEEE Conf. on Industrial Electronics and Applications*, May 2006, pp. 1–6.

[6] R. S. Choras, "Hand shape and hand gesture recognition," in *Proc. of the IEEE Symp. on Industrial Electronics and Applications*, Kuala Lumpur, Oct. 2009, pp. 145–149.

[7] Y. Fang, J. Cheng, K. Wang, and H. Liu, "Hand gesture recognition using fast multi-scale analysis," in *Proc. of the 4th Intl. Conf. on Image and Graphics*, Aug. 2007, pp. 694–698.

[8] H. Kawasaki and T. Mouri, "Design and control of five-fingered haptic interface opposite to human hand," *IEEE Trans. on Robot.*, vol. 23, no. 5, pp. 909–918, 2007.

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

[10] P. Garg, N. Aggarwal, and S. Sofat, "Vision based hand gesture recognition," *World Academy of Science, Engineering and Technology*, vol. 49, pp. 972–977, 2009.

[11] C. Harshith, K. R. Shastry, M. Ravindran, M. Srikanth, and N. Lakshmikhanth, "Survey on various gesture recognition techniques for interfacing machines based on ambient intelligence," *Intl. Jour. of Computer Science and Engineering Survey*, vol. 1, no. 2, pp. 31–42, 2010.

[12] G. R. S. Murthy and R. S. Jadon, "A review of vision based hand gestures recognition," *International Journal of Information Technology and Knowledge Management*, vol. 2, no. 2, pp. 405–410, 2009.

[13] Y. Wu and T. S. Huang, "Vision-based gesture recognition: A review," in *Gesture-Based Communication in Human-Computer Interaction: Proc. of Intl. Gesture Workshop*, ser. LNCS, 1999, vol. 1739, pp. 103–114.

[14] A. Chaudhary, J. L. Raheja, K. Das, and S. Raheja, "Intelligent approaches to interact with machines using hand gesture recognition in natural way: A survey," *International Journal of Computer Science and Engineering Survey*, vol. 2, no. 1, pp. 122–133, Feb. 2011.

[15] J. P. Wachs, M. Klsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Comm. of ACM*, vol. 54, no. 2, pp. 60–71, Feb. 2011.

[16] A. Savaris and A. von Wangenheim, "Comparative evaluation of static gesture recognition techniques based on nearest neighbor, neural networks and support vector machines," *J. Braz. Comp. Soc.*, vol. 16, no. 2, pp. 147–162, Apr. 2010.

[17] J. Fan, W. Xu, Y. Wu, and Y. Gong, "Human tracking using convolutional neural networks," *IEEE Trans. on Neur. Netw.*, vol. 21, no. 10, pp. 1610–1623, Oct. 2010.

[18] L. Yun and Z. Peng, "An automatic hand gesture recognition system based on viola-jones method and svms," in *Proc. of the 2nd Intl. Workshop on Computer Science and Engineering*, Oct. 2009, pp. 72–76.

[19] C. Nebauer, "Evaluation of convolutional neural networks for visual recognition," *IEEE Trans. on Neur. Netw.*, vol. 9, no. 4, pp. 685–595, 1998.

[20] Y. LeCun, F.-J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Proc. of IEEE Conf. on CVPR*, vol. 2, 2004, pp. 97–104.

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

[22] D. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Proc. of 22nd Intl. Joint Conf. on Artificial Intelligence*, 2011, pp. 1237–1242.

[23] D. Scherer, A. Muller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Proc. of the Intl. Conf. on Artificial Neural Networks*, 2010, pp. 92–101.

[24] 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. D. Caro, F. Ducatelle, E. Ferrante, A. Föster, J. M. Gonzales, J. Guzzi, V. Longchamp, S. Magnenat, N. Mathews, M. M. 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," IRIDIA, Brussels, Belgium, Tech. Rep. 14-11, July 2011.

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

[26] D. H. Wiesel and T. N. Hubel, "Receptive fields of single neurones in the cat's striate cortex," *J. of Physio.*, vol. 148, pp. 574–591, 1959.

[27] ——, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. of Physio.*, vol. 160, pp. 106–154, Jan. 1962.

[28] P. Sebastian, Y. V. Voon, and R. Comley, "The effect of colour space on tracking robustness," in *Proc. of the 3rd IEEE Conf. on Industrial Electronics and Applications*, Singapore, Jun. 2008, pp. 2512–2516.

[29] J. Nagi, K. S. Yap, S. K. Tiong, S. K. Ahmed, and M. Malik, "Nontechnical loss detection for metered customers in power utility using support vector machines," *IEEE Trans. on Pow. Deliv.*, vol. 25, no. 2, pp. 1162–1171, Apr. 2010.

[30] Z. G. Y. Liu and Y. Sun, "Static hand gesture recognition and its application based on support vector machines," in *Proc. of the 9th ACIS Intl Conf. on Software Engineering, Artificial Intelligence, Networking and Distributed Computing*, 2008, pp. 517–521.

[31] D. Das, M. Ghosh, C. Chakraborty, M. Pal, and A. K. Maity, "Invariant moment based feature analysis for abnormal erythrocyte recognition," in *Proc. of Intl. Conf. on Systems in Medicine and Biology*, Dec. 2010, pp. 242–247.

[32] S. Jian-Fang and S. Bei, "Research on image recognition based on invariant moment and svm," in *Proc. of 1st Intl. Conf. on Pervasive Computing, Signal Processing and Applications*, 2010, pp. 598–602.

[33] S. Lazebnik, C. Schmid, and J.Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. of the IEEE Conf. on CVPR*, 2006, pp. 2169–2178.

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

[35] X. Zhu, "Shape recognition based on skeleton and support vector machines," in *Proc. of Intl. Conf. on Intelligent Computing*, 2007, pp. 1035–1043.

[36] 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*, 2006, pp. 64–69.

[37] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proc. of the IEEE Conf. on CVPR*, Amsterdam, Jul. 2007, pp. 401–408.

[38] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of the IEEE Conf. on CVPR*, 2005, pp. 886–893.

[39] Y. Ren and F. Zhhang, "Hand gesture recognition based on meb-svm," in *Proc. of the Intl. Conf. on Embedded Software and Systems*, May 2009, pp. 344–349.

[40] Q. Le, J. Ngiam, Z. Chen, D. Chia, P. Koh, and A. Ng, "Tiled convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2010.