Journal of VLSI Signal Processing 23, 497–511 (1999)

Face and Eye Detection by CNN Algorithms

DAVID BALYA* AND TAMÁS ROSKA
Analogical and Neural Computing Laboratory, Computer and Automation Research Institute, Lágymányosi u. 11, Budapest, H-1111, Hungary

Abstract. A novel approach to critical parts of face detection problems is given, based on analogic cellular neural network (CNN) algorithms. The proposed CNN algorithms find and help to normalize human faces effectively while their time requirement is a fraction of the previously used methods. The algorithm starts with the detection of heads on color pictures using deviations in color and structure of the human face and that of the background. By normalizing the distance and position of the reference points, all faces should be transformed into the same size and position. For normalization, eyes serve as points of reference. Other CNN algorithm finds the eyes on any grayscale image by searching characteristic features of the eyes and eye sockets. Tests made on a standard database show that the algorithm works very fast and it is reliable.

1. Introduction

This paper deals with a novel approach to the face detection problem, based on cellular neural networks (CNN) processing. It describes a completely new method for the localization and normalization of faces, which is a critical step of this complex task but hardly ever discussed in the literature. The proposed CNN algorithm is capable of finding and helping normalize human faces effectively while its time requirement is a fraction of the previously used methods (see Table 1).

Recognition of faces is a remarkable example for the ability of humans to perform complex visual tasks. Even animals can solve this task. Pigeons find their mates easily though they look very similar to us. Our recognition ability develops over several years of childhood. This is important in several aspects of our social life such as estimating the facial-expression of people we interact with. It has probably played an important role in the course of evolution.

Face recognition, a problem that has been considered a challenge since the very first days of computer vision, has experienced a revival recently, after nearly twenty years of silence. Finding proper solution to the problem was one of the goals of the fifth generation artificial intelligence project. Several methods have been proposed but none of them are considered to be perfect in all aspects such as accuracy, speed etc.

In our approach the complex task is divided into subtasks. The literature deals mainly with the representation of faces i.e. the phase of the problem that comes after finding and normalizing the face qualitatively as well as geometrically.

The first step, however simple it may seem, constitutes in reality a difficult and sophisticated preprocessing sub-task. Most known systems spend more than half of the total computation time on this step, which is due to the quality of the whole process depending on the success of the detection. The present paper utilizes the potential of a new tool, the Cellular Neural Network (CNN) computer, in which an array of programmable analog processing cells perform parallel computation on corresponding pixel values of a given image [1–5]. Templates control functioning of a CNN-UM (Cellular Neural Network and Universal Machine) and the programming of a CNN is done by organizing the series of templates [6].

The human face as input information could be used by various systems [7]. Biometrics identification systems present an obvious example because human faces are straightforward and comfortable objects for identification. Security systems, e.g. ATMs, could use faces as keys which can not be lost and are very difficult to forge. In working with image databases, such as

*D. Balya is also with the Technical University of Budapest.

Table 1. Comparing some eye detection algorithms.

Tool	Accuracy	Speed	Remarks
PDBNN [8]	96.4%	500 ms SUN Sparc20	It needs many images for training
DT, features [23]	96%	<1 sec	Part of a face recognition system
MIP [24]	95%	91 sec SGI INDY	Find noses too
Templates [25]	94%	50 ms TMS320C40	Poorer on static images
Neural Nets [26]	93%	120 sec SUN Sparc20	Face detection
CNN	98%	3 ms CNN chip	See Section 1 and 6

television and video, searching for given visual objects is an inherent task. The actual identification task often requires the witness to scrutinize thousands of images including line drawings or sketches. Obviously, this is too much for a successful search. However, the images can effectively be used for the control of alertness.

The algorithm described in the paper starts with the detection of the head on a colored picture using deviations in color and structure of the human face and that of the background. Reference points are needed for normalization of the face. The eyes were found the most reliable for this purpose. By normalizing the distance and position of reference points, all faces can be transformed into the same dimension and position.

The proposed CNN algorithm finds the eyes on any grayscale image by searching for some characteristic features of the eyes and eye sockets. Test made on a standard database (15 randomly selected faces from the FERET face database and 15 other images) shows that the algorithm works very fast (see Section 6) and that it is reliable. Its error (non-detection) is a few percent (in one case the algorithm did not find the eyes, though in several cases it found more objects than the two eyes).

In this paper, we introduce the following topics: in Section two we discuss the face recognition steps. In Section three the head-detection problem is described and a CNN algorithm is proposed as solution. The fourth section talks about face normalization. The fifth section is devoted to the eye localization. Section six describes the details of the developed eye-finder CNN algorithm, while conclusion is drawn in Section seven.

2. Face Recognition Overview

Face recognition is described as follows identify the person(s) in a picture that is given on arbitrary media. Pictures can come from any scanned in photograph or digital graphics on computer as well as a shot of a video film. Face recognition is a complex task, which can be divided into several steps (Fig. 1).

In order to recognize objects, one must successfully locate the objects and extract useful features from them. For face recognition, the first step is to detect the presence of human head. This detection must not take a long time; it should use only one or two properties which can be checked fast. In video processing it means real-time because this is only the first filtering of the frames. The method should have low false rejection rate (FRR). We give a simple color based detection algorithm in section three.

Head localization is the next step in the procedure. For localizing the human head one could use more complicated features. This is a critical step in automated face recognition. CNN algorithms (see Section 6) can aid the complex feature extraction.

Further processing begins with a complex task, termed as capturing the face. The head position information is known from the previous step but the face should be normalized. The head is transformed in order to create canonical images, in which the face is scaled and rotated to a standard size. The process up until to this stage requires about half of the computation time of the whole face recognition [8]. It needs space and processing power in a traditional way, therefore if this normalization can be speeded up, the time of the whole procedure will be reduced proportionally. The CNN is an appropriate tool for accomplishing the task (see Section 4).

It should be determined what features are the best to represent the object. Two criteria are used to select a good feature vector: it must convey enough information for distinguishing the object from others, and it must be as invariant as possible. Computer vision research groups have developed many different methods for face representation. Several methods of different complexity are described in the literature ranging from PC computation power to

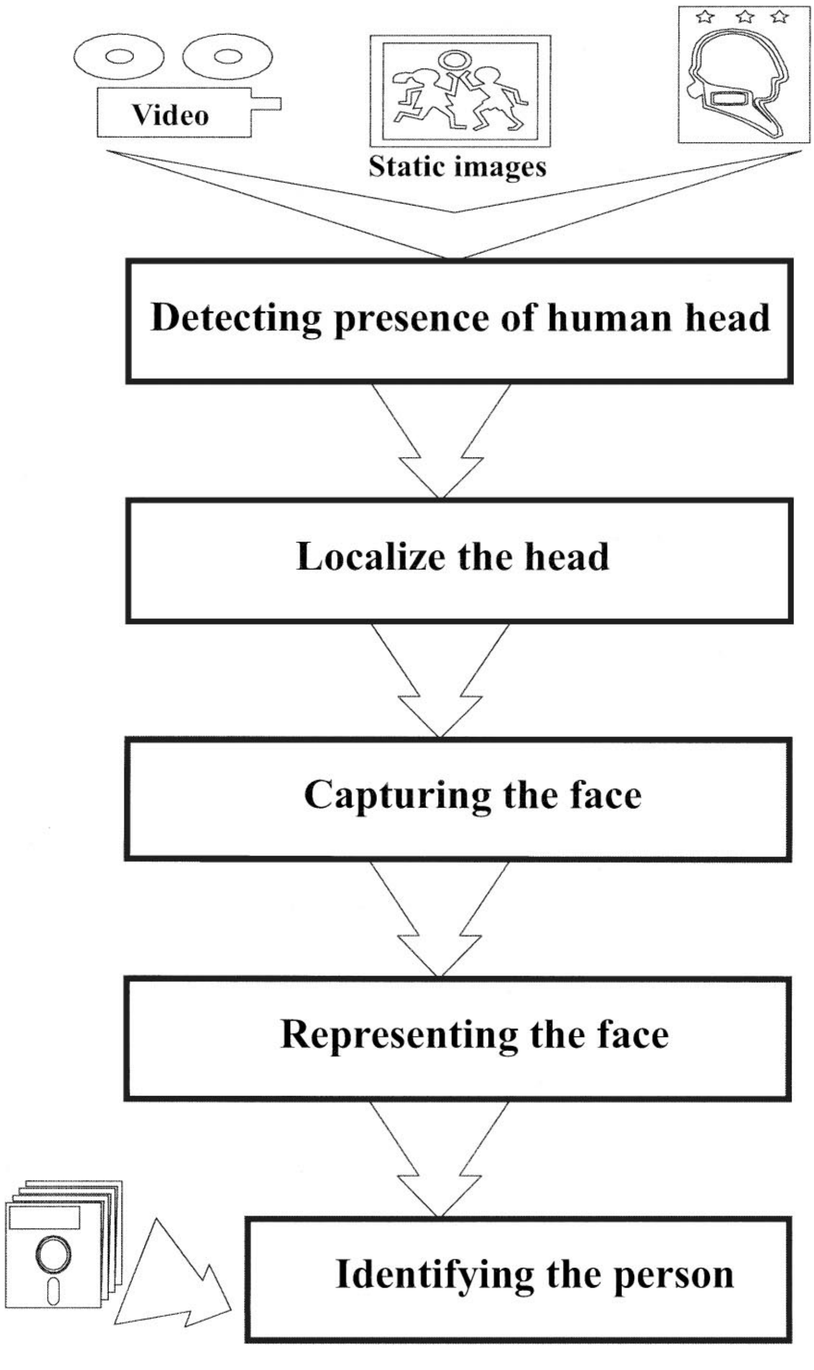

Figure 1. Stages of face recognition. Face recognition as a complex activity can be divided into several steps from detection of presence to database matching. The literature deals mainly with the representation and identification of faces.

supercomputing and from traditional methods [9] to soft computing paradigm [10] and statistics [11].

The feature vector can then be used to recognize a face from a given database. The database contains feature vectors of the people stored in it and/or their standard picture(s). The process identifies the person whose feature vector and that of the agent face has a minimum distance [12] if the difference is lower than a given threshold. The goal is to minimum false acceptance (FAR) *and* false rejection (FRR).

3. Head Detection

The head detection problem can be described as finding the location and size of every human face within the image. The head can be detected if its representative region is observed [13]. It is obvious that the whole frontal facial area contains information for recognition. The job of deciding which part of the object is the representative region is done before the detection and recognition processes begin. The developed CNN algorithm is based on natural human color information.

The complexity of head localization originates from the nature of humans and their environment.

- The size of the face in the scene can be diverse, therefore we need size invariant procedure to mask the heads.
- The background clutter causes difficulty especially if the head of the person is relatively small in the image. The texture of the background and clothes may lead to false localization as well.
- Diverse head poses, inclination and rotation produce very different visual objects (e.g. profile versus frontal views). This problem can be turned to our benefit if the image is from a live video. Then we can test liveliness with different face poses so the face recognition system will be safer. Intruders are not able to use static pictures for identifying themselves.

3.1. Human Head Features

The human head has some basic features. Many features such as bilateral symmetry, elliptical outline, not uniform field (as some background) are characteristic to humans.

First, we need to localize the person in the picture. In this task we can use color information. Human skin is never blue or green therefore we may assume that orange is in the center and that there is a broad spectrum around this center, which can accommodate yellow and white as well as light brown. Color recognition is a fast method for detecting the presence of human head [14, 15].

3.2. A Simple Color Head Localization CNN Algorithm

Detection of some head features can be done simply using CNN algorithms and it also solves most of the problems described previously. The CNN can perform size invariant algorithms, therefore it can extract heads of very different sizes. The background can usually be removed with skin color information. Clothes rarely have the same color as the face color.

At first, we apply color filter which involves some *threshold* template and logic instructions [16]. Then the gradient field is computed and the smallest objects are erased using the *small killer* template, because the human skin is non-uniform. After running of the algorithm, we get a mask of the color picture which indicates the possible head fields (Fig. 2). All the templates used in this task can be found in the CSL (CNN template library) [17].

4. Face Normalization

The possible head positions are known from the head detection step. The face capturing stage normalizes those faces (Fig. 3). Producing faces of standard size and direction is an important and time-consuming step. We were able to construct a CNN algorithm which significantly helps to solve this task.

Reference points should be found in the face to normalize it. One of the most fundamental references is eye position, because eyes are significantly different from other parts of the face and every human has got two of them. Once the eye positions are determined, all other important facial features, such as positions of nose and mouth, can easily be determined. Translation dependency can be eliminated by setting the origin of co-ordinates to a point that can be detected with sufficient accuracy in each image (e.g. eyes). We developed a CNN algorithm for localizing the eye areas (see Section 6).

Additional features should be extracted from the detected head. Detection of the nose line and that of

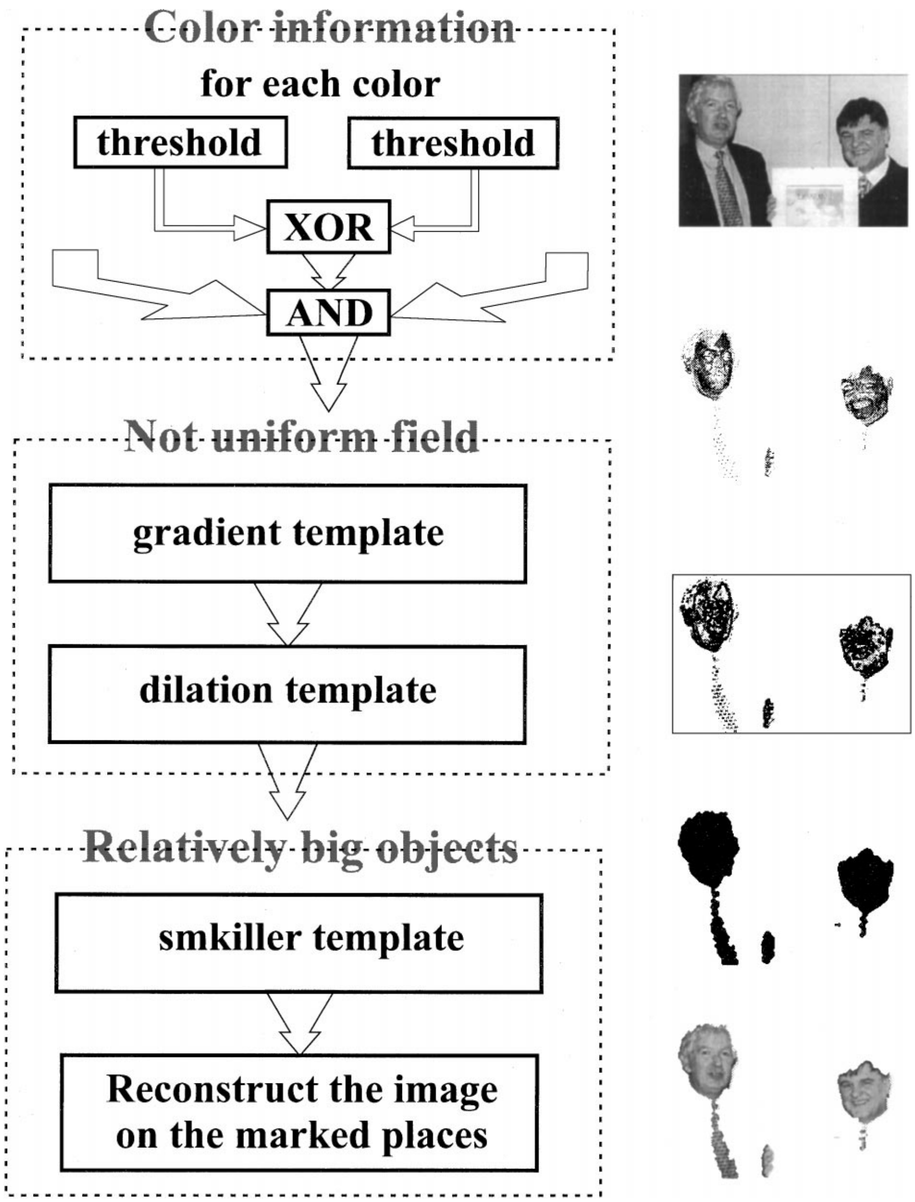

Figure 2. CNN algorithm for color head detection. The CNN algorithm is based on human color information and non-regularity of human skin. At first, we apply the color filter, then the gradient field is computed, finally the small areas are deleted. The output is a mask which indicates the possible heads on the color image. The templates can be found in the CNN template library and in Table 2.

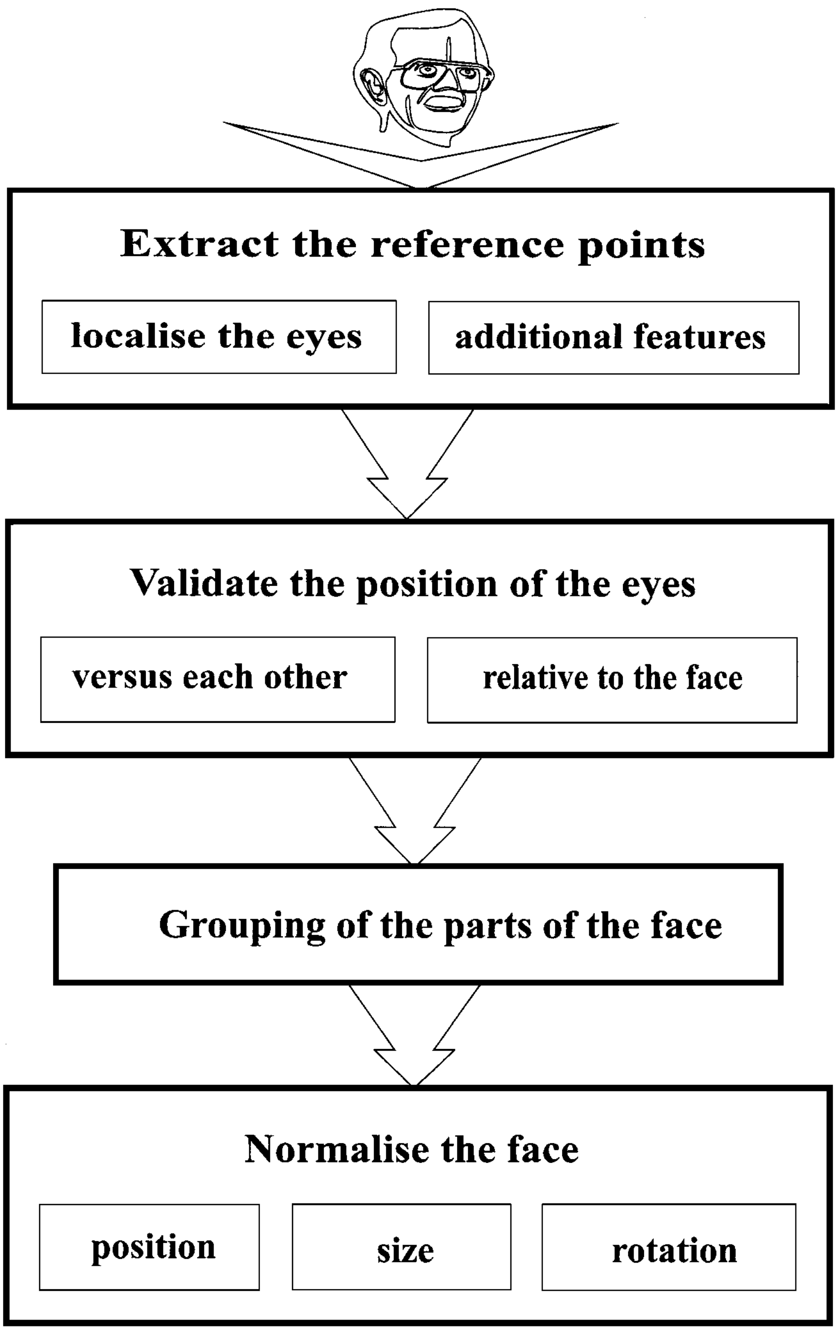

Figure 3. The stages of normalization. It is an important and time-consuming step to make standard size and direction faces. This process claims about half of the total computation time of the whole face recognition task until today. Once the eye positions are determined, all other important facial features can easily be determined.

the mouth area are not too complicated but their result is less reliable than detecting the position of the eyes. However, information on nose and mouth could be used to validate the position of the eyes.

One of the most critical issues in using a vector of features lies in that of proper normalization. The extracted features must be normalized in order to be independent of position, scale, and rotation of the face in the image. Scale and rotation invariance can be achieved by properly setting the interocular distance and the direction of the eye-to-eye axis. Normalization should rotate the head with the angle between the nose line and eye-to-eye axis. It should scale the face to adjust the distance between the eyes to a standard size so the whole face will be in a canonical form.

5. The Eye Localization

One of the most important tasks in face identification is eye localization. The eyes as reference points should be found in order to normalize the face. Eye detection is a more difficult task than head localization because the eye areas are usually very small. We propose an algorithm to solve this problem in a wide spectrum of images.

The classical approaches are based on holistic features (image blocks) such as eigenfaces [18], Probabilistic Decision Based Neural Network [8], Multi Level Perceptron [19], and the convolution of the image with a linear or non-linear filter estimated from training data. These methods are very sensitive to the statistical characteristics of the training patterns (e.g. the existence of appropriate negative examples). Besides, they generalize poorly to varying image capture conditions.

5.1. Problems of Eye Detection

The three sources of problem are the quality of the image, the countenance of the person, and the position of the camera.

- The picture of the face is view dependent. The image of the face varies according to the viewing direction. Diverse head poses, inclination and rotation produce very different visual objects.
- The nonrigidity of the person produces different facial expressions, therefore the images will be different from a fixed viewpoint, too. The eyes may be open, closed, or semi-closed. The occurrence of facial details such as glasses (which often blur the eye areas), moustaches and beards or face-painting disturb the normal face.
- With the same viewpoint and the same facial expression, the image can still be different due to diverse lighting environments. The lighting direction and intensity pose problems in all fields of computer vision, the solution to which requires space-variant illumination normalization [20] or adaptive histogram equalization [21].

5.2. Human Eye Features

Several properties of grayscale eyes can be useful. Large vertical gradients are characteristic features and the eyes are dark spots, which are relatively small and round objects. These properties are local and independent of view. The CNN is mainly a space invariant computing system, and thus is well suited to implement local feature detectors. Glasses do not disturb detection because the frames of the glasses are bigger than the extended eye areas. The detection will be successful if the eyes are open or semi-closed. Strange parts on the face (e.g. hand, glasses, and painting) or the hairy fields do not create confusion for the procedure. The developed algorithm uses two additional features: the two eyes are located about in the same vertical position and they are inside the face.

6. The Eye Detection CNN Algorithm

First we assume that of the face occupies the main part of the image. The algorithm can be split into five different parts according to basic properties (Fig. 4). The algorithm runs in less than three ms on a VLSI CNN chip [22].

$$(200 \text{ (normal template)} + 360 \text{ (propagating template)}) \\ * 5\,\mu\text{s (time constant)} < 3\,\text{ms}$$

Vertical gradients are useful to detect the eyes, and the vertical position is estimated using anthropometric standards. A vertical gradient map is computed first, which results in the first mask.

The eyes are dark spots which can be indicated by the *junction* template. The algorithm can detect the eyes in every image where eyeballs are bigger than twice by two pixels. On the original image, the junction

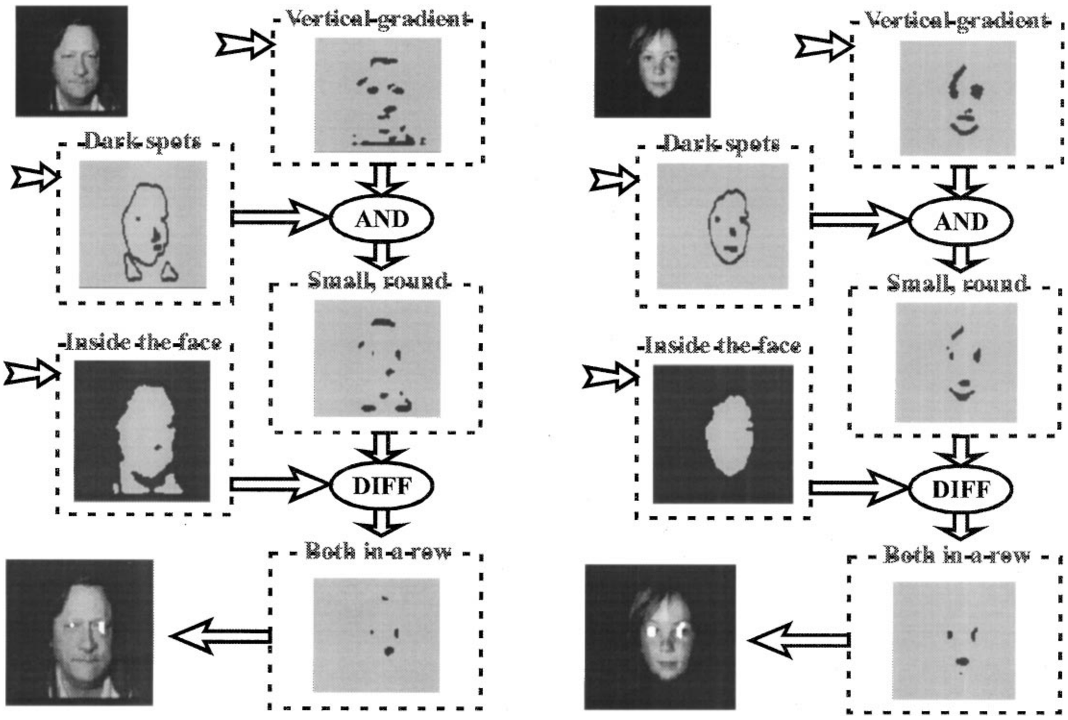

Figure 4. The logic of the CNN eye detection algorithm. The algorithm can be split into five different parts according to basic properties. The input of the algorithm is a grayscale picture. Temporary images and the output are binary. The eyes are marked white spot on the original grayscale image. See Figs. 5–9 for details.

template searches for and joins the dark fields this is the second mask.

The eye-area shape possesses both of the above-defined properties. Therefore, we should calculate the intersection of the previous two fields. The convex corners are filled out to get round objects from that intersection. Adding the first and second masks together (logical and), we get the possible eye areas. Next we create round objects. Then the picture gets eroded in some steps, but objects that have been removed completely are reconstructed at the same time so we get the third mask.

The eyes are relatively small objects inside the face with definite edges. Near the edge of the face some false indicator can usually be found. To remove those we distinguish the grayscale face area and create the vertical gradient of the mask, make it wider, and subtract it from the third mask.

Finally, a global feature is used. Every person has two eyes horizontally in a row by looking into the camera. The image of the appropriate objects is the output of the algorithm; these spots are not necessarily only the eyes but all possible areas (including eyes) are marked.

The input of the algorithm is a grayscale picture while the temporary images and the output are binary (only black and white pixels are allowed). The figures show the details of the stage of the algorithm on the left side and give an example on the right (Figs. 5–9). For better illustration the eyes are marked by white on grayscale images and working pictures use white-gray transformation for better visibility (Figs. 4–9). The templates can be found in CSL (CNN Template Library [17]) a summary of which is shown in Table 2.

6.1. *Gradient Computing*

We perform gradient analysis by partitioning the gradient map in terms of edge directions. There are two main directions in face pictures: horizontal and vertical. Horizontal gradients are useful to detect the left and right boundaries of face and nose, while vertical gradients are useful to descry the head top, eyes, nose base and mouth. A pixel is considered to be in the vertical edge map if the magnitude of the vertical component of the gradient at that pixel is greater than the

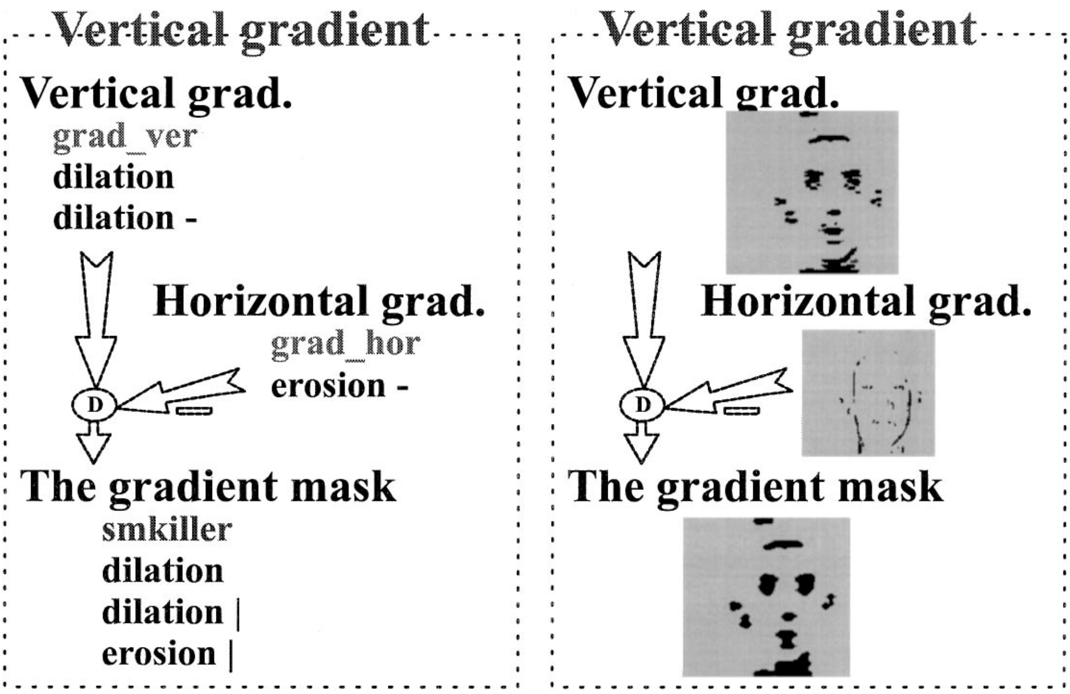

Figure 5. Vertical gradient map. Gradients are useful to detect the eyes. The figure shows the details of the stage of the algorithm on the left side and gives an example on the right. Working pictures use white-gray transformation for better visibility. You can find the input image on Fig. 9. The templates and its functions can be found in Table 2.

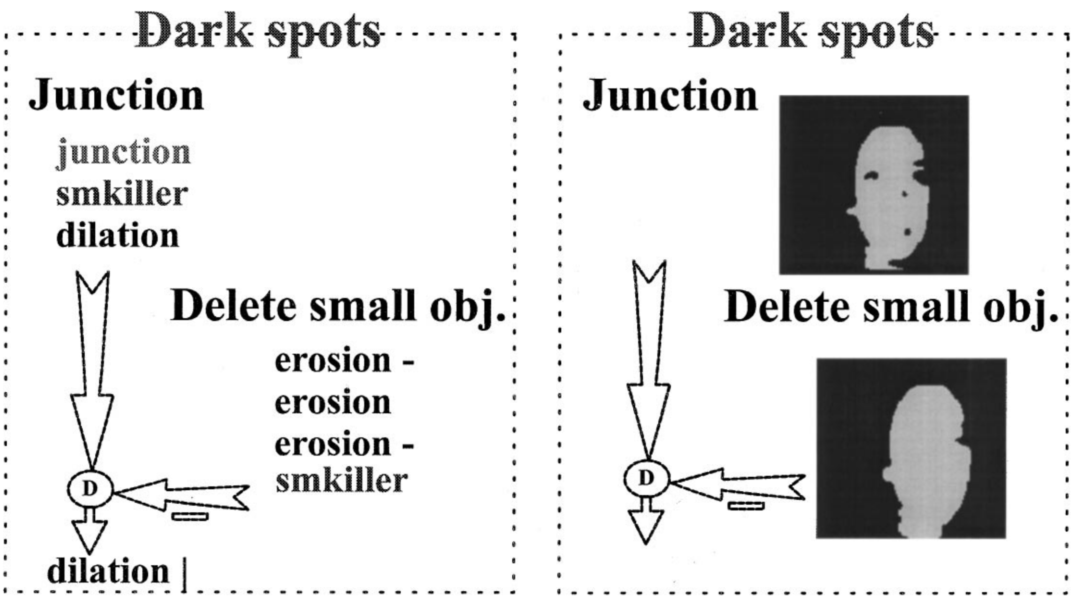

Figure 6. Detection of dark fields. The eyes are dark spots which are indicated by the junction template. The figure shows the details of the stage of the process on the left side and give the require CNN templates on the right.

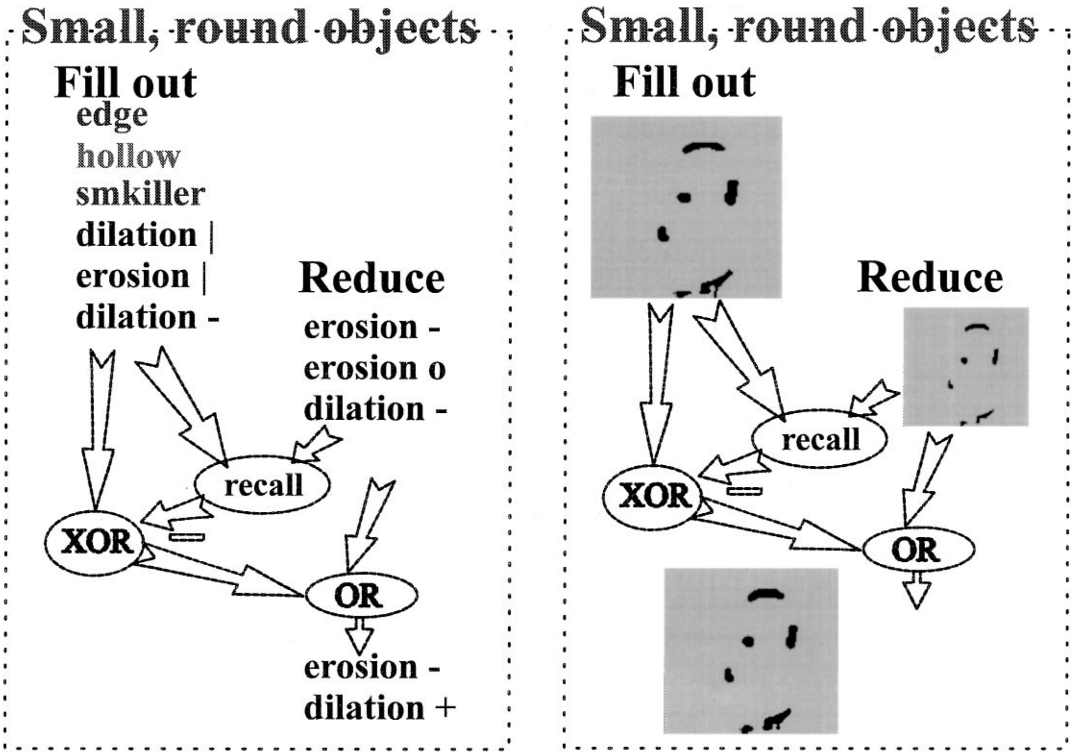

Figure 7. Detection of small round spots. The details of detection are on the left side and the example is on the right. The templates can be found in the CNN template library. The *hollow* template is the most suitable for fill out the corners, because the template produces many convex and round objects.

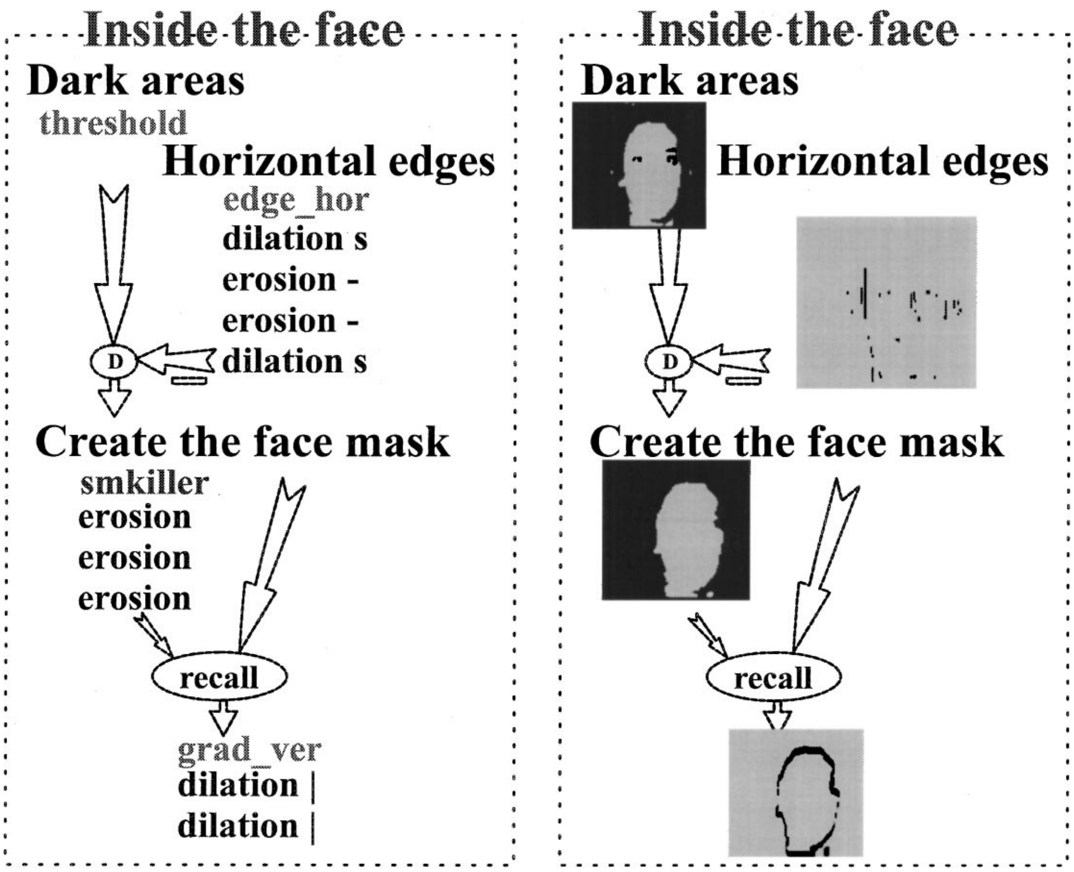

Figure 8. Grayscale head detection. The eyes are relatively small objects inside the face with definite edges. This stage of the face detection algorithm could be an independent routine, which works on images with mainly black background. It creates a mask of the inner face. The figure shows the details of the algorithm on the left side and gives an example on the right.

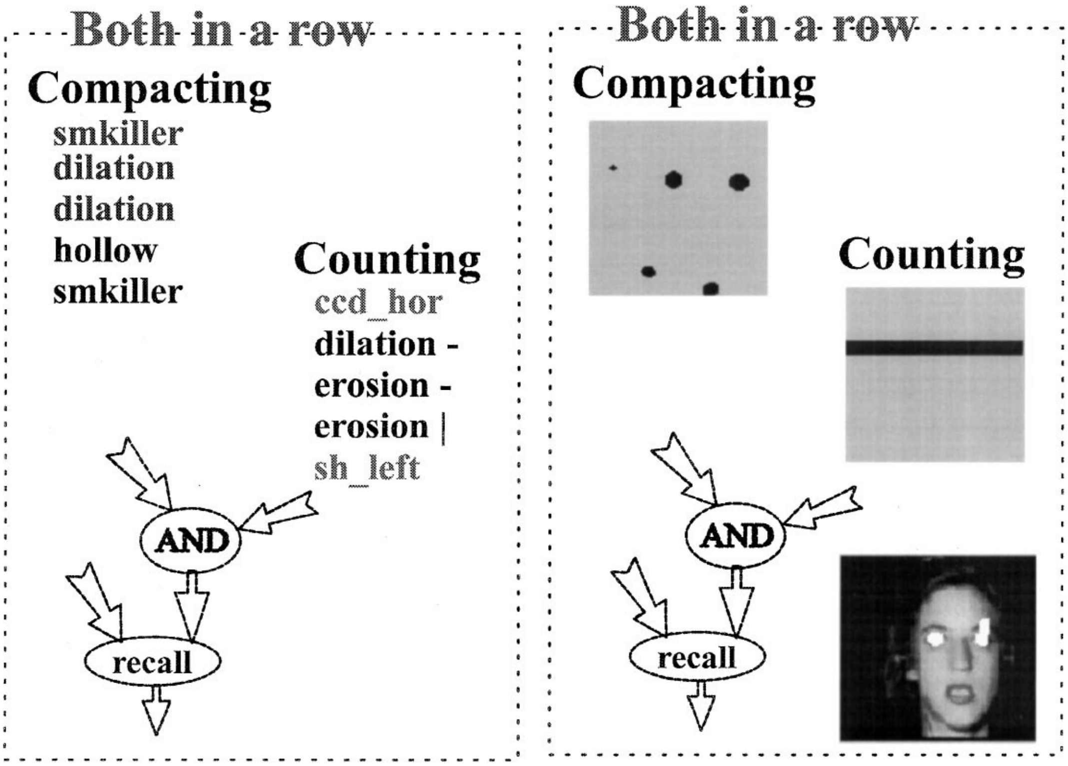

Figure 9. More than one object horizontally in a row. Every person has two eyes in a horizontal row by seeing in the camera, therefore the task is to count the objects in a row and if it is less than two delete it. The figure shows the details of the process on the left side and gives the output of the whole CNN algorithm on the right side. The eyes are marked white spots on the original grayscale image.

horizontal one. The output pixel will be black if the gradient intensity is above a selected threshold.

The CNN template library contains *gradient* and *edge* templates, which need to be adjusted in accordance with direction. Gradient computation is an intensity sensitive non-linear template. We have used the following vertical gradient template (*grad_ver*):

$$A = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \quad B = \begin{array}{|c|c|c|}\hline 0 & b & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & b & 0 \\ \hline \end{array}$$

$$z = \boxed{-1.5} \quad b = \begin{cases} 2, & \text{if } |u_{ij} - u_{kl}| > 2 \\ |u_{ij} - u_{kl}|, & \text{otherwise} \end{cases}$$

The template detects areas where the gradient is larger than a given value. The edge detection template is a simple linear template.

Vertical gradients are computed, because the neighborhood of the eyes is characterized by large gradients. Horizontal gradients are computed too, and are subtracted from the previous mask because the picture could have diagonal lines indicated also with vertical gradient. Finally we delete the small objects and enlarge those that remain after *small kill*ing (Fig. 5).

6.2. *Detection of Dark Fields*

The background of the image is usually dark and so the *junction* template marks those pixels. Some *erosion* on the image deletes the small inner face pixels (eyes) therefore the large homogeneous fields will remain. The modified image is then subtracted from the original one to yield an output that contains the relatively small and dark areas only (Fig. 6).

Dark fields with appropriate round shape (e.g. a square dark field is not suitable) should be searched and joined in the original image to mark the dark spots. The *junction* template is better for this than the *threshold* template; both can be used on grayscale images. The *junction* template is originally made for binary pictures but it worked successfully for grayscale images too.

Table 2. The applied CNN templates and its effects.

Template	Function
ccd_hor	It indicates the number of objects in a row at the right side of the image. The result is some black pixels separated with one white pixel.
dilation	It is a basic binary morphological operator; the objects are made wider.
dilation − dilation | dilation +	The mathematical morphology employs a structuring element. The shape of the structuring element is indicated with a sign after the name of the template (e.g. |, -).
dilation s	It is a special vertical line filter, see details in Section 6.4.
edge	The contour of the objects can be detected.
erosion	It is a basic binary morphological operator; the objects are made thinner.
erosion | erosion o erosion −	The mathematical morphology employs a structuring element. The shape of the structuring element is indicated with a sign after the name of the template (e.g. |, -).
gradient	The pixel will be black if the gradient intensity is above a selected threshold.
grad_hor grad_ver	It is detecting areas where the directional gradient is larger than a given value.
hollow	It fills out the convex corners.
junction	Dark fields with appropriate shape (e.g. a square dark field is not suitable) can be searched and joined by this template.
recall	It reconstructs the marked objects.
sh_left	Each black pixel is spread horizontally from right to left.
smkiller (small killer)	The small killer template deletes all separated pixels (white pixels in black surroundings and black pixels in white environment).
threshold	It marks the pixels that have values higher than a given level.
logdiff (D)	Local boolean operator: the output pixel is black only if the state pixel is black and the input pixel is white.
AND	The output pixel is black only if both pixels are black.
OR	The output pixel is white only if both pixels are white.
XOR	The output pixel is black if the color of the pixels is not the same.

The applied junction template is:

$$A = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \quad B = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 6 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

$$z = \boxed{-3}$$

The output is a binary picture on which binary morphological templates can operate. It is easy to delete the unnecessary pixels by a good direction resolving *erosion* template. Mathematical morphology employs a structuring element, the shape of which is indicated with a sign after the name of the mathematical morphology template (e.g. |, -).

6.3. *Small Round Spots*

Adding the first and second masks together, we get the possible eye areas. The shape of the eye is not concave but round, so long lines and concave objects should be deleted.

The linear *edge* template is utilized for detecting the contour of the objects. We should fill out the convex corners to get many round objects. The *hollow* template is the most suitable for this task, according to the experiments, because the template produces many convex and round objects.

If the object is too big or too long, the edge of the object will remain after the filling. They should therefore be deleted with the *smkiller* template. The fragmented objects close enough to each other are joined with a single *dilation* template.

The great advantage of using the *smkiller* template is that it not only deletes the separated pixels but also makes the bigger objects more compact. It provides simple noise reduction, too.

In the next stage the picture gets eroded in two steps, but the *recall* template reconstructs completely removed objects at the same time, the result of which is the third mask. This last mask contains all the possible eye-areas because the algorithm deleted only those fields, which are certainly not the neighborhood of the eyes (Fig. 7).

6.4. Grayscale Head Detection

This stage could be an independent CNN algorithm which works only with uniform and mainly black background. It creates a mask of the inner face. The *threshold* template computes the dark parts of the original image, which becomes the background mask. The black fields will indicate the surroundings of the face as well as the dark parts of it, assuming the image satisfies the condition mentioned above. If the eyes are too close to the edge of the face and the light direction is not perfect, the shadow may join the mask. The method calculates the horizontal edges of the original image and subtracts its vertical lines from the background mask to compensate for the information loss. We used the following *dilation* template instead of the common dilation for vertical line filtering:

$$A = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \quad B = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

$$z = \boxed{1}$$

Small and medium-sized objects are erased with *erosion* templates. Afterwards the remaining object gets recalled, and this is expected to give the inverted face-mask (Fig. 8). The applied *threshold* and *smkiller* templates can be found in the CNN template library. The direction sensitive edge template (*edge_hor*) is:

$$A = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \quad B = \begin{array}{|c|c|c|}\hline -1 & 0 & -1 \\ \hline -1 & 6 & -1 \\ \hline -1 & 0 & -1 \\ \hline \end{array}$$

$$z = \boxed{-0.5}$$

6.5. The Eyes are Horizontally in a Row

The processed image contains only one face because this was the goal of the face detection stage. Most of the pictures show a frontal view of the person, the total profile view is quite rare. The eyes are located in the same vertical position if the angle of the rotation of the person's head is only a few degrees. This global property can be utilized in this case.

The task is to count the objects in a row and if there is only one delete it. The objects should be compacted and magnified. The algorithm uses the *hollow* and the *small killer* templates because the concave objects give false result.

The *ccd_hor* template indicates the objects in a row at the right side of the image plane. The result of a row is some black pixel separated with one white pixel. The black pixels show the number of the connected components (spots) in the row. The row should be deleted if there is only one black pixel in it. A morphological horizontal *close* operator does the erasure, which joins the dark pixels and deletes the separated ones.

Overlapping objects can be filtered using the vertical erosion (*erosion* |) template. The black pixels indicate the searched rows. They are spread horizontally with the *sh_left* template; therefore the appropriate objects are marked.

The algorithm recalls the spots, which are the final output of the process (Fig. 9). The previous steps ensure that all of the possible eye areas marked, which implies that the eyes are found, too.

7. Conclusions

A novel approach to the face detection problem is given, based on analogic cellular neural network algorithms. It describes a completely new method for the finding and helping the normalization of faces, which is a critical step of this complex task but hardly ever discussed in the literature. Though still in an experimental stage, we found that the suggested CNN algorithm finds and normalizes human faces effectively while its time requirement is a fraction of the previously used methods.

The algorithm starts with the detection of heads on color pictures using deviations in color and structure of the human face and that of the background. Reference points are needed for normalization of the face. The eyes were found the most reliable for this purpose. By normalizing the distance and position of the reference points, all faces could be transformed into the same size and position.

Another CNN algorithm finds the eyes on any grayscale image by searching some characteristic features of the eyes and eye sockets. The investigated features are useful for other recognition and detection problems too, not only in face detection. The parts of the algorithm can be used in various tasks as a subroutine.

Tests made on standard database show that the algorithm works very fast and it is reliable. The time-requirement of the algorithm is about 3 ms which is acceptable for real-time applications (e.g. security identification, image query, etc.).

Since the suitability of the algorithms are proved the next step is to develop a complete system and a hardware implementation.

Acknowledgments

The support of the OTKA (Hungarian National Scientific Fund, grant No. T026555) and the Hungarian Academy of Sciences is acknowledged.

References

1. L.O. Chua and L. Yang, "Cellular neural networks: Theory," *IEEE Trans. on Circuits and Systems*, Vol. 35, pp. 1257–1290, 1988.
2. L.O. Chua and T. Roska, "The CNN paradigm," *IEEE Trans. on Circuits and Systems I*, Vol. 40, pp. 147–156, 1993.
3. F. Werblin, T. Roska and L.O. Chua, "The analogic cellular neural network as a bionic eye," *Journal of Circuit Theory and Applications*, Vol. 23, pp. 541–569, 1995.
4. T. Roska, "CNN chip set architectures and the visual mouse," *IEEE Int. Workshop CNNA '96*, Sevilla, pp. 487–492, 1996.
5. L.O. Chua and T. Roska, "Cellular neural networks: Foundations and primer," *Lecture notes for curse EE129 at U.C.Berkeley*, 1997.
6. T. Roska and L.O. Chua, "The CNN universal machine: An analogic array computer," *IEEE Trans. on Circuits and Systems II*, Vol. 40, pp. 163–173, 1993.
7. S.Y. Kung, "Neural networks for intelligent multimedia processing," *Proc. IEEE*, Vol. 86, No. 6, pp. 1244–1272, 1998.
8. S.-H. Lin, S.Y. Kung, and L.-J. Lin, "Face recognition/detection by probabilistic decision-based neural network," *IEEE Trans. on Neural Networks*, Vol. 8, No. 1, pp. 114–132, 1997.
9. T. Kanade, "Picture processing by computer complex and recognition of human faces," Technical Report, Kyoto University, Dept. of Information Science, 1973.
10. N. Intrator, D. Reisfeld, and Y. Yeshurun, "Face recognition using a hybrid supervised/unsupervised neural network," *Patt. Recog. Lett.*, Vol. 17, pp. 67–76, 1996.
11. P. Penev and J. Atick, "Local Feature Analysis: A general statistical theory for object representation," *Network: Comp. Neural Sys.*, Vol. 7, pp. 447–500, 1996.
12. I. Szatmári and T. Roska, "The CNN implementation of wave type metric for image analysis and classification," *IEEE Int. Workshop on Nonlinear Dynamics of Electronic Systems (NDES'98)*, Budapest, pp. 137–140, 1998.
13. A.H. Gee and R. Cipolla, "Fast visual tracking by temporal consensus," Univ. Cambridge, Dept. Engineering Report TR 207, 1995.
14. C.H. Lee, J.S. Kim, and K.H. Park, "Automatic human face location in a complex background using motion and color information," *Patt. Recog.*, Vol. 29, pp. 1863–1876, 1996.
15. E. Littmann and H. Ritter, "Adaptive color segmentation: A comparison of neural and statistical methods," *IEEE Trans. Neural Networks*, Vol. 8, pp. 175–185, 1997.
16. T. Roska, A. Zarándy, and L.O. Chua, "Color image processing using CNN," *Proc. Eur. Conf. on Circuit Theory and Design, Part I, Elsevier*, Amsterdam, pp. 57–62, 1993.
17. T. Roska, L. Kék, L. Nemes, Á. Zarándy, M. Brendel, and P. Szolgay, (Eds.), "CNN software library" (Templates and Algorithms) Version 7.2, DNS-CADET-15, Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, 1998.
18. M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, Vol. 3, pp. 71–86, 1991.
19. K.K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *AI Memo 1521/Cbcl*, MIT, Cambridge, MA, 1994.
20. P.J. Philips and Y. Bardi, "Efficient illumination normalisation of facial images," *Patt. Recog. Letters*, Vol. 17, pp. 921–927, 1996.
21. M. Csapody and T. Roska, "Adaptive histogram equalization with CNN," *IEEE Int. Workshop CNNA '96*, Sevilla, pp. 81–86, 1996.
22. G. Linán, S. Espejo, R.D. Castro, E. Roca, and A. Rodrigez-Vázquez, "CNNUC3: A mixed-signal 64×64 CNN universal chip," *Int. Conf. on Microelectronics for Neural, Fuzzy, and Bio-Inspirated Systems (MicroNeuro99)*, Granada, pp. 61–68, 1999.
23. S. Gutta, J. Huang, B. Takacs, and H. Wechsler, "Face recognition using ensembles networks," *Int. Conf. on Pattern Recognition*, Vienna, Austria, 1996.
24. M.S. Lew, "Information theoretic view-based and modular face and modular face detection," *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition (FG96)*, 1996.
25. B. Scassellati, "Finding eyes and faces with foveated vision system," AAAI, 1998.
26. H. Rowley, S. Baluja and T. Kanade, "Human face detection in visual scenes," CMU-CS-95-158, Carnegie Mellon University, 1995.

David Balya was born in Budapest, Hungary. He received the M.Sc. degree in computer science from Technical University of Budapest, Budapest, Hungary, in 1999. Currently, he is with the Neuromorphic Information Technology Graduate Center, where he is working towards the Ph.D. degree. His research interests include nonlinear dynamical systems, especially CNN, neural networks, artificial intelligence and application of mathematics and computer science in social sciences.

Tamás Roska (M'87-SM-90-F'93) received the Diploma in electrical engineering from the Technical University of Budapest, Budapest, Hungary, in 1964 and the Ph.D. and D.Sc. degrees in 1973 and 1982, respectively. From 1964 to 1970 he was with the Measuring Instrument Research Institute, Budapest. From 1970 to 1982 he was with the Research Institute for Telecommunication, Budapest. Since 1982 he has been with the Computer and Automation Institute of the Hungarian Academy of Sciences, where he is Head of the Analogic and Neural Computing Research Laboratory. Since 1989 he has been a Visiting Scholar at the Department of Electrical Engineering and Computer Sciences and the Electronics Research Laboratory and was recently a Visiting Research Professor at the Vision Research Laboratory of the University of California at Berkeley. His main research areas are cellular neural networks, nonlinear circuits and systems, and analogic spatiotemporal supercomputing. Since 1975 he has been a member of the Thecnical Committee on Nonlinear Circuits and Systems of the IEEE Circuits and Systems Society. Recently, he has served twice as Associate Editor of the IEEE Transactions on Circuits and Systems. He received the IEEE Fellow Award for contributions to the qualitative theory of nonlinear circuits and the theory and design of programmable cellular neural networks. In 1993 he was elected a Member of the Academia Europaea (European Academy of Sciences, London) and the Hungarian Academy of Sciences. In 1999 he became the Founding Chair of the Technical Committee on Cellular Neural Networks and Array Computing of the IEEE Circuits and Systems Society.
email: roska@lutra.sztaki.hu