The Fourteenth International Symposium on Artificial Life and Robotics 2009 (AROB 14th '09),
B-Con Plaza, Beppu, Oita, Japan, February 5 - 7, 2009

Feature Map Sharing Hypercolumn Model for Shift Invariant Face Recognition

Saleh Aly [1], Naoyuki Tsuruta[2], Rin-Ichiro Taniguchi[1]
[1] Department of Intelligent Systems, Kyushu University
744, Motooka, Nishi-ku, Fukuoka, 819-0395
Email: {aly,rin}@limu.is.kyushu-u.ac.jp

[2] Department of Electronics Engineering and Computer Science, Fukouka University
8-19-1, Nanakuma, Jonan-ku, Fukuoka 814-0180
Email: tsuruta@tl.fukuoka-u.ac.jp

Abstract

In this paper, we propose a shift invariant pattern recognition mechanism using feature sharing Hypercolumn model (FSHCM). To improve the recognition rate of Hypercolumn model (HCM) a shared map among a set of locally neighborhood maps is constructed in the feature extraction and feature integration layers. The shared maps help the network to increase its ability to deal with wide translation and distortion variations. The proposed framework uses FSHCM neural network to perform feature extraction step, and linear support vector machine for recognition task. The effectiveness of proposed approach is verified by using the misaligned ORL face database.

1 Introduction

Most face recognition algorithms assume aligned face images by facial features (e.g. by eye centers). In many applications, automatic alignment by facial features is still open problem. As the misalignment problem remains a bottleneck to the performance of face recognition, we propose a shift invariant pattern recognition mechanism using Hypercolumn model neural network.

Understanding how visual cortex recognize objects is a critical and essential task to design and implement an invariant pattern recognition system. Because humans and primates outperform the best machine vision systems with respect to almost any measure. Building a system that emulates object recognition in cortex has always been an attractive area of research.

Hierarchical models have been shown to outperform single-template object recognition systems [1]. There are two well-known models, which are developed to emulate the structure and behavior of visual cortex, neocognitron (NC) [2] and Standard model (SM) [3]. The complex structure of these models is one reason for its unpopularity. The Neocognitron (NC) is a massively parallel hierarchical neural network, designed primarily for 2-D pattern recognition. It was inspired by Hubel and Wiesel's serial model of biological vision [4]. It has been experimented on character recognition and show a shift and distortion tolerance. The standard model (SM) composed of a hierarchy of feedforward layers of neuron-like units, which perform either (1) a tuning computation (weighted linear sum or Gaussian template matching) to increase feature selectivity or (2) a nonlinear pooling operation based on a maximum operation to increase response invariance to translation and scaling.

Recently, applications of artificial neural networks have been expanded into general image recognition problems such as face recognition and visual surveillance. In these applications, there are three factors should be taken into consideration in order to achieve high recognition rate:

1. *Dimensionality reduction:* the distribution of visual objects in the original image space almost lies in a low-dimension subspace, which is always lower than the dimension of image space. Therefore, it is necessary to reduce the dimensions in order to eliminate the redundancy in the data.

2. *Invariance:* the appearance of the objects almost affected by many kinds of image variations. These variations may occur separately or simultaneously such as scale, shift, rotation, illuminations, pose variations. Therefore robust object recognition system should show some invariance against these factors.

3. *Network parameters:* there are two different pa-

rameters associated with designing an efficient neural network for object recognition problem, which are structure and training parameters. Deciding the appropriate parameters is highly depend on the distribution of the training data and the kind of required invariance.

Solving these problems is the target of Hypercolumn model. Hypercolumn model [5] has a similar architecture with Neocognitron and standard model. However, the learning strategy of the network is based on Kohonen learning algorithm. Self-organizing map (SOM) [6] is the fundamental component in building the Hypercolumn network.

The main contribution in this paper, is to replace the small-size neighbor maps in feature extraction and feature integration layers with one large shared map. The feature map is trained to share among a set of neighbor maps. This large feature map can be created by two different methods. In the first method, shared features are learned from the aggregation of all training examples covered by all shared maps. While the second method learn the shared map by applying neighborhood learning technique among neighbor maps.

The remainder of the paper is organized as follows: In section 2, the structure and the learning algorithm of HCM neural network is presented. Section 3 describes methods of creating shared feature maps. Experimental results, and conclusions are given in the following two sections,.

2 Hypercolumn model

2.1 HCM structure

The HCM network shown in Figure (1) is derived from the NC by replacing each C-cells and the lower directly connected S-cells with a two-layer hierarchical self-organizing map HSOM network [7]. The first map in the HSOM cell stands for feature extraction with size N_n^{FEL}, while the second one is used for feature integration with size N_n^{FIL}. The number of HSOM in X and Y directions are denoted by N_{HSOM}^X and N_{HSOM}^Y respectively. The input field for each feature extraction map is slightly shifted and overlapped by a certain number of neurons. Feature integration layer is an SOM map, whose input is the index of the winner in the lower feature extraction map. In the feature integration map, therefore, all shifted and distorted patterns are mapped to the same neuron, since the number of neurons in the feature integration map is

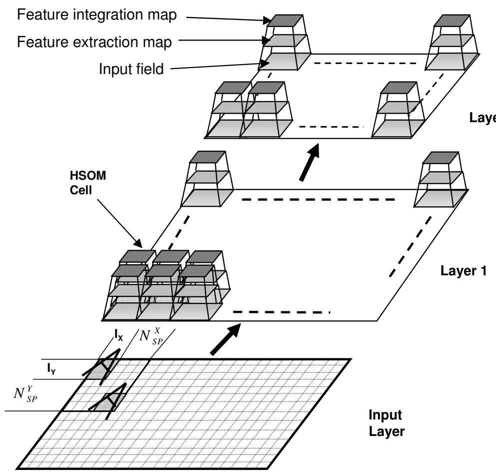

Figure 1: HCM Network structure

smaller than the number of neurons in the feature extraction map. Consequently, the index of the winners from lower feature integration maps are presented to the next higher feature extraction map. These local features are pilled up hierarchically in the higher layers to tolerate image scale, shift, rotation, and distortion variations.

2.2 HCM Training

The HCM uses the unsupervised learning algorithm of the competitive neural networks to construct its feature maps. The learning process is applied layer-by-layer starting from the bottom layer, where the normal learning algorithm of the HSOM is used to train each unit in the map. All HSOMs in the same layer can be trained in parallel. After presenting the input pattern to the HCM network, the training algorithm find the best matching winner from the first feature extraction map. Each feature extraction layer has a local input field with the size of $(I_X \times I_Y)$. The competition step in the feature extraction layer is slightly different than the original competitive step in the Kohonen algorithm. In which the competition is performed through all shifted patterns, the shift step size and the number of shifts in X and Y directions are denoted as $N_S^X \times N_S^Y$ and $N_{SS}^X \times N_{SS}^Y$ respectively. The winner pattern used to update the weights of the winner neuron and its neighbors. The input field for the next HSOM is overlapped with the input field of the neighbor HSOMs, and the size of overlapping in both X and Y directions are decided by the parameters $N_{OP}^X \times N_{OP}^Y$. These parameters are decided experimentally to tolerate shift, scale and distortion variations for the HCM network. The feature map for each HSOM cell is constructed by learning all patterns from the region covered by the shifted input field, however, due to expression and rotation variations these features may be repeated in the nearest feature maps. To solve this problem, the feature map for all neighbors HSOM should share one large feature map.

3 Feature map sharing implementation

In order to deal with wide shift and distortion variations in the local features, a shared feature map among a set of neighbor maps is constructed. There are two proposed methods for implementation, In the first one, all training examples presented to the neighbor maps are used to train one shared map with a sufficiently large number of neurons. The second method, apply the idea of neighborhood learning to train neighbor maps.

3.1 Method 1: Training examples aggregation

A shared map is constructed among set of neighbor maps by learning from all examples presented to the neighbor maps. Therefore the training algorithm repeat the original SOM learning algorithm for each input pattern in the region covered by all shared maps.

1. Find the best matching neuron c using a similarity measure between the input and all the map's neurons, where c is the desired winner and should satisfy:

$$\|x_i - w_c\| = arg\min_j(\|x_i - w_j\|) \quad (1)$$

2. Update the weight vector of the winner c and also all its topological neighborhood in the map towards the prevailing input according to the rule:

$$w_j(t+1) = w_j(t) + h_{cj}(t)[x_i(t) - w_j(t)] \quad (2)$$

$$h_{cj}(t) = \alpha(t).\exp\left(-\frac{\|r_c - r_i\|}{2\sigma^2(t)}\right) \quad (3)$$

where $h_{cj}(t)$ is the neighborhood kernel function around the winner c at time $t, \alpha(t)$ is the learning rate and is decreased gradually toward zero and $\sigma^2(t)$ is a factor used to control the width of the neighborhood kernel.

As the learning process finished the constructed feature map is copied to all neighbor maps. We have to mention that each map has its different input field although they shared the same map.

3.2 Method 2: Neighborhood map learning

The main idea for the second method, is to construct the shared feature map by applying the neighborhood learning trick in the feature map level. Which implies that there are two neighborhood learning, the original neighborhood learning in the Kohonen algorithm among neurons in the map and the proposed one between neighbor maps. Therefore the second training algorithm differ than the first method, and can be summarized as follows.

- Find the best matching unit as stated in equation (1). Moreover, the shift mechanism in the HCM is applied to find the winner pattern and the winner neuron by including all shifted patterns in the competition step.
- Update the winner neuron using the winner pattern by applying equation (2).
- Update the neuron at the same position in all neighbor maps by the winner pattern using small learning rate.

The above steps are repeated for each HSOM cell in the HCM network and for every map in the HSOM cells, (i.e. feature extraction map and feature integration map).

4 EXPERIMENTAL RESULTS

4.1 ORL Face Database

The ORL database was collected between 1992 and 1994 [8]. It contains ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement).

4.2 Face recognition using misaligned face database

The aim of this experiment is to determine the performance of FSHCM algorithm to solve misaligned face recognition problem. In this experiment, ORL face database is divided into two halves; one-half used as a training data and the other part for blind test.

The training data for HCM consists of 200 images, the first five images for each person. All images in the database are scaled to the size 48×48 pixels and photometrically normalized with histogram equalization method without any geometric normalization.

The proposed face recognition system consists of 2−layers HCM neural network for feature extraction stage and linear SVM classifier at the top of the network for recognition stage. Table (1) summarizes the parameters used in the training of HCM. In our experiment, the first method is used to construct all shared maps in the first layer of HCM neural network. This system give 89% recognition accuracy using features extracted from the first layer of HCM network. However, the second layer features give slightly lower accuracy rate (88.5%) than the first layer features. The decrease in the accuracy due to the utilization of cropped input image in the test phase, however, for coarse cropped faces the features from second layer expected to give better performance than features from first layer.

4.3 Shift invariant feature extraction

In this experiment, the capability of FSHCM neural network to deal with affine transformed input images is examined. A set of artificial face images is generated using affine transformed training data. The scaling parameter for the transformed test data has the following values $\{0.95, 0.97, 1.05, 1.07\}$, rotation parameter has values of $\{2°, 4°, -2°, -4°\}$, and the translation parameter take the values $\{1, 2, -1, -2\}$ pixels in both X and Y directions. The recognition accuracy for all affine transformed face images using the same network structure in the previous experiment is 98%. This higher rate of accuracy indicate that FSHCM features exhibit large tolerance to affine transformed input images. As the number of shifted pixels due to scale, rotation, and translation variations are more than 10% of the image size.

5 CONCLUSION

A modified version of Hypercolumn model has been proposed. Shared feature maps are proposed to recover wide variation in shift and distortion. In order to perform the classification step in the final stage of our pattern recognition system, a simple linear classifier such as linear SVM algorithm is used to classify the extracted features. The performance of the modified network shows a reasonable results to deal with affine transformed images from ORL face database.

Table 1: HCM Training Parameters

Parameter	Layer 1	Layer 2
$I_X \times I_Y$	3×3	2×2
$N^X_{HSOM} \times N^Y_{HSOM}$	23×23	11×11
$N^X_S \times N^Y_S$	2×2	0×0
$N^X_{SS} \times N^Y_{SS}$	1×1	0×0
$N^X_{OP} \times N^Y_{OP}$	2×2	2×2
N^{FEL}_n	30×16	40×28
N^{FIL}_n	8×6	12×10

References

[1] B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio. Categorization by learning and combining object parts. *Advances in Neural Information Processing Systems*, 14, 2002.

[2] K. Fukushima. Neocognitron. *Biological Cybernetics*, 36(4):193–202, 1980.

[3] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(3):411–426, 2007.

[4] D. Hubel and T. Wiesel. Receptive fields, binocular interaction and functional architecture in the cats visual cortex. *J. of Physiology*, 160:105–154, 1962.

[5] N. Tsuruta, R. Taniguchi, and M. Amamiya. Hypercolumn model: A combination model hierarchical self- organizing maps and neocognitron for image recognition. *System and Computer in Japan*, 31(2):49–61, 2000.

[6] T. Kohonen. *Self-Organizing Maps.* Springer, 1997.

[7] J. Lampinen and E. Oja. Clustering properties of hierarchical self organizing maps. *Journal of Math. Imaging and Vision*, 2(2-3):261–272, 1992.

[8] F. Sammaria and A. Harter. Parameterization of a stochastic model for human face identification. *In 2nd IEEE workshop on applications of computer vision*, 1994.