Face Detection Based on Fuzzy Cascade Classifier with Scale-invariant Features

Yafeng Deng, and Guangda Su

Department of Electronic Engineering,
Tsinghua University, Beijing, China 100084

dyf02@mails.tsinghua.edu.cn, sugd@ee.tsinghua.edu.cn

Abstract

Viola et al. have introduced a rapid object detection framework based on a boosted cascade of simple feature classifiers. In this paper we extend their work and achieve two contributions. Firstly, we propose a novel feature definition and introduce a feature shape mask to represent it. The defined features are scale-invariant which means the features can be rescaled easily and reduce the performance degradation introduced by rounding. The feature shape mask can be computed efficiently and expanded conveniently, which can simulate feature shapes used by others and thus enriches the haar-like feature pool. Secondly, we present an improved cascade-structured classifier which is called fuzzy cascade classifier. The cascade-structured classifier owns the disadvantage of neglecting confidence of the prior stage classifiers while only using the binary output of prior stages. Motivated by fuzzy theory, we expand the output of each stage to three states: face, non-face, and potential face and set probability being face to each candidate window to make full use of the information of prior stages. Merged by voting, we improve the hit rate at similar false alarm rate.
Keyword: Face detection, AdaBoost, Scale-invariant feature, Fuzzy cascade classifier

I Introduction

Face detection is the first step of any face processing system. The accuracy and computation complexity are both the most important performance of a face classifier. Viola et al. [1] have proposed a framework for robust and extremely rapid object detection, which achieved an equivalent accuracy of the state-of-the-art results [2] while being distinguished from others in its ability to detect face extremely rapidly. Recently, many extensions and improvements have been made by expanding features [3], [4], [5] and polishing up features selection procedure [4].

There are many attributions of the high performance of Viola et al.'s method. To improve the accuracy, they used not raw pixel values but simple features to reduce the in-class variability and increase out-of-class variability. Learning with AdaBoost, thousands of important features are selected and combined to construct classifier with a good classification performance. To speed up, a new image representation called integral image is proposed to allow fast feature evaluation, and with a cascade classifier structure, most of the false candidate windows are rejected by simple classifiers, while only a little amount of promising windows are processed with complex computation to reject negative samples similar to positive.

Based on Viola et al.'s framework, our work extends theirs in such two following ways:

Firstly, we propose a novel definition of features, and the novel defined features are scale-invariant meaning that the feature can be computed in same manner when features are rescaled to detect different size of object and because the features are normalized by acreage, the performance degradation introduced by rounding rescaled coordinates to nearest integer position is reduced. In addition, the features are convenient to correct illumination and need not to deal with intensity average. Further more, we expand the feature set with a shape mask, which is very flexible and controllable, with which, we can select and design more features conveniently. The shape mask can simulate the features used by [1], [3], [5] and enriches the haar-like feature pool very much because there is 3^9 of different shape masks!

Secondly, we rectify the cascade structure proposed by Viola et al. [1] to build an improved cascade-structured classifier which improves hit rate at same false alarm rate. The cascade-structured classifier in [1] owns the disadvantage of omitting information of the prior stages. In [9], B. Wu et al. have proposed a nesting-structured cascade to enhance cascade-structured classifier in [1]. To make full use of prior stages information, we set the output of every stage classifier to three states: face, non-face, and potential face and set possibility being face to all potential face detections according to the number of stages they passed and the confidence of the final rejecting stage classifier. At the last several stages, if a candidate window is rejected by the current stage, we do not reject it as non-face, while setting them as potential faces and processing them in the merging procedure. In merging procedure, we merge neighboring potential detections to create valid faces by voting. If sum of the possibility of potential face detections at same position with similar size is larger than a threshold, we output a valid face detection. If the sum of probability is smaller than threshold, we reject potential windows as non-face. Experimental results show that the approach improves the hit rate at same false alarm ratio. To be convenient, we call the cascade structure proposed in [1] as original cascade structure, and call our novel improved cascade structure with fuzzy outputs as fuzzy cascade structure.

The remainder of the paper is organized as follows: Section II defines the novel features with introduction of the advantages of the rectified definition and presents the convenient shape mask. The fuzzy cascade structure is introduced in section III. Section IV provides the experimental results and conclusions are given in section V.

II. Features

Rather than using pixel intensity directly, most objects classifiers use features to present object. Features can make classification easier by reducing the in-class variability while increasing the out-of-class variability than raw pixel intensity [3]. Further more, features is more expressive to represent objects with similar computational complexity. Haar-like features are used by Papageorgou et al. [6] to detect pedestrians and faces, and then Viola et al. have developed a fast and robust face detection approach with three kinds of rectangle features. Some other previous works [3], [4], [5] have used similar features, and for convenience, we call them haar-like features.

All of the above haar-like features can be looked upon as being composed of several rectangle features and the feature value is defined as the intensities difference of the neighboring regions. Unlike the above works, we rectified the feature definition and designed a feature shape mask to build and represent different features, with which we can build tremendous features according

different classification work. The features which are also combined with rectangle can be calculated efficiently and easier to expand.

A. Feature definition

Unlike Viola et al. [1], we define the features as the average intensities difference between several positive and negative neighboring rectangles. We suppose that there are two rectangle sets R^p and R^n, where $R^p=\{r_1^p, r_2^p, ...r_{m_p}^p\}$ is called positive rectangles set because their signs are positive, and $R^n=\{r_1^n, r_2^n, ...r_{m_n}^n\}$ is called negative rectangles set because theirs are negative. Our features are defined in the form:

$$f(R^p, R^n) = \frac{\sum_i sum(r_i^p)}{\sum_i w_i^p h_i^p} - \frac{\sum_j sum(r_j^n)}{\sum_j w_j^n h_j^n}. \quad (1)$$

The pixel sum of r is denoted by sum(r), and w/h is the width/height of the rectangle. The features have several good properties, which are proved as following. Since rectangle is the basic unit of our features, we would like to discuss the properties of rectangle feature firstly.

B. Rescaling properties of basic rectangle feature

A image rectangle in the window is specified by r(l, t, w, h), where l/t is the left/top and w/h is the width/height of the rectangle. The pixel sum of r is denoted by sum(r), where sum(r) is defined as $\text{sum(r)}=\sum_{y=t}^{t+h-1}\sum_{x=l}^{l+h-1} I(x,y)$ and I(x, y) is the gray intensity of pixel(x, y).

Let us consider such situation when a image r(l, t, w, h) is zoomed out with scale $\frac{1}{s}$(s>1) to get a smaller image r'(r', t', w', h') and thus $s=\frac{w}{w'}=\frac{h}{h'}$. Shown in fig. 1, the procedure is approximate to that two digital images r and r' are all got by sampling from an original continuous image with different spatial sample frequency $(\frac{1}{\Delta x}, \frac{1}{\Delta y})$ and $(\frac{1}{s\Delta x}, \frac{1}{s\Delta y})$.

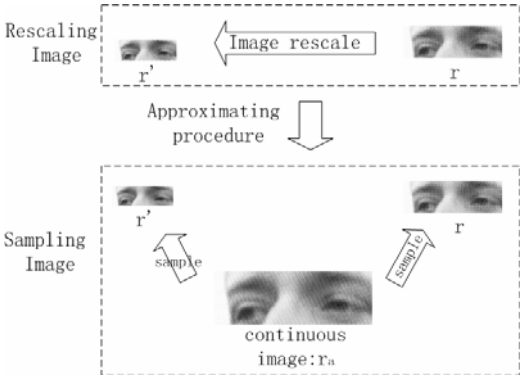

Fig. 1. Approximating image rescaling to image sampling

Let r_a denote a continuous image, and we sample it to get r with spatial sample frequency $(\frac{1}{\Delta x}, \frac{1}{\Delta y})$. Let $F(\omega_1, \omega_2)$ denote the discrete two-dimensional Fourier transform of the sampled image, and according to Fourier transform convolution theorem, there exists the relationship

$$F(\omega_1, \omega_2) = \frac{1}{\Delta x \Delta y}\sum_{k_1}\sum_{k_2} F_a(\frac{\omega_1 - 2\pi k_1}{\Delta x}, \frac{\omega_2 - 2\pi k_2}{\Delta y}), \quad (2)$$

where $F_a(\omega_1,\omega_2)$ denotes continuous two-dimensional Fourier transform of original image. Supposing that the sampling rate is greater than the Nyquist rate, from (2), we get $F(0,0)=\frac{1}{\Delta x}\frac{1}{\Delta y}F_a(0,0)$. According to discrete Fourier transform definition, $F(0,0)=sum(r)$. Thus we get $sum(r)=\frac{1}{\Delta x}\frac{1}{\Delta y}F_a(0,0)$. Suppose that we get r' by sampling r_a with sample frequency $(\frac{1}{s\Delta x},\frac{1}{s\Delta y})$, and thus $sum(r')=\frac{1}{s\Delta x}\frac{1}{s\Delta y}F_a(0,0)$. So we get

$$\frac{sum(r)}{sum(r')}=s^2. \tag{3}$$

Equation (3) shows the approximate relationship between intensities sum of original rectangle image and that of rescaled rectangle image.

C. Scale-invariant property

Let us assume the basic unit for the presence of an object is a window called candidate window. Suppose that there are two candidate windows, and window 2 is got by rescaling window 1 with scale s(s>1). Shown as fig. 2, we use feature sets R^p and R^n to calculate features in window 1, and the corresponding rescaled feature sets R'^p and R'^n to calculate features in window 2. To window 2, equation (1) has the form

$$f(R'^p,R'^n)=\frac{\sum_i sum(r_i'^p)}{\sum_i w_i'^p h_i'^p}-\frac{\sum_j sum(r_j'^n)}{\sum_j w_j'^n h_j'^n}, \tag{4}$$

Where $s=\frac{w_i'^p}{w_i^p}=\frac{h_i'^p}{h_i^p}=\frac{w_i'^n}{w_i^n}=\frac{h_i'^n}{h_i^n}$. According to equation (4) and (2), we get $f(R'^p,R'^n)=f(R^p,R^n)$. Thus the rescaled features in corresponding rescaled image windows are same as the original features in the original image and we need only to calculate rescaled features in candidate windows of different size, and no more other operations are required. The property is called scale-invariant in the paper.

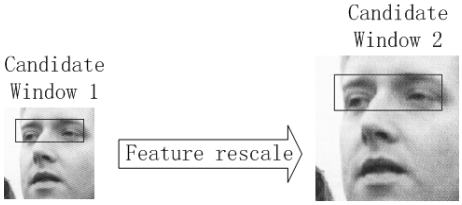

Fig. 2. Rescaling features

D. Illumination-invariant property

Rainer lienhart et al. [3] used $I'(x,y)=\frac{I(x,y)-u}{c\sigma}$, $c\in R^+$ to correct illumination. According to equation (1), when we translate a image with equation $I'(x,y)=\frac{I(x,y)-u}{c\sigma}$, the corresponding feature in translation image is $f'(R^p,R^n)=\frac{f(R^p,R^n)}{c\sigma}$, so our features only need to be normalized by $c\sigma$. We call the property as illumination-invariant which means the features are robust to the change of illumination of the whole image.

E. Feature shape mask

Above works have presented many feature shapes, some of which are shown as a, b, c in fig. 3. We unify the feature shapes and generalize them to a shape mask composed of 9 blocks. In the nine blocks, positive features set is composed of white blocks, negative features set is composed of gray blocks, and black ones are left to be not used. Using this feature mask we can get tremendous feature shapes (almost 3^9 cases) which is very convenient. The shape mask can present most of the prior features used by others similarly, some of which are shown as d, e, f in fig. 3.

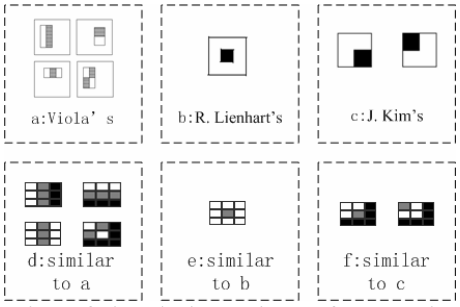

Fig. 3. Examples of simulating others' features by shape mask

III. Improved structure of cascade

In [1], cascade structure was used to speed up, with which most of the false candidate windows are rejected by simple classifiers, while only a little amount of promising windows are processed with complex computation to reject negative samples similar to positive. To make more use of the prior stages information, we use fuzzy output and voting to improve face hit rate with similar false alarm rate.

Original cascade classifier owns the disadvantage of omitting the information of the prior stages. To make full use of information, we set the output of last several stage classifiers to three states: face, non-face, and potential face. Shown as fig. 4, at the last m stages, if a candidate window is rejected by the current stage, we do not reject it as non-face, while setting their output as potential face and giving it a probability being face according to both its passed stages number and confidence of the final rejected stage classifier. All the potential face would be processed in the merging procedure.

When all non-face candidate windows are rejected, the face and potential face detections are merged to create final valid faces. Since face windows are not as sensitive to changes in position and scale as non-face, one valid face might create several detected candidates, while false detected non-face often appearing individually. Among the potential detections, several intersecting ones are introduced by one face which can not pass all the stages and by voting, we can expect some of the neighboring potential faces can be merged to create a final valid face while rejecting the individual potential faces created by non-face.

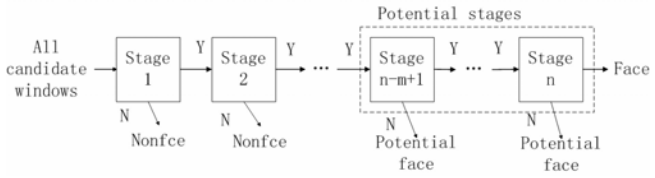

Fig. 4.The structure of fuzzy cascade classifier

The whole face merging procedure is composed of two steps. In first step, we merge detections at same position with similar size by voting. First, the set of face and potential face windows are partitioned into disjoint subsets and two detections are in the same subset if the centers distance and size ratio of two windows are in the limited range. In each subset, if there is face detection, yields a single final detection, and outputs the final bounding region as the average of corners of all face detections in the subset. If there is no face detection, add all face probability of the potential faces together, and if the probability sum is larger than a threshold, yields a final detection, and output the bounding region as the average of corners of all potential face detections. If the probability sum is smaller than threshold, reject all the potential windows as non-face. In second step, we combine overlapping valid detections to reduce false detections. Because in nature, the situations such as one face is in another or two faces are intersected by almost 1/2 of the whole acreage are hard to appear, we integrated overlapping faces to avoid unnatural situations and thus reduce false detections.

IV. Face detection in real world

Before we implement the approach in face detection, we will present some more details of feature selection and training methods.

A. Feature selection and fast computation

One feature is decided by three factors. The shape mask denotes the feature's shape which thus decides the feature's calculation manner, and the position and size including width and height distinguish features by calculating difference of intensities sum of neighboring regions at different position with different size.

Since the raw feature set is infinitely large, for practical reasons, the shapes are selected as follows. The 9 basic blocks should be same as each other in size, while the width and height of one block can be different. Due to each shape, we shift the position and change the size, but limit the outline in candidate rectangle, and we can get $\left(\frac{1}{2}(H+2)(\frac{H}{3}-1)+1\right)\left(\frac{1}{2}(W+2)(\frac{W}{3}-1)+1\right)$ features totally for one shape, where W/H is the width/height of the candidate windows. The shape can be in 3^9 different forms and since the feature shapes are tremendous, considering expressional power and computation, only 16 feature shapes are selected to train, some of which are shown in fig. 3.

Like Viola, we get pixel sum of rectangle by integral image as following format: $sum(r(l, t, w, h)) = II(l-1,t-1)+II(l+w-1, t+h-1)-II(l - 1, t + h)-II(l+w-1,t-1)$, where II(x, y) is the (x, y) value in integral image. Getting pixel sum of one rectangle is called a rectangle operation in this paper. It seems that our features need 9 rectangle operations, but by integrating neighboring regions, it can be reduced greatly and each feature used in the paper only needs 2 or 3 rectangle operations and a little addition/subtraction operations. For each scale level, we record the reference coordinate of the rescaled features to the top-left of integral image and the sum of acreage of the positive/negative rectangles of the features in look-up-table (LUT), after looking up the value of the rescaled rectangle's coordinate and areas, we calculate features with reference coordinate. Since the feature is average-invariant, we only use image variance σ to correct illumination, which can be got using integral images of both original image and image squared.

One of the advantages of haar-like features is that they can easily be rescaled which avoids to calculate a pyramid of images and thus accelerates the detector greatly. When rescaling, we must round the coordinate to nearest integer, which would degrade the performance of Viola's features

[3]. But due to our features, we need only to round the rescaled coordinates to nearest integer and store the corresponding acreage, and because the features have been normalized by acreage, the rounding error would be decreased very much.

B. Candidate weak classifiers and training methods

We use Real AdaBoost [8] to train every stage classifiers. Domain-partitioning weak hypotheses are used to build weak classifiers. To minimize the upper bound on training error, each candidate weak classifier is computed from the weighted histograms of face and non-face on several disjoint blocks, and to avoid the value in the histograms might be very small or even zero, the weak classifiers prediction is smoothed as [8]

$$h(x) = \frac{1}{2}\ln(\frac{W_{+1}^{j} + \varepsilon}{W_{-1}^{j} + \varepsilon}), \tag{5}$$

where ε is a small positive constant, and W_{+1}^{j}, W_{-1}^{j} are the weighted histograms of features of face and non-face examples in block j. To save training time, we use weight trimming method to select examples with larger weights to train while neglecting the samples with smaller weight [7].

C. Experimental Results

All experiments are performed on the complete CMU frontal face test set of 130 grayscale pictures with 510 frontal faces [2]. The hit and false alarm criterion is same as that defined by Rainer Lienhart [3].

We use 1000 original frontal faces to create 5000 face patterns with random rotating about ±10 degree, random shifting up to ±1 pixel, and random mirroring, while 3000 non-face samples passed all prior stages are used to train the current stage which is called bootstrap. All the samples are resized to 24×24. With training algorithm for building a cascade detector [1], we get a cascade-structured classifier with 34 stages which is called original cascade classifier here. Due to each stage, the hit rate is no less than 0.998, and the false alarm rate is no more than 0.4. Beginning with the 24×24 scale, we set the scale factor to 1.15 to build the rescaled features, and by shifting the windows with 1.15 times of the candidate windows' size, we scan all the potential positions.

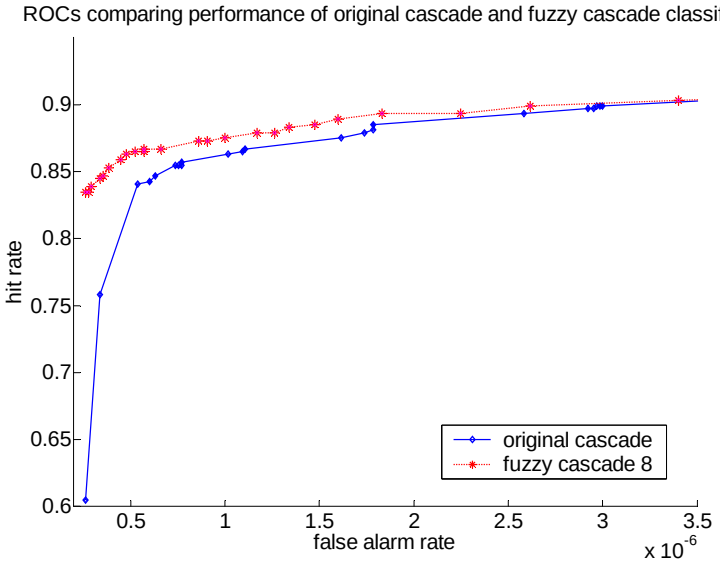

Fig. 5. ROCs comparing performance of original cascade and fuzzy cascade classifier

We set the final 8 stages of original cascade classifier as potential stages to get a fuzzy cascade classifier. The Receiver Operating Curves (ROCs) shown in fig. 5 compares the performance of fuzzy cascade classifier and the original cascade classifier. From the ROCs, we can see that the hit rate of voting cascade classifier outperforms that of the original one with same false alarm rate. Especially when the false alarm rate is lower than 5E-7, the hit rate is increased more than 5%. But with the false alarm rate increasing, the curve of voting cascade inclines to the original cascade curve, which is because that when the threshold decreases to some extent, all the candidates pass the stages before the first potential stage can pass all the following stages, both the fuzzy cascade classifier and the original cascade classifier use the stages before the first potential stage only. Some of the detection examples are shown in fig. 6.

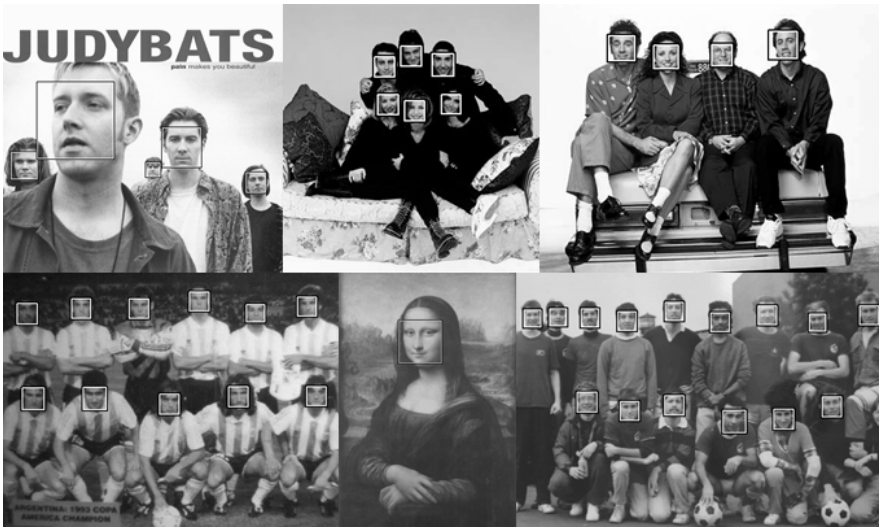

Fig. 6. Some detection examples

V. Conclusions

The paper is based on the work of Viola et al. In the paper, firstly, we propose a novel feature definition and introduce a feature shape mask to represent it. The defined features are scale-invariant which means the features can be rescaled easily and reduce the performance degradation introduced by rounding the coordinates to nearest integer. In addition, the novel features are robust to whole image illumination changing and illumination correction can be processed with integral images of both original image and image squared. The feature shape mask can be computed efficiently and expanded conveniently, which can simulate feature shapes used by others [1], [3], [5] and thus enrich the haar-like feature pool. Fast features computation methods and an approach to create candidate weak classifiers from features are also introduced.

Secondly, we introduce a novel cascade structure called fuzzy cascade based on cascade structure proposed in [1]. With fuzzy output and voting, we promote the hit rate with same false alarm rate.

Although the examples and experiments are based on face detection, both the features and the novel cascade structure can be implemented in object detection conveniently.

References

[1] P. Viola and M. Jones. Robust real time object detection. In IEEE ICCV Workshop on Statistical and Computational Theories of Vision (July 13, 2001), Vancouver, Canada.

[2] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. In IEEE Patt. Anal. Mach. Intell. (1998), volume 20, pages 22-38.

[3] R. Lienhart, A. Kuranov and V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In DAGM'03, 25th Pattern Recognition Symposium (2003), pages 297-304.

[4] S.Z. Li, Z.Q. Zhang, H. Shum, and H.J. Zhang. FloatBoost learning for classification. In NIPS 15 (December 2002).

[5] J. Kim, Y. Sung, and S. Kee. A fast and robust face detection based on module switching network. In Sixth IEEE International Conference on Automatic Face and Gesture Recognition (May 17-19, 2004), Seoul, Korea.

[6] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In International Conferenceon Computer Vision (1998).

[7] J. Friedman，T. Hastie and R. Tibshirani，Additive logistic regression：a statistical view of boosting，Technical Report，Stanford University (1998).

[8] E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory (1998), pages 80-91.

[9] B. Wu, H.Z. AI, C. Huang and S.H. Lao. Fast rotation multi-view face detection based on real AdaBoost. In Sixth IEEE International Conference on Automatic Face and Gesture Recognition (May 17-19, 2004), Seoul, Korea.

Yafeng Deng received his master degree in Information and Communication System in Electronic engineering, department of Tsinghua University in July, 2005. Currently, he is a member of the research center of Vimicro Corporation and his interests include pattern recognition and computer vision.

Guangda Su received the B.S. degree in electronic engineering from Tsinghua University in 1977, major in wireless technology. He has famous lecture on “Image Processing System” (for undergraduate) and “Parallel Image Processing Techniques” (for graduate). His projects are successfully done include: General Image Processing System, Image Processing System for Camera, Ultrasonic Image Processing System, Image Restoration System, Computer face Combination System, Face recognition System. He received ministry level scientific awards five times. Main research area is image recognition and high speed image processing.