A Neural Scheme for Robust Detection of Transparent Logos in TV Programs

Stefan Duffner and Christophe Garcia

France Telecom Division Research & Development,
4, rue du Clos Courtel, 35512 Cesson-Sévigné, France,
{stefan.duffner, christophe.garcia}@francetelecom.com

Abstract. In this paper, we present a connectionist approach for detecting and precisely localizing transparent logos in TV programs. Our system automatically synthesizes simple problem-specific feature extractors from a training set of logo images, without making any assumptions or using any hand-made design concerning the features to extract or the areas of the logo pattern to analyze. We present in detail the design of our architecture, our learning strategy and the resulting process of logo detection. We also provide experimental results to illustrate the robustness of our approach, that does not require any local preprocessing and leads to a straightforward real time implementation.

1 Introduction

In the last decade, we have entered the digital era, with the convergence of telecommunication, video and informatics. Our society (press agencies, television channels, customers) is producing daily extremely large and increasing amounts of digital images and videos, making it more and more difficult to track and access this content, with traditional database search engines, requiring tedious manual annotation of keywords or comments. Therefore, automatic content-based indexing has become one of the most important and challenging issues for the years to come, in order to face the limitations of traditional information systems. Some expected applications are [6, 7, 9]: Information and entertainment, video production and distribution, professional video archive management including legacy footages, teaching, training, enterprise or institutional communication, TV program monitoring, self-produced content management, internet search engines and video conference archiving and management.

The recent progresses in the field of object detection and recognition tend to make possible a large range of applications that require accessing the semantic content and identifying high-level indices, regardless of the global context of the image, in order to ease automatic indexing and provide more intuitive formulation of user requests. For instance, human face detection can now be considered as a very mature tool, even though progresses have still to be made for full-profile view detection and accurate facial feature detection, for allowing robust face recognition. Recently, Garcia and Delakis [2] proposed a near-real time neural-based face detection scheme, named "Convolutional Face Finder" (CFF) that has been designed to precisely locate multiple faces of minimum size 20x20 pixels and variable appearance, rotated up to 30 degrees in image plane and turned up to 60 degrees, in complex real world images. A detection rate of 90.3% with 8 false positives have been reported on the CMU test set, which are the best results published so far on this test set. Locating a face and recognizing it [10] tend to appear as a required functionality in state-of-the-art systems, working with professional videos or personal digital image collections. Another important expected functionality is superimposed text detection and recognition.

Even though lots of progresses have been recently made in the field of object detection and recognition, most approaches have focused on image of objects variable in scale and orientation, but with small variation in shape or global gray level appearance. There is a still a lot to be done in the case of deformable 3D object detection but also in the case of very variable object texture appearance.

In this paper, we will focus on the specific case of transparent object detection in images, which is a very challenging problem. We propose a general solution that will be evaluated on the problem of transparent logo detection in video programs. For illustration purposes, we will focus on the detection of the logo of the France 2 television channel (FR2 logo), as shown in Fig. 1. Note that the proposed method is very generic and can be applied to other logos in a straightforward way. Most logo detection approach consider opaque logos superimposed on video frames. In that case, pixels inside the logo boundaries keep approximately the same values from one frame to the next, with a certain amount of noise due to video coding. Only pixels outside the logo boundaries are variable. In the case of transparent logo, pixels inside the logo boundaries also change depending on the video underneath. If temporal constancy of pixel inside opaque logo can ease the detection process, by temporal gradient analysis [4] or low level based pattern matching techniques [4, 1, 11], this is not the case for transparent logos, where all pixels strongly vary at the same time depending on the background.

To face this challenge, we propose an image-based approach that is designed to precisely detect transparent patterns of variable size, in complex real world video images. Our system is based on a convolutional neural network architecture [3], directly inspired from our face detector, the Convolutional Face Finder (CFF) described in [2]. It automatically derives problem-specific feature extractors, from a large training set of logo and non-logo patterns, without making any assumptions about the features to extract or the areas of the logo patterns to analyze. Once trained, our system acts like a fast pipeline of simple convolutions and subsampling modules, that treat the raw input image as a whole, for each analyzed scale, and does not require any costly local preprocessing before classification. Such a scheme provides very high detection rate with a particularly low level of false positives, demonstrated on difficult videos, maintaining a near real time processing speed.

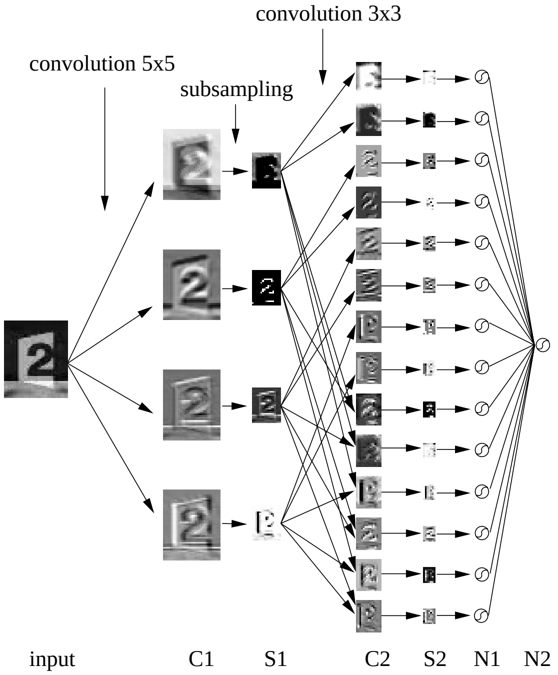

Fig. 1. The convolutional architecture

The remainder of the paper is organized as follows. In section 2, we describe the architecture of the proposed transparent logo detection system. In sections 3 and 4, we explain in detail the way we train and apply the built detector. In section 5, we assess the performance of our approach by analyzing its precision. Some experimental results obtained on images of complex scenes are also presented to demonstrate the effectiveness and the robustness of the proposed approach. Finally, conclusions are drawn.

2 System Architecture

The convolutional neural network, shown in Fig. 1, consists of a set of three different kinds of layers. Layers Ci are called convolutional layers, which contain a certain number of planes. Layer C1 is connected to the retina, receiving the image area to classify as logo or non-logo. Each unit in a plane receives input from a small neighborhood (biological local receptive field) in the planes of the previous layer. The trainable weights (convolutional mask) forming the receptive field for a plane are forced to be equal at all points in the plane (weight sharing). Each plane can be considered as a feature map that has a fixed feature detector that corresponds to a pure convolution with a trainable mask, applied over the planes in the previous layer. A trainable bias is added to the results of each convolutional mask. Multiple planes are used in each layer so that multiple features can be detected.

Once a feature has been detected, its exact location is less important. Hence, each convolutional layer Ci is typically followed by another layer Si that performs local averaging and subsampling operations. More precisely, a local averaging over a neighborhood of four inputs is performed followed by a multiplication by a trainable coefficient and the addition of a trainable bias. This subsampling operation reduces by two the dimensionality of the input and increases the degrees of invariance to translation, scale, and deformation of the learnt patterns.

The different parameters governing the proposed architecture, i.e., the number of layers, the number of planes and their connectivity, as well as the size of the receptive fields, have been experimentally chosen. Practically, different architectures have been iteratively built, trained, and tested over training sets. We retained the architecture that performed efficiently in terms of good detection rates and especially in terms of false alarm rejection, while still containing an acceptable number of free parameters.

Layers C1 and C2 perform convolutions with trainable masks of dimension 5x5 and 3x3 respectively. Layer C1 contains four feature maps and therefore performs four convolutions on the input image. Layers S1 and C2 are partially connected. Mixing the outputs of feature maps helps in combining different features, thus in extracting more complex information. In our system, layer C2 has 14 feature maps. Each of the four subsampled feature maps of S1 is convolved by two different trainable masks 3x3, providing eight feature maps in C2. The other six feature maps of C2 are obtained by fusing the results of two convolutions on each possible pair of feature maps of S1. Layers N1 and N2 contain simple sigmoid neurons. The role of these layers is to perform classification, after feature extraction and input dimensionality reduction are performed. In layer N1, each neuron is fully connected to exactly one feature map of layer S2. The unique neuron of layer N2 is fully connected to all the neurons of the layer N1. The output of this neuron is used to classify the input image as logo or non-logo. For training the network, we used the classical backpropagation algorithm with momentum modified for being used in convolutional networks as described in [3]. Desired responses are set to -1 for non-logo and to +1 for logo.

In our system, the dimension of the retina is 38x46. Because of weight sharing, the network has only 1147 trainable parameters. Local receptive fields, weight sharing and subsampling provide many advantages to solve two important problems at the same time: the problem of robustness and the problem of good generalization, which is critical given the impossibility of gathering in one finite-sized training set all the possible variations of the logo pattern. This topology has another decisive advantage. In order to search for a specific pattern, the network must be replicated (or scanned) at all locations in the input image, as classically done in detection approaches [5, 8]. In our approach, since each layer essentially performs a convolution with a small-size kernel, a very large part of the computation is in common between two neighboring logo window locations in the input images. This redundancy is naturally eliminated by performing

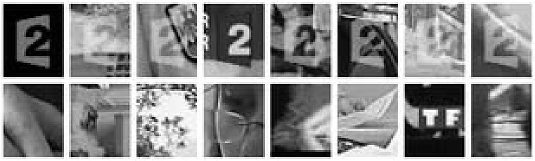

Fig. 2. Some samples of the training set. The last row shows initial negative examples.

the convolutions corresponding to each layer on the entire input image at once. The overall computation amounts to a succession of convolutions and non-linear transformations over the entire images.

3 Training Methodology

The FR2 logo examples used to train the network were collected from various video segments, during a 12 hour broadcast of the FR2 TV channel. Some of the $1,993$ collected FR2 logo images are shown in the first row of Fig. 2. Collecting a representative set of non-logos is more difficult as virtually any random image could belong to it. A practical solution to this problem consists in a bootstrapping strategy [8], in which the system is iteratively re-trained with false alarms produced when applied to a set of video images, that do not contain the targeted logo. In the proposed approach, we improved this strategy. Before proceeding with the bootstrapping, an initial training set of $2,313$ non-logo patterns was built by randomly cropping images from video frames. Some non-logo patterns (negative examples) are shown in Fig. 2. The proposed bootstrapping procedure is presented in table 1. In step 1, a validation set is built and used for testing the generalization ability of the network during learning and, finally, selecting the

1. Create a validation set of 400 logo images and 400 non-logo images randomly extracted and excluded from the initial training set. It will be used to choose the best performing weight configuration during steps 3 and 8.
2. Set $BIter = 0$, $ThrFa = 0.8$.
3. Train the network for 60 learning epochs. Use an equal number of positive and negative examples in each epoch. Set $BIter = BIter + 1$.
4. Gather false alarms from a set of 300 video frames with network answers above *ThrFa*. Collect at maximum $5,000$ new examples.
5. Concatenate the newly created examples to the non-logo training set.
6. If $ThrFa \geq 0.2$ set $ThrFa = ThrFa - 0.2$.
7. If $BIter < 6$ go to step 3.
8. Train the network for 60 more learning epochs and exit.

Table 1. The proposed bootstrapping scheme.

weight configuration that performs best on it. This validation set is kept constant through all the bootstrapping iterations, in contrast with the training set which is updated. In step 3, the backpropagation algorithm is used with the addition of a momentum term for neurons belonging to the N1 and N2 layers. Stochastic learning was preferred versus batch learning. For each learning epoch, an equal number of examples from both classes are presented to the network giving no bias toward one of the two classes.

The generation of the new patterns that will be added to the non-logo training set is carried out by step 4. The false alarms produced in this step force the network, in the next iteration, to refine its current decision boundary for the FR2 logo class. At each iteration, the false alarms, giving network answers greater than *ThrFa*, and therefore strongly misclassified, are selected. As the network generalizes from these examples, *ThrFa* is gradually reduced until reaching 0. In this way, some redundancy is avoided in the training set. The learning process is stopped after six iterations, when convergence is noticed, i.e. when the number of false alarms remains roughly constant. This procedure helps in correcting problems arising in the original algorithm proposed in [8] where false alarms were grabbed regardless of the strength of the network answers. Finally, the controlled bootstrapping process added around $21,000$ non FR2 logo examples to the training set.

4 Logo Localization

Fig. 3. depicts the process of logo localization. In order to detect FR2 logo

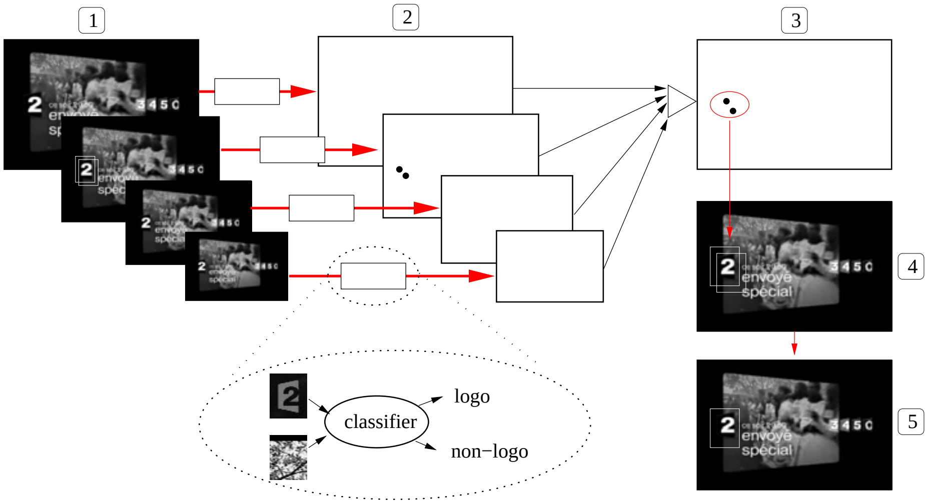

Fig. 3. Multi-scale logo localization

patterns of different sizes, the input image is repeatedly subsampled via a factor of 1.2, resulting in a pyramid of images.

As mentioned earlier, each image of the pyramid is entirely convolved at once by the network. For each image of the pyramid, an image containing the network results is obtained. Because of the successive convolutions and subsampling operations, this image is approximately four times smaller than the original one. This fast procedure may be seen as corresponding to the application of the network retina at every location of the input image with a step of four pixels in both axis directions, without computational redundancy.

After processing by this detection pipeline, logo candidates (pixels with positive values in the result image) in each scale are mapped back to the input image scale (step 3). They are then grouped according to their proximity in image and scale spaces. Each group of logo candidates is fused in a representative logo whose center and size are computed as the centroids of the centers and sizes of the grouped logos, weighted by their individual network responses. After applying this grouping algorithm, the set of remaining representative logo candidates serve as a basis for the next stage of the algorithm, in charge of fine logo localization and eventually false alarm dismissal.

To do so, a local search procedure is performed in an area around each logo candidate center in image scale-space (step 4). A reduced search space centered at the logo candidate position is defined in image scale-space for precise localization of the logo candidate. It corresponds to a small pyramid centered at the logo candidate center position covering ten equally distant scales varying from 0.8 to 1.5 times the scale of the logo candidate. For every scale, the presence of a logo is evaluated on a rescaled grid of 16×16 pixels around the corresponding logo candidate center position. We observed that true logos usually give a significant number of high positive responses in consecutive scales, which is not often the case for non logos. In order to discriminate true logos from false alarms, it resulted efficient to take into account both number and values of positive answers. We therefore consider the volume of positive answers (the sum of positive answer values) in the local pyramid in order to take the classification decision. Based on the experiments described in the next section, a logo candidate is classified as logo if its corresponding volume is greater than a given threshold *ThrVol* (step 5). The bottom-right image of Fig.3 shows the position and size of the detected logo after local search.

5 Experimental Results

We tested the trained logo detection system on two sets containing images extracted from TV programs. The first set consists of 800 images each containing one FR2 logo. The other consists of 257 images not containing any FR2 logo. Fig. 4 shows a ROC curve for the first set. This curve presents the detection rate as a function of the number of false alarms while varying the volume threshold *ThrVol*. One can clearly notice that, for a low number of false alarms, the detector attains a high detection rate. For example, if we allow 10 false alarms (for

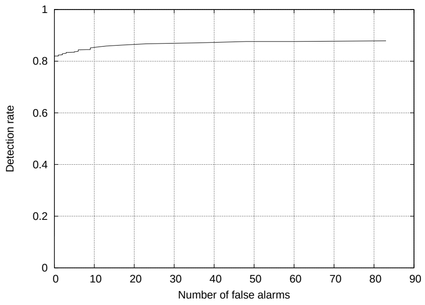

Fig. 4. Detection rate on the first test set versus the number of false alarms for varying volume threshold *ThrVol*.

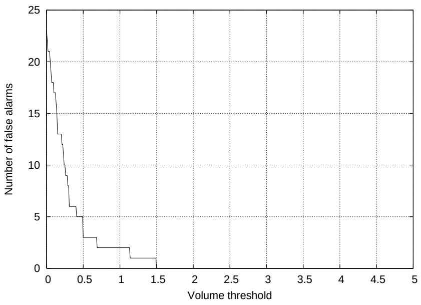

Fig. 5. Number of false alarms on the second test set versus the volume threshold *ThrVol*.

the 800 images) the system detects about 85% of the logos, which seems to be the maximal detection rate that can be reached on these test images, with the proposed architecture. An important point is that we obtain a good detection rate of 82% with no false alarm, for values of *ThrVol* above 1.5. Note that for 13 test images (≈ 1.6%), we judged the logo invisible to the human eye, but we still counted these examples as undetected. Fig. 6 shows some images with false alarms for a very low *ThrVol*.

In the second experiment, we applied the FR2 logo detector on the second test set that does not contain any image displaying the logo. The curve in Fig. 5 shows the number of false alarms as a function of the volume threshold *ThrVol*. One can notice that this number of false alarm decreases very quickly as *ThrVol* increases, and that no false alarm are produced for *ThrVol* above 1.5. For illustration purposes, Fig. 7 shows some images of the first test set with detected transparent logos. There are examples containing logos of very low contrast due to light background. Other examples show logos over a high contrasted non-uniform background which considerably falsifies the logo contours in the image region. There are also some examples of FR2 logos of different sizes at different positions in the image.

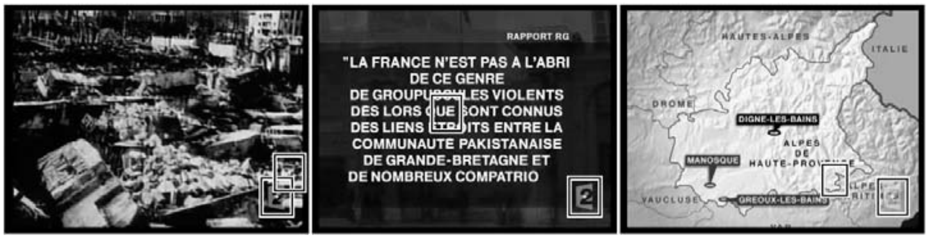

Fig. 6. Some results with false alarms.

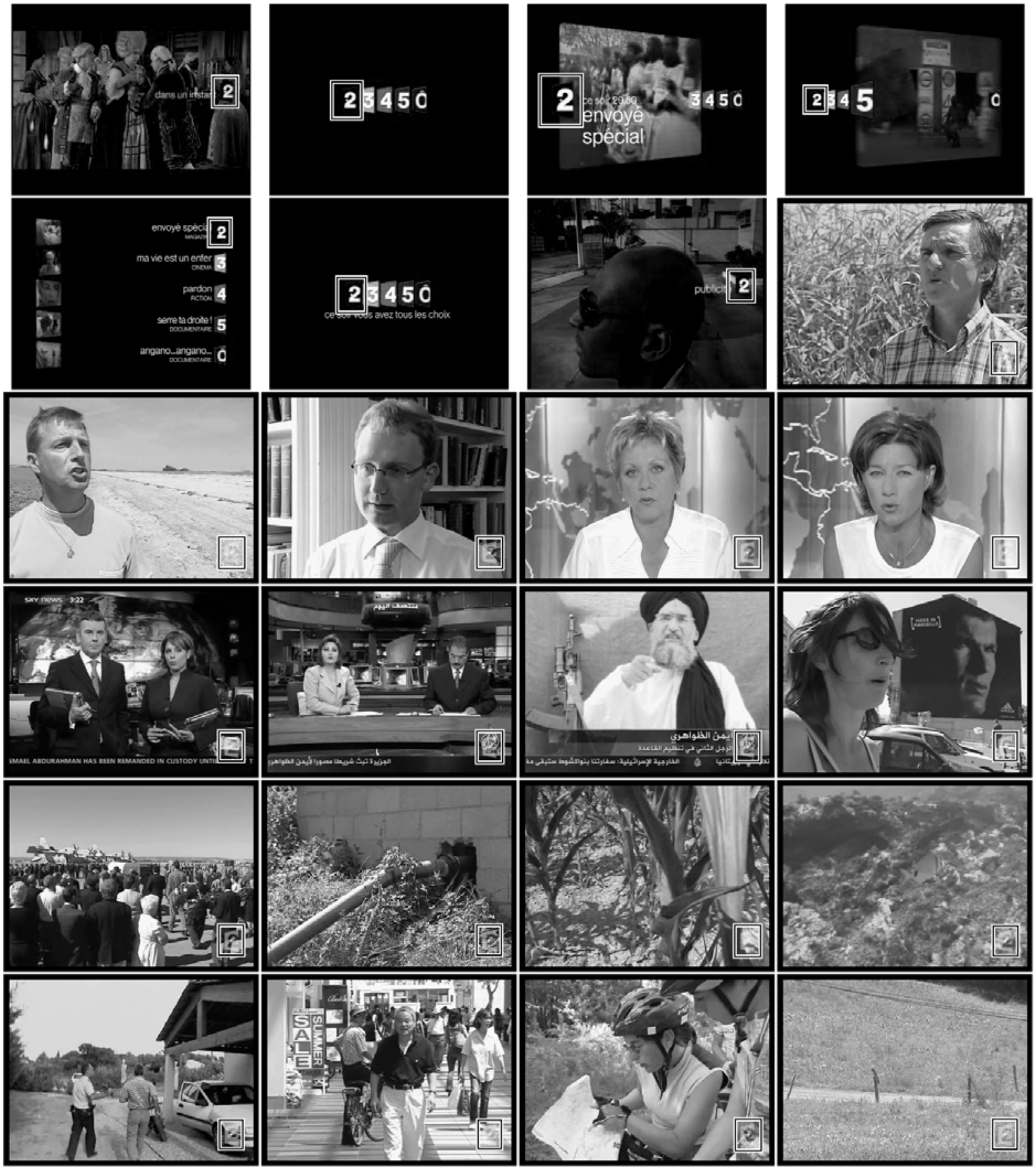

Fig. 7. Some results of logo detection on "France 2" TV programs.

6 Conclusion

Our experiments have shown that a multi-resolution scheme based on convolutional neural networks is very powerful for transparent logo detection. Indeed, this approach does not require any heuristic regarding image preprocessing, low level measures to extract or segmented shape analysis. The detection rate is very high even in cases where the transparent logo is very poorly contrasted because of the video background. Due to its convolutional nature and the use of a single network, our approach is very fast and can be easily embedded in real time on various platforms. Moreover, recent experimental results tend to show that multiple transparent logos can be handled through the use of a single light convolutional architecture. As an extension of this work, we are currently considering the detection of animated deformable transparent logos.

References

1. R.J.M. den Hollander and A. Hanjalic. Logo recognition in video stills by string matching. In *Proceedings of International Conference on Image Processing (ICIP)*, pages 517–520, 2003.
2. C. Garcia and M. Delakis. Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1408–1423, 2004.
3. Y. LeCun, L. Bottou, , Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
4. H. Pan, B. Li, and M. Ibrahim Sezan. Automatic detection of replay segments in broadcast sports programs by detection of logos in scene transitions. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2002.
5. H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.
6. H. Sanson. Video indexing: Myth and reality. In *Proceedings of International Workshop on Content-Based Multimedia Indexing*, 2005.
7. C.G.M. Snoek and M. Worring. A state-of-the-art review on multimodal video indexing. In *Proceedings of the 8th Annual Conference of the Advanced School for Computing and Imaging*, 2002.
8. K.K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51, 1998.
9. R. C. Veltkamp and M. Tanase. Content-based image retrieval systems: A survey. *IEEE Image Processing*, 1(1):100–148, 2001.
10. M. Visani, C. Garcia, and J.M. Jolion. Bilinear discriminant analysis for face recognition. In *Proceedings of International Conference on Advances in Pattern Recognition (ICAPR 2005)*, 2005.
11. K. Zyga, R. Price, and B. Williams. A generalized regression neural network for logo recognition. In *Proceedings of International Conference on Knowledge-Based Engineering Systems and Allied Technologies*, 2000.