In Proceeding of International Conference on Artificial Neural Networks (ICANN), Madrid, Spain,
LNCS 2415, pp. 1319-1324, Springer, 2002

Learning Face Localization using Hierarchical Recurrent Networks

Sven Behnke

Freie Universität Berlin, Institute of Computer Science
Takustr. 9, 14195 Berlin, Germany
behnke@inf.fu-berlin.de, www.inf.fu-berlin.de/~behnke

Abstract. One of the major parts in human-computer interface applications, such as face recognition and video-telephony, consists in the exact localization of a face in an image.
Here, we propose to use hierarchical neural networks with local recurrent connectivity to solve this task, even in presence of complex backgrounds, difficult lighting, and noise. Our network is trained using a database of gray-scale still images and manually determined eye coordinates. It is able to produce reliable and accurate eye coordinates for unknown images by iteratively refining an initial solution.
The performance of the proposed approach is evaluated against a large test set. The fast network update allows for real-time operation.

1 Introduction

To make the interface between humans and computers more pleasant, computers must adapt to the users. One important step for many adaptive applications, like face recognition, lip reading, reading of the users emotional state, and video-telephony is the localization of the user's face in a captured image.

A recent survey on face detection can be found in [4]. Many localization techniques rely on image motion or skin color which are not always available. In [9] multiresolution window scanning in combination with a neural network is used to detect faces in gray-scale static images. Such sequential search techniques are computationally expensive. Many methods preprocess the data intensively to extract facial features and match them with predefined models [5, 6].

In this paper, we present a method that uses a hierarchical neural network with recurrent local connectivity to localize a face in gray-scale still images. The network operates by iteratively refining an initial solution. We present images directly to the network and train it to do the job.

2 Face Database and Preprocessing

To validate the performance of the proposed approach for learning face localization, we use the BioID data base [5]. This database consists of 1521 images that show 23 individuals in front of various complex office backgrounds with uncontrolled lighting. The persons differ in gender, age, and skin color. Some of them

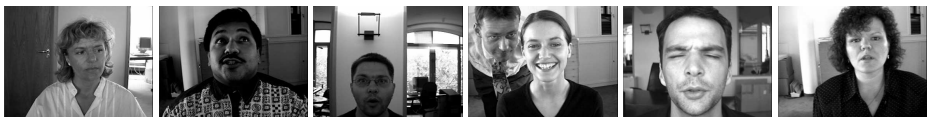

Fig. 1. Some face images from the BioID data set.

wear glasses or a beard. Since the face size, position, and view, as well as the facial expression vary considerably, the dataset can be considered challenging.

Such real world conditions are the ones that show the limits of current localization techniques. For instance, while the hybrid localization system described in [5] correctly localizes 98.4% of the XM2VTS database [7] that has been produced under controlled conditions, the same system localizes only 91.8% of the BioID faces. Figure 1 shows some images from the dataset.

The gray-scale BioID images have a size of 384×288 pixels. To reduce border effects, we lowered the contrast towards the sides of the image. To limit the amount of data, the image is subsampled to 48×36, 24×18, and 12×9 pixels as shown in Fig. 2(b). In addition to the images, manually labeled eye positions are available. Fig. 2(a) shows the marked eye positions for a sample image. We produce a multi-resolutional Gaussian blob for each eye (see Fig. 2(b)).

3 Network Architecture

The preprocessed images are presented to a hierarchical neural network that is structured as Neural Abstraction Pyramid [1]. As can be seen in Figure 3, the network consists of four layers. Each layer contains excitatory and inhibitory quantities. Each quantity is computed at a 2D-grid of locations by $\sum$-units that share a common weight template. The resolution of the layers decreases from L_0 (48×36) to L_2 (12×9) by a factor of 2 in both dimensions. L_3 has only one unit per quantity. The number of quantities per layer increases when going from L_0 (4+2) to L_2 (16+8). L_3 contains 10 excitatory and 5 inhibitory quantities.

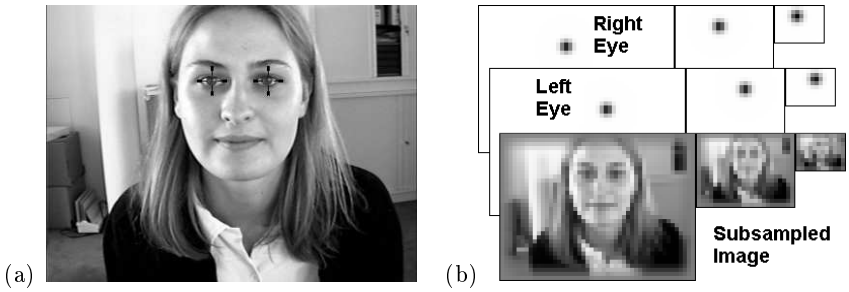

Fig. 2. Preprocessing: (a) original image with marked eye positions; (b) eye positions and subsampled framed image in three resolutions.

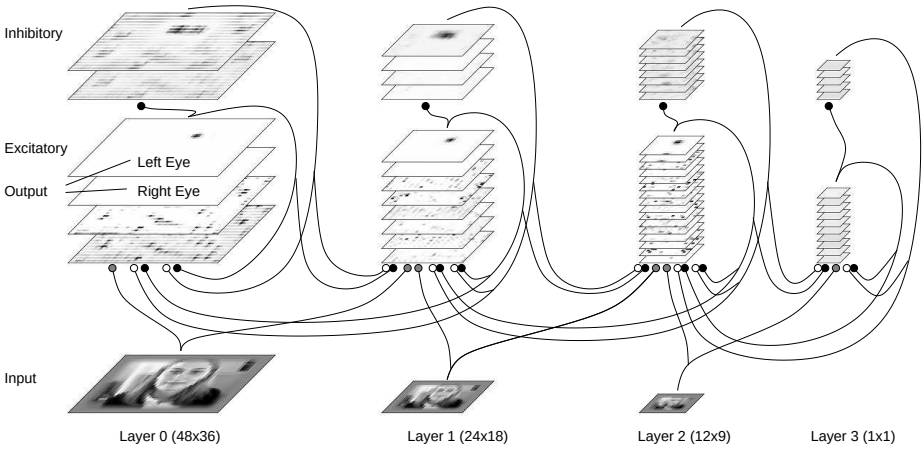

Fig. 3. Sketch of the hierarchical network architecture with local recurrent connectivity.

The network's connectivity is recurrent and local. Each unit receives input from only a small window of units that correspond to similar locations in the layer below (forward weighs), in the same layer (lateral weights) and from the layer above (backward weights). Weights from excitatory units are non-negative. Weights from inhibitory units are non-positive. Input weights can have any sign.

The excitatory units of L_0-L_2 receive input from 4×4 windows of the quantities one layer below. They look at 5×5 input units and at a 3×3 neighborhood at the same layer. The backward weights have a window size of 2×2. Connections between L_2 and the topmost L_3 are different. Both, forward- and backward weights have a 12×9 window size. Inhibitory quantities look only at 5×5 windows of all excitatory quantities at the same layer. Of course, in L_3 this reduces to 1×1. The update step $(t+1)$ of a unit for quantity q at position (x, y) in L_z is done as follows:

$$v^{t+1}_{x,y,z,q} = \sigma \left[\sum_{j \in \mathcal{L}(i)} \mathcal{W}(j)\, v^{t}_{\mathcal{X}(j,x),\mathcal{Y}(j,y),\mathcal{Z}(j,z),\mathcal{Q}(j)} + \mathcal{B}(i) \right] . \tag{1}$$

$\mathcal{L}(i)$ is the set of links of the associated template $i = \mathcal{T}(z, q)$, and $\mathcal{B}(i)$ is the template bias. $(\mathcal{X}(j,x), \mathcal{Y}(j,y), \mathcal{Z}(j,z), \mathcal{Q}(j))$ describe location and quantity of the input for link j, and $\mathcal{W}(j)$ is its weight. The output function $\sigma(x) = \mathsf{ln}(1 + e^{\beta x})/\beta$ is here a smooth approximation to the rectifying function $\mathsf{max}(0, x)$. In addition, a start value $\mathcal{V}^0(i)$ for initialization at $t = 0$ is needed for each template.

4 Supervised Training

Training recurrent networks is difficult due to the non-linear dynamics of the system. The backpropagation through time algorithm (BPTT) [10] unfolds the network in time and applies the backpropagation idea to compute the gradient

of an error function. For face localization, we present a static input $\mathbf{x}_k$ to the network and train it to quickly produce the desired output $\mathbf{y}_k$.

The network is updated for a fixed number $T = 10$ of iterations. The output error δ_k^t, the difference between the activity of the output units $\mathbf{v}_k^t$ and the desired output $\mathbf{y}_k$, is not only computed at the end of the sequence, but after every update step. In the error function we weight the squared differences progressively, as the number of iterations t increases:

$$E = \sum_{k=1}^{K} \sum_{t=1}^{T} t \, \|\mathbf{y}_k - \mathbf{v}_k^t\|^2. \tag{2}$$

Minimizing the error function with gradient descent faces the problem that the gradient in recurrent networks either vanishes or grows exponentially in time depending on the magnitude of gains in loops [3]. Hence, it is very difficult to determine a learning constant that allows for both stability and fast convergence.

For that reason, we decided to employ the RPROP algorithm [8], that maintains a learning rate for each weight and uses only the sign of the gradient to determine the weight change. We modify not only the weights in this way, but adapt the biases and start values as well. To accelerate the training, we initially worked with randomly chosen subsets of the training set, as described in [2].

5 Experimental Results

We divided the BioID data set randomly into 1000 training images and 521 test examples. Figure 4 shows the development of the trained network's output over time when the test image from Fig. 2 is presented as input. One can see that the blobs signaling the locations of the eyes develop in a top-down fashion. After the first iteration they appear only in the lowest resolution. This coarse localization is used to bias the development of blobs in lower layers. After five iterations, the network's output is close to the desired one. It does not change significantly during the next iterations that take 22ms each on a P4 1.7GHz PC.

The generation of stable blobs is the typical behavior of the network. To evaluate its performance, one has to estimate eye coordinates from the blobs and to compute a quality measure by comparison with the given coordinates.

We estimate the position of each eye separately by finding the output unit with the highest activity in the corresponding high resolution output. For all units in a 7×7 window around it, we segment the units belonging to the blob by comparing their activity with a threshold that increases with greater distance from the center. The weighted mean location of the segmented units is the

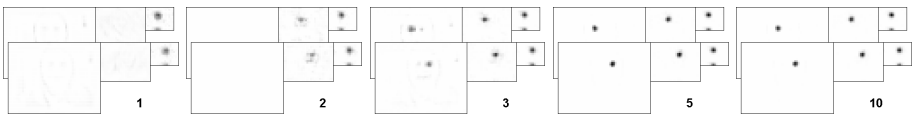

Fig. 4. Recall. Shown are the activities of the network's output over time.

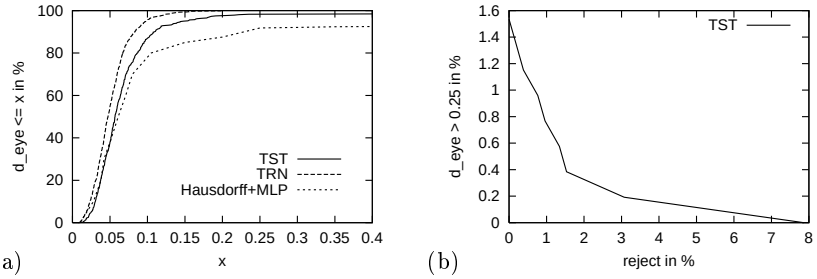

Fig. 5. Localization performance: (a) percentage of examples having small d_{eye} for the proposed method (TRN, TST) and for the hybrid system (Hausdorff+MLP)[5]; (b) rejecting the least confident examples lowers the number of mislocalizations.

estimated eye position. After transforming these eye positions into the original coordinate system, we compute a scale independent relative error measure as suggested in [5]: $d_{eye} = \mathsf{max}(d_l, d_r)/||C_l - C_r||$, where d_l and d_r are the distances of the estimated eye positions to the given coordinates C_l and C_r. A relative distance $d_{eye} < 0.25$ is considered a successful localization, since $d_{eye} = 0.25$ corresponds approximately to the half width of an eye.

Figure 5(a) shows the network's localization performance for the training set (TRN) and the test set (TST) in comparison to the data taken from [5] (Hausdorff+MLP). All training examples have been localized successfully. The performance on the test set is similar. Only 1.5% of the test examples have not been localized accurately enough. Compare this to the 8.2% mislocalizations of the reference system.

A detailed analysis of the network's output for the mislocalizations showed that in these cases the output deviates from the one-blob-per-eye pattern. It can happen that no blob or that several blobs are present for an eye. By comparing the activity of a segmented blob to a threshold and to the total activity a confidence measure is computed for each eye. Both are combined to a single confidence. In Figure 5(b) one can see that rejecting the least confident test examples lowers the number of mislocalizations rapidly. When rejecting 3.1% of

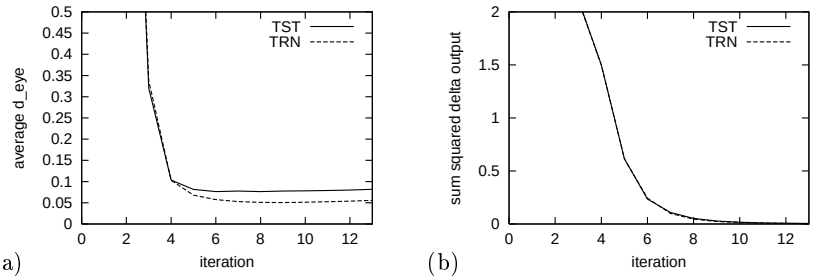

Fig. 6. Performance over time: (a) average distance d_{eye}; (b) sum of squared changes in the network's output.

the images, only one mislocalization is left. The average localization error of the accepted examples is $d_{eye} = 0.06$. That is well within the area of the iris and corresponds to the accuracy of the given coordinates.

Figure 6 illustrates the network's performance over time. The average error d_{eye} drops rapidly within the first five iterations and stays low afterwards. The changes in the network's output are large during the first iterations and decrease even when updated longer than the ten steps it has been trained for.

6 Conclusions

In this paper we presented an approach to face localization that is based on a hierarchical neural network with local recurrent connectivity. The network is trained to solve this task even in the presence of complex backgrounds, difficult lighting, and noise through iterative refinement.

We evaluate the network's performance on the BioID data set. It compares favorably to a hybrid reference system that uses a Hausdorff shape matching approach in combination to a multi layer perceptron.

The proposed method is not limited to gray-scale images. The extension to color is straight forward. Since the network works iteratively, and one iteration takes only a few milliseconds, it would also be possible to use it for real-time face tracking by presenting image sequences instead of static images.

References

1. S. Behnke and R. Rojas. Neural Abstraction Pyramid: A hierarchical image understanding architecture. In *Proc. IJCNN'98–Anchorage*, pages 820–825, 1998.
2. Sven Behnke. Learning iterative image reconstruction in the Neural Abstraction Pyramid. *International Journal on Computational Intelligence and Applications, Special Issue on Neural Networks at IJCAI-2001*, 1(4):427–438, 2001.
3. Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. on Neural Networks*, 5(2):157–166, 1994.
4. Erik Hjelmas and Boon Kee Low. Face detection: A survey. *Computer Vision and Image Understanding*, 83:236–274, 2001.
5. O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz. Robust face detection using the Hausdorff distance. In *Third Int. Conf. on Audio- and Video-based Biometric Person Authentication, Halmstad, Sweden*, pages 90–95. Springer, 2001.
6. D. Maio and D. Maltoni. Real-time face localization on gray-scale static images. *Pattern Recognition*, 33:1525–1539, 2000.
7. K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. XM2VTSDB: The extended M2VTS database. In *Second Int. Conf. on Audio and Video-based Biometric Person Authentication*, pages 72–77, 1999.
8. Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proceedings of the International Conference on Neural Networks – San Francisco, CA*, pages 586–591. IEEE, 1993.
9. H. A. Rowley, S. Baluja, and T. Kanade. Neural network based face detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20:23–38, 1998.
10. R. Williams and J. Peng. An efficient gradient-based algorithm for on-line training of recurrent network trajectories. *Neural Computation*, 2(4):491–501, 1990.