Artificial Neural Network Based Face Detection

Project Final Report

Xunzhe Wen
Electrical and Computer Engineering
University of Ottawa
Ottawa, ON, Canada
xwen055@uottawa.ca

***Abstract*—This project aims to implement the functional task of detecting upright, frontal face in the images by using the artificial neural network. The face detection system we have implemented based on a previously-trained neural network. The objective image was firstly divided into lots of sub-windows, which would be pre-processed, and each of those window went through the trained neural network to make a judgment whether each sub-window contains a face; then the system would arbitrates among the sub-windows in order to converge the overlapping detected faces. A retinal connection to the neural networks would be biologically motivated and presented. We chose the different components of the training set to evaluate the development of the system.**

Keywords—Face detection; computer vision; artificial neural networks, machine learning.

I. INTRODUCTION

Face detection is a necessary first-step in face recognition systems, with the purpose of localizing and extracting the face region from the background. It also has several applications in areas such as content-based image retrieval, video coding, video conferencing, crowd surveillance, and intelligent human–computer interfaces.

The human face is a dynamic object and has a high degree of variability in its appearance, which makes face detection a difficult problem in computer vision. Neural networks have become a popular technique for pattern recognition problems, including face detection. Neural networks today are much more than just the simple multi-layer perceptron. The first advanced neural approach which reported results on a large, difficult dataset was by Rowley.

II. LITERATURE REVIEW

A. Retina Receptive Field

The receptive field of an individual sensory neuron is the particular area of the sensory space in which a stimulus will trigger the firing of that neuron. In the human visual system, receptive fields are volumes in visual space. Visual receptive fields were described in two dimensions, as circles, squares, or rectangles. In the case of binocular neurons in the visual cortex, receptive fields cannot extend to optical range. Instead, they are restricted to a certain interval of distance from where the eyes are concentrated.

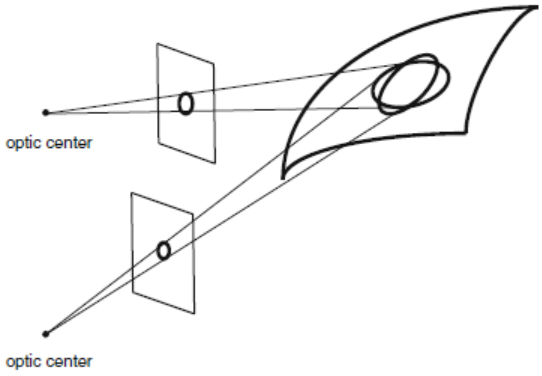


Figure1. consider a vision system that is restricted to using rotationally symmetric image operations over the spatial image domain only. If such a vision system observes the same three-dimensional object from two different views, then the back-projections of the receptive fields onto the surface of the object will in general correspond to different regions in physical space over which corresponding information will be weighed differently [1].

The receptive field is often identified as the region of the retina where the action of light alters the firing of the neuron. In retinal ganglion cells, this area of the retina would contain all the photoreceptors, all the cells of rods and cones from one eye that are connected to this particular ganglion cell via bipolar cells, horizontal cells, and amacrine cells. In binocular neurons in the visual cortex, it is necessary to specify the corresponding area in both retinas. Hubel and Wiesel advanced the theory that receptive fields of cells at one level of the visual system are formed from input by cells at a lower level of the visual system [2]. In this way, small, simple receptive fields could be combined to form large, complex receptive fields. Receptive fields have been mapped for all levels of the visual system from photoreceptors, to retinal ganglion cells, to lateral geniculate nucleus cells, to visual cortex cells, to extra cortical

cells. Studies based on perception do not give the full picture of the understanding of visual phenomena, so the electrophysiological tools must be used, as the retina is an outgrowth of the brain.

- *Computational Model of Biological Receptive Fields:*

  Regarding visual receptive fields in the lateral geniculate nucleus, DeAngelis et al. report that most neurons have approximately circular centre-surround group in the spatial domain and that most of the receptive fields are separable in space and time domain. There are two main classes of temporal responses for such cells: a "non-lagged cell" is defined as a cell for which the first temporal lobe is the largest one, a "lagged cell" is defined as a cell for which the second lobe dominates [3].

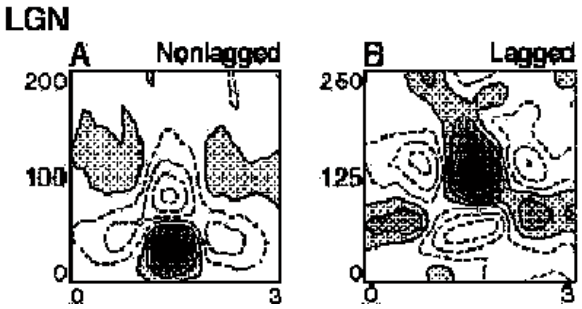


Figure 2. Examples of space–time separable receptive field profiles in the LGN as reported by DeAngelis and Anzai. There are two main categories of such cells; a for a non-lagged cell, the first temporal lobe dominates, while b for a lagged cell the second temporal lobe is strongest. In terms of the spatio-temporal receptive field model in the project, non-lagged cells can be modelled by first-order temporal derivatives, while the shape of lagged cells resembles second-order temporal derivatives (horizontal dimension: space x, vertical dimension: time t) [4].

Such temporal response properties are typical for first and second order temporal derivatives of a time-causal temporal scale space representation. The spatial response shows a high similarity to a Laplacian of a Gaussian, leading to an idealized receptive field model of the form. Within the above-mentioned spatio-temporal scale-space theory, it can approximate the qualitative shape of these circular center-surround receptive fields in the LGN with the following idealized model [1]:

$$h_{LGN}(x_1, x_2, t; s, \tau) = \pm(\partial_{x_1 x_1} + \partial_{x_2 x_2}) g(x_1, x_2; s) \partial_{t^n} h(t; \tau) \quad (1)$$

Where $\pm$ determines the polarity, which contains the on-center or off-surround with the off-center or on-surround; $\partial_{x_1 x_1} + \partial_{x_2 x_2}$ denotes the spatial and Laplacian operator; $g(x_1, x_2; s)$ denotes a rotationally symmetric spatial Gaussian; ∂_{t^n} denotes a temporal derivative operator with respect to a possibly self-similar transformation of time [5]; $h(t; \tau)$ is a temporal smoothing kernel over time corresponding to the time-causal smoothing kernel

- *Receptive fields in the context of neural networks:*

  The term receptive field is also used in the applications and researches of artificial neural networks, most often in relation to traditional artificial neural networks. When used in this field, the term adopts a meaning reminiscent of receptive fields in actual biological nervous systems. Convolutional neural networks have an unique and distinct architecture, designed to mimic the way in which real animal brains are understood to function; instead of having every neuron in each layer connect to all neurons in the next layer, the multilayer perceptron, the neurons are arranged in a 3- dimensional structure in such a way as to take into account the spatial relationships between different neurons with respect to the original data. Since convolutional neural networks are used primarily in the field of computer vision, the data that the neurons represent is typically an image; each input neuron represents one pixel from the original image.

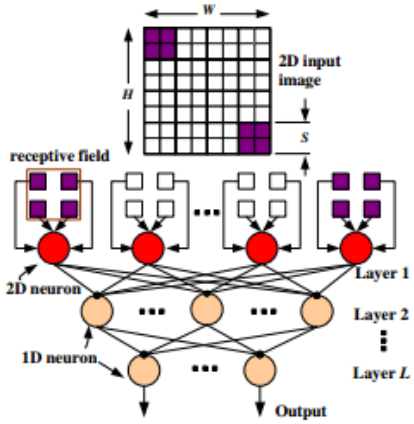


Figure 3. Receptive field neural network model.

The first layer of neurons is composed of all the input neurons; neurons in the next layer will receive connections from some of the input neurons, but not all, as would be the case in a multilayer perceptron and in other traditional neural networks. Hence, instead of having each neuron receive connections from all neurons in the previous layer, convolutional neural networks use a receptive field-like layout in which each neuron receives connections only from a subset of neurons in the previous and lower layer. The receptive field of a neuron in one of the lower layers encompasses only a small area of the image, while the receptive field of a neuron in subsequent and higher layers involves a combination of receptive fields from several neurons in the layer before.

In this way, each successive layer is capable of learning increasingly abstract features of the original

image. The use of receptive fields in this fashion is thought to give the convolutional neural networks an advantage in recognizing visual patterns when compared to other types of neural networks.

B. Artificial Neuron Network

Artificial neural networks have proven to be a powerful computational tool for many tasks: pattern classification, data clustering, function approximation, data compression, to name a few. The growing computation powers have supported new and complex network architectures for solving difficult cognitive tasks.

- *The availability of using ANN in face detection:*

  All of us human are able to aware if something is a face or not, A precise mathematical model may be built, but this kind of thinking is overly complicating the task more than is necessary. So let's treat the gray scale image of one face as a whole object. What we observe is that the distribution of intensity of pixels has some regular patterns, although maybe hard to be modeled as the explicit function, so we need to come up with an algorithm that detects these patterns. Furthermore, the face that has one eye closed, with the mouth open or shut is also a face. Thus, the algorithm should also be adaptive to different kinds of patterns, intuitively say, the algorithm should be able to generalize the principle behind the problem. And for ANN, which stands for artificial neural network, has exactly the property we want. ANNs gather their knowledge by detecting the patterns and relationships in data and learn (or to say: be trained) through experience, not from programming [6]. ANNs are bio-inspired by real neural net-work in human brains. The human brain is a powerful tool in pattern recognition. When we look at a tree, we know it is a tree because the neurons in the visual cortex of our brain have generated a similar input pattern on previous signal processing stages, and this input pattern can trigger the link between specific pattern to 'tree' at the conscious level. Our brain contains countless number of neurons which are interconnected at a great scale, thus we can learn and recognize an almost endless variety of input patterns. ANNs process the data in the same manner as human brain. Although with the rapid advance of ANN brought by Information theory or statistics, the structure of ANN is far different from neural networks in real animal. But they do share some common points: (1) Non-linear mapping: ANNs that sufficient number of hidden units can simulate any non-linear continuous function at any given accuracy. (2) The parallel processing: The information in ANNs is stored and processed in a parallel way, which gave it strong robustness and fast processing speed. (3) Adaptive learning: While in training, ANNs can extract regular patterns and memorize it within it's the connection weights. So to say, ANN has ability to generalize, and ANN can also learn online. (4) ANNs can deal with both quantified and logistic inputs, thus it can integrate traditional engineering techniques. (5) Multi variables: the input dataset and output dataset variables are arbitrary, ANN provided a viable and universal way to depict between single variable and multi-variable systems, without considering the decouple questions with each sub-system [7]. These properties altogether made it possible to detect human face with ANN.

- *ANN algorithm in the term Machine Learning*:

  So we know we can detect face with ANN now, but how is ANN built in a computer, with codes and data? The concept: Machine learning. Inductive machine learning is the process of learning a set of rules from instances (examples in a training set), or more generally speaking, creating a classifier that can be used to generalize from new instances. The process of applying supervised ML to a real-world problem [8]. In the term machine learning, there are many types of questions and an amount of practical algorithms. There are typically these four main category of problems: Regression, Classification, Clustering, Dimension Reduction .As for the algorithms to deal with them, based on whether the algorithms need to operate under the direction of human expert or previous knowledge, they can be divided into supervised learning and unsupervised learning. The family of ANN algorithms are probably the most famous supervised learning algorithm, here we use ANN for classification, although it can also solve other types of problems.

  The structure of the ANN is like it is a biological inspiration: a computational model formed by hundreds of artificial neurons, connected with coefficients (weights) by which neural structure is constituted. They are also known as processing elements (PE) as they process information [9]. Each PE has weighted inputs, transfer function and one output. In the concept of machine learning, the most significant thing of PE is an equation which balance inputs and outputs, different equations can be chosen based on different requirements, the sigmoid function is typically the most widely used one. ANNs are also called connectionist models as the connection weights represent the memory of the system. Although even one single neuron can accomplish a certain information processing procedure, the power of neural computations eventually comes from all the connections of neurons in a network. The supposed intelligence of artificial neural networks is a matter of argument. Artificial neural networks rarely have more than a few hundred or a few thousand processing elements, while the human brain has about 100 billion neurons.

  To consider the great difference of a modern ANN with a human brain in complexity, we know that the ANN is still far away from the creative capacity of the human brain. The human brain is of much more complexity and, unfortunately, the mechanism of how brain works are still not well known, people are still debating on how the consciousness and cognitive come into existence. Anyway, ANNs are capable of processing extensive numbers of data sets and to make

predictions that are sometimes surprisingly accurate. This will not give them intelligence in the common 'human' sense of the word, so the term computer intelligence may be a better way of describing these systems. There are many types of neural networks designed and new ones are invented every week but all can be described by the transfer functions of their neurons, by the learning rule, and by the connection formula.

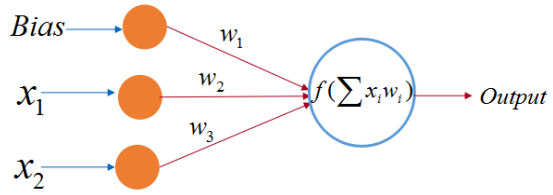


Figure 4. The model of one artificial neuron.

The artificial neuron unit is the basic functional part of the ANN which is designed to simulate the function of the biological neuron. The input signals along with the bias unit signal, also called inputs together, multiplied by the weights (adjusted) are first summed (combined) and then passed through a transfer function to produce the output for that neuron. The activation function is the weighed sum of the neuron's inputs and the most commonly used transfer function is the sigmoid function:

$$g(z) = \frac{1}{1+e^{-wx}} \tag{2}$$

Where w is the vector which indicates the weight of connectivity x will be the m-by-n matrix which represents the input vector, where m is the number of features, n is the number of inputs.

The goal in supervised learning is to predict one or more target values from one or more input variables [6]. Supervised learning is kind of a form of the regression algorithm that relies on samples of data: input and output from the training data. This kind of network structure is a system of fully connected neurons organized in these three layers, 1.input layer, 2.output layer,3.hidden layers(between 1and2,which is a black-box system after being settled, so called hidden layer).

The input layer neurons receive data from a data file. The output neurons provide ANN's response to the input data. Hidden neurons will communicate only with other neurons. They are to form the part of the large internal pattern that determines a hidden function to depict or solve the problem. Theory says that most functions can be approximated using a single hidden layer [10].

The information that goes from one PE to another will be contained within a set of weights. Some of the interconnections are strengthened and some are weakened, so that a neural network will output a more accurate answer. The most commonly used learning algorithm is back propagation algorithm. The error in prediction is fed backwards through the network to adjust the weights and minimize the error, thus preventing the same error from happening again. This process is continued with multiple training sets until the error is minimized across many sets. This results in the mapping of inputs to outputs via an abstract hidden layer.

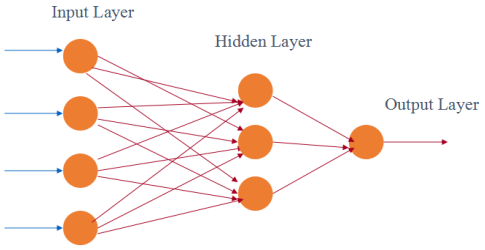


Figure 5. The model of layers in ANN.

How this process is being done is via the algorithm called gradient descent. The gradient descent is an optimization algorithm, also called steepest descent. It is the simplest and oldest method to solve unconstrained optimization problems. Concretely, the algorithm can be programmed into one line of code:

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(w_0,...w_n) \tag{3}$$

Where w_j is the weight of connectivity for a specific connection j, and α is the learning rate which controlled the speed of convergence of descent (If α is too large, algorithm may fail to converge).Function J here is called cost function, it computes the difference between predictions and the outputs in Euclidean space. Thus, by computing this equation in each iteration, the w_j will be closer and closer to its global optima. In a back-propagation neural network, this is how the connectivity value w_j adapts itself.

The number of neurons in the hidden layer also have an impact on the connections. During training procedure, the input data are used to adjust by the connection weights. Too few hidden neurons will hinder the learning process and too many will depress prediction abilities through over-fitting. As in the figure showed below, we can see if we under-fit the model, the prediction error will be high, and if we over-fit the model, the simulation of training will be especially good, but it loses the ability to generalize, thus the error

in validation set and the test set will be high. And in implementation, we can see the main problem we are facing is this over fitting problem.

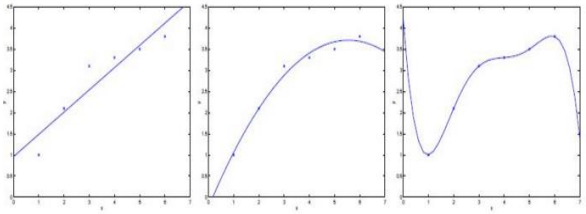


Figure 6. The example of under fit (left), right fit (median), over fit (right).

By increasing the number of the hidden neurons the ANN more closely follows the topology of the training data set. However, exceeding an optimum number results in tracing the training pattern too closely. When the ANN produces the desired output (i.e. is trained to a satisfactory level) the weighted links between the units are saved. These weights are then used as an analytical tool to predict results for a new set of input data. This is a recall or prediction phase when network works only by forward propagation of data and there is no backward propagation of error. The output of a forward propagation is the predicted model for the validation data. Pattern association is usually supervised learning. ANNs compete well with statistical methods in pattern recognition, especially when the systems contain high level of noise and variation. All above introduced the model and algorithm of ANN, with it, we can begin collecting data to train it for our face-detection purpose.

III. METHODS

The basic function of the entire system is a filter which receives 20 by 20 pixel region of the image as an input and generates an output ranging from 1 to -1, signifying the presence or absence of a face, namely. In order to detect faces anywhere in the objective image, the filter is applied at every location in the objective image. To further detect faces larger than the window size, the input image is repeatedly subsampled to decrease the size, and then the filter would be applied at each size of these subsampled images [11].

Therefore, the artificial neural network based face detection system would be introduced in the following three parts: The pre-processing stage was firstly applied on the inputs of the every single image; then the training progress would be introduced and discussed in terms of their performance. Finally the merging and arbitrating stage would be implemented in order to eliminate the overlapping detections.

A. Pre-processing

The actual image quality would be affected by the position of the photographer, the conditions of the external light source and so on. Therefore, some of the facial features would also be covered or shrunk to some degree. The preprocessing stage attempts to fix this gap to compensate and equalize the intensity values across the window.

- *Linear Fitting Process.*

  We fit a function which varies linearly across the window to the intensity values in the whole region inside the window. Pixels near the edge of the image could be considered as the background, so those intensity values should be ignored in the processing when the filter is going to compute the lighting variation across the face in the objective images. The linear function which used to fit the objective image would approximate the overall brightness of each part of the window and can be also subtracted from the window to make a compensation for a variety of lighting conditions.

  1). Initialize two vectors, which represents the sampled intensity values in the middle of the window image, in horizontal and vertical direction respectively;

  2). Coefficient undetermined linear functions would be applied to the two vectors respectively, so that the linear function would be represent the overall trend of the intensity values, and the fitting coefficients would be recorded;

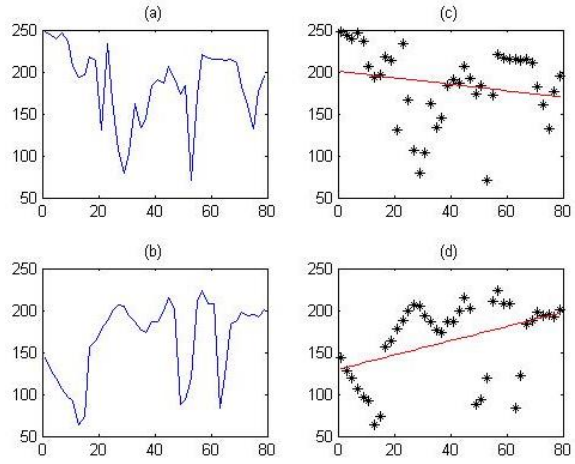


Figure 7. The result of the fitting function processing.(a) the intensity variation in horizontal; (b) the intensity variation in vertical; (c) and (d) the linear function fitting the sampled intensity values.

3). Generate a new correcting plate image, which intensity values in horizontal and vertical would be distributed as the fitting linear presents;

4). Converge the correcting plate image and the original window image to achieve the compensation.

- *Histogram Equalization*:

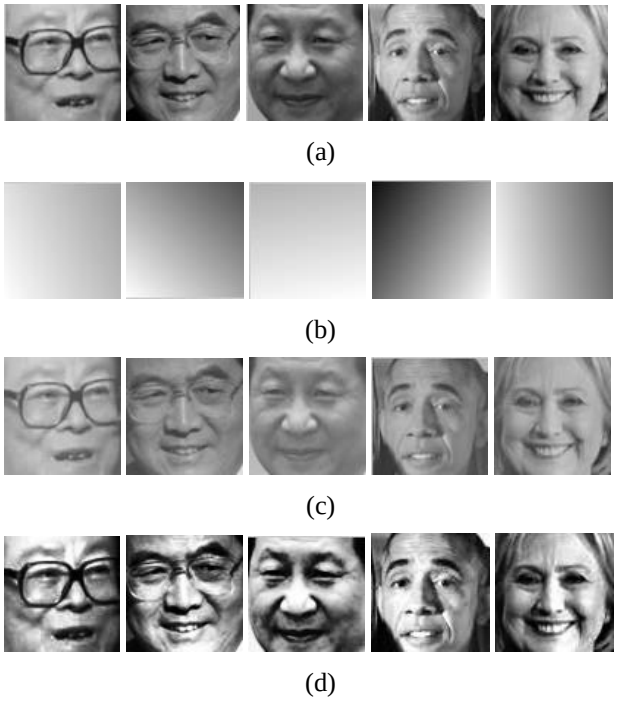
  Then, histogram equalization is performed, which nonlinearly maps the intensity values to expand the range of intensities in the window. The histogram is computed for pixels inside the region in the window. This compensates for differences in camera input gains, as well as used to enhance contrast. It is not necessary that contrast will always be increase in this.

  1). First we have to calculate the probability mass function of all the pixels in this image;

  2). The next step involves calculation of cumulative distributive function;

  3). The last step, in which we have to map the new gray level values into number of pixels, and map these new values you are onto histogram.

In order to ignore the background, which could influence the facial feature during the pre-processing, a background mask could be used to eliminate this kinds of factors. The examples of the pre-processing results are shown in figure below.



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

Figure 8. The steps in pre-processing a window region. First, a linear function is fit to the intensity values in the windows, and then subtracted out, correcting for some extreme lighting conditions. Then, histogram equalization is applied to the entire window. (a) Original window; (b) Best fit linear function; (c) Lighting corrected window; (d) Histogram equalized window.

B. *Training*

- *Gather training data:*

  First, the positive training data were downloaded from the database provided by university of Massachusetts (FDDB), then, each one of the positive training example was cropped manually out of the original image.

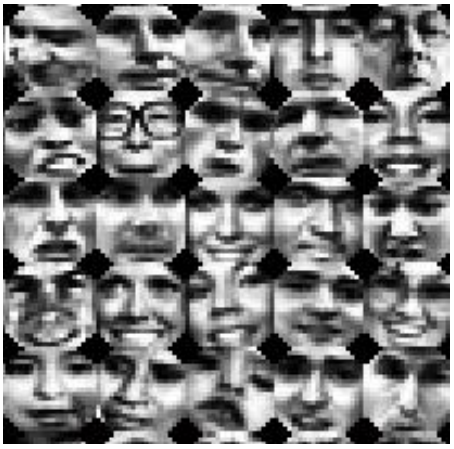


Fig.9. The sample Positive Training Data.

A total number of 700 images of face was cropped.(as in Fig.9). For the manual cropping was too exhausting, the face data set could be enlarge quickly by introducing rotation and mirroring [11]. And with mirroring, rotating by 15 degree in left & right direction respectively (as in Fig 10, the pre-processing step is followed right after), the positive training data was enlarged 6 times. Thus, a total number of 4200 positive examples were generated.

Fig.10. The sample Positive Training Data.

As for negative training examples, Practically any image can serve as a non-face example because the space of non-face images is much larger than the space of face images [11]. Based on this conclusion, the negative training examples could be generated by "slicing" the picture that have no faces in it at all, and each of the "slice" could serve as a negative example. To do the "slicing", a sliding window is applied, by going through all the image, multiple negative examples could be generated with 1 original image. And by doing so, a total number of 11114 non-face training examples were generated. In addition,500 pictures in which pixel values are randomly generated are also included as negative training examples in order to enhance it is robustness. With the method mentioned previously, so far 4200 positive training data that contained face are generated, and a total number of 11614 of negative training data that are non-face are gathered also (11114 of them are cropped automatically by sliding window,

500 of them are randomly generated noise-like image), so the total number of training data we got is 15814.

Data/Amount	Positive	Negative
Cropped	4200	11114
Random	N/A	500
Total	4200	11614

Table 1. The positive and negative training data

The ANN can now be trained as all the training data generated. Here the toolbox in Matlab2013b is used to simplify the process. In training, the toolbox would automatically break the input data set into three categories: Training set, Validation set and Test set. Training set is a set of examples used for learning, that is to fit the weights in ANN. Validation set is used to tune the architecture(numbers of hidden neurons) of ANN. Test set has no direct influence over ANN structure, it is a set of examples used only to assess the performance (generalization ability) of a fully specified ANN. Although the toolbox did most of the job in building and training ANN, what's the number of neurons that fits best for the task is vague. So in experiment, different numbers of hidden units were chosen (10, 30, 50, 70, 100 and 200) and performance were compared. The table and figure for the performance are listed below.

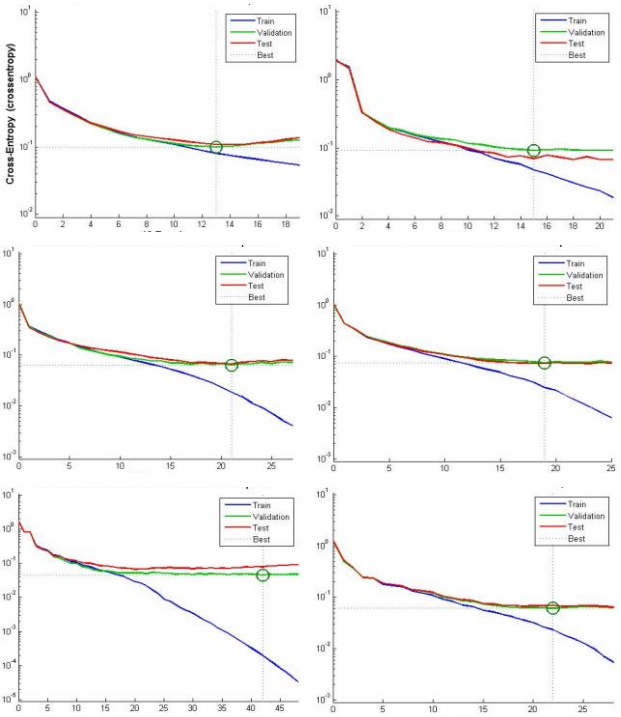


Fig.11 The entropy curve of performance (From top-down, left-right, the neuron are 10, 30, 50, 70, 100, 200).

From the entropy curve, it showed an over-fitting problem at every parameter we choose. However, the error for validation and test set are roughly the same, and they are all relatively small enough to be acceptable. In order to find a relatively better network, a customized validation set was built, the face images and non-face images that are different from the ones in training set. In this new validation set, there are 50 images with face, and 81 non-face images as negative examples. And an arbitration was made by evaluation with this new validation set.

No. of Neurons	Ideal	10	30	50	70	100	200
False-Negative (%)	0	6	8	4	8	4	8
False-Positive (%)	0	3.7	1.2	1.2	3.7	1.2	2.4
Accuracy (%)	100	95.4	96.2	97.7	94.7	97.7	95.4
F_1 score (%)	0	4.58	2.09	1.05	5.06	1.85	3.69

Table2.The statistical analysis of different ANNs

The False-Negative here means the algorithm misses a face, the False-Positive means the algorithm mistakes the background as a face. The accuracy is computed by computing the whole validation data divided by all the false detection. And the F_1 score .If chose an ideal number of neurons, the false detection is 0%, and an accuracy of 100% would be met. The F_1 score is a metric the measure the classifier. The number should be smaller the better. From the statistical analysis, the ANN with 50 hidden units, and 100 hidden units showed same performance, the reason may due to the insufficient number of images in validation data. Here we've chosen the neuron network with 100 hidden units. For it has relatively high accuracy and low F1 score. And it will generalize better when bigger data come in, when compared with ANN with 50 hidden units.

C. Arbitrating and Merging

After the previous steps, there are many false detections of faces and the overlapping detections of the same face. In this step, the arbitration could be used to decrease the narrow down the wrong detections of a single neural network, and then, using the morphology idea to eliminate the overlapping detection of the same face [11]. In this project, we develop a different method to achieve this processing.

- *Arbitration among the multiple networks:*

In order to reduce the number of false positives, we can apply multiple networks and arbitrate between their outputs to produce the final decision. Each network is trained in a similar manner, but with random initial weights, random initial nonface images, and permutations of the order of presentation of the scenery images. The detection and false-positive rates of the individual networks will be quite close. However, because of different training conditions, the networks will have different biases and will make different errors. We figured out a new method to achieve the arbitration step, which take the computation and the time consuption into consideration.

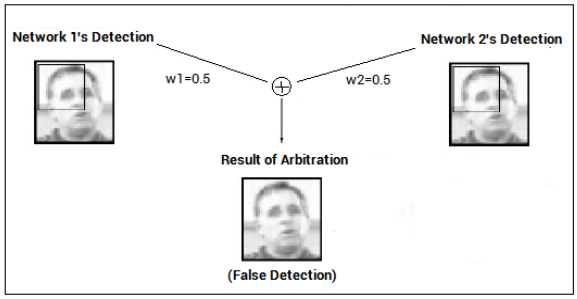


Fig.12. The weighted outputs from two networks over the same positions can improve detection accuracy.

1). We established a branch new neural network, trained by a set of data. The new database contains the positive face data in the previous training processing, and the new set of negative samples generated by applying the sliding on another non-face image;

2). Then, the two neural networks would make their predictions for the same window image respectively, and weights, which we chose 0.5 for each neural network, were assigned with their output;

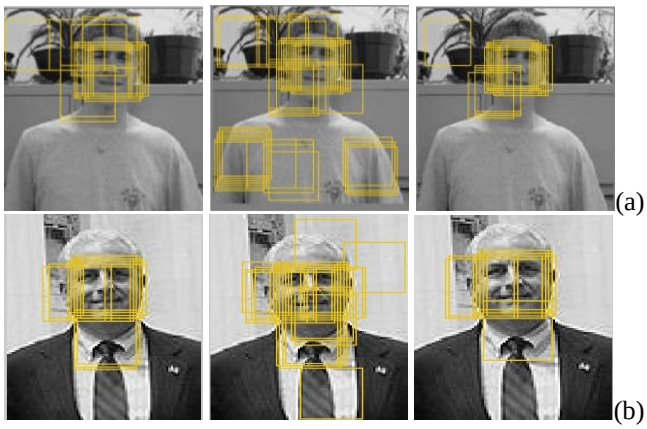


Fig.13. The weighted outputs (a) (b) the left were generated by Network 1, the median were generated by Network 2, the right were generated by the arbitration of Network 1 and Network 2.

3). The final decision would determine the presence of the face or not, and could eliminates false detections as well.

To test this hypothesis, we applied two separate neural networks to arbitrate among the their multiple detections. For a single image, the arbitration processing examines a small window with the sliding every single pixels. For each window, we counted the decision values of detections by weighting at each of scales, and chose 0.99 as the threshold of the prediction, which can result in an accurate prediction number for each window.

- *Merging Overlapping Detection:*

Most faces are detected at multiple nearby positions, while false detections often occur with less consistency. This observation leads to a merging method which can eliminate many overlapping detections. For each location, the number of detections within a specified neighborhood of that location can be counted. If the number is above a threshold, then that location is classified as a face. This project used the morphology to narrow down area of the overlapping area, and could further eliminate the false detections, which presents in some isolated pixels. To merge the overlapping detections, a distance based clustering of the presenting detection can be proposed. The same face detection could be varied around a dense area, therefore, the distance from the same corner or the centroid to a constant fixed pixels would be clustered into the same class. The up and left corner of the nearby detections defines the location of the detection result box, thereby in the experiments section, this distance differences will be processing referred to as "threshold". If a clustered particular location is significantly varied with respect to the other clustered detection location, which can indict that there is a new detection presence. Then the correctly identified as a face, and all other detection locations which overlap it, because they are likely to be errors and can therefore be eliminated.

1). According to the outputs of the arbitration neural networks, a mapping image, which presenting the high presence probabilities of the face detection (more than 98%), can be generated;

2). Morphology method, which using a fixed structure element, can play a role of close operation, the isolation of a few false detection could be eliminated, and the computation in the further merging step could be significantly reduced as well;

3). Create and initialize an vector, so that the vector can be used to store the distance from the top left corner of every single detection to the image's origin;

4). Find the distance clusters and locate the significantly changed pixels, final map the coordinates to the image, to accomplish the face detection.

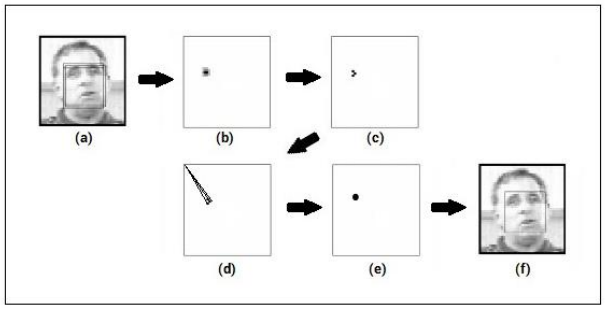
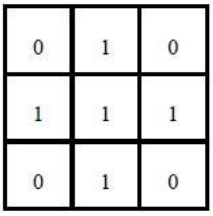
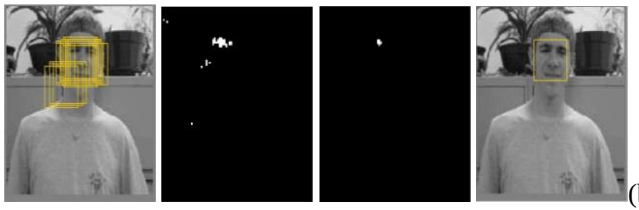


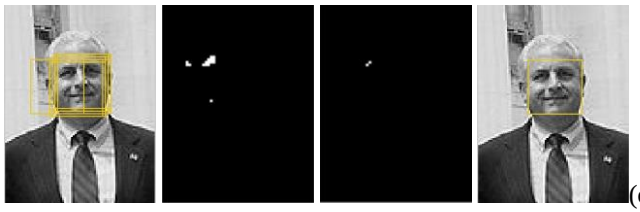
Fig.14. The diagram of merging. (a) The arbitrated image; (b) The mapping image with the high probabilities of presence of a face detection; (c) Mapping image after close operation; (d) Clustering in terms of the distance; (e) Overlapping eliminated mapping image. (f) Face detection image after merging processing step.



(a)



(i) (ii) (iii) (iv) (b)



(i) (ii) (iii) (iv) (c)

Fig.15. The results of the merging processing. (a) The structure elements used in the close operation implementation; In the (b) and (c), (i) The result from the previous arbitration step; (ii) The mapping image with the high probabilities of presence of a face detection; (iii) Mapping image after close operation; (iv) Overlapping eliminated image.

IV. EXPERIMENTAL RESULTS

All the functions and process steps mentioned above can be operated one after another sequentially, however, a graphical user interface (GUI) was built and serve as an integration of the whole programs. The important buttons are introduced as follows:

A. Crop Button

The Crop Button here enables users to manually crop a rectangle region in the input image that roughly matches the area with the face in the image. It is realized by the "getrect" Function in Matlab. The reason why this button is included here is because the unacceptably high computational cost, to sample an image with a small window, according to the size of the original image, the sample may go between 1000 and 20000times.Not to mention that the size of the face is not a constant, thus an image has to go under multiple times of down-sampling to be adjusted to the size of the face. In implementation, to sample a relatively small image, the computational cost is merely acceptable (about 1-2minutes.)If the input image become large (e.g. 1000*800 pixels), it would generate about 27000 sub-sampled images, and the total running time would go even more than 15 minutes. By introducing this crop button, because the size of the face is manually decided, the re-size ratio is also settled, thus, the input image is only down-sampled one time. And the time cost would be reduced greatly. (Although in the future, after we come up with a way to reduce the computational cost, we would cancel this function and do the face-detection in an automatic way.)

B. Face-Detect Button

The Face-Detect button enables the users to inspect the performance of the trained neuron network in an easy way. One can load the arbitrary image and crop any-part of the image, or some typical image which neuron network always make mistakes in classifying. How is it realized is by loading two trained neuron network: Both of them are with 100 hidden units, but one of them are trained with more negative examples. (So it is output may have higher precision but less recall, just to compensate the other neuron network)An arbitration part as introduced in the previous chapter is made, and the threshold at which we would accept it is a human face can also be adjusted (the value is chosen to be 0.98 in an empirical way).

C. Auto-FaceDetect Button

The auto face detect button passes a window through the down-sampled image, and the area which was determined as face would be recorded. And based on those region, the rectangles are plotted to highlight the face. The down-sample ratio is calculated based on the manual cropped image. Because the training data are all 20*20 images. The user cropped image would also to be considered as a 20*20 image. Thus the bigger value of real height and width of the user cropped image is compared to 20 and determine the ratio. We assume all the face in one image have similar size. Thus, by passing through a fixed-size sliding window, the face could be detected. As the face detected, the position would be recorded into a matrix, and in the same time, the rectangle command in Matlab plots the highlighted face area.

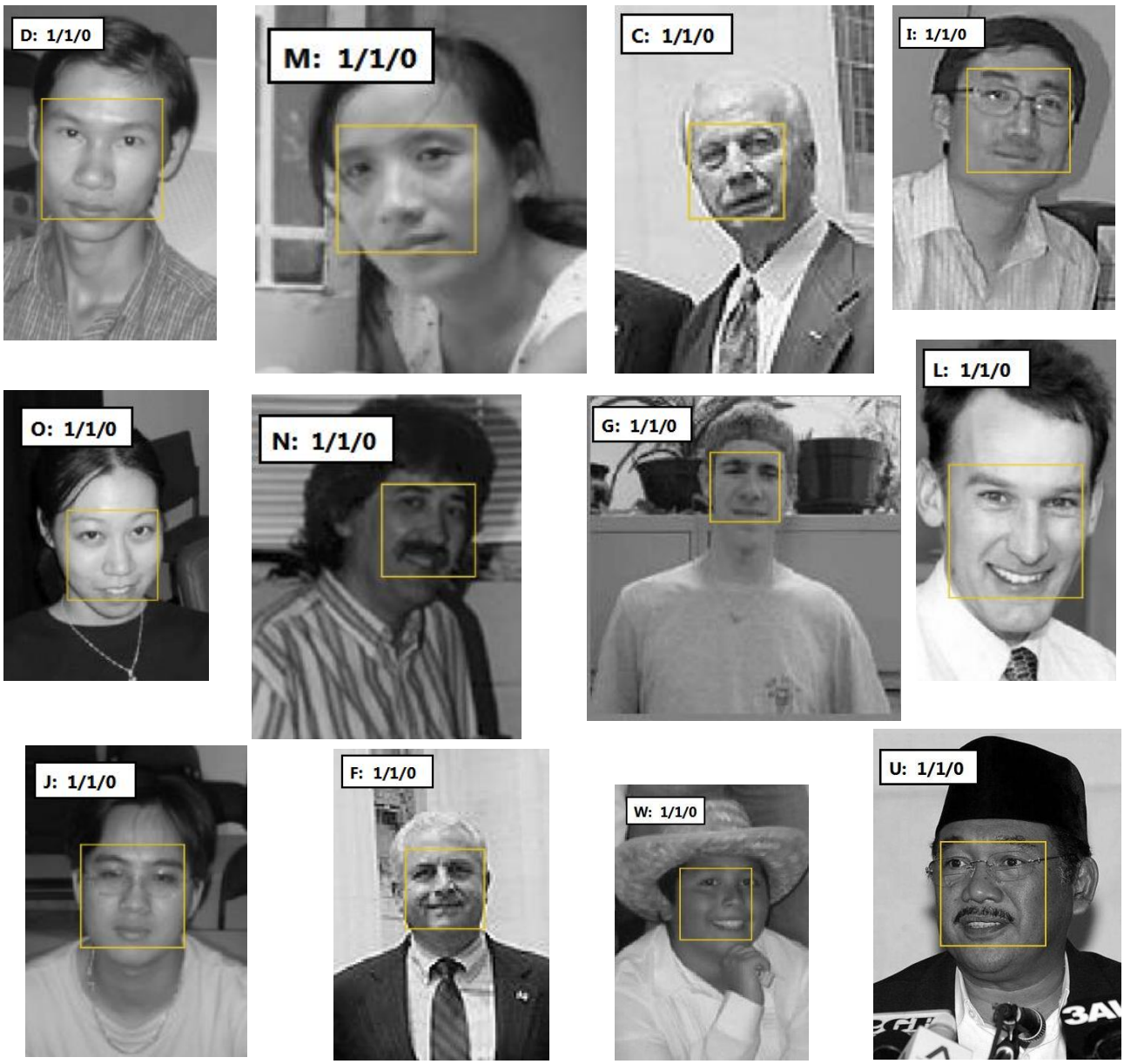


Fig.16 Output obtained from the system described above (arbitrating from two neural network). For the notation on the top-left corn, three numbers are shown: the number of faces in the image, the number of faces detected correctly, and the number of false detection. For portrait image that only has one or two faces, the accuracy is high, it seldom gives out false positive/negative results.

D. Merging Button

The merging button here merges all the rectangles belong to one face. The reason and the method are introduced in the previous chapter.

A standard process of using this GUI is described below: First, the user would load an test image (original image), then, the user should crop the rectangle area that are similar to face, after that, clicking "FaceDetect" button he could inspect the probability of face in that area, and he can decided if the cropped area is good or not. And the user could also click the "Auto-FaceDetect" button, the software would resize the image based on the ratio calculated in the crop step, then a small window would be passed over all portions of the image, and the detection would be made

Fig. 17. The faces are missed under these conditions: profile or large angle (in A).Group photo with multiple faces, this problem is caused by the merging step, (in B, P, Q, S), T was detected with a certain false detections, caused by the disadvantages of the merging processing, which we will discuss in the following chapters.

within each and every one of the sub-sampled window. The rectangle is plot in each iteration. After the Detection finishes, one can click the merge button to see a clear highlighted face image.

V. COMPARISON & DISCUSSION

Because face detection techniques requires a priori information of the face, they can be effectively organized into two broad categories distinguished by their different approach to utilizing face knowledge. The techniques in the first category make explicit use of face knowledge and follow the classical detection methodology in which low level features are derived prior to knowledge-based analysis.

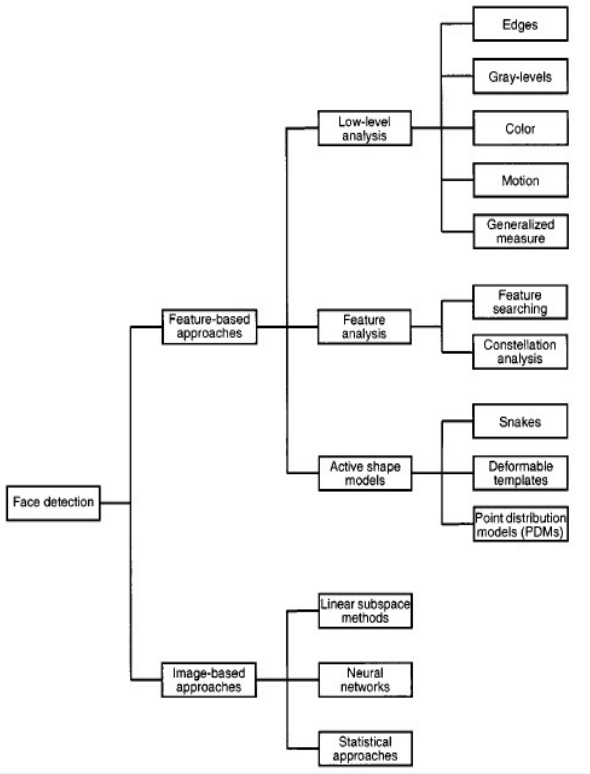


Fig.18. The face detection divided into approaches [13].

The apparent properties of the face such as skin color and face geometry are exploited at different system levels. Typically, in these techniques face detection tasks are accomplished by manipulating distance, angles, and area measurements of the visual features derived from the scene. Since features are the main ingredients, these techniques are termed the feature-based approach. Taking advantage of the current advances in pattern recognition theory, the techniques in the second group address face detection as a general recognition problem. Image-based representations of faces, for example in 2D intensity arrays, are directly classified into a face group using training algorithms without feature derivation and analysis. Unlike the feature-based approach, these relatively new techniques incorporate face knowledge implicitly into the system through mapping and training schemes [12].

Given a typical face detection problem in locating a face in a cluttered scene, low-level analysis first deals with the segmentation of visual features using pixel properties such as gray-scale and color. Because of the low-level nature, features generated from this analysis are ambiguous. In feature analysis, visual features are organized into a more global concept of face and facial features using information of face geometry.

Most of the image-based approaches apply a window scanning technique for detecting faces. The window scanning algorithm is in essence just an exhaustive search of the input image for possible face locations at all scales, but there are variations in the implementation of this algorithm for almost all the image-based systems. Typically, the size of the scanning window, the subsampling rate, the step size, and the number of iterations vary depending on the method proposed and the need for a computationally efficient system.

VI. CONCLUSION&FUTURE WORK

A. Conclusion

From the final result and comparison with other system, many advantages and disadvantages of this face detection system had emerged. First, for this system is trained based on the intensity distribution of face, it gave the system significant robustness in classifying face: gray-scaled face, cartoon face, etc. As showed in the result images, when classifying portrait pictures, this system showed excellent performance, it seldom misses faces, even if so, it was introduce by the flaw of later steps (e.g. the flaw in erosion and merging). The disadvantages are also obvious, first and most severe one is that the computational cost is incredibly high. In implementation, to process one signal image would cost 3-15 minutes. When the face is small (compared to the input image), in order to search and sample every portion of image, the sliding window operation could repeat for over 20000 times. Apparently it is unacceptable, this only way to reduce the computational cost is to set a gap parameter to let the sliding window "jump" instead of sampling pixel by pixel. But it is a trade-off between time and accuracy, for the window now could "jump" over the face area, and increase the false negative rate, for now, there are no sound way to solve this problem. Another problem is within the neural network itself, for the pattern of face is relatively obvious, but the feature of non-face image is hard to generalize. When the input images has symmetry structure, face-like intensity distribution caused by lighting condition or other factor, the neural network also accept them as a face, which could be hard to eliminate. In conclusion, the neural network face detection system implemented here so far has high accuracy in detecting faces in portrait image, and the input image has no constrains. The drawbacks are that the computational cost is too high.

Face detection system	*CMU-130*	*CMU-125*	*MIT-23*	*MIT-20*
Schneiderman & Kanade E^a [14]		94.4%/65		
Schneiderman & Kanade W^b [14]		90.2%/110		
Yang *et al.* –FA [15]		92.3%/82		89.4%/3
Yang *et al.* –LDA [15]		93.6%/74		91.5%/1
Roth *et al.* [16]		94.8%/78		94.1%/3
Rowley *et al.* [11]	86.2%/23		84.5%/8	
Feraud *et al.* [17]	86%/8			
Colmenarez & Huang [18]	93.9%/8122			
Sung & Poggio [19]			79.9%/5	
Lew & Huijsmans [20]			94.1%/64	
Osuna *et al.* [21]			74.2%/20	
Lin *et al.* [22]			72.3%/6	
Gu and Li [23]			87.1%/0	

Table.3. The performance of several popular used face detection algorithms (with their database) [13].

Figure 19. The face detection system integrated in the GUI interface.

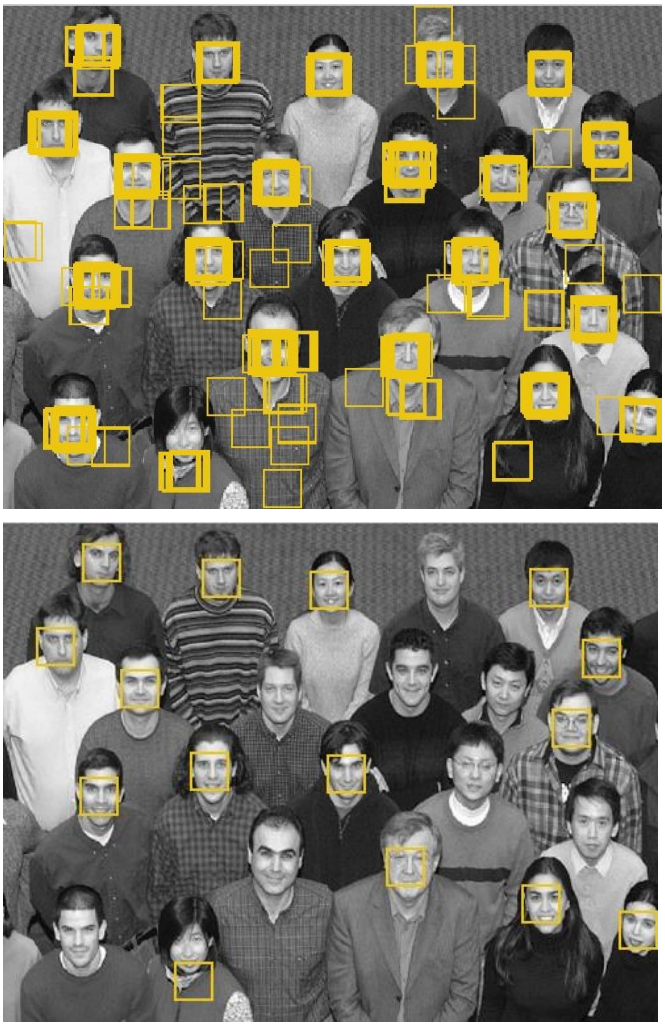


Fig.19. The missing face reveal the disadvantages of this system.

The training processing step was limited by the insufficient datasets, therefore, a number of windows with a similar facial intensity distribution would be detected as the face; the merge processing step using a close operation, but the improvements can be the structure elements, which would filter some isolated false detections, however, the true faces can be ignored by this processing as well. A distance based merging method then applied, when it comes into the cases that the faces are arranged along the concentric circles, these faces would be merged into only one single detections, with other faces lost in this processing. Those aspects we discussed could significantly affected the performance of the system, especially for a large amount of faces integrated into a single image. Therefore there can be improvements in the further implementation.

B. Future work

A great proportion of false positive/negative was caused by the neural network itself. By collecting more data and use the bootstrap algorithm introduce in Rowley's paper, the performance of ANN can be better without doubt. Then, a better algorithm such as hierarchical clustering could be applied for a better merging performance. Also, as we see in other papers that there is a way to make the input face image rotation in-invariant and translation invariant, with this feature, we can set a bigger gap value for sliding window to reduce the computational cost greatly. At last, we would get the program to automatically down-sample the image and detect all the faces in input images, better in a real-time manner, in order to fully complete the function of this face-detection software.

REFERENCES

[1] Lindeberg, Tony. "A computational theory of visual receptive fields." Biological cybernetics 107.6 (2013): 589-635.

[2] Hubel, David H., and Torsten N. Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex." The Journal of physiology 160.1 (1962): 106-154.

[3] DeAngelis GC, Ohzawa I, Freeman RD (1995) Receptive field dynamics in the central visual pathways. Trends Neurosci 18(10):451–457.

[4] DeAngelis, G. C., and A. Anzai. "A modern view of the classical receptive field: Linear and non-linear spatio-temporal processing by V1 neurons." The visual neurosciences 1 (2004): 704-719.

[5] Lindeberg T (2011) Generalized Gaussian scale-space axiomatics comprising linear scale-space, affine scale-space and spatio-temporal scale-space. J Math Imaging Vis 40(1):36–81.

[6] G. Agatonovic-Kustrin, S., and R. Beresford. "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research."Journal of pharmaceutical and biomedical analysis 22.5 (2000): 717-727.

[7] Sivanandam, Manu. Introduction to artificial neural networks. vikas publishing House PVT LTD, 2009.

[8] Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." (2007): 3-24.

[9] J. Zupan, J. Gasteiger, Anal. Chim. Acta 248 (1992) 1–30.

[10] B.D. Ripley, Pattern Recognition and Neural Networks, Cambridge University Press, Cambridge, 1996.

[11] Rowley, Henry A., Shumeet Baluja, and Takeo Kanade. "Neural network-based face detection." Pattern Analysis and Machine Intelligence, IEEE Transactions on 20.1 (1998): 23-38.

[12] E. Viennet and F. Fogelman Souli ́e, Connectionist methods for human face processing, in Face Recognition: From Theory to Application. Springer-Verlag, Berlin/New York, 1998.

[13] E. Hjelmas and B.K. Low, "Face Detection: A Survey," Computer Vision and Image Understanding, vol. 83, pp. 236-274, 2001.

[14] H. Schneiderman and T. Kanade, A statistical model for 3D object detection applied to faces and cars, in IEEE Conference on Computer Vision and Pattern Recognition, 2000.

[15] M.-H. Yang, N. Ahuja, and D. Kriegman, Face detection using mixtures of linear subspaces, in Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 2000.

[16] D. Roth, M.-H. Yang, and N. Ahuja, A SNoW-based face detector, in Advances in Neural Information Processing Systems 12 (NIPS 12), MIT Press, Cambridge, MA, 2000.

[17] R. Feraud, O. Bernier, J.-E. Viallet, and M. Collobert, A fast and accurate face detector for indexation of face images, in Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 2000.

[18] A. J. Colmenarez and T. S. Huang, Face detection with information-based maximum discrimination, in IEEE Proc. of Int. Conf. on Computer Vision and Pattern Recognition, 6 1997.

[19] K.-K. Sung and T. Poggio, Example-based learning for view-based human face detection, IEEE Trans. Pattern Anal. Mach. Intelligence 20, 1998, 39–51.

[20] M. S. Lew and N. Huijsmans, Information theory and face detection, in Proc. of International Conference on Pattern Recognition, 1996.

[21] E. Osuna, R. Freund, and F. Girosi, Training support vector machines: An application to face detection, in IEEE Proc. of Int. Conf. on Computer Vision and Pattern Recognition, 6, 1997.

[22] S.-H. Lin, S.-Y. Kung, and L.-J. Lin, Face recognition/detection by probabilistic decision-based neural network, IEEE Trans. Neural Networks 8, 1997, 114–132.

[23] Q. Gu and S. Z. Li, Combining feature optimization into neural network based face detection, in Proceedings of the 15th International Conference on Pattern Recognition, 2000, Vol. II, p. 4A.