Block Independent Component Analysis for Face Recognition

Lei Zhang, Quanxue Gao and David Zhang
Biometrics Research Center
Dept. of Computing, The Hong Kong Polytechnic University
E-mail: cslzhang@comp.polyu.edu.hk

Abstract

This paper presents a subspace algorithm called block independent component analysis (B-ICA) for face recognition. Unlike the traditional ICA, in which the whole face image is stretched into a vector before calculating the independent components (ICs), B-ICA partitions the facial images into blocks and takes the block as the training vector. Since the dimensionality of the training vector in B-ICA is much smaller than that in traditional ICA, it can reduce the face recognition error caused by the dilemma in ICA, i.e. the number of available training samples is greatly less than that of the dimension of training vector. Experiments on the well-known Yale and AR databases validate that the B-ICA can achieve higher recognition accuracy than ICA and enhanced ICA (EICA).

1. Introduction

Subspace analysis techniques have been widely used in face recognition [1-4, 7-11]. They represent a face image as a linear combination a set of optimal bases under some criteria. The task of face recognition is implemented in the space spanned by those bases, which is usually a subspace of the original face space. Principal component analysis (PCA) is one of the most widely used subspace analysis methods [4]. It computes the optimal subspace in the sense of minimum mean square error. As a supervised learning technique, Linear Discriminant Analysis (LDA) [3] tries to find a subspace that maximizes the between class distance and minimizes the within-class distance. Different from PCA and FLD, independent component analysis (ICA) [5] tries to find a subspace spanned by a set of independent bases.

PCA was first used by Kirby and Sirovich [4] to represent human faces and it was found that a face image could be reconstructed approximately as a weighted sum of a small collection of basis facial images plus a mean face image. Based on this research, Turk and Pentland [7] developed the well-known Eigenface method. Most of the PCA based face recognition methods need to stretch the image matrix to a vector before calculating the principle components (PCs). In [2], Yang *et al* proposed a two dimensional PCA (2D-PCA) scheme by projecting the image matrix, but not the stretched image vector, onto a set of basis vectors. PCA exploits only the second-order statistics of the dataset. The high-order statistics, which can be very useful to the face representation and recognition, are not exploited in PCA.

ICA, as an extension of PCA, was originally developed for blind source separation and it has been widely used in signal processing, medical image analysis and pattern recognition [5-6, 8-12]. The objective of ICA is to seek for a set of linear bases which are as independent as possible in the sense of high-order, other than the second-order, statistics. The independent components (ICs) obtained by projecting the face images onto the subspace spanned by these bases can reflect better the intrinsic properties and local characteristics of the facial dataset [9]. In a word, ICA can remove the high-order statistical dependencies to produce a sparse and independent code for subsequent pattern discrimination.

Bartlett *et al* [8] first applied ICA to face recognition and found that high-order statistical information is useful for representing and identifying faces. Since then, ICA has gained more interests in face modeling and recognition [9-11]. Liu [9] analyzed the performance of ICA and proposed an enhanced ICA (EICA) method, which implements ICA in a reduced PCA space to improve retrieval performance. Pong *et al* [11] studied the effect of the number of ICs on recognition accuracy. They indicated that not all ICs are useful for recognition and proposed an algorithm to select ICs.

The above ICA-based methods stretch the whole 2D facial image matrix into a 1D vector before computing ICs. The dimensionality of the resulting

14th International Conference on Image Analysis and Processing (ICIAP 2007)
0-7695-2877-5/07 $25.00 © 2007 IEEE

vector is usually very high and this leads to a dilemma of ICA: the number of available training samples is much less than that of the dimension of the underlying vector. This dilemma makes it very difficult to estimate accurately the statistics of the underlying face vector. The estimation error of the face vector statistics will then deteriorate the accuracy of face recognition.

To reduce the effect of the dimensionality dilemma in ICA, we propose a block ICA (B-ICA) scheme in this paper. The whole image is portioned into many sub-images, i.e. blocks, of the same size, and then a common demixing matrix for all the blocks is calculated. Compared with ICA, whose training vector is stretched from the whole image, B-ICA stretches only part of the face image as the training vector. B-ICA greatly dilutes the dimensionality dilemma of ICA because the dimension of the training vector is much smaller so that the statistics can be more accurately estimated. Our experimental results show that B-ICA achieves higher recognition accuracy than ICA and it is also computationally more efficient.

The rest of this paper is organized as follows. Section 2 briefly reviews ICA. Section 3 describes the proposed B-ICA algorithm. Section 4 conducts experiments to test the proposed method and Section 5 concludes the paper.

2. Independent component analysis

PCA exploits only the second-order statistics and it is optimal for datasets which are of Gaussian distribution. In general, however, the distribution of facial images is non-Gaussian. ICA can be viewed as an extension of PCA to deal with non-Gaussian datasets. In this section, we briefly review the concepts and computation procedure of ICA. For more details, please refer to [5-6, 12].

Suppose an *n*-D vector $\vec{x}=\left[x_1,x_2,\cdots,x_n\right]^T$ can be represented as the linear combination of m $\left(m\le n\right)$ elements $s_1,s_2,\cdots,s_m$, which are statistically independent (or as independent as possible), then the noise-free model of ICA is

$$\vec{x}=Q\vec{s} \tag{1}$$

where $\vec{s}=\left[s_1,\cdots,s_m\right]^T$ is the vector of ICs, Q is an unknown $n\times m$ mixing matrix..

In general, ICs s_i , $i=1,2,...,m$, and the mixing matrix Q are unknown. ICA aims to find a demixing matrix W such that

$$\vec{u}=W\vec{x}=WQ\vec{s} \tag{2}$$

is a good estimation of $\vec{s}$, with possible permutation and rescaling. Fig. 1 illustrates the procedure. If $E\left\{s_is_i\right\}=1$ for all ICs s_i , $i=1,2,...,m$, the ICs will be uniquely determined except for their signs [6].

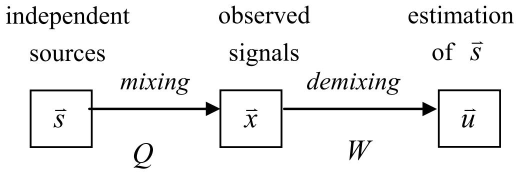

Figure 1. The model of ICA.

Many algorithms to implement ICA have been developed based on nonlinear decorrelation or maximum non-Gaussianity [6, 12]. Among them, the Fast-ICA [12] algorithm has been dominantly used. It is implemented in two steps. The first step is whitening. The covariance matrix of $\vec{x}$ is

$$\textstyle\sum_x=E\left[\left(\vec{x}-E\left[\vec{x}\right]\right)\left(\vec{x}-E\left[\vec{x}\right]\right)^T\right] \tag{3}$$

where $E\left[\bullet\right]$ is the expectation operator. Let $V=\left[\vec{\gamma}_1,\vec{\gamma}_2,...,\vec{\gamma}_n\right]$ and $\Lambda=diag\left(\lambda_1,\lambda_2,\cdots,\lambda_m\right)$, where $\vec{\gamma}_i$ and λ_i are the eigenvectors and the corresponding eigenvalues of $\sum_x$, respectively, and $\lambda_1\ge\lambda_2\ge\cdots\ge\lambda_m$. Then the whitened data can be calculated as

$$\vec{y}=\left(V\Lambda^{-1/2}\right)^T\vec{x}=P^T\vec{x} \tag{4}$$

where $P=V\Lambda^{-1/2}$ is called the whitened matrix. $\vec{y}$ is decorrelated and has an unit variance.

In the second step, the whitened data $\vec{y}$ is used to compute the demixing matrix W_d via kurtosis such that the components of $\vec{u}=W_d\vec{y}$ are (almost) independent. Denote by $\vec{w}$ the column vector of W_d . The kurtosis of the projection of $\vec{y}$ onto $\vec{w}$ is

$$kurt(\vec{w}^T\vec{y})=E\left[\left(\vec{w}^T\vec{y}\right)^4\right]-3\left(E\left[\left(\vec{w}^T\vec{y}\right)^2\right]\right)^2 \tag{5}$$

To make the components of $\vec{u}$ as independent as possible, we want to find a $\vec{w}$ to maximize $kurt(\vec{w}^T\vec{y})$ under constraint

$$E\left[\left(\vec{w}^T\vec{y}\right)^2\right]=1 \tag{6}$$

A Lagrangian coefficient λ is introduced to $kurt(\vec{w}^T\vec{y})$ to solve the optimization problem

14th International Conference on Image Analysis and Processing (ICIAP 2007)
0-7695-2877-5/07 $25.00 © 2007 IEEE

$$kurt(\vec{w}^T\vec{y}) = E\left[\left(\vec{w}^T\vec{y}\right)^4\right] - 3\left(E\left[\left(\vec{w}^T\vec{y}\right)^2\right]\right)^2 + \lambda\left(1 - E\left[\left(\vec{w}^T\vec{y}\right)^2\right]\right) \quad (7)$$

Differentiate Eq. (7) with respect to $\vec{w}$ and let it be 0, we have

$$\vec{w} = \frac{2}{\lambda}\left(H^{-1}E\left[\vec{y}\left(\vec{w}^T\vec{y}\right)^3\right] - 3\vec{w}\right) \quad (8)$$

where $H = E\left[\vec{y}\vec{y}^T\right]$. $\vec{w}$ can be calculated iteratively as follows:

$$\vec{w}^*(t) = H^{-1}E\left[\vec{y}\left(\vec{w}(t-1)^T\vec{y}\right)^3\right] - 3\vec{w}(t-1) \quad (9)$$

$$\vec{w}(t) = \frac{\vec{w}^*(t)}{\sqrt{\vec{w}^*(t)^T H\vec{w}^*(t)}} \quad (10)$$

Using (9) and (10), we can calculate all the column vectors $\vec{w}_i$, $i = 1,\cdots,m$, of W_d and then the demixing matrix W_d is obtained as $W_d = \left[\vec{w}_1 \ \cdots \ \vec{w}_m\right]$. We let $W = W_d \cdot P$ and call W the demixing matrix of $\vec{x}$ because $\vec{u} = W_d\vec{y} = W_d \cdot P\vec{x} = W\vec{x}$.

3. Algorithm of Block ICA (B-ICA)

3.1. Idea of B-ICA

The matrix-to-vector stretching procedure in the conventional ICA makes the statistics estimation difficult and inaccurate because the training sample size is relatively very small compared with the high dimensionality of the training vector. This problem also exists in the PCA-based face recognition. To dilute this small sample size problem, Yang *et al* [2] proposed a 2D-PCA scheme. They projected the 2D face image onto a set of vectors using $\vec{y} = A\vec{x}$, where A is the face image, $\vec{x}$ is the projection vector and $\vec{y}$ is the projected vector. 2D-PCA actually takes each row of the face image as a training vector and then finds a common projection subspace which applies to all row vectors. Since the dimension of each row vector is significantly less than that of a face image vector, the small training sample size problem is diluted. The experimental results in [2] validate that 2D-PCA achieves better face recognition performance than PCA, especially when the training sample size is small.

The idea of 2D-PCA can be applied to ICA by setting the row of the facial image as the training vector. Though this strategy has been proved to be able to improve the recognition performance, it actually restricts the flexibility of setting other forms of training vector. In this paper, we propose a block ICA (B-ICA) algorithm. It can be seen that setting the row of the face image as the training vector is just a special case of vector setting in B-ICA.

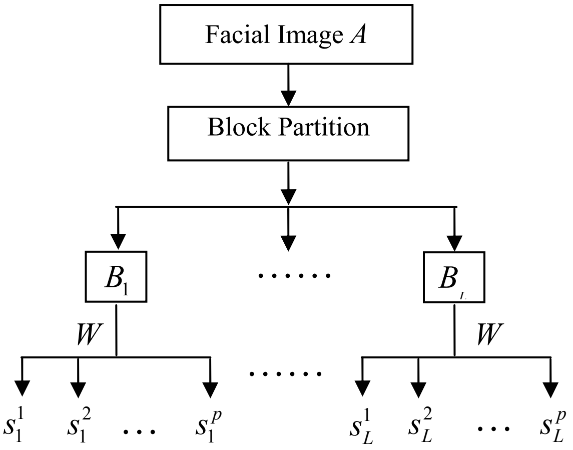

Figure 2. Illustration of B-ICA.

The 2D face image can be viewed as a super-class and it can be portioned into many blocks with the same size, which can be viewed as sub-classes. Instead of finding a demixing matrix for the whole face image, we find a common demixing matrix W for all the sub-classes. Fig. 2 illustrates the procedure of B-ICA. The face image A is evenly partitioned into L blocks: B_1, B_2, …, B_L. Then we stretch the block to a vector and take it as the training vector, whose dimension is much smaller than the original face image. A common demixing matrix W for all the blocks is computed, by which p ICs (s_l^1, s_l^2, …, s_l^p) of block B_l are extracted. Finally the set of ICs of the original image A is set as $S = \left\{s_l^j \,|\, l = 1,2,...,L; j = 1,2,...,p\right\}$ and S will be used for face recognition.

3.2. Implementation of B-ICA

Denote by $A \in R^{M\times N}$ a face image and by $T = \left\{ \ A_k \,|\, k = 1,2,\cdots,K \ \right\}$ the training set, which has K face image samples. We partition A into L blocks B_1, B_2, …, B_L, whose size is $m\times n$. We stretch B_l to a $m\cdot n\times 1$ vector, denoted by $\vec{b}_l$. Then the training set can be rewritten as

$$T = \left\{ \ \vec{b}_l^k \,|\, k = 1,2,\cdots,K; l = 1,2,\cdots,L \ \right\}$$

where $\vec{b}_l^k$ means it is the l^{th} row of the k^{th} face sample. We define a common covariance matrix $G \in R^{m \cdot n \times m \cdot n}$ of vector $\vec{b}_l$ as

$$\begin{aligned} G &= \frac{1}{L}\sum_{l=1}^{L} E\left[\left(\vec{b}_l - \bar{\bar{b}}_l\right)\left(\vec{b}_l - \bar{\bar{b}}_l\right)^T\right] \\ &= \frac{1}{L \cdot K}\sum_{l=1}^{L}\sum_{k=1}^{K}\left(\vec{b}_l^k - \bar{\bar{b}}_l\right)\left(\vec{b}_l^k - \bar{\bar{b}}_l\right)^T \end{aligned} \qquad (11)$$

where $\bar{\bar{b}}_l = \frac{1}{K}\sum_{k=1}^{K}\vec{b}_l^k$.

Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{n \cdot m}$ be the eigenvalues of G and $\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_{n \cdot m}$ the corresponding eigenvectors. The whitened matrix P on face image B_l is

$$P = V\Lambda^{-1/2} \qquad (12)$$

where $V = \left[\vec{v}_1 \;\; \vec{v}_2 \;\; \cdots \;\; \vec{v}_{n \cdot m}\right]$ and $\Lambda = diag\left(\lambda_1, \lambda_2, \cdots, \lambda_{n \cdot m}\right)$. We denote by $\vec{z}_l^k$ the whitened vector of each training sample $\vec{b}_l^k$

$$\vec{z}_l^k = P^T\left(\vec{b}_l^k - \bar{\bar{b}}_l^k\right) \qquad (13)$$

The demixing matrix $W_d \in R^{p \times n \cdot m}$ of the whitened data is computed by taking $\vec{z}_l^k$ as inputs. $p \leq n \cdot m$ is the number of ICs we set for $\vec{b}_l^k$. The Fast-ICA algorithm introduced in Section 2 is used to compute W_d from $\vec{z}_l^k$. Then $W = W_d \cdot P^T$ is the demixing matrix of vector $\vec{b}_l$.

Projecting $\vec{b}_l$ onto W, we get the B-ICA output of block B_l as

$$\vec{s}_l = \left[s_l^1, s_l^2, ..., s_l^p\right]^T = W\left(\vec{b}_l - \bar{\bar{b}}_l\right) \qquad (14)$$

Compared with $\vec{b}_l$, whose dimension is $m \cdot n \times 1$, the B-ICA output $\vec{s}_l$ preserves the most important ICs of $\vec{b}_l$ but with a smaller dimension $p \times 1$. The B-ICA result of the original face image A is

$$S = \left[\vec{s}_1^T, \vec{s}_2^T, ..., \vec{s}_L^T\right]^T \qquad (15)$$

3.3. Face classification

By partitioning each training image A_k ($k = 1, 2, \cdots, K$) into L blocks and projecting those blocks onto the subspace determined by demixing matrix W, we obtain the IC set S_k for image A_k . Given an input face image I to be recognized, we can compute its IC set S^* using W. The nearest neighbor classifier is used for recognizing I . The Euclidian distance between S and S^* is defined by

$$d\left(S^*, S_k\right) = \sqrt{\sum_{i=1}^{p \cdot L}\left(S^*(i) - S_k(i)\right)^2} \qquad (16)$$

If $d\left(S^*, S_t\right) = \min_k d\left(S^*, S_k\right)$, then the input image I is judged belonging to the class of face image A_t . Other distance metrics, such as cosine, L^1 and Mahalanobis distances, can also be used.

4. Experimental results

Two well-known face databases, Yale and AR, are used to test the proposed method when there are facial variations over time, facial expressions, illumination and occlusion, etc. The widely used subspace methods, including PCA, 2D-PCA, ICA and EICA, are employed for comparison.

4.1. Experiments on Yale database

The Yale face database was built at the Yale Center for Computational Vision and Control. It contains 165 grayscale images of 15 persons. The images demonstrate variations in lighting condition (left-light, center-light, right-light), facial expression (normal, happy, sad, surprised, and wink), and with/without glasses. All the images are manually cropped to 32×32 in the following experiments.

In the first experiment, we select 5 images per person for training and use the remaining images for testing. Thus the total number of training images is 75 and the number of testing images is 90. In the second experiment the training and testing datasets are exchanged. For B-ICA, we use different block sizes to test its performance. Table 1 lists the top correct recognition rate (CRR) values, the corresponding dimensions of feature vector, and the sizes of demixing matrices for these methods in the two experiments.

From table 1, we see that the top recognition rates of PCA, 2D-PCA, ICA and EICA are 67.78% (65.33%), 72.222% (68.00%), 66.67% (65.56%) and 68.89% (66.67%), respectively. (The values in the parentheses denote the results of the second experiment). B-ICA achieves higher recognition rate than the above methods. When the block size is *m*=2, *n*=16, B-ICA achieves the highest CRR values 75.56% (69.33%) for the two experiments.

The full demixing matrices in ICA and EICA are of size 1024×1024 (32×32=1024), while the full demixing matrix is of size 32×32 for the B-ICA with block size 2×16. From table 1, it is also seen that B-ICA needs a smaller number of features while achieves better recognition accuracy than other methods.

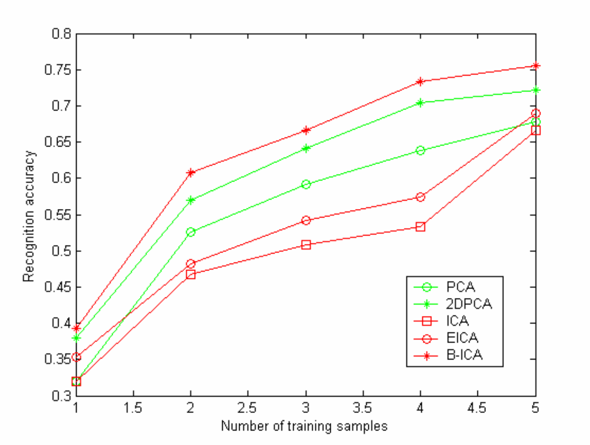

Figure 3. Recognition accuracies of PCA, 2DPCA, ICA, EICA, B-ICA.

To evaluate more extensively the performance of B-ICA, we increase the number of training samples from 1 to 5 per individual and use the remaining images for testing. Fig. 3 plots the highest CRR curves of PCA, 2D-PCA, ICA, EICA and B-ICA versus the number of used training samples. We can see that B-ICA always achieves the best performance.

4.2. Experiments on AR database

The AR database is used in this section to evaluate the performance of B-ICA under variations of illumination, facial expression and time. AR database contains over 4000 color face images from 126 people (70 men and 56 women), including frontal views of faces with different facial expression, lighting conditions and occlusions. The pictures of most persons were taken in two sessions, separated by two weeks. Each session contains 13 color images per person and 120 individuals (65 men and 55 women) participated in both sessions. In our experiment, the images of these 120 individuals are selected and used. The face portions of those images were manually cropped to 50×40 pixels.

We perform experiments by varying the number of training samples per person. In the k^{th} test, we randomly select k images for each person for training and use the remaining samples for testing. The top CRR values of different methods at different number of training samples and the corresponding number of features are listed in Table 2. (The CRR values of B-ICA is with block size m=2, n=20.) It is shown that B-ICA achieves much better result than other methods.

5. Conclusion

A new face recognition method called block independent component analysis (B-ICA) was presented in this paper. B-ICA partitions the face image into small blocks of the same size and then computes a common demixing matrix for those blocks. Because the dimensionality of the training vector is much smaller than that in the traditional ICA, B-ICA significantly dilutes the dilemma in ICA: the dimensionality of training vector is very high but the size of training samples is relatively very small. Experiments on Yale and AR databases show that B-ICA achieves higher recognition accuracy than ICA, EICA and other methods.

6. References

[1] X. Wang and X. Tang, A unified framework for subspace face recognition, IEEE Trans. on Pattern Anal. Mach. Intell. 26(9) (2004)1222–1228.

[2] J. Yang, D. Zhang, A. F. Frangi, and J. Y. Yang, Two-dimensional PCA: a new approach to appearance-based face representation and recognition, IEEE Trans. on Pattern Anal. Mach. Intell., 26 (1) 2004 131-137.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, IEEE Trans. on Pattern Anal. Machine Intell., 19 (1997) 711–720.

[4] M. Kirby and L. Sirovich, Application of the KL Procedure for the Characterization of Human Faces, IEEE Trans. on Pattern Analysis and Machine Intelligence, 12 (1) (1990)103-108.

[5] P. Comon, Independent component analysis, a new concept? Signal Processing, 36 (1994) 287–314.

[6] A. Hyvärinen, Survey on independent component analysis, Neural Computing Surveys, 2 (1999) 94–128.

[7] M. Turk and A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience, 3 (1) (1991) 72-86.

[8] M. S. Bartlett, J. R. Movellan and T. J. Sejnowski, Face recognition by independent component analysis, IEEE Trans. on Neural Networks, 13 (6) (2002) 1450–1464.

[9] C. Liu, Enhanced independent component analysis and its application to content based face image retrieval, IEEE Trans. on System, man and Cybernetics, Part B, 34 (2) (2004) 1117-1127.

[10] C. Liu and H. Wechsler, Independent Component Analysis of Gabor Features for Face Recognition, IEEE Trans. on Neural Networks, 14 (4) (2003) 919-928.

[11] P. C. Yuen and J. H. Lai, Face representation using independent component analysis, Pattern Recognition, 135 (2002) 1247-1257.

[12] A. Hyvärinen and E. Oja, A fast fixed-point algorithm for independent component analysis, Neural Computation, 9 (7) (1997) 1483–1492.

Table 1. The top recognition rates, the corresponding dimensions of feature vectors and the sizes of projection matrices for PCA, 2D-PCA, ICA, EICA and B-ICA on Yale database. The values in parentheses are the result for the second experiment.

Algorithms	**Top recognition rate (%)**	**Dimension of features**	**Size of projection matrix**
PCA	67.78 (65.33)	53 (31)	1024×1024
2D-PCA	72.22 (68.00)	160 (128)	32×32
ICA	66.67 (65.56)	72 (80)	1024×1024
EICA	68.89 (66.67)	44 (32)	1024×024
B-ICA(m=16,n=16)	62.22 (57.33)	24 (28)	256×256
B-ICA(m=8,n=16)	64.44 (64.00)	16 (40)	128×128
B-ICA(m=8,n=8)	66.67 (66.67)	48 (80)	64×64
B_ICA(m=4,n=4)	73.33 (65.33)	192 (192)	16×16
B-ICA(m=2,n=16)	**75.56 (69.33)**	34 (34)	32×32
B-ICA(m=2,n=32)	74.44 (66.67)	32 (32)	64×64
B_ICA(m=1,n=32)	74.44 (65.33)	160 (64)	32×32

Table 2 Comparison of the recognition rates of PCA, 2D-PCA, ICA, EICA and B-ICA on AR database. The values in parentheses are the corresponding number of features.

Training samples	***PCA***	***2D-PCA***	***ICA***	***EICA***	***B-ICA (m=2, n=20)***
1	53.97% (49)	63.93% (550)	52.03% (100)	59.03% (50)	**64.87%** (350)
2	54.90% (50)	63.81% (500)	57.88% (97)	61.18% (60)	**65.66%** (350)
3	65.62% (50)	71.56% (500)	65.29% (85)	66.67% (58)	**71.67%** (400)
4	63.75% (49)	70.91% (500)	63.75% (81)	66.55% (40)	**72.01%** (350)
5	62.30% (49)	69.44% (450)	62.26% (65)	68.77% (37)	**71.35%** (250)
6	73.92% (50)	85.46% (700)	80.17% (53)	83.63% (34)	**85.96%** (400)
7	73.90% (49)	86.05% (700)	83.68% (62)	83.82% (50)	**88.07%** (450)
8	72.50% (49)	85.32% (650)	83.47% (59)	84.23% (45)	**88.01%** (450)
9	70.94% (50)	85.25% (700)	81.52% (58)	82.45% (35)	**88.33%** (400)