

Face Recognition on Smartphones Via Optimised Sparse Representation Classification

Yiran Shen^{†‡}, Wen Hu[‡], Mingrui Yang[‡], Bo Wei^{†‡}, Simon Lucey[‡] and Chun Tung Chou[†]

[†]School of Computer Science and Engineering
University of New South Wales, Sydney, Australia
Email: {yrshen, bwei, ctchou}@cse.unsw.edu.au

[‡]CSIRO Computational Informatics, Brisbane, Australia
Email: {wen.hu, mingrui.yang, simon.lucey}@csiro.au

Abstract—Face recognition is an element of many smartphone apps, e.g. face unlocking, people tagging and games. Sparse Representation Classification (SRC) is a state-of-the-art face recognition algorithm, which has been shown to outperform many classical face recognition algorithms in OpenCV. The success of SRC is due to its use of ℓ_1 optimisation, which makes SRC robust to noise and occlusions. Since ℓ_1 optimisation is computationally intensive, SRC uses random projection matrices to reduce the dimension of the ℓ_1 problem. However, random projection matrices do not give consistent classification accuracy. In this paper, we propose a method to optimise the projection matrix for ℓ_1 -based classification¹. Our evaluations, based on publicly available databases and real experiment, show that face recognition based on the optimised projection matrix can be 5-17% more accurate than its random counterpart and OpenCV algorithms. Furthermore, the optimised projection matrix does not have to be re-calculated even if new faces are added to the training set. We implement the SRC with optimised projection matrix on Android smartphones and find that the computation of residuals in SRC is a severe bottleneck, taking up 85-90% of the computation time. To address this problem, we propose a method to compute the residuals approximately, which is 50 times faster but without sacrificing recognition accuracy. Lastly, we demonstrate the feasibility of our new algorithm by the implementation and evaluation of a new face unlocking app and show its robustness to variation to poses, facial expressions, lighting changes and occlusions.

Keywords—Face Recognition, Smartphones, Random Matrices, Sparse Representation, Android, JavaCV/OpenCV, Face Unlocking

I. INTRODUCTION

Face recognition is an important research problem in computer vision. With the availability of Open Source Computer Vision (OpenCV: opencv.org) on both Android and iOS, face recognition has also found its way into many smartphone apps such as people tagging and mobile gaming. In fact, we have observed more than 500 Android apps making use of face recognition. As an example, a recent update of Android includes a face unlocking app which replaces traditional password-access control with face recognition.

There are three face recognition algorithms in OpenCV: EigenFace [1], FisherFace [2] and LBPFace [3]. All these three algorithms use feature extractions (where different algorithms

use different set of features) followed by the nearest neighbourhood classifier (NNC) to match the test image with a best fit image in the training set. Recently, Wright, et al. [4] propose a new face recognition classifier based on sparse representation classification (SRC). SRC introduces a few new ideas into the face recognition algorithms. First, SRC uses image pixels directly which means feature selection is not required. Second, it uses ℓ_1 optimisation to make face recognition *robust* to lighting changes and occlusion. SRC has been shown in [4] to outperform NNC-based algorithms such as EigenFace and FisherFace. However, although ℓ_1 optimisation makes SRC robust, it introduces intensive computational cost [5]. In order to reduce the processing time while preserving the accuracy of recognition, SRC uses random projection matrices to reduce the dimensionality of ℓ_1 minimisation. In particular, SRC uses random Bernoulli or Gaussian matrices because of their information preserving properties, inspired by the recent theory of compressive sensing (CS) [6], [7]. We will refer to the SRC algorithm based on random projection matrices as rand-SRC.

Although random projection matrices can significantly reduce the computation time of SRC, the classification accuracy of rand-SRC is *not* consistent. In fact, accuracy of rand-SRC can change by up to 15% depending on the random projection matrix used. In this paper, we propose a novel strategy to optimise the projection matrix to remove the variability in the performance of SRC. We will refer to SRC that uses an optimised projection matrix as opti-SRC. In addition to providing consistent classification accuracy, opti-SRC is, respectively, 9-12% and 5-17%, more accurate than rand-SRC and the OpenCV algorithms. In other words, for a given level of accuracy, opti-SRC requires less number of projections compared to rand-SRC. From a smartphone point of view, this translates to a reduced computational resource requirements for opti-SRC. The contributions of this paper are:

- We propose a novel strategy to optimise the *row coherence* of a projection matrix while preserving low *mutual coherence*. The resulting combinatorial optimisation problem has a large search space and we propose two efficient off-line heuristics based on greedy and tabu search. The SRC algorithm based on the optimised projection matrix significantly outperforms its counterpart based on random projection matrices as well as face recognition algorithms in OpenCV. The optimised projection matrix is also robust in the sense that it does not have to be re-

¹Wen Hu was a visiting researcher at SICS (Swedish ICT) during the course of this work

optimised even if new faces are added to the training set after optimisation. We also show that our optimised projection matrix outperforms other existing methods of optimising projection matrices [8], [9].

- We implement opti-SRC on Android platforms to evaluate its efficiency. We find that the computation of residuals in SRC is a severe bottleneck, taking up 85–90% of the computation time. To address this problem, we propose a method to compute the residuals approximately. The method reduces the residual computation time by 50-fold while maintains the classification accuracy of opti-SRC.
- We evaluate the efficiency, in terms of computation time and energy consumption, of opti-SRC on three different smartphone models. We find the efficiency of opti-SRC is comparable to the OpenCV algorithms.

The organisation of this paper is as follows. We provide a brief introduction to SRC and matrix coherences in Section II. In Section III, two challenges and solutions on the design of face recognition system on smartphones are discussed. Section IV evaluates the performance of opti-SRC using two publicly available databases. Section V evaluates opti-SRC on different smartphone platforms. Section VI discusses related work and Section VII concludes the paper.

II. TECHNICAL BACKGROUND

In this section, we introduce the rand-SRC face recognition algorithm in [4] and discuss the roles of matrix coherences in sparse representation.

A. Sparse Representation Classifier (SRC)

In [4], the authors formulate the face recognition as a sparse representation problem computed via ℓ_1 optimisation. The formulation uses a random projection matrix for dimensionality reduction. The steps of rand-SRC are:

(i) **Dictionary and Sparse Representation** To model face recognition as a sparse representation problem, one needs to first build a dictionary \mathcal{D} . We assume there are K subject classes and T training images per class. All the images used should be scaled into the same size. Each training image consists of p pixels and is vectorised into a p -dimensional column vector. We then assemble the vectorised training images of the i^{th} subject in a $p \times T$ sub-dictionary D_i . A $p \times KT$ dictionary $\mathcal{D} = [D_1, D_2, \dots, D_K]$ is then formed from the K classes. Let y denotes a vectorised test image, then its representation under the dictionary \mathcal{D} is obtained by solving the following linear equation with the knowledge of y and \mathcal{D} :

$$y = \mathcal{D}\theta \quad (1)$$

where the unknown vector θ contains $n = KT$ unknowns which is equal to the number of columns in \mathcal{D} . If the vectorised test image y belongs to the k^{th} class, then *ideally* y is within the space spanned by the T vectors in D_k class and independent of the other classes. If the ideal condition holds, then the representation vector θ for y has the form:

$$\theta = [0, 0, \dots, \alpha_{k,1}, \alpha_{k,2}, \dots, \alpha_{k,T}, \dots, 0, 0, \dots, 0]^T \quad (2)$$

where \cdot^T denotes the matrix transpose, and the non-zero elements appear only in those positions related to the k^{th} class in \mathcal{D} . If the number of classes K is large, then θ is a *sparse* vector if the ideal condition holds.

(ii) **Random Projections** Because the dimension p of the image vectors is huge, solving Equation (1) can be computationally expensive. A random projection matrix can be applied to improve the computational efficiency while preserving recognition accuracy. The random projection matrix Φ in this application is generated from a Gaussian distribution with zero mean and unit variance [4]. Incorporating an $m \times p$ Gaussian matrix Φ in equation (1), we have

$$\Phi y = \Phi \mathcal{D} \theta. \quad (3)$$

where $m \ll n$ makes the systems of linear equations under-determined. Since we are looking for a sparse representation θ , we aim to solve the following ℓ_0 optimisation problem

$$\hat{\theta} = \arg \min \|\theta\|_0 \quad \text{subject to } \Phi y = \Phi \mathcal{D} \theta \quad (4)$$

where $\hat{\theta}$ is the sparse representation of y under dictionary \mathcal{D} and $\|\cdot\|_0$ represents the ℓ_0 norm, which counts the number of non-zero coefficients in $\hat{\theta}$. We remark that a basic requirement is that each test image must have a *unique sparsest representation* under training set \mathcal{D} ; this will be discussed further in Section II-B2. The optimisation problem (4) is *NP-hard* [10], which means no known algorithms can solve the problem within polynomial time.

(iii) **ℓ_1 Optimisation** Inspired by the recent theory of CS, the solution of ℓ_0 optimisation in Equation (4) can be well approximated by the following ℓ_1 optimisation problem,

$$\theta_{\text{opt}} = \arg \min \|\theta\|_1 \quad \text{subject to } \|\Phi y - \Phi \mathcal{D} \theta\|_2 < \epsilon \quad (5)$$

where ϵ is a small positive value used to account for noise. The solution θ_{opt} from the ℓ_1 optimisation is used in the following classification procedure.

(iv) **Minimal Residual for Classification** After obtaining the coefficient vector θ_{opt} , we can determine the class of the test vector y by using residuals. The residual for class i is:

$$r_i = \|y - \mathcal{D}_i \theta_{\text{opt}}^{(i)}\|_2 \quad (6)$$

where $\theta_{\text{opt}}^{(i)}$ is a T -dimensional vector containing the T elements in θ_{opt} related to class i . Then the final classification is determined by

$$\hat{i} = \arg \min_{i=1,2,\dots,K} r_i, \quad (7)$$

i.e., the class having the minimal residual among all classes.

B. Random Matrices and Coherence

Though random projection matrices can significantly reduce the dimension of the optimisation problem, they introduce substantial variation in classification accuracy too. Moreover, random projection matrices are not optimal. We will show later on that an optimised projection matrix can significantly improve the classification accuracy of SRC. We will optimise projection matrix using row coherence and this section provides background on matrix coherences.

1) *Matrix Coherences*: We consider the coherences of the sensing matrix A , where $A = \Phi\mathcal{D}$, i.e. the product of the projection matrix Φ and dictionary \mathcal{D} .

Mutual Coherence Let a_i and a_j denote the i^{th} and j^{th} columns of A respectively. The mutual coherence $\mu(A)$, which is the coherence between columns, is defined as:

$$\mu(A) = \max_{i < j} \frac{|a_i^T a_j|}{\|a_i\|_2 \|a_j\|_2}. \quad (8)$$

In words, $\mu(A)$ is the maximum absolute value of cross-correlations between the columns of A which stays in between 0 and 1.

Row Coherence Let \tilde{a}_i and \tilde{a}_j denote the i^{th} and j^{th} rows of A respectively. The row coherence is defined as:

$$\nu(A) = \max_{i < j} \frac{|\tilde{a}_i \tilde{a}_j^T|}{\|\tilde{a}_i\|_2 \|\tilde{a}_j\|_2}, \quad (9)$$

which is the maximum absolute value of cross-correlations between the rows of the matrix A .

Mutual coherence has been well studied in literature [11], [12], while row coherence has not attracted as much attention. In fact, *both* mutual coherence and row coherence affect the classification accuracy of SRC as discussed in Section II-B2.

2) *The Importance of Coherences*: A basic requirement for SRC to work is that each test image must have a unique sparsest representation under training set \mathcal{D} in (4). According to [13], the uniqueness of sparsest representation requires the following two conditions to hold:

$$\mu(A) \leq \frac{c_0}{\log n}, \quad s \leq \frac{c_0 n}{\log n \cdot \|A\|_2^2} \quad (10)$$

where $\mu(A)$ is the mutual coherence, c_0 is a constant and s is the sparsity of θ . $\|A\|_2$ is the 2-norm of the sensing matrix (also known as spectral norm) which equals to the largest singular value of A . These two conditions guarantee that a unique sparsest representation exists with a probability of $1 - O(n^{-1})$.

The second condition of (10) can be rewritten as:

$$\frac{1}{\|A\|_2^2} \geq \frac{\log n \cdot s}{c_0 n} \quad (11)$$

which means a smaller spectral norm for A makes it easier to satisfy this condition.

We now impose the standard condition that the columns of A has unit norm. By applying the theory of tight frames [14], the spectral norm of the $m \times n$ matrix A (which equals to the redundancy of A) has the property:

$$\|A\|_2^2 \geq n/m \quad (12)$$

where the equality holds if the rows of the matrix A are orthogonal to each other or in other words, the row coherence of A is zero. This inequality suggests that a way to reduce $\|A\|_2$ is to increase the number of projections m . Alternatively, we can reduce $\|A\|_2$ by making the matrix A more orthogonal by reducing its row coherence. This will be the approach taken by this paper. In particular, we propose a method to minimise the row coherence while preserving low mutual coherence. Intuitively, a sensing matrix with a lower row coherence means

there is less overlap in the information provided by different projections. In fact, we will show a lower row coherence leads to better classification accuracy.

III. CHALLENGES AND PROPOSED SOLUTIONS

There are a number of challenges to produce robust and efficient face recognition via SRC on smartphones. We will point out the challenges and present the proposed solutions.

A. Challenge I: Random Matrices

As mentioned above, the recognition accuracy of rand-SRC varies substantially with different random projection matrices. To solve this challenge, we propose an approach to optimise the projection matrix by minimising the row coherence while preserving relatively low mutual coherence of the sensing matrix. Moreover, opti-SRC requires a smaller number of projections compared to rand-SRC to achieve the same recognition accuracy. This means opti-SRC requires less computational requirement than rand-SRC on smartphones.

1) *Optimising Row Coherence in Finite Space*: From the discussion in Section II-B2, a good sensing matrix requires *both* low row coherence and mutual coherence. A naïve method is to minimise the row coherence alone. While such a method can reduce row coherence, it can at the same time increase the mutual coherence significantly, which is undesirable. To address this problem, we propose an optimisation approach in a finite feasible space to minimise the row coherence of the sensing matrix while preserving a low mutual coherence.

It is well understood that random sampling strategies, such as projection matrices Φ generated through either i.i.d. Gaussian or Bernoulli distributions, produce sensing matrices with relatively low mutual coherence [15].

Let us assume we have a finite (but still very large) set Ω of projection matrices stemming from either a random Gaussian or Bernoulli distribution. We know that any projection matrix Φ stemming from the set Ω will result in a sensing matrix A which with high probability has relatively low mutual coherence. Furthermore, we know from Equation (9) that the row coherence of A can be minimised by choosing the sampling matrix Φ that minimises

$$\arg \min_{\Phi} \nu(\Phi\mathcal{D}) \quad \text{subject to } \Phi \subseteq \Omega. \quad (13)$$

The strength of this strategy lies in its ability to directly reduce the row coherence of A while with high probability maintaining a relatively low mutual coherence. The high probability guarantees that the sensing matrix A will have low mutual coherence stems from the fact that Ω is finite. For example, if Ω contained the infinite set of all possible projection matrices drawn from a Gaussian distribution then there clearly exists (but is extremely unlikely when drawn from a random distribution) a sensing matrix that has zero row coherence, but unity mutual coherence. Since the objective in Equation (13) is searching for the Φ with the lowest row coherence of the sensing matrix in the infinite set, it would inevitably choose this projection matrix which would result in a sensing matrix A that has the high mutual coherence.

By constraining Ω to a finite set of randomly generated projection matrices one then guarantees with overwhelmingly

high probability that all projection matrices in the set will produce a sensing matrix with relatively low mutual coherence. It is clear that this probabilistic guarantee on the mutual coherence is a function of the size of the finite set Ω . It is also clear that since the set Ω is finite and random the set is not convex making the objective in Equation (13) non-convex. Further, the search space for this combinatorial optimisation problem is enormous (even though Ω is finite) and is in fact substantially complex in computation. We remark that there is another reason why we restrict ourselves to find the optimal projection matrix within the set of Bernoulli or Gaussian matrices, rather than the set of all matrices. This will be discussed in Section III-B.

2) *Efficient Solutions*: Because the optimisation problem (13) is NP-hard. We propose two efficient algorithms based on: (i) greedy, and (ii) tabu search to minimise the objective although neither strategy is guaranteed of finding the global minima.

For both strategies we explored the role of the set size Ω versus performance of recognition accuracy. We found that the larger the chosen set the better the recognition accuracy so in all our experiments we chose the largest finite set Ω possible for a tractable solution. This result is not entirely unexpected due to the strong probabilistic guarantees of a random sampling forming sensing matrices with good mutual and row coherence.

Algorithm 1 Greedy Projection Matrix Optimisation

- 1: Input: Search space \mathcal{S}_n , Dictionary \mathcal{D} , the number of rows of the projection matrix m .
 - 2: Initialisation: traverse the search space to find two row vectors $\hat{\phi}_1$ and $\hat{\phi}_2$ such that their coherence is minimal, $\hat{\Phi} = \{\hat{\phi}_1, \hat{\phi}_2\}$, $\mathcal{S}_n = \mathcal{S}_n \setminus \{\hat{\phi}_1, \hat{\phi}_2\}$, $i = 2$
 - 3: **while** $i \leq m$ **do**
 - 4: $\hat{\phi}_j = \arg \min_{\hat{\phi}_j \in \mathcal{S}} \max_{\hat{\phi}_k \in \hat{\Phi}} |(\hat{\phi}_j \mathcal{D})^T (\hat{\phi}_k \mathcal{D})|$
 - 5: // $\hat{\phi}_j$ and $\hat{\phi}_k$ are two rows from \mathcal{S} and $\hat{\Phi}$ respectively
 - 6: $\hat{\Phi} = \hat{\Phi} \cup \{\hat{\phi}_j\}$
 - 7: $\mathcal{S}_n = \mathcal{S}_n \setminus \{\hat{\phi}_j\}$
 - 8: $i++$
 - 9: **end while**
 - 10: Output: optimised projection matrix formed by the m vectors in $\hat{\Phi}$
-

Greedy Algorithm (Algorithm 1) The greedy algorithm starts with searching for two rows with the minimal coherence (Line 2). This is achieved by traversing the whole search space. The search space \mathcal{S}_n in Line 1 is a large finite set of projection vectors randomly generated from Gaussian distributions. In the subsequent iterations, it chooses the vector from the remaining vectors that minimises the maximum row coherence between this vector and the ones already chosen (Lines 4–7). When the desired number of vectors is reached (Line 3), the iteration is terminated. The chosen vectors then form the projection matrix (Line 10).

Tabu Search Algorithm (Algorithm 2) The solution of the greedy algorithm is known to converge to a local optimum which heavily depends on the starting point. This local optimum could be significantly different from the global optimum. To address this problem, we propose another algorithm based on tabu search [16] to find an improved and robust solution.

Algorithm 2 Tabu Search Projection Matrix Optimisation

- 1: Input: Dictionary \mathcal{D} , search space \mathcal{S}_n and number of rows of the projection matrix m .
 - 2: Initialisation: allocate four empty lists: L_{in} , L_{out} , \bar{L} and \hat{L} , stop criteria $StabilityLimit = 500$, $BestCov = Inf$, $i = 0$
 - 3: **while** $i < StabilityLimit$ **do**
 - 4: $CurrentCov = Inf$, $j = 0$
 - 5: Randomly choose m vectors to form $\hat{\mathcal{P}}_m$, record the indices list of $\hat{\mathcal{P}}_m$ as \hat{L} ($\hat{L} \cap L_{out} = \emptyset$, $\hat{L} \cap L_{in} = L_{in}$). The lists are first-in-first-out;
 - 6: **while** $j < m$ **do**
 - 7: $\hat{\mathcal{P}}_{m-1} = \hat{\mathcal{P}}_m \setminus \{\hat{\mathcal{P}}_m\{j\}\}$
 - 8: $\mathcal{S}_{n-m+1} = \mathcal{S}_n \setminus \hat{\mathcal{P}}_{m-1}$
 - 9: $Cov_j = \min_{s_d \in \mathcal{S}_{n-m+1}} \max_{p_k \in \hat{\mathcal{P}}_{m-1}} |(p_k \mathcal{D})^T s_d \mathcal{D}|$
 - 10: // s_d and s_k are rows from \mathcal{S}_{n-m+1} and $\hat{\mathcal{P}}_{m-1}$ respectively.
 - 11: **if** $Cov_j < CurrentCov$ **then**
 - 12: $CurrentCov = Cov_j$, $PotentialObj = \hat{\mathcal{P}}_m\{j\}$
 - 13: Store indices of current solution $\hat{\mathcal{P}}_m$ in \bar{L}
 - 14: **end if**
 - 15: $j++$
 - 16: **end while**
 - 17: **if** $CurrentCov > BestCov$ **then**
 - 18: $BestCov = CurrentCov$; $BestList = \bar{L}$
 - 19: Clear L_{out} , push $PotentialObj$ in L_{in}
 - 20: $i = 0$
 - 21: **else**
 - 22: Push $\hat{\mathcal{P}}_m$ in L_{out}
 - 23: $i++$
 - 24: **end if**
 - 25: **end while**
 - 26: Outputs: The indices list of the optimised projection matrix in $BestList$ chosen from the search space \mathcal{S}_n
-

Tabu search is metaheuristic algorithm for combinatorial optimisation problems. It enhances the performance of the local search algorithm by exploring the unreached area of the greedy algorithms. The computation complexity of tabu search is $O(n^2)$ [17] which means it can be solved in polynomial time.

Tabu search algorithm utilises memory structures by defining the neighborhood of the current solution and two tabu lists. The size of the neighbourhood can be as large as the size of the search space. It can be tuned according to the processing capability of the devices. The tabu lists store the specific atoms. In our case, one tabu list stores the row indices which can not be visited for a specific number of iterations and the other list stores the row indices that cannot be removed from the solution for another number of iterations. The number of iterations are determined by the size of the tabu lists. The tabu lists should be large enough to avoid cycles and we tuned them as half of the size of search space and the solution respectively. Details of the algorithm are shown in Algorithm 2.

This algorithm aims to pick m feasible rows (to form the projection matrix) in the search space to minimise the row coherence of the sensing matrix. Lines 1-2 represent the initial setup which specifies some parameters and creates two empty tabu lists. The proper algorithm starts in Line 3 by randomly choosing m rows in the search space satisfying the constraints

of the tabu lists. This loop will continue searching for an optimal solution until the solution has not been updated by a number of *StabilityLimit* iterations. During the inner loop presented in Lines 6-16, it updates one of the rows in the chosen matrix by one row in the defined neighbourhood to find the local minimum (this is called *Intensification* stage). When the new minimum is larger than the one found previously, the algorithm explores the unreached area in the next iteration in *Diversification* stage (Lines 17-23); otherwise, when *Aspiration Level condition* (Line 17) is satisfied, it will reset the status of the algorithm (counter of the iteration, tabu list). The algorithm will terminate when the minimum has not been updated by a number of *StabilityLimit* (in Line 3) iterations. *StabilityLimit* should be large enough to guarantee the robustness of the algorithm. During the search progress, the algorithm will not search the recently reached area due to constraints of the tabu lists. By these approaches, tabu search can avoid converging to a local minimum by exhaustively searching the new area.

With the greedy algorithm or tabu search algorithm, we can obtain an optimised projection matrix. We replace the random matrix used in SRC approach with the optimised matrix to improve the performance of recognition accuracy. To distinguish SRC with different matrices, we call original SRC as rand-SRC and our approach as opti-SRC.

B. Challenge II: Residual Calculation

The SRC in [4] was designed to solve the face recognition problem assuming abundant computing power is available. However, if the algorithm is to be implemented on embedded systems (e.g., smartphones), we will need to optimise the computations in SRC to improve its efficiency. Our evaluation of SRC on various smartphone platforms shows that the computation of residuals in Equation (6) is a severe bottleneck, accounting for 85–90% of SRC’s running time. The computation of residual r_i in Equation (6) is of the order $\mathcal{O}(p + pT)$ where in particular p is the number of pixels in the training image, which is generally a large number. Instead of computing the exact r_i using (6), we propose to compute an approximate r_i by making use of the **Johnson-Lindenstrauss (J-L) Lemma** [18]. Let Φ be the $m \times p$ (with $m \ll p$) projection matrix used in the ℓ_1 optimisation, which is either Gaussian or Bernoulli distributed, the J-L Lemma provides an approximation of the residual by:

$$\|y - \mathcal{D}\theta_{\text{opt}}^{(i)}\|_2 \approx \|\Phi y - \Phi \mathcal{D}\theta_{\text{opt}}^{(i)}\|_2. \quad (14)$$

Since both Φy and $\Phi \mathcal{D}$ are used in the ℓ_1 optimisation problem (5), they have already been computed. Also, both of them have m rows, so the computation on the right-hand side of (14) has a complexity of $\mathcal{O}(m + mT)$, which is much lower because $m \ll p$. We will refer to the computation of residuals using the right-hand side of (14) as *compressed residuals*.

Note that the validity of the J-L Lemma requires that the projection matrix is either Gaussian or Bernoulli distributed. This is another reason why we have restricted the optimisation problem in Section III-A1 to search for a Gaussian or Bernoulli projection matrix.

The classification is then carried out using the compressed residuals after they have been computed. Our experiments

show that the recognition accuracy of using compressed residuals is **exactly the same** as that of using full residual calculations when sufficient number of projections are used.

IV. PERFORMANCE EVALUATION ON PUBLICLY AVAILABLE DATABASES

A. Goals, Metrics and Methodology

The aim of this section is to evaluate the classification accuracy of opti-SRC, and compare it with rand-SRC as well as three face recognition algorithms in OpenCV: EigenFace, FisherFace and LBPFace. The comparison makes use of two publicly available databases: the Extended Yale database B [19] and the AT&T database [20]. Both databases are popular choices [4], [21] to evaluate the recognition accuracy of face recognition algorithms.

For fair comparison between different algorithms, a common parameter m is used as the number of projections in the two SRC-based algorithms as well as the number of features in the three algorithms in OpenCV. For the SRC-based algorithms, projections play the same role as features.

For opti-SRC, compressed residuals (14) are used. The classification results for using full residual calculations (6) is identical to that of using compressed residuals and are therefore not shown.

We compare opti-SRC against two existing methods for optimising projection matrix. We also investigate the tolerance of opti-SRC to training set changes. The evaluation in this section is carried out on a computer. Experimental results on smartphones will be presented in the next section.

In this paper, we use the percentage of correct recognition as the performance metric (same as [4]) for the recognition evaluation, which is simply the number of true recognitions over the total number of tests. We express the recognition accuracy of different methods under different number of projections/ features and plot the recognition accuracy curves of different face recognition methods.

B. Extended Yale Database B

The Extended Yale Database B (Ext-YaleB) contains $K = 38$ subjects under 9 poses and 64 illumination conditions. We choose the first $T = 32$ images from each subject as the training images and the following 10 images forms the test set. We remark that our choice of training images from each subject is different from [4], where 32 images were chosen randomly from each subject. Because the images are sorted by the time that they are captured, our choice is more consistent with the face recognition practice on smartphones where training images are added well before the test images. We have reproduced the results in [4] and observed that their choice tends to produce a better recognition accuracy because it distributes more similar face images into the training and test sets (empirically, images of one subject taken on temporal contiguity share more information. They are more likely to be classified into the same class).

The evaluation in [22] shows that face recognition can tolerate some degree of resolution reduction. We scale the images from the original size of 192×168 to 96×84 (i.e.,

4 times reduction), which does not affect the recognition accuracy. (We also try to reduce the resolution by 16 times, but the accuracy decreases significantly). This setting gives an dictionary \mathcal{D} of size $8,064 \times 1,216$. We choose to reduce the image resolution because of two considerations: first, the high resolution of the image will introduce intensive computation especially for the ℓ_1 solver; second, the resolution of the embedded cameras varies among the smartphones, a face recognition system using low resolution images can be readily used on different smartphones.

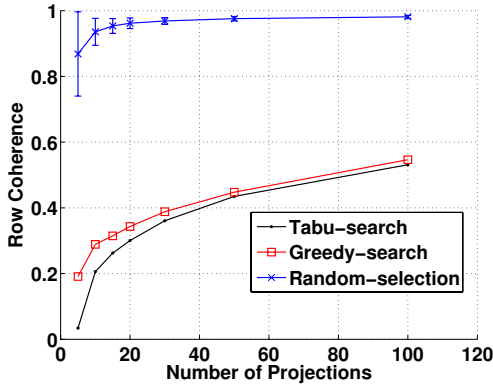


Fig. 1: YaleB: comparison of row coherence of the sensing matrices

1) *Coherence Analysis:* We compare the row coherence of the sensing matrix $\Phi\mathcal{D}$ where Φ comes from minimising (13) or is randomly generated. We vary the number of projections from 10 to 100. For each number of projection, we find an optimised Φ using greedy algorithm with the search domain Ω consisting of Gaussian distributed vectors and compute the row coherence of the sensing matrix using the optimised Φ . We repeat the process with tabu search. We also generate 30 random Gaussian matrices, and calculate the mean and standard deviation of their row coherence for the corresponding sensing matrices. The results are plotted in Figure 1. The x -axis is the number of projections from 10 to 100 and the y -axis is the row coherence of the sensing matrix. Figure 1 shows that tabu search provides sensing matrices with the smallest row coherence. Our proposed greedy and tabu search algorithms can reduce the row coherence of the sensing matrix by up to 4 times when the number of projections is 10 and tabu search performs better than greedy search. Moreover, our algorithms can provide more consistent row coherence while the random approach shows a large variance (error bars in the figure). We have also calculated the mutual coherence of the sensing matrix with the optimised projection matrix. The result in Figure 2 shows that our optimisation approach will not increase the mutual coherence (actually it reduces the mutual coherence a bit). Note that for face recognition, \mathcal{D} is a dictionary consisting of face image vectors as its columns. The faces in the same class can be very similar to each other. So the mutual coherence can be high (over 0.99 in random selection approach). The large row coherence in Figure 1 arises for a similar reason: images have high spatial coherence. The adjacent pixels tend to have similar pixel values which lead to high row coherence.

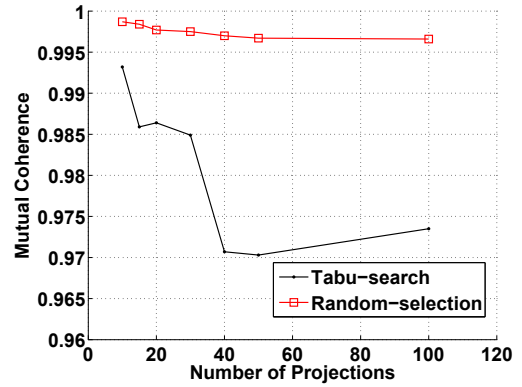


Fig. 2: YaleB: comparison of mutual coherence of the sensing matrices

2) *Recognition Accuracy:* To prove our strategy on minimising the coherence of the sensing matrix gives good face recognition accuracy, we compare opti-SRC (with optimised projection matrix from greedy and tabu search respectively), rand-SRC and three algorithms from OpenCV. The results from the rand-SRC are represented by the average and standard deviation over 30 independent trials where a different random projection matrix is used in each trial. In Figure 3, the x -axis represents the number of features/projections ranging from 20 to 200 and the y -axis is the recognition accuracy. The results show that opti-SRC with projection matrix from tabu and greedy search algorithms improves the recognition accuracy by up to 12% compared with the best of the other algorithms, and opti-SRC with projection matrix from tabu search performs better than that from greedy search. The results give strong evidence that our strategy of minimising the row coherence of the sensing matrix delivers a face recognition algorithm with consistently high accuracy. From a different point of view, for a given accuracy, opti-SRC needs less projections than rand-SRC; this translates to less computation resource requirement for opti-SRC.

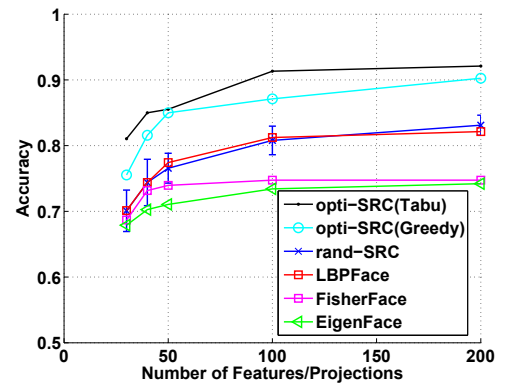


Fig. 3: YaleB: comparison of face recognition accuracy of different methods

3) *Stability to Training Set Evolvment:* For many face recognition applications, new classes are constantly added to the training set. This means the dictionary \mathcal{D} evolves constantly

over time. Since the projection matrix Φ is optimised for a given dictionary \mathcal{D} (see Equation (13)), we investigate whether it is necessary to re-optimize Φ if new classes are added to the dictionary.

For this evaluation, we begin with an initial training set containing 16 classes. We use tabu search to obtain an optimised projection matrix based on the initial dictionary. We then add training classes to the dictionary: 4 classes at a time until 36 classes and then 2 additional classes are added to reach 38 classes. We consider two different strategies. The first strategy, which we call opti-SRC-fixed, uses the optimised projection obtained from the initial dictionary with 16 classes even when the dictionary evolves. The second strategy, called opti-SRC-adaptive, re-optimises the projection matrix each time new classes are added to the dictionary. The number of projections is kept at 100.

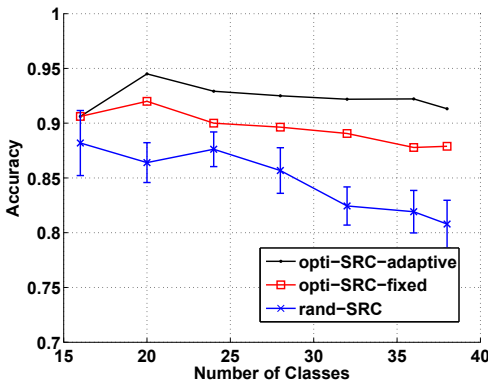


Fig. 4: YaleB: training set evolution

Figure 4 compares the classification accuracy of opti-SRC-adaptive, opti-SRC-fixed and rand-SRC. It is not surprising that opti-SRC-adaptive gives the best performance while rand-SRC is the worst. The interesting observation is that opti-SRC-fixed gives fairly stable performance in spite of the increase in number of classes. In fact, the performance of opti-SRC-fixed and opti-SRC-adaptive differs by at most 3%. Moreover, part of the degradation in the performance of opti-SRC-fixed is due to the use of the same number of projections as the number of classes increases. This is a very encouraging result which shows that re-optimisation of projection matrix is only required occasionally. Note that there is performance improvement when the number of classes increases from 16 to 20. One explanation is the newly added test images bear some similarity to those in the dictionary.

4) *Comparison with Other Projection Matrix Optimisation Approaches:* The idea of computing a projection matrix that is optimally designed for a certain signal class is not new. Elad [9], and Duarte-Carvajalino and Sapiro [8] recently proposed strategies for learning projection matrix directly from a dictionary to improve the performance of signal recovery in CS. The algorithm in [9] aims to find a projection matrix Φ to reduce the mutual coherence of the sensing matrix $A = \Phi\mathcal{D}$ while that in [8] finds a Φ so that $A^T A$ is close to the identity matrix. There is another work on designing projection matrix with orthogonal rows. However, the “orthoprojectors” proposed in [23] aim to make the rows of projection matrix

orthogonal. They do not take the dictionary into account. Therefore, we only compare with the approaches in [9] and [8]. For this comparison, we use the optimised projection matrices from these two algorithms for face recognition; note that residual calculations use (6) because it is not known whether these optimised projection matrices satisfy the J-L Lemma. The comparison is against opti-SRC based on tabu search and rand-SRC. The results are shown in Figure 5, which shows that the optimised projection matrix from [9] and [8] cannot improve the recognition accuracy. It is probably because the requirements on Φ for signal recovery and classification are different.

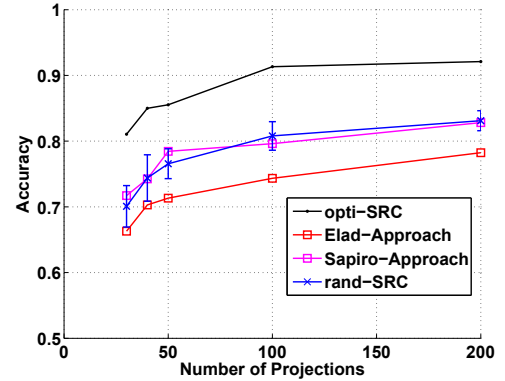


Fig. 5: YaleB: comparison of different optimised matrices

C. AT&T Face Database

AT&T (ATT) database is another commonly used database for evaluating face recognition algorithms. There are 40 subjects in the database and each subject contributes 10 face images. The database contains various lighting conditions, facial expression (open/closed eyes, smiling/neutral) and eye wear (glasses/no glasses). The size of each image is 92×112 .

We use the first half of the images of each subject as training and the rest as test. Therefore, the training set \mathcal{D} is a matrix of size $10,304 \times 200$. The evaluations conducted for this database is the same as those in Section IV-B.

1) *Coherence Analysis:* For coherence analysis, we compare the performance of the two algorithms (tabu and greedy search) on reducing the row coherence of the sensing matrix with the ATT database. The results are shown in Figure 6 which demonstrate that both algorithms reduce the row coherence dramatically.

2) *Performance Evaluation:* Figure 7 compares the recognition accuracy of opti-SRC (greedy and tabu search), rand-SRC and three algorithms from OpenCV. Our proposed approach achieves consistently better recognition accuracy. The improvement is up to 9% compared with the second best algorithm.

3) *Stability to Training Set Evolution:* We also check the stability of our approach to the evolution of training set using the ATT database. As shown in Figure 8, when the number of classes is growing from 16 to 40, the performance of opti-SRC-fixed is comparable to opti-SRC-adaptive, while both of these algorithms perform much better than rand-SRC.

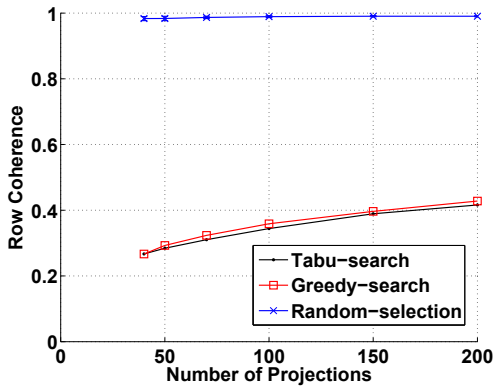


Fig. 6: ATT: comparison of row coherence of the sensing matrices

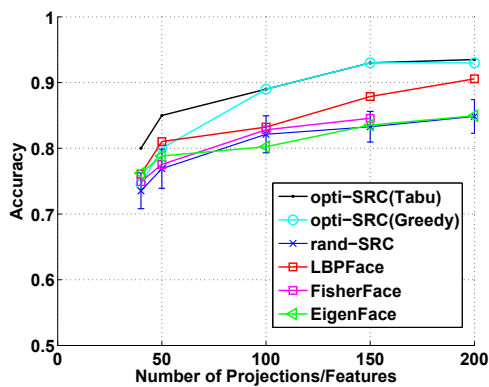


Fig. 7: ATT: comparison of face recognition accuracy of different methods

D. Conclusions of this Section

From the results of this section, we can conclude that 1) opti-SRC has significantly better recognition accuracy compared with the state-of-the-art face recognition algorithms; 2) the optimised projection matrix can tolerate the evolution of the training set which makes it more convenient to use on smartphones; 3) projection matrix derived from tabu search provides consistently better recognition accuracy than greedy search. Therefore we apply tabu search in our experiments on smartphone to optimise the projection matrix.

V. EXPERIMENT ON SMARTPHONES

A. Experiment Description

The aim of this section is to evaluate the classification accuracy of opti-SRC, and compare it with rand-SRC as well as the OpenCV algorithms on the smartphones. Ten subjects, 7 males and 3 females, with different skin tones are recruited for the experiment.² We take 40 photos of each subject in random poses, lighting conditions and expressions. We then

²Ethical approval for carrying out this experiment has been granted by the University New South Wales (Reference Number 08/2013/72) and CSIRO (Reference number 083/13)

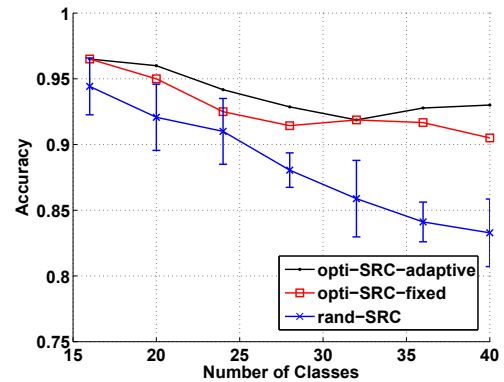


Fig. 8: ATT: training set evolution

use the Android OS *face-detection* API to automatically crop the face region of the images to a size of 92×56 and store them on the phones' memory. The resolution of the face images is much lower than that of the embedded cameras on off-the-shell smartphones. We will refer to this database as *Private database*.

We implement opti-SRC on the Android platform. The ℓ_1 minimisation uses the ℓ_1 -homotopy solver [24] because it is much more efficient than linear programming: complexity of $\mathcal{O}(s^3 + smn)$ (where s is the sparsity of the solution, with $s \ll n$) versus $\mathcal{O}(n^3)$.

In this section, we present experimental results on computation time and resource consumption (Section V-D), and recognition accuracy (Section V-E) of opti-SRC and OpenCV algorithms on the chosen smartphone platforms. We have also designed a group face unlocking app (Section V-F) with the opti-SRC and compared it against the Android built-in face unlocking app on under various different conditions to test its robustness.

B. Platforms



Fig. 9: Android smartphones used in the experiment; from left to right: Samsung Galaxy S3, Google Nexus 4 and HTC OneX

In order to demonstrate the feasibility of opti-SRC, we implement it on three different popular smartphones: Samsung Galaxy S3, Google Nexus 4, and HTC OneX (shown in

Figure 9). The specifications of these smartphones relating to the application are shown in Table I.

TABLE I: Smartphones specifications relating to the face recognition application

	Galaxy S3	Nexus 4	HTC OneX
CPU	Quad 1.4 GHz	Quad 1.5 GHz	Quad 1.5 GHz
Storage	16 GB	8 GB	16 GB
RAM	1GB	2 GB	1GB
Screen	4.7 inches	4.7 inches	4.7 inches
Battery	2, 100 mAh	2, 100 mAh	1, 800 mAh
Android OS	v4.0.4	v4.2.2	v4.1.1

C. Preliminary experiments

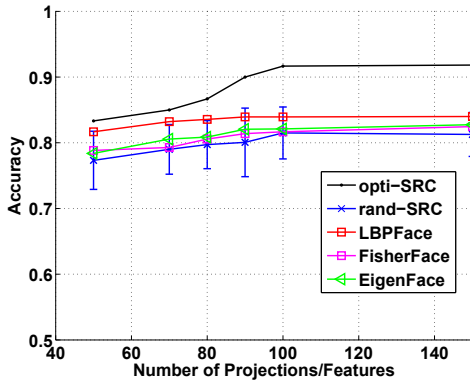


Fig. 10: Face recognition accuracy comparison under different number of projections/features with private database

1) *Number of Projections*: We conduct an experiment using Private database A on a computer to choose a suitable number of projections/features for the face recognition algorithms. This experiment is conducted in the same way as those in Section IV where we study the effect of the number of projections on the recognition accuracy. We randomly pick 20 images from each subject to form the training set (we call this *Private-Training*) and the rest of the images form the test set. Therefore, the dictionary is a matrix of size $5,152 \times 200$. The results are shown in Figure 10. We again find that opti-SRC outperforms the other algorithms significantly. The recognition accuracy of all the algorithms levels off after 100 projections. We therefore choose to use 100 projections for further experiments on the smartphones. The size of the projection matrix is $100 \times 5,152$.

2) *Compressed Residual*: According to challenge II in Section III-B, the computation of residuals using the full-size image vectors as in equation (6) is computationally intensive (complexity of $\mathcal{O}(p + pT)$ where $p = 5152$ is the size of the image) and we propose to approximate the residual calculations by equation (14) (complexity of $\mathcal{O}(m + mT)$ where $m = 100$ is the number of projections). We measure the computation time (per test image classified) of both methods of residual calculations on the smartphones and show the results in Table II. We see that, by using compressed residuals, the computation time is reduced by a factor of 45–50 times. This is consistent with the complexity expressions which predict a $\frac{p}{m} \approx 52$ times reduction.

Table III shows that the computation time of opti-SRC without the residual calculation component (i.e. face detection and ℓ_1 minimisation components only) is 200–350ms for the three platforms. If the original method of residual calculations is used, then it will take up 85-90% of the computation time, which is a severe bottleneck. By using compressed residuals, the total classification time per image is reduced by about 6 times.

TABLE II: Time for original/compressed residual calculations per face classification

	Galaxy S3	Nexus 4	HTC OneX
Compressed Residual	42 ms	33 ms	43 ms
Original Residual	1, 923 ms	1, 926 ms	2, 105 ms

D. Resource Consumption on Smartphones

Table III shows the resource consumption (running time and energy consumption) of opti-SRC on different smartphones. The running time is obtained from the console of Eclipse development environment and the energy consumption is evaluated using the PowerTutor (powertutor.org) app on Android.

TABLE III: Resource consumption of opti-SRC on different smartphones

	Galaxy S3	Nexus 4	HTC OneX
Face Detection (ms)	100	70	97
ℓ_1 (ms)	240	136	247
Residual (ms)	42	33	43
Total (ms)	382	239	387
Display Energy (mJ)	1, 500	1, 040	1, 300
CPU Energy (mJ)	175	139	246

In terms of processing speed, Table III shows that the SRC component (ℓ_1 minimisation + compressed residual calculation) is the dominant factor of time consumption which is about 3 times of the face detection component. Therefore, the use of compressed residual can drastically reduce the total processing time. Another noticeable result is that most of the energy is consumed by the display which is inevitable for visualisation of the applications. The CPU only takes 10-15% of the total energy consumption.

To provide a benchmark for comparison, we also implement the OpenCV based face recognition applications on the smartphones. Because the original OpenCV library was coded in C++, we use the Java Native Interface (JavaCV-v0.6) which is based on OpenCV-v2.4.6. The average processing time for detecting and recognising one face is shown in Table IV. ℓ_1 minimisation is known to be computational intensive. However, with the power of dimensionality reduction and compressed residual, opti-SRC achieves the same order

TABLE IV: Speed of face detection and recognition with feature based recognition methods in OpenCV

	Galaxy S3	Nexus 4	HTC OneX
EigenFace (total)	177 ms	143 ms	161 ms
FisherFace (total)	221 ms	159 ms	186 ms
LBPFace (total)	268 ms	220 ms	257 ms

of speed as the algorithms in OpenCV, which are known to be fast. It means that our algorithm is quite efficient for face recognition. Note that our ℓ_1 -homotopy implementation uses Java which is known to be less efficient than C++. The speed of classification can be further improved with C++ implementation but it is out of the scope of this paper.

E. Recognition Accuracy on Smartphones

In this section, we evaluate the recognition accuracy of opti-SRC, rand-SRC and OpenCV algorithms using a *new* test set. This new test set has 400 photos, 40 from each subject. (Note the 400 photos from this new test set are different from those in the Private database. These photos are taken at two different locations with lighting conditions: bright (above 100 lux) and dark (below 50 lux) respectively. The luminance can be measured by using the light sensor embedded on the smartphones. Twenty photos are taken for each lighting condition per subject. These 20 photos come from 20 different (*expression, pose*) pairs where expression is neutral, happy, sad or neutral with glasses; and pose is front, up, down, left or right as shown in Figure 11. All the photos are manually labelled by their lighting condition, expression and pose. This new test set presents a rich variety of situations to test the face recognition algorithms. Also, there is mobility when the experiments were conducted. During the experiments the subjects walk around in the lab and take pictures freely.



Fig. 11: Face samples of different poses

For this evaluation, the training set is Private-Training used in Section V-C1. Note that there are only 20 images per subject in the training set, but there are 40 images per subject in the test set. Therefore, the test set presents a much richer variations than the training set. The number of projections/features remains as 100.

TABLE V: The recognition accuracy of opti-SRC under different subsets of the test set.

Neutral	Happy	Sad	Glasses	Dark	Bright	Total
92%	89%	89%	90%	85%	95%	90%

Table V shows the recognition accuracy of opti-SRC using different subsets of the test set. For example, the accuracy under *Happy* means we take the 100 photos (= 10 subjects \times 2 lighting conditions \times 5 poses) in the test set with the label

Happy and test whether opti-SRC can recognise the subject in the photo correctly. For the accuracies under *Dark* and *Bright*, 200 test photos are used in each subset. From the results we can see that, opti-SRC is more sensitive to the change of expressions than the small occlusion (wearing glasses). This is reasonable because the change of expressions is related to much larger area of face than the small occlusion. Another observation is that the well-lighted environment can improve the recognition accuracy (about 10% improvement).

For the methods in OpenCV and rand-SRC, the classification accuracy of EigenFace, FisherFace, LBPFace and rand-SRC is only 81- 82% which is significantly lower than those of opti-SRC (90%).

F. Group Face Unlocking

Face unlocking is a built-in security feature in the recent Android OS release. It uses face, rather than password, for unlocking the smartphone. Here we design a new face unlocking app for smartphones based on opti-SRC. Different from the built-in face unlocking function, our app considers the situation that the smartphones can be shared by a small group of users. We call our app group face unlocking.

1) *Resistance to Intruders*: Because a security system should have "zero" tolerance to intruders, we include a criterion to validate the recognition results so that intruders are rejected. In our experiments, we choose 3 subjects as the "true" persons in the group unlocking apps and consider the other 7 subjects as intruders. There are $K = 3$ training classes where each class corresponds to a subject in the group.

We apply opti-SRC as before and compute the compressed residual for each class. Let r_j^c be the compressed residual of class j . If a person, say person 1, in the group is presented for recognition, we expect the residual r_1^c is much lower than r_2^c and r_3^c . We can define the confidence of recognition as:

$$confidence = \left(\frac{1}{K} \sum_{j=1}^K r_j^c - \min_{j=1, \dots, K} r_j^c \right) / \frac{1}{K} \sum_{j=1}^K r_j^c \quad (15)$$

The confidence is in the range $[0, 1]$ and should be close to 1 if a subject in the group is presented for recognition. On the other hand, if it is an intruder, the confidence should be close to 0.

The unlocking algorithm will first use opti-SRC to perform a recognition. After that, a confidence level is calculated and a person will be considered to be an intruder if the confidence level is lower than a pre-defined threshold. The threshold can be tuned according to the security level of the application. There is a trade-off between the security level and the recognition of true users: a higher threshold makes the system more resistant to intruders but true users are also more likely to be rejected. We use a threshold of 0.16 in the following experiment so that no false positives (an intruder is recognised as user) are shown with the test of the 7 intruders.

2) *Robustness to variations*: In this part, we evaluate the recognition accuracy of our group unlocking app under many different conditions to test its robustness. This is important because a good face unlocking app needs to recognise the right user under different facial expressions and environment

variations. We compare our group unlocking based on opti-SRC against the built-in face unlocking app under different variations, including lighting, distance, occlusion, expression and poses. During the evaluation, we carefully control the factors so that only one type of variation is present at a time. Unlike most of the Android OS, the built-in face unlocking is *closed source*, therefore we cannot use exactly the same face image to compare these two apps. Instead, we have to capture 2 images for each test and make these 2 images as similar as possible. We again use 20 face images for each class as the training set for opti-SRC and we train the face unlocking with the default conditions mentioned below.

Lighting Variation: The capability of recognition under different lighting conditions is the key for the face unlocking app because the users will use the app under different environment. The robustness to the lighting changes is evaluated in 5 locations with different illumination intensity which are around 5 lux, 20 lux, 100 lux, 200 lux and 800 lux respectively during the experiments. The default lighting illuminance under which the face unlocking is trained is around 100 lux.

Distance variation: The users may stand at different distances from the camera when they are being recognised. We gradually change the distance from 20–60 cm to evaluate the robustness to the distance changes. The default distance is around 40 cm.

Occlusion: We test the face unlocking apps for the different ratio of occlusions. The users wear either normal prescription glasses, headphones or large black sunglasses during the test. The faces in the training set do not contain any occlusions.

Expression Variation: Expression is another common variation when the face images are taken. The expressions of neutral, happy and sad are used during the test. The default expression is neutral.

Pose Variation: Pose changes should be addressed because direction of the faces to the camera cannot be always strictly frontal. To compare the robustness of the face unlocking apps to the variation of the pose, the users gradually change their pose of face from front to two directions (up, down) until the false recognition appears (the angles to the gravity direction can be measured with built-in sensor in smartphones). The front pose is used as default pose.

The evaluation results are shown in Table VI. The pose change limit is demonstrated by angles to the gravity direction. It is the largest angles that the face unlocking apps can recognise the users correctly. (We are not able to do experiments for the left and right poses because they cannot be directly measured. However, our experience shows that opti-SRC can tolerate larger pose angles in right and left directions.) The check/cross markers (\checkmark/\times) represent the face unlocking app recognises the face of users correctly/wrongly under the corresponding variance (true positive/false negative).

The results shown in Table VI demonstrate that our group unlocking app is more robust than the built-in app for lighting and pose changes and the two face unlocking apps achieve the same results on the expression, distance and occlusion changes.

TABLE VI: The comparison of our approach and built-in face unlocking apps on different variance conditions

lighting	5 lux	20 lux	100 lux	200 lux	800 lux
built-in	x	\checkmark	\checkmark	\checkmark	x
opti-SRC	\checkmark	\checkmark	\checkmark	\checkmark	x
distances	20 cm	30 cm	40 cm	40 cm	60cm
built-in	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark
opti-SRC	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark
occlusions	none	headphone	glasses	sunglasses	
built-in	\checkmark	\checkmark	\checkmark	x	
opti-SRC	\checkmark	\checkmark	\checkmark	x	
expressions	neutral	happy		sad	
built-in	\checkmark	\checkmark	\checkmark	\checkmark	
opti-SRC	\checkmark	\checkmark	\checkmark	\checkmark	
poses	up		down		
built-in	17°		40°		
opti-SRC	20°		36°		

VI. RELATED WORK

With the fast development of smartphone technologies, applications of smartphones in sensing and embedded systems are becoming more popular. For example, in [25], the smartphone is used as a sensing system to measure the heart rate of people during workout while listening to the music. Another sensing system is used to monitor the expenditure of calories during activities [26]. Because the sensors are embedded on the earphones or smartphones, they do not introduce extra accessories during sensing. These mobile systems can provide useful information for people in realtime with little overhead.

There are also many other utilities of smartphones. For instance, with the power of mobility, smartphones are used as the data mules in [27]. By the implementation of mobile-phone-based data muling system, the smartphones can be used to monitor remote sensor networks. The effect of muling on energy consumption is also studied in the paper. The study on energy consumption and efficiency of mobile systems are important because the limitation of power supply and processors performance. [28] studies the balancing of energy, latency and accuracy on smartphones and [29] presents analytical and empirical characterisations of imaging sensors in embedded cameras in order to optimise their energy consumption. Other examples of smartphone utilities include security and authentication [30], [31], visualisation [32] and localisation [33].

Random matrices are often used to reduce the dimensionality of the problem while preserving the accuracy of the applications. They have been applied to speed up background subtraction on embedded system [34] and cross-correlation computation in sensor networks [35]. In [5], SRC is used for acoustic classification and a *column reduction* procedure is proposed to reduce the dimension of ℓ_1 minimisation. Note that column reduction in [5] is complementary to the techniques of projection matrix optimisation and compressed residuals proposed in this paper; all three can be applied to improve the performance of SRC. Other application of random projection matrix is to enable efficient moisture data collection in sensor networks [36] and privacy preservation of voice data [30].

Other related work include face recognition algorithms [3], [2], [1], [4] and projection matrix optimisation methods [9], [8]. We have already discussed them in the main text.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we address the challenges of performing face recognition *accurately* and *efficiently* on smartphones by designing a new face recognition algorithm called opti-SRC. A key idea behind opti-SRC is a novel strategy to optimise the projection matrix so that the resulting sensing matrix has both low row and mutual coherence. We show that opti-SRC is 5-17% more accurate than the state-of-the-art face recognition algorithms. The use of optimised projection matrix means that opti-SRC can achieve the same recognition accuracy using a lower dimension projection matrix. This translates to a lower computation requirement on smartphones. In addition, we propose the use of compressed residuals in order to significantly reduce the computational time of opti-SRC on resource constrained smartphones. We compare the running time of opti-SRC against OpenCV algorithms on three smartphone platforms and find they have comparable running time.

Considering the inconvenience of offline optimisation, we may investigate the choice of offloading the optimisation task to a cloud or do it locally. Offloading the task to the cloud can always be a potential choice for mobile systems. However, it is known that transmission consumes lots of energy. Also, if the smartphones cannot access to Wifi, it has to use 3G to offload the task. The energy consumption and delay will increase significantly when the signal strength is low. A lot of careful investigations need to be done on the resource consumption when we introduce cloud in this system.

REFERENCES

- [1] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *CVPR' 91*. IEEE, 1991, pp. 586–591.
- [2] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *PAMI*, vol. 19, no. 7, pp. 711–720, 1997.
- [3] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *PAMI*, vol. 28, no. 12, pp. 2037–2041, 2006.
- [4] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *PAMI*, vol. 31, no. 2, pp. 210–227, 2009.
- [5] B. Wei, M. Yang, Y. Shen, R. Rana, C. T. Chou, and H. Wen, "Real-time classification via sparse representation in acoustic sensor networks," in *Sensys' 13*. ACM, 2013.
- [6] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *TIT*, pp. 489–509, 2006.
- [7] D. Donoho, "Compressed sensing," *TIT*, pp. 1289–1306, 2006.
- [8] J. Duarte-Carvajalino and G. Sapiro, "Learning to Sense Sparse Signals: Simultaneous Sensing Matrix and Sparsifying Dictionary Optimization," *ITIP*, pp. 1395–1408, 2009.
- [9] M. Elad, "Optimized projections for compressed sensing," *IEEE Transactions on Signal Processing*, vol. 55, no. 12, pp. 5695–5702, 2007.
- [10] A. Krause and C. Guestrin, "Optimizing Sensing: From Water to the Web," *Computer*, vol. 42, pp. 38–45, 2009.
- [11] E. Candes and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse problems*, vol. 23, no. 3, p. 969, 2007.
- [12] D. Donoho, M. Elad, and V. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *TIIT*, pp. 6–18, 2006.
- [13] J. A. Tropp, "The sparsity gap: Uncertainty principles proportional to dimension," in *CISS' 10*. IEEE, 2010, pp. 1–6.
- [14] R. Calderbank, S. Howard, and S. Jafarpour, "Construction of a large class of deterministic sensing matrices that satisfy a statistical isometry property," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 358–374, 2010.
- [15] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, pp. 253–263, 2008.
- [16] F. Glover and M. Laguna, *Tabu Search*. Norwell, MA, USA: Kluwer Academic Publishers, 1997.
- [17] G. Paul, "An efficient implementation of the robust tabu search heuristic for sparse quadratic assignment problems," *European Journal of Operational Research*, vol. 209, no. 3, pp. 215–218, 2011.
- [18] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Conference in modern analysis and probability*, 1984.
- [19] A. Georghiadis, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *PAMI*, vol. 23, no. 6, pp. 643–660, 2001.
- [20] *AT&T Laboratories Cambridge, The AT&T Database of Faces*, 2002. [Online]. Available: <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>
- [21] D. Cai, X. He, Y. Hu, J. Han, and T. Huang, "Learning a spatially smooth subspace for face recognition," in *CVPR '07*. IEEE, 2007, pp. 1–7.
- [22] B. Boom, G. Beumer, L. J. Spreeuwers, and R. N. Veldhuis, "The effect of image resolution on the performance of a face recognition system," in *ICARCV' 06*. IEEE, 2006, pp. 1–6.
- [23] M. A. Davenport, M. B. Wakin, and R. G. Baraniuk, "The compressive matched filter," Tech. Rep. TREE 0610, Rice University, Tech. Rep., 2006.
- [24] M. Salman Asif and J. Romberg, "Dynamic updating for l_1 minimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 421–434, 2010.
- [25] S. Nirjon, R. F. Dickerson, Q. Li, P. Asare, J. A. Stankovic, D. Hong, B. Zhang, X. Jiang, G. Shen, and F. Zhao, "Musicalheart: A hearty way of listening to music," in *Sensys' 12*. ACM, 2012, pp. 43–56.
- [26] A. Zhan, M. Chang, Y. Chen, and A. Terzis, "Accurate caloric expenditure of bicyclists using cellphones," in *Sensys' 12*. ACM, 2012, pp. 71–84.
- [27] U. Park and J. Heidemann, "Data muling with mobile phones for sensor networks," in *Sensys' 11*. ACM, 2011, pp. 162–175.
- [28] D. Chu, N. D. Lane, T. T.-T. Lai, C. Pang, X. Meng, Q. Guo, F. Li, and F. Zhao, "Balancing energy, latency and accuracy for mobile sensor data classification," in *Sensys' 11*. ACM, 2011, pp. 54–67.
- [29] R. LiKamWa, B. Priyantha, M. Philipose, L. Zhong, and P. Bahl, "Energy characterization and optimization of image sensing toward continuous mobile vision," in *Mobisys' 13*. ACM, 2013.
- [30] B. Liu, Y. Jiang, F. Sha, and R. Govindan, "Cloud-enabled privacy-preserving collaborative learning for mobile sensing," in *Sensys' 12*. ACM, 2012, pp. 57–70.
- [31] P. Gilbert, J. Jung, K. Lee, H. Qin, D. Sharkey, A. Sheth, and L. P. Cox, "Youprove: authenticity and fidelity in mobile sensing," in *Sensys' 11*. ACM, 2011, pp. 176–189.
- [32] C.-W. You, N. D. Lane, F. Chen, R. Wang, Z. Chen, T. J. Bao, M. Montes-de Oca, Y. Cheng, M. Lin, L. Torresani *et al.*, "Carsafe app: Alerting drowsy and distracted drivers using dual cameras on smartphones," *Mobisys' 13*, 2013.
- [33] J. Qiu, D. Chu, X. Meng, and T. Moscibroda, "On the feasibility of real-time phone-to-phone 3d localization," in *Sensys' 11*. ACM, 2011, pp. 190–203.
- [34] Y. Shen, W. Hu, J. Liu, M. Yang, B. Wei, and C. T. Chou, "Efficient background subtraction for real-time tracking in embedded camera networks," in *Sensys' 12*. ACM, 2012, pp. 295–308.
- [35] P. Misra, W. Hu, M. Yang, and S. Jha, "Efficient cross-correlation via sparse representation in sensor networks," in *IPSN' 12*. ACM, 2012, pp. 13–24.
- [36] X. Wu and M. Liu, "In-situ soil moisture sensing: measurement scheduling and estimation using compressive sensing," in *IPSN' 12*. ACM, 2012, pp. 1–12.