Name and affiliation of the authors

Régis Vaillant	Laboratoire Central de Recherches Thomson-CSF Domaine de Corbeville 91404 Orsay Cedex France e-mail: `vaillant@thomson-lcr.fr`
Christophe Monrocq	Laboratoire Central de Recherches Thomson-CSF Domaine de Corbeville 91404 Orsay Cedex France e-mail `monrocq@thomson-lcr.fr`
Yann Le Cun	Room 4G-332 AT&T Bell Laboratories Crawford Corner Road Holmdel NJ 07733 USA e-mail: `yann@neural.att.com`

An Original Approach for the Localization of Objects in Images

Abstract

In this article, we present an original approach for the localization of objects in an image. Our approach is neuronal and it includes two steps. In the first step, a rough localization is performed by presenting each pixel with its neighbourhood to a neural net which is able to indicate if this pixel and its neighbourhood are the image of the search object. This first filter is not very discriminant for the position. From its result, we can select areas which might contain an image of the object. In the second step, these areas are presented to another neural net which can determine the exact position of the object in each area. We apply this algorithm to the problem of localizing faces in images.

1 Introduction

The detection and localization of faces in an image has many applications in various domains: surveillance, TV audience polling.... We propose a new method for this task which

- does not require any hypothesis on the position of the face in the image, or on its scale,
- does not require any hypothesis on the background.
- can be implemented to operate at a fraction of video rate (5 to 10 images per second) with current technology.

The main idea of our method is to train a neural network to detect the presence or absence of a face in its input window, and to scan this network over at all possible locations in the image. Because of the nature of the neural network architecture we used, this process can be done very efficiently without requiring to actually recompute the entire network state at each location. The scanning is performed on several versions of the image at various scales, resulting in an efficient, scale independent detector and locator.

Several approaches to this problem have been proposed in the literature. There are two main classes of methods:

- The first kind of approaches relies on the use of a synthetic model of a face. In [2], the authors represent a face as a combination of two parallel lines, which are the sides of the faces and two arcs of a circle for the chin and the top of the face. Yuille and al [10] suggest to represent each part of the face as a deformable element which is searched in the image by minimising an energy. Vincent and al [9] locates these different parts using neural nets. Craw and al [1] have a similar approach. This kind of techniques have the following difficulties: the computation time for adjusting the model could be long and the choice of the initial position of the model is quite difficult.

- The second kind of approaches relies on building a classifier which processes constant size images, and indicates if it corresponds to a face or not. Turk [7, 8] uses a principal component analysis. Neural nets are used in [5].

2 The Data Base

In order to detect some specific elements in an image, it is necessary to describe the primitives that must be detected in a way which is compatible with the used algorithm. One of the main advantages of some advanced neural net architectures is their ability to process raw (or almost raw) images. The problem of finding (and computing) the appropriate representation for the classifier is greatly facilitated. Our database is composed of many examples of small-size images of "faces" and "non-faces".

2.1 Formation of the data base: image acquisition

Twenty eight volunteers of both sexes, and various ages, were asked to walk towards a camera, starting from a distance of 5 meters from the camera, to a distance of about 3 meters from the camera. The subjects were asked to talk, and change facial expression, and head attitude, while walking. To make the problem simpler, we ask the subjects who wore glasses to take them off. Indeed the glasses reflect light and can introduce highlight in the images. Because of the varying distance of the subjects from the camera, the size of the observed faces had widely varying sizes (the ratio of the size between the different images of the sequence is 3). To take into account the variations in lighting conditions, we acquired two sequences of images: in the first one, there is only one light behind the camera, in the second one, there were also more diffuse lighting. A supplementary sequence, without faces, was acquired.

The images were smoothed with a zero-mean Laplacian filter. They were also normalized for the mean and the standard deviation. The mean of the pixels of each image is set to 0 and the standard deviation to 1.

2.2 Formation of the data base: extraction of patches

As we briefly mentioned in the introduction, the neural net is given a small window taken from the input image, and is asked to activate its output if a face is present in the window. The size of the window was chosen to be 20×20 pixels. We chose this size because it is close to the minimum resolution which allows unambiguous distinction between faces and non-faces. In [6], it is mentioned that a size 16×16 is the lower limit such the human can detect a face.

To handle the scale variation, three approaches are possible. The first one is to train the neural net to detect faces independent of their size in the window, the second one is two train a separate neural net for each range of size, and combine their outputs. The third approach, which is the one we used, is to use a single neural network, and scan it over several versions of the input image at various resolutions. The outputs from the network at various scales are then combined.

To create examples of 20×20 pixel images of faces and non-faces, we manually segmented the whole database by entering for each image, the point $m_1 = (x_1, y_1)$ between the eyes, and the point $m_2 = (x_2, y_2)$ at the center of the mouth. The second point gives information about the orientation and scale of the face. The area of the face in the original image was reduced to a patch of size 20×20 using appropriate scaling factors chosen among a discrete set of scaling factors. As we will see later the same scaling factors were used when the algorithm was applied to the whole images. We uses 7 different scaling factors. Consequently, the faces did not always fill completely the patch of size 20×20. This patch was included in a bigger patch of size 48×32 (figure 1).

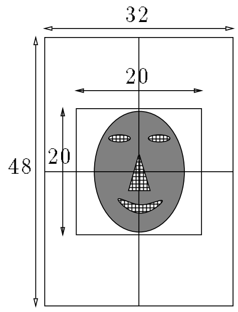


Figure 1: Geometry of the patches that have been extracted of the data base

The data base contains 1792 patches with a face. We have formed an equal number of images without faces, that we will call background patches, using the sequence of 32 images.

3 The training

3.1 General principle

Several neural net architectures were tried. The simplest one has no hidden layer, while the others have multiple convolutional hidden layers [4]. These networks were trained with the backpropagation algorithm taking into account the shared weights. The networks had the following points in common:

- an input layer of size 20×20. Each of the neurons of this layer is fed with the corresponding pixel of the patch.
- an output layer which contains only one neuron. This neuron indicates if the presented patch corresponds to a face or not.

3.1.1 A neural net without hidden layer

The neural net does not include any hidden layer. We use it for analyzing the complexity of our problem. It contains 401 weights.

3.1.2 A shared-weight neural net

This neural net comprises 3 hidden layers. Each of the hidden layer is divided into 4 small images (or feature maps). This net uses shared weights following the ideas described in [3, 4]. Figure 2 shows the architecture of the neural net. Each neuron of each map of the first hidden layer is connected to 5×5 neurons of the input layer. The weights are shared in the map. Each neuron of each map of the second hidden layer is connected to 2×2 neurons of the corresponding map of the first hidden layer. The weight are shared. The neuron of each map of the third hidden layer is connected to each neurons of the corresponding map of the second hidden layer. The neuron of the output layer is conected to the four neurons of the third hidden layer. This net has 1157 free parameters (but much more connections because of the weight sharing). There are many well-known advantages to using shared weight neural net architecture (fewer free parameters, better generalization, distortion invariance). In our context, shared-weight architectures have another decisive advantage. For our application, the network must be replicated (or scanned) over a large image (say 256 by 256 pixels). Now, since each layer of the network essentially performs a convolution (with a small-size kernel), a large part of the computation is in common between to networks applied at two neighboring locations. This redundancy can be eliminated by performing the convolution corresponding to each layer *on the entire image at once.* The overall computation amounts to a succession of convolutions and non-linear transformations over the entire image.

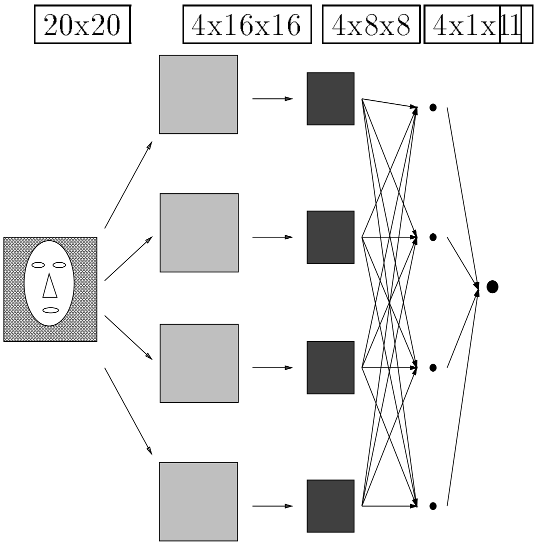


Figure 2: Architecture of the neural net

3.2 What must the neural net learn ?

Once, the architecture of the net is chosen, the question is: what must the neural net learn ? We propose 3 answers:

- Training with the goal of performing a complete localization: the elements of the database are presented to the neural net. If the presented patch corresponds to a perfectly centered face, the desired output is α, else the desired output is $-\alpha$. This is the natural choice which makes a direct use of the two classes of our problem.

- Training with the goal of performing a rough localization: The elements of the database are presented to the neural net as a perfectly centered patch or a shifted patch. This means that we feed the input layer of the neural net with an image extracted from the patch 48×32 whose size is 20×20 and whose center is placed at (x, y) pixels of the center of the patch of the database. If the image is perfectly centered, the desired output is α. It the image is shifted, the desired output is $\alpha(2e^{-\lambda\sqrt{x^2+y^2}} - 1)$. The desired output is an exponentially decreasing function of the shift. If the presented image is a background, the desired output is $-\alpha$. Our goal is to train the net to give a medium answer when it encounters a shifted face and to give a maximal answer when it encounters a perfectly centered face. So when the neural net will be applied to a complete image, the obtained answer will be smooth all around the face. The areas of the image which correspond to face will be easy to detect. The drawback is that the position of the center of the face will not be very precise.

- Training with the goal of performing a precise localization: The elements of the data base are presented perfectly centered with a desired output α or shifted with a desired output of $-\alpha$. The background images are not presented. This neural net must be able to localize precisely the center of the face if the input layer is fed with faces more or less centered.

We can also notice that the two last training techniques multiply the number of patterns in the data base. Indeed, $336 = (48-20) \times (32-20)$ different images are formed from each original image of the data base. Even if these images are not completly independent patterns, we can assume that the generalization rate is correctly estimated and we do not have overfitting.

3.3 Results of the training

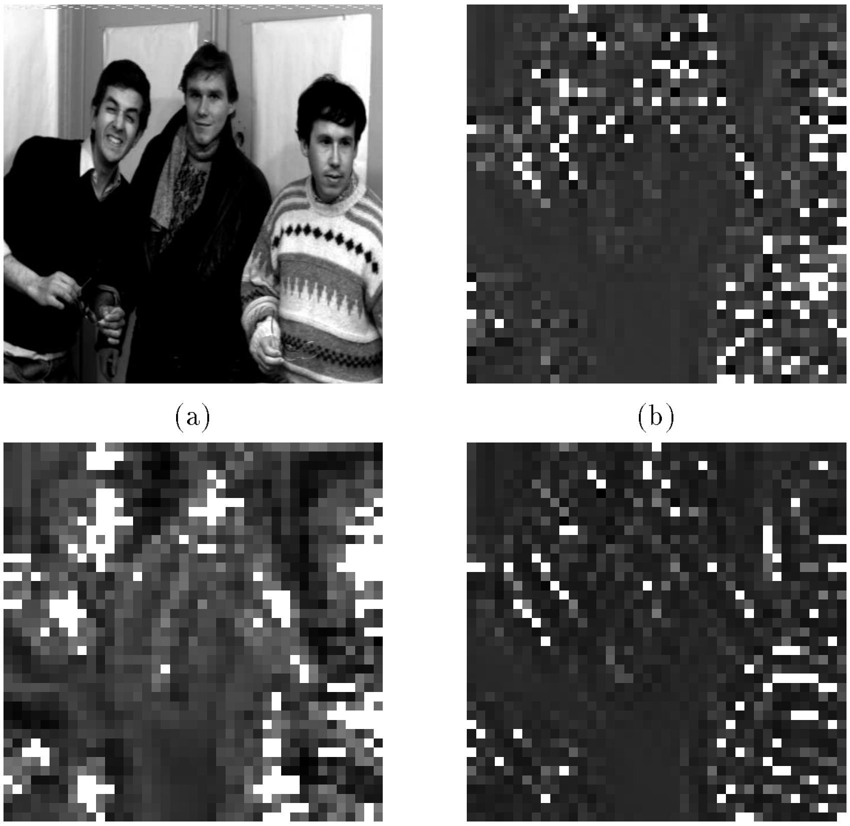
In a first step, we have tested the two neural nets which are described in the section 3.1 and the use the two first learning methods which are presented in the section 3.2. The learning set is formed with half of the database and the test set is formed with other part of the database. With a classical workstation (Sun4 SPARC), several hours are needed for some of the training sessions.

3.3.1 Training for a complete localization

Figure 4 shows the evolution of the quadratic error and the rate of well classified example. The quadratic error is defined as: $\frac{1}{2N}\sum_{n=1}^{N}(d^n - o^n)^2$. o^n is the obtained output when the example n is presented and d^n is the desired output. An example is assumed to be well classified if o^n and d^n have the same sign. These values are measured after a presentation of the whole training base to the neural net.

First, we can note that the two neural nets have results which are quite equivalent. On the test set, the quadratic error decreases quickly and the rate of correct recognition increases towards 96%.

These results could appear very satisfactory. In fact, they are not. Indeed, we plan to segment an image in areas that correspond to a face and in areas that do not correspond to a face. We apply the neural net with shared weight to a standard image whose size is 256×256. The output will be an image of size $126 \times 126 = 15876$. The size of the output image is different of the input as there is one hidden layer which subsamples its input, thereby dividing the size of the input by 2. If



(a) (b) (c) (d)

(a) The image.

(b) The neural net with shared weight trained for a complete localization.

(c) The neural net with shared weight trained for a rough localization.

(d) The neural net with shared weight trained for a precise localization.

Figure 3: Neural nets applied to an image

the rate of image which are well classified is 96%, the image will include 635 positive answers which will probably correspond to false alarms. This result cannot be exploited: there are too many false alarms.

As example, the figure 3(b) shows the obtained result when the shared weight nets is applied to the image of the figure 3(a). The grey-level of the pixels is proportional to the answer of the neural net. This image is scaled so that its resolution is 86×86. There are 95 pixels with a positive answer (5.6% of the points of the output image).

3.3.2 Training for a rough localization

Figure 5 shows the quadratic error and the rate of well classified examples in the case of a rough localization. We can note the following points:

- The first net does not succeed to learn. The quadratic error and the rate of well classified examples stops changing significantly after a few iterations.
- In the case of the second net, the error and the recognition rate decreases more slowly than in the case of complete localization.
- There is no overfitting. The generalisation rate does not decrease at the end of the training. In the case of the training for the complete localization, they decrease at the end of the training.

These various remarks indicate that this problem is more difficult than the previous one.

The generalisation rate obtained at the end of the training phase is lower than the rate we obtained in the case of the complete localization. Consequently, when we will apply our net to a complete image, it will produce a greater number of false alarms. On the other hand, the false alarms can be easily separated from the correct alarms. Indeed, when a face is present in the image, there is a complete area where the neural net gives a positive answer.

Figure 3(c) shows the obtained result when this net is applied to the image 3(a). There are 181 pixels with a positive answer. It is equivalent to the estimated generalization rate of 90%. It is important to note the distribution of positive answer. They are grouped in about 10 areas. Each of these areas could be considered as an hypothesis for the detection of a face and could be subject to further processing as explained later.

3.4 Training for a precise localization

Figure 3(d) shows the obtained results when the shared weight net is applied to the whole image. There are 79 pixels with a positive answer. They are scattered on the whole image. This is normal as the net has not been trained to give identical answer for a pixel and its neighbours.

4 Application to images

We wish to have an algorithm for the localization of faces in images which does not make any hypothesis on the scale of the face in the image. The system we described in the previous sections requires that the face is observed to a fixed size. We apply these results to an image where the size of observed faces is unknown by processing this image at several resolutions with the same network. The complete algorithm is:

- Several versions of the the original image are created at different scales (the set of scaling factors is determined in advance). The shared-weight neural nets trained for rough localization is scanned over each of the images. Figure 6 shows the output of the net for each scale.
- We look for "blobs" of positive values in the output maps produced by the network. Each of the blobs is considered as a good candidate (an hypothesis) for fine detection of faces (see figure 7).
- We apply the neural net trained for a precise localization to each hypothesis and we search for the one that gives the maximal answer. If it is larger than some threshold[1], the hypothesis is assumed to be valid and the point with maximal answer is taken at the center of the face.
- The different valid hypothesis which corresponds to a single face are grouped. Indeed, a single face can be detected at two different scales. It is quite frequent, as the used resolutions are not very different, and the faces in the database are not very precisely normalised to the same scale. To group the different valid hypothesis, we consider the area that they describe in the original image. If two hypothesis are conflicting, i.e. their corresponding areas intersect, we retain only the one corresponding to the highest answer. Figure 8 shows the set of the retained hypothesis. A rectangle is drawn, in the initial image, around the area associated to each hypothesis. The size of the rectangle is computed from the resolution at which the hypothesis was formed.

5 Conclusion

We have presented an algorithm for the detection of faces in images using shared-weight replicated neural networks. In a first step, a first neural net forms rough hypotheses about the position of faces. These hypotheses are then verified in the second step using a second neural network. We have also shown that the algorithm applies to images where the size of the faces is unknown a priori.

The computational time which is necessary for the complete processing of an image is reasonable. With a classical workstation (Sun4 SPARC) an image of size 256×256 is treated in 6 second (smoothing and normalization of the image included). It is interesting to note that this algorithm

[1]At the stage, some of the hypothesis may be removed

could be easily installed on a more specialized machine as the major part of the operations is based on convolutions with kernels of size 5×5 or 8×8. Of course the example of time given below have been obtained with an implementation using this property of the neural net. Using a net of 6 different machines, we are able to process one image each second and to present a "live" demonstration.

In this paper, we assume that the face are well oriented in the image. It is possible to eliminate this assumption by following an approach similar to the one used for the scale problem. A net is trained to be insensitive to the precise orientation of the face. The network is scanned over several versions the image rotated by various angles (say every 20 degrees).

This kind of segmentation algorithm can be applied to other problems where the objects to be detected cannot be characterized easily by its outline or by classical primitives in image processing: car detection, ... Very little problem-specific hand-crafting is necessary: constructing a database of positive and negative examples suffices.

References

[1] Ian Craw, David Tock, and Alan Bennett. Finding Face Features. In *Second European Conference on Computer Vision*, April 1992.

[2] Venu Govindaraju, Sargur N. Srihari, and David B. Sher. A Computational Model for Face Location. In *Third International Conference on Computer Vision*, 1990.

[3] Y. Le Cun. *Modèles Connexionnistes de l'Apprentissage.* PhD thesis, Université Pierre et Marie Curie, Paris, France, 1987.

[4] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Back-propagation applied to handwritten zipcode recognition. *Neural Computation*, 1(4), January 1990.

[5] J.L. Perry and J.M. Carney. Human Face Recognition Using a Multilayer Perceptron. In *Internationale Conference Neural Networks*, January 1990.

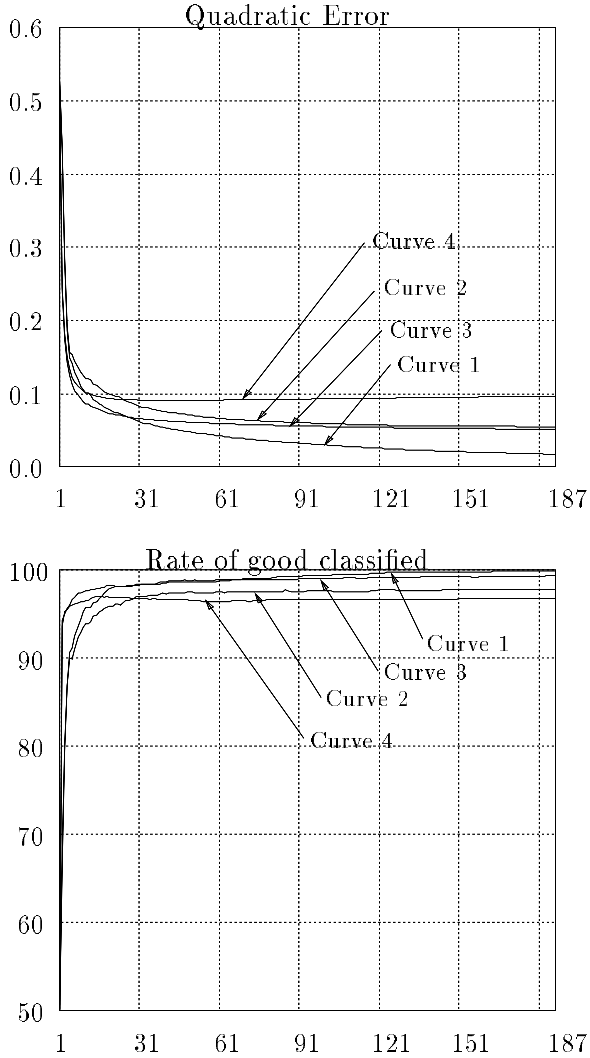
[6] Ashok Samal and Prasana Iyengar. Automatic recognition and analysis of human faces and facial expressions: a survey. *Pattern Recognition*, 25(1):65–77, 1992.

[7] Mathew A. Turk and Alex P. Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1):72–86, 1991.

[8] Matthew Alan Turk. *Interactive-Time Vision: Face Recognition as a Visual Behavior.* PhD thesis, MIT Artificial Intelligence Laboratory, September 1991.

[9] J.M. Vincent, J.B. Waite, and D.J. Myers. Precise location of facial features by hierarchical assembly of neural nets. In *Second Artificial Conference on Artificial Neural Networks*, pages 69–73, 1991.

[10] Alan L. Yuille, David S. Cohen, and Peter W. Hallinan. Feature extraction from faces using deformable templates. In *Computer Vision and Pattern Recognition*, 1989.



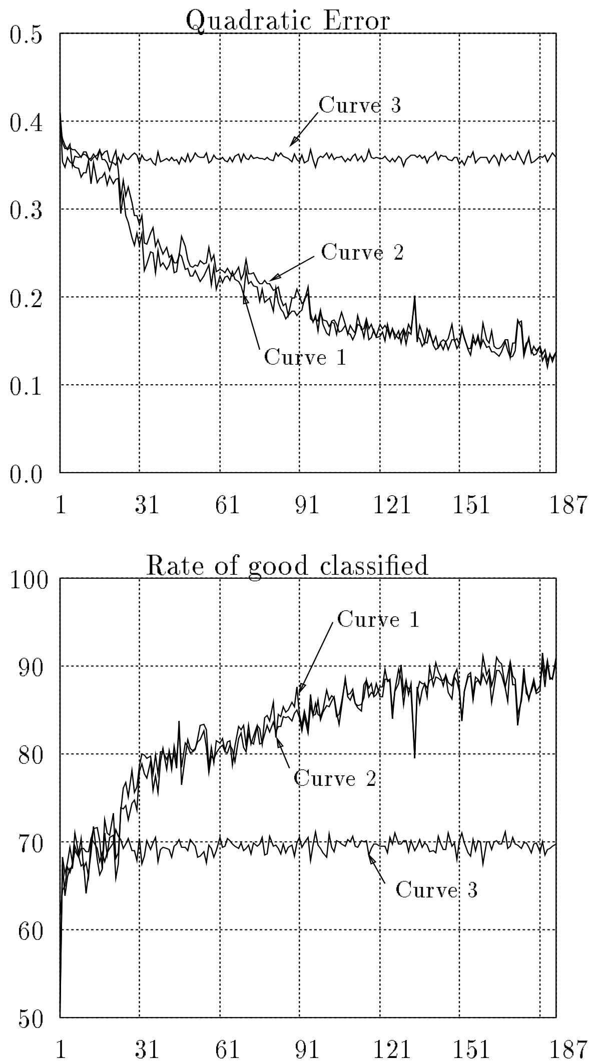
Curve 1 : Training with the shared weight neural net

Curve 2 : Generalization with the shared weight neural net

Curve 3 : Training with the completely connected neural net

Curve 4 : Generalization with the completely connected neural net

Figure 4: Training for a complete localization



Curve 1 : Training with the shared weight neural net

Curve 2 : Generalization with the shared weight neural net

Curve 3 : Training with the completely connected neural net

Figure 5: Training for a rough localization

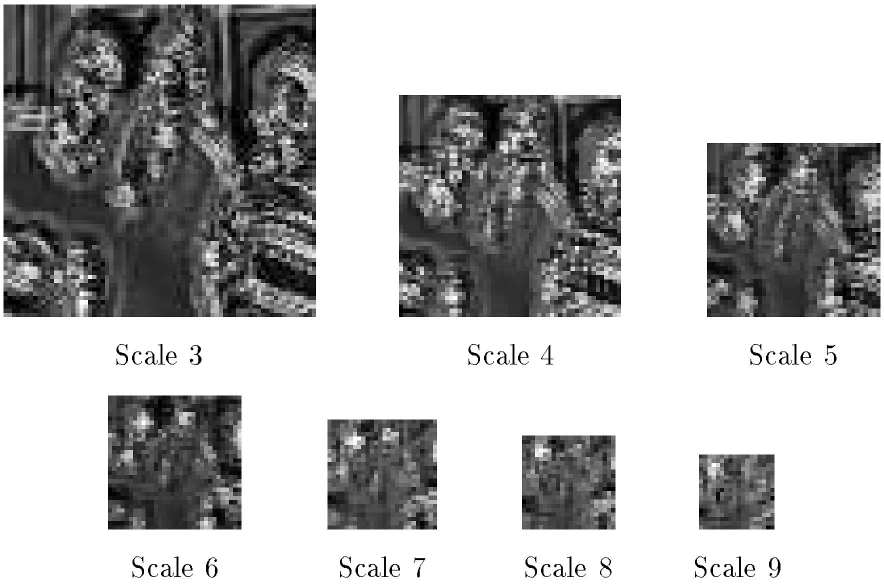


Figure 6: These images are the result of the shared weight neural net trained for a rough localization. The input image is treated with several resolutions. Hypothesis are formed in the area with a positive answer which are sufficiently large.

Figure 7: The hypotheses

Figure 8: Localization : the hypotheses that have been retained at the end of the processing.