Robust Face Recognition using Sparse Representation with a Novel Greedy Sparse Recovery Technique

RAFIKA HARRABI
Computer Engineering Department,
Faculty of Computers and Information Technology,
University of Tabuk,
Tabuk 47512,
SAUDI ARABIA

Abstract: - This paper presents an efficient face recognition framework based on the Fast Reduced Sampling Matching Pursuit (FRSMP) algorithm. Unlike conventional greedy sparse recovery methods, which may select an inappropriate number of components per iteration, the proposed approach introduces an adaptive double threshold mechanism to determine relevant elements in a data-driven manner. Furthermore, it employs the Sherman Morrison Woodbury identity to iteratively update the least squares solution, thereby avoiding expensive large scale matrix inversions and significantly reducing computational complexity. The proposed method is extensively evaluated in terms of recognition accuracy, execution time, iteration count, and reconstruction error, and compared with state of the art techniques. When integrated into the kernel based sparse representation classification (KSRC) framework, FRSMP demonstrates notable performance gains, achieving high recognition rates (97.41% on LFW, 98.67% on GT, and 95.37% on AR) with reduced computational cost. The results confirm that the method outperforms existing sparse representation approaches while maintaining an effective balance between accuracy and elliciency, making it suitable for practical and resource constrained applications.

Key-Words: - Face Recognition, Sparse Representation Classification, Compressive Sensing, Fast Reduced Sampling Matching Pursuit (FRSMP), Kernel Methods, Sherman Morrison Woodbury Formula.

Received: March 21, 2026. Revised: May 16, 2026. Accepted: June 14, 2026. Published: November 3, 2026.

1 Introduction

Recent research has increasingly focused on compressive sensing and sparse representation as effective frameworks in signal and image processing, [1], [2]. Their ability to reconstruct high-dimensional data from a limited number of measurements has enabled successful applications in image compression, denoising, and face recognition. Nevertheless, face recognition remains a challenging task under practical conditions such as illumination changes, pose variations, and partial occlusion. In this regard, kernel-based sparse representation classification has shown strong robustness when dealing with distorted or incomplete facial images.

Originally proposed in [3], KSRC maps input samples into a high-dimensional kernel feature space via a nonlinear transformation. Within this space, sparse coding is performed by solving an l_1-norm minimization problem, which enhances the method's robustness to noise and occlusion.

A primary limitation of KSRC, however, lies in its high computational burden, resulting from the large dimensionality introduced by kernel mappings. To address this issue, dimensionality reduction techniques are frequently applied before performing sparse representation. Methods such as Principal Component Analysis (PCA) [4], Linear Discriminant Analysis (LDA) [5], and Random Projection (RP) [6] help lower computational demands while preserving the most discriminative features of the data.

Sparse reconstruction methods are commonly divided into two principal categories, one of which relies on convex optimization techniques such as LASSO [7], BPDN [8], and GPSR [9]. Although these approaches provide reliable and accurate solutions, they typically involve considerable computational cost. Consequently, their use is often restricted to problems of limited or moderate size.

Greedy search algorithms are widely used to reduce computational complexity while retaining acceptable reconstruction accuracy, with representative approaches including OMP, StOMP,

SWOMP, SaKOMP, CoSaMP, SP, and RMP [10], [11], [12], [13], [14], [15], [16], which experimentally achieve a good balance between speed and performance.

However, the performance of many greedy algorithms strongly depends on the heuristic rules used for atom selection at each iteration. Inappropriate selection may result in choosing too many or too few atoms, which can degrade the reconstruction accuracy.

This behavior may increase reconstruction time and compromise signal estimation accuracy. Additionally, conventional sparse recovery methods that employ least-squares updates often require computing the pseudo-inverse of matrices at every iteration, leading to high computational costs for large-scale data. To mitigate this issue, incremental matrix inversion techniques based on the Sherman Morrison formula [17] and its extension, the Sherman Morrison Woodbury formula [18], have been developed. These approaches allow efficient updates to the inverse when columns are added or removed. However, to date, no studies have explored inverse update strategies that can handle the insertion or deletion of arbitrary columns at any position within the matrix, particularly in the context of greedy sparse reconstruction algorithms.

In this study, we propose a practical and scalable solution to the sparse recovery problem via a novel greedy algorithm termed Fast Reduced Sampling Matching Pursuit (FRSMP). The FRSMP approach addresses key limitations of traditional sparse recovery methods by adaptively selecting an appropriate number of atoms at each iteration through a double-thresholding strategy, while simultaneously updating the matrix inverse using the Sherman–Morrison–Woodbury (SMW) formula. This eliminates the need for computationally expensive full-scale matrix inversions, resulting in a substantial improvement in efficiency.

The method achieves both faster computation and higher recognition accuracy by combining the adaptive element selection mechanism with the SMW-based inverse update procedure. This design allows for iterative sparse signal estimation without resorting to large matrix inversions, making it highly suitable for real-time and large-scale applications.

This work can be viewed as a natural extension of previous research, [19], where FRSMP was initially introduced for compressive sensing. That earlier study focused on algorithmic design and theoretical analysis, showing that FRSMP attains faster convergence and higher reconstruction accuracy under both noisy and noiseless conditions compared with classical greedy algorithms such as OMP [10], StOMP [11], CoSaMP [14], and SP [15].

Here, we extend the application of FRSMP to face recognition, particularly within the kernel-based sparse representation classification (KSRC) framework. By integrating FRSMP into KSRC, we develop a face recognition pipeline capable of efficiently processing large-scale image datasets while maintaining robustness. Extensive experiments on benchmark datasets, including LFW, AR, and GT, demonstrate that the proposed KSRC_FRSMP method achieves both superior recognition rates and significantly reduced classification time, highlighting its suitability for practical deployments.

The primary contribution of this work lies in the development of FRSMP with a double-thresholding mechanism for adaptive element selection, effectively addressing the instability and arbitrary selection issues common in traditional greedy pursuit methods. Additionally, the framework leverages the SMW formula for efficient matrix inverse computation, removing the need for repeated full-scale inversions and drastically lowering computational cost.

Overall, the proposed approach delivers high recognition performance with reduced processing time and improved scalability in high-dimensional kernel feature spaces. The method is grounded in adaptive and efficient sparse representation, where FRSMP dynamically selects an appropriate number of support elements and updates the inverse matrix iteratively. This design ensures computational scalability while maintaining accuracy, enabling the system to outperform several state-of-the-art face recognition methods that rely on conventional l_1-norm solvers.

To assess the performance of the proposed approach, we incorporate FRSMP into the KSRC framework for face recognition. Experimental evaluations on standard benchmark datasets, including LFW, GT, and AR, indicate that the KSRC_FRSMP method surpasses current state-of-the-art techniques in both recognition accuracy and computational efficiency, maintaining robust performance even under noisy conditions.

Section 2 reviews CS, KSRC, and common greedy pursuit algorithms; Section 3 presents the proposed FRSMP method; Section 4 reports and analyses experimental results demonstrating the effectiveness of FRSMP with KSRC; and Section 5 concludes the paper by summarizing the main contributions and discussing its computational efficiency for real-time face recognition.

2 Kernel Sparse Representation

2.1 Compressed Sensing

Let $x \in R^N$denote an s-sparse signal, that is, a signal containing at most $s$nonzero elements. In the context of compressed sensing (CS), the goal is to reconstruct $x$from a limited number of linear measurements. In practice, this acquisition process is commonly modeled as follows:

$$y = Ax \quad (1)$$

where $y \in R^M$denotes the measurement vector, while $A \in R^{M \times N}$represents the sensing matrix, with the dimensionality condition $s < M < N$ being satisfied. Given this underdetermined system, the recovery of the original sparse signal is not straightforward. However, Donoho [1] showed that exact reconstruction of $x$can be achieved by solving an l_1-norm minimization problem, expressed as

$$\hat{x} = argmin_x \|x\|_1 \quad s.t. \quad y = Ax \quad (2)$$

Accurate recovery requires the sensing matrix A to satisfy the Restricted Isometry Property (RIP), which limits the distortion of sss-sparse signals during measurement. Specifically, RIP holds if there exists a constant $\delta_s \in [0,1]$ such that the Euclidean norms of sparse vectors are approximately preserved, allowing reliable signal reconstruction.

$$\hat{x} = argmin_x \|x\|_1 \quad subject\ to \quad y = Ax \quad (3)$$

2.2 Kernel Sparse Representation-Based Classifier

Consider a training dataset composed of Cdistinct classes, where each class $i$contains N_c facial images of dimension M. The training samples associated with class $i$can be arranged into a matrix defined as:

$$A_i = [a_{i,1}, a_{i,2}, .., a_{i,j}, .., a_{i,Nc}] \in R^{M \times Nc} \quad (4)$$

where each column vector $a_{i,j} \in R^M$represents the j^{th} facial image belonging to class i. By concatenating the class-specific matrices, the complete dictionary $A$is constructed as:

$$A = [a_1, a_2, .., a_N] \in R^{M \times N}, \text{where } N = \sum_{i=1}^{C} n_i \quad (5)$$

In practice, this dictionary serves as a compact representation of all training samples across classes. The Kernel Sparse Representation–Based Classifier (KSRC) [3] operates by projecting both the training samples and a given test sample into a high-dimensional feature space Hthrough a nonlinear mapping $\phi(\cdot): R^M \to H$. Within this kernel-induced space, the dictionary is implicitly transformed, enabling more effective discrimination among classes:

$$\Phi(A) = [\phi(a_1), \phi(a_2), \ldots, \phi(a_N)]^T \in R^{D \times N} \quad (6)$$

The underlying assumption of KSRC [20], [21] is that a test sample y can be approximately represented as a sparse linear combination of the mapped training samples in the feature space. This leads to the following sparse optimization problem:

$$\hat{x} = argmin_x \|x\|_1 \quad s.t \quad \phi(y) = \phi(A)x \quad (7)$$

Direct computation in the high-dimensional space $H$is generally impractical. Therefore, the above formulation is transformed into an equivalent kernel-based representation using inner products defined by a kernel function. The sparse coding problem is then reformulated as:

$$\hat{x} = argmin_x \|x\|_1 s.t\ \|B^T k(A,y) - B^T K x\|_2^2 \leq \varepsilon \quad (8)$$

where $K = \phi(A)^T \phi(A) \in R^{N \times N}$denotes the kernel Gram matrix, $k(A,y) = \phi(A)^T \phi(y) \in R^N$represents the kernel similarity vector between the test sample and the training set, $B \in R^{N \times d}$is a projection matrix employed for dimensionality reduction, and ε is a small tolerance parameter controlling reconstruction error.

Once the sparse coefficient vector $\hat{x}$is obtained, the reconstruction residual for each class $i$is computed independently as:

$$r_i(y) = \| B^T k(A,y) - B^T K \delta_i(\hat{x}) \|_2 \quad (9)$$

where $\delta_i(\hat{x})$is an operator that preserves only the coefficients associated with class i, while setting all other entries to zero.

Finally, the identity of the test sample is determined according to the minimum reconstruction error criterion:

$$label(y) = arg\ min_i\ r_i(y) \quad (10)$$

In the proposed framework, the conventional l_1-norm solvers used within KSRC are replaced by the Fast Reduced Sampling Matching Pursuit (FRSMP) algorithm. This substitution significantly reduces computational complexity while maintaining

discriminative sparse representations, thereby enhancing both the efficiency and scalability of the KSRC classifier for large-scale face recognition tasks.

2.3 Greedy Pursuit Algorithms

Owing to their computational efficiency, greedy pursuit techniques are commonly adopted in sparse signal recovery. Under the measurement model $y = Ax$, they iteratively build an estimate of the sparse vector x through adaptive selection of columns from A that match the measurements.

At each iteration, the residual is updated, and the most correlated columns of A are chosen to update the support and estimate the sparse coefficients.

Among the earliest and most influential greedy approaches is Orthogonal Matching Pursuit (OMP), [10]. OMP selects a single atom per iteration specifically, the one exhibiting the maximum correlation with the current residual and updates the residual through an orthogonal projection. While OMP is straightforward and effective for low sparsity levels, its requirement for one iteration per nonzero coefficient often results in a large number of iterations and increased computational cost in high-dimensional settings.

To accelerate convergence, several extensions of OMP have been proposed that allow the selection of multiple atoms at each iteration. Stagewise Orthogonal Matching Pursuit (StOMP) selects atoms whose correlation magnitudes exceed a threshold derived from the residual energy, whereas Stagewise Weak Orthogonal Matching Pursuit (SWOMP) retains atoms whose correlations surpass a fixed fraction of the maximum correlation value. These stagewise strategies reduce the number of iterations but may suffer from the inclusion of irrelevant atoms, especially in noisy environments, [11].

Greedy methods with pruning improve robustness by expanding a candidate atom set and then retaining only the most significant coefficients. CoSaMP [14] and SP [15] follow this strategy, with CoSaMP using a larger-than-expected candidate set before pruning, while SP restricts the set size to the assumed sparsity level. Although this enhances stability, reliance on a predefined sparsity may lead to over- or under-selection when the true sparsity is unknown or signal-dependent.

The Reduced-Set Matching Pursuit (RMP) [16] employs an adaptive double-thresholding strategy to improve atom selection by first identifying highly correlated atoms and then retaining only the most relevant ones. While this reduces the influence of weak or noisy components, RMP still requires solving a least-squares problem with explicit matrix inversion during reconstruction.

A key computational bottleneck in greedy pursuit methods is the repeated least-squares estimation at each iteration, as the growing support set makes pseudo-inverse computations increasingly expensive, especially for large-scale or real-time applications.

Greedy pursuit algorithms offer a practical trade-off between accuracy and efficiency, but heuristic atom selection and repeated matrix inversions limit their scalability. These challenges motivate the proposed Fast Reduced Sampling Matching Pursuit (FRSMP), which combines adaptive selection with efficient inverse updates to improve accuracy and computational performance.

3 The proposed Method

Most existing greedy pursuit algorithms employed for sparse signal recovery rely on conventional ℓ_1-norm minimization strategies and have shown satisfactory performance in a variety of applications. Nevertheless, a fundamental drawback of these approaches lies in their unregulated atom selection process. At each iteration, they may either include an excessive number of atoms or fail to select a sufficient set, which leads to unnecessary computational overhead and degradation in reconstruction accuracy. Moreover, the repeated computation of least-squares solutions significantly increases processing cost, especially when dealing with high-dimensional data.

To address these limitations, this work introduces a novel greedy sparse recovery technique termed Fast Reduced Sampling Matching Pursuit (FRSMP). The proposed method is specifically designed to enhance both the accuracy and efficiency of sparse representation by combining adaptive atom selection with an efficient inverse matrix update mechanism. Unlike traditional greedy algorithms, FRSMP dynamically adjusts the number of selected atoms at each iteration using a double-thresholding strategy, thereby avoiding arbitrary or heuristic selection rules.

The proposed approach is tightly integrated within the Kernel Sparse Representation-based Classification (KSRC) framework. Compressed sensing theory focuses on reconstructing sparse signals from a limited number of measurements, while KSRC exploits nonlinear kernel mappings to improve class separability in a high-dimensional feature space. By incorporating FRSMP into KSRC,

sparse coefficients can be computed efficiently without repeatedly solving large-scale ℓ_1-norm optimization problems, which significantly reduces computational complexity.

A key advantage of the proposed approach is its ability to perform iterative least-squares updates without full matrix inversions. FRSMP employs incremental inverse updates based on the Sherman Morrison Woodbury (SMW) identity [18], reducing memory usage and computational cost, and making it well suited for large-scale and real-time face recognition tasks.

Beyond efficiency, the framework improves robustness by reducing sensitivity to noise, illumination variations, and occlusions. Its adaptive atom selection retains only informative components, yielding sparse representations that are more stable and discriminative than those of conventional greedy pursuit methods.

The FRSMP algorithm can be viewed as an extension of earlier work introduced in [19], where the method was initially developed for general compressed sensing problems. That study primarily focused on algorithmic formulation and theoretical analysis, demonstrating faster convergence and improved reconstruction accuracy compared with classical greedy techniques such as OMP, StOMP, CoSaMP, and SP. In the present work, the scope of FRSMP is extended to face recognition by embedding it within the KSRC framework, thereby enabling efficient sparse coding in kernel-induced feature spaces.

Before presenting the complete FRSMP algorithm, the next subsection introduces the inverse matrix update strategy that forms the core computational component of the proposed method. In many greedy pursuit algorithms, the least-squares estimation step requires computing the inverse of a growing matrix at each iteration, which becomes prohibitively expensive as the support size increases. To overcome this issue, the proposed approach leverages the SMW formula [18] to incrementally update the inverse when atoms are added to or removed from the support set, eliminating the need for repeated full matrix inversions.

Let $\Phi \in R^{d\times N}$ denote the sensing matrix, defined in this work as $B^T K$, where B and K represent the basis and kernel matrices, respectively.

3.1 Update Inverse Matrix

Let $\Phi \in R^{d\times t}$ be a symmetric positive definite matrix, where $t < d$. In many sparse recovery algorithms, it becomes necessary to update the inverse of this matrix when a subset of columns is either appended or removed during the iterative process. Recomputing the inverse from scratch after each modification is computationally expensive and impractical for large-scale problems.

To efficiently handle such updates, the proposed method employs the Sherman Morrison Woodbury (SMW) identity, which provides a closed-form expression for updating the inverse of a matrix subject to a low-rank modification. For a matrix expressed as $B + VCV^T$, the inverse can be written as:

$$(B + VCV^T)^{-1} = B^{-1} - B^{-1}(C^{-1} + V^T B^{-1}V)^{-1}V^T B^{-1} \quad (11)$$

This formulation is particularly effective when the update rank q , representing the number of added or removed columns, is much smaller than t. In this case, inverse updates are significantly cheaper than full matrix inversion, leading to notable reductions in execution time, as confirmed by the experimental results.

Consider a symmetric matrix R partitioned as:

$$R = [B + V\ V^T\ C\ ] \quad (12)$$

To compute the inverse of the partitioned matrix R, we rely on the general inverse formula for block matrices [18], which can be expressed as follows [22]:

$$R^{-1} = [F_{11}^{-1}\ - F_{11}^{-1}VC^{-1}\ - C^{-1}V^T F_{11}^{-1}\ F_{22}^{-1}\ ] \quad (13)$$

where

$$F_{11} = B - VC^{-1}V^T \quad (14)$$

$$F_{22} = C - V^T B^{-1}V \quad (15)$$

The inverses of F_{11} and F_{22} can be efficiently updated using the Sherman–Morrison–Woodbury (SMW) identity, avoiding full inversion and reducing computational overhead, especially for large matrices in iterative processes.

$$F_{11}^{-1} = B^{-1} + B^{-1}VF_{22}^{-1}V^T B^{-1} \quad (16)$$

$$F_{22}^{-1} = C^{-1} + C^{-1}V^T F_{11}^{-1}VC^{-1} \quad (17)$$

Unlike conventional greedy pursuit methods that recompute a full matrix inverse at every iteration, the use of the SMW identity enables incremental inverse updates. This strategy significantly lowers computational complexity and

ensures scalability in high-dimensional kernel spaces, which is essential for the KSRC framework.

The following subsections describe how the inverse matrix is updated when columns are either added to or removed from the sensing matrix.

3.1.1 Updating Inverse Matrix by Adding Columns

Assume that, at iteration i, a set of new columns collected in matrix U is appended to the previous sensing matrix Φ_{i-1}. The updated matrix is given by: $\Phi_i = [\Phi_{i-1}\ U]$.

The associated Gram matrix becomes:

$$B_i = B_{i-1} + UU^T = \Phi_{i-1}^T\Phi_{i-1} + UU^T = \left[\Phi_{i-1}^T\ U^T\right][\Phi_{i-1}\ U] \tag{18}$$

which can be written in block form as:

$$B_i = \left[\Phi_{i-1}^T\Phi_{i-1}\ \Phi_{i-1}^T U\ U^T\Phi_{i-1}\ U^T U\right] \tag{19}$$

By applying block matrix inversion together with the SMW formula, the inverse B_i^{-1} can be updated efficiently without computing a full inverse:

$$B_i^{-1} = [F_{11}^{-1} - B_{i-1}^{-1}\Phi_{i-1}^T UD^{-1} \quad -D^{-1}U^T\Phi_{i-1}B_{i-1}^{-1}\ D^{-1}\ ] \tag{20}$$

where

$$F_{11}^{-1} = B_{i-1}^{-1} + B_{i-1}^{-1}\Phi_{i-1}^T UD^{-1}U^T\Phi_{i-1}(B_{i-1}^{-1})^T \tag{21}$$

$$D = U^T U - U^T\Phi_{i-1}B_{i-1}^{-1}\Phi_{i-1}^T U \tag{22}$$

This update process is summarized in **Algorithm 3 (Annex)**.

3.1.2 Updating Inverse Matrix by Deleting Columns

When a subset of columns, represented by the matrix P, is removed, the current sensing matrix can be expressed as: $\Phi_i = [\Phi_{i+1}\ P]$, where $\Phi_{(1)}$denotes the matrix after deletion. By applying a similar block-partitioning strategy, the inverse $B_{(1)}^{-1}$can be efficiently derived from B_i^{-1}using the Sherman–Morrison–Woodbury (SMW) identity, without the need to recompute the entire inverse.

The corresponding Gram matrix takes the form:

$$B_i = \left[\Phi_{i+1}^T\Phi_{i+1}\ \Phi_{i+1}^T P\ P^T\Phi_{i+1}\ P^T P\right] \tag{23}$$

The inverse of the partitioned matrix B_i^{-1} can be expressed via its sub-blocks, allowing efficient incremental updates that reduce computational cost while preserving numerical stability in iterative sparse reconstruction.

$$B_i^{-1} = \left[F_{11}^{-1} - B_{i+1}^{-1}\Phi_{i+1}^T PD^{-1} \quad -D^{-1}P^T\Phi_{i+1}B_{i+1}^{-1}\ D^{-1}\right] \tag{24}$$

where:

$$F_{11}^{-1} = B_{i+1}^{-1} + B_{i+1}^{-1}\Phi_{i+1}^T PD^{-1}P^T\Phi_{i+1}(B_{i+1}^{-1})^T \tag{25}$$

Let $Z = B_{i+1}^{-1}\Phi_{i+1}^T P$. Eq. (25) can be simplified as:

$$F_{11}^{-1} = B_{i+1}^{-1} + ZD^{-1}Z^T \tag{26}$$

Finally, the inverse matrix for the next iteration is obtained as:

$$B_{i+1}^{-1} = F_{11}^{-1} - ZD^{-1}Z^T \tag{27}$$

This deletion-based inverse update is also summarized in **Algorithm 3**.

3.2 Fast Reduced Sampling Matching Pursuit (FRSMP)

The FRSMP algorithm improves sparse recovery through two key mechanisms. Its adaptive double-threshold atom selection dynamically chooses atoms based on the residual's dominant components, reducing over- or under-selection and enhancing reconstruction accuracy compared to traditional greedy methods.

The second feature of FRSMP is its efficient iterative least-squares estimation using incremental inverse updates. By applying the Sherman–Morrison–Woodbury (SMW) identity, FRSMP avoids costly pseudo-inverse computations, reducing the $O(t^3)$ inversion cost and greatly improving computational efficiency.

At each iteration, FRSMP adaptively selects atoms using a dual-threshold rule based on the residual. The chosen indices are combined with previous ones to update the support set T_i, enabling progressive and stable signal reconstruction.

Once the support set has been updated, the sparse coefficient vector is estimated by solving a least-squares optimization problem restricted to the selected atoms. This estimation typically involves

computing the pseudo-inverse of the sensing sub-matrix Φ, given by

$$\Phi_T,\Phi_T^{\dagger} = (\Phi_T^T\Phi_T)^{-1}\Phi_T^T, \quad (\text{illegible})$$

where and $\Phi_T \in R^{d\times t}$ (k<<s<<d).

By integrating the inverse update strategy described in Section 3.1, FRSMP avoids recomputing this pseudo-inverse from scratch at each iteration.

A central advantage of the FRSMP framework is its adaptive double-thresholding mechanism, which automatically regulates the number of selected dictionary atoms in a data-driven manner. This feature addresses common shortcomings observed in classical greedy algorithms such as OMP, CoSaMP, and SP, where fixed or heuristic selection rules often lead to unstable support evolution. Consequently, FRSMP achieves more reliable support updates, improved reconstruction accuracy, and enhanced robustness to noise and modeling uncertainties.

In general, directly computing the inverse of the nonsingular matrix $(\Phi_T^T\Phi_T)$entails a computational complexity on the order of $O(t^3)$, which becomes increasingly prohibitive as the support size grows. To alleviate this burden, FRSMP leverages the matrix inversion lemma to significantly reduce the associated computational overhead. Rather than recomputing the inverse at each iteration, the algorithm incrementally updates the inverse matrix as new atoms are added to the support.

Following the estimation stage, a pruning operation is applied to retain only the most significant coefficients. The columns of Φ corresponding to discarded coefficients are removed, and the inverse matrix is updated accordingly. This pruning step ensures compact support representation and prevents unnecessary growth of the support set.

The residual is updated iteratively until a stopping criterion is met, either when its change is below a threshold Φ or a maximum iteration count is reached. At iteration i-th, the sensing sub-matrix and support set are denoted by Φ_i and T_i, respectively.

Algorithm 2 (Annex) provides a high-level overview of FRSMP, followed by a stepwise explanation showing how each iteration refines signal reconstruction efficiently and robustly. Identification: Compute the correlation vector g between the current residual $r$and the columns of the sensing matrix Φ.

1) Support Merger: Select an initial candidate set containing the indices of the largest entries in the correlation vector. This set is then refined using the following thresholding rule:

$$\Omega = \left\{j: \left|g_j\right| \geq t_s * \frac{\|r_J\|_2}{\sqrt{d}}\right\}, j = 1\ldots N \quad (28)$$

where r_J represents the residual restricted to the candidate index set J, and t_s is a user-defined threshold parameter. The indices in Ω are incorporated into the updated support set T_i.

2) Estimation: Update the support set T_i and estimate the sparse signal using least squares, with the inverse matrix updated incrementally.
3) Pruning: Retain only the $s$largest coefficients of the estimated signal and set the remaining entries to zero. Remove the corresponding columns from Φ_i and update the inverse matrix accordingly.
4) Sample Update: Update the residual according to:

$$r = y - \Phi\hat{x} \quad (29)$$

The iterative process continues until the stopping criterion ε is satisfied. A complete summary of the FRSMP algorithm is presented in **Algorithm 4 (Annex).**

4 Experimental Results

This section reports a comprehensive set of experiments designed to assess the effectiveness of the proposed approach in face recognition scenarios. The experimental evaluation focuses on demonstrating the advantages of the FRSMP algorithm in comparison with several established greedy pursuit methods, considering both ideal (noise-free) conditions and scenarios affected by noise.

All experiments were performed in MATLAB (R2016). Algorithm performance was assessed using reconstruction error, runtime, and the number of iterations to convergence. Together, these metrics provide an objective evaluation of both accuracy and efficiency, enabling a fair comparison with existing methods and offering insight into the practical convergence behavior of the proposed approach.

4.1 Comparisons of Greedy Pursuit Algorithms

For fair and unbiased evaluation, the proposed method was compared with several established

greedy pursuit algorithms, namely OMP, StOMP, SWOMP [11], CoSaMP, SP, and RMP [16]. Performance was primarily assessed using the reconstruction error (RE), which provides a quantitative measure of signal recovery accuracy:

$$RE = \frac{\|x-\hat{x}\|_2}{\|x\|_2} \quad (30)$$

Here, x denotes the original s-sparse signal, while $\hat{x}$ represents its s-sparse approximation. All reported results are averaged over 100 independent simulation runs.

In each experimental trial, a synthetic s-sparse signal of length $n = 1024$was generated, with nonzero entries drawn independently from a Gaussian distribution $N(0,1)$. The sensing matrix $A \in R^{m\times n}$ was constructed by sampling its elements independently from a normalized Gaussian distribution $N(0,1/m)$, with the number of measurements fixed at $m = 256$.

For RMP, the threshold parameters were set to $\alpha = 0.75$and $\beta = 0.5$as suggested in [16]. SWOMP and StOMP were configured with $\alpha = 0.7$and $t_s = 2.5$, respectively. In the FRSMP algorithm, the threshold parameter was set to $t_s = 3$. The stopping criterion was defined either as reaching a maximum of 100 iterations or when the residual norm changed by less than 10^{-8}.

Figure 1 (Appendix) compares the algorithms under noiseless conditions. As shown in Figure 1(a) in Appendix, all methods exhibit low reconstruction error for sparsity levels below 80, while performance differences become evident beyond 100. Notably, FRSMP attains the lowest reconstruction error at higher sparsity levels. The results in Figure 1(b) in Appendix further confirm its efficiency, as FRSMP converges with substantially fewer iterations than the competing methods.

This behavior is expected for OMP, as selecting a single atom per iteration leads to an iteration count that closely follows the sparsity level. Although SWOMP and StOMP reduce the number of iterations, the absence of a pruning mechanism degrades their reconstruction accuracy. CoSaMP and SP maintain relatively stable performance, but their reliance on fixed atom selection restricts their adaptability. Finally, Figure 1(c) in Appendix illustrates the reconstruction time versus sparsity. FRSMP demonstrates a clear computational advantage, especially for sparsity levels above 50, which can be attributed to its adaptive atom selection strategy and efficient inverse-update mechanism. Overall, the noiseless experiments confirm that FRSMP provides improved reconstruction accuracy, faster convergence, and reduced computational cost compared to conventional greedy pursuit methods.

The robustness of the algorithms was also assessed under additive Gaussian noise:

$$y = Ax + e \quad (31)$$

Here, e denotes Gaussian white noise with independent entries drawn from $N(0, 10^{-SNR/10})$. In this experiment, the sparsity level is fixed at $s = 70$, while the Signal-to-Noise Ratio (SNR) is varied from 10 dB to 70 dB to evaluate the robustness of the algorithms under different noise conditions.

As illustrated in Figure 2(a) in Appendix, both FRSMP and RMP consistently produce the smallest reconstruction errors across the full range of SNR values. In contrast, SWOMP and StOMP exhibit poorer performance because they do not incorporate a pruning mechanism; any erroneous indices introduced due to noise remain in the support, which increases the reconstruction error.

Figure 2(b) in Appendix indicates that FRSMP requires the fewest iterations to converge, while Figure 2(c) in Appendix shows that it achieves the shortest reconstruction time among all tested methods. CoSaMP is excluded from these comparisons because its computational cost is substantially higher, stemming from its strategy of selecting a large and fixed number of elements at each iteration.

Overall, the results under noisy conditions confirm that FRSMP not only maintains superior reconstruction accuracy but also remains computationally efficient even at low SNR levels, surpassing all other considered greedy algorithms.

4.2 Evaluation of Face Recognition Methods

Face recognition experiments were conducted on three standard datasets: LFW [24], GT [23], and AR [25]. For each dataset, images were divided into training and testing sets to assess the algorithms' generalization to unseen samples and their recognition performance under diverse facial variations.

Each experiment was repeated 10 times to obtain average recognition rates and computation times. To assess both the accuracy and efficiency of the proposed face recognition approach, its performance was compared with several existing methods [26], [27], including Sparse Representation-based Classification (SRC) [28], KSRC, Extended SRC (ESRC) [29], Kernel ESRC (KESRC) [30], Fast KESRC (FKESRC) [31], and Kernel Coordinate Descent with Virtual Dictionary

(KCDVD) [32]. All experiments were implemented in MATLAB.

The proposed method (KSRC_FRSMP) was compared with SRC [28], KSRC, ESRC [29], KESRC [30], FKESRC [31], and KCDVD [32]. For algorithms requiring sparse coding, the $l1-norm$ minimization was implemented using the $l1_ls$ solver [32], [33].

For clarity in the following discussion, we denote the KSRC method that uses l1_ls for ℓ1-norm minimization as KSRC_l1ls, and the KSRC method combined with our FRSMP algorithm as KSRC_FRSMP.

All kernel-based algorithms in our experiments including KSRC_l1ls, KESRC, FKESRC, KCDVD, and our proposed KSRC_FRSMP classifier use the Gaussian kernel $k(x,y) = e^{(-\|x-y\|^2/2\sigma^2)}$, where the kernel width σ is defined as $\sigma = r \times d$. Here, d represents the average Euclidean distance among the training samples, and r is a kernel ratio chosen separately for each dataset.

To ensure a consistent comparison, all face images undergo dimensionality reduction using Principal Component Analysis (PCA). The KSRC_FRSMP method applies the same reduction procedure to maintain uniformity.

4.2.1 Evaluation of Face Recognition Methods on the LFW face database

The Labeled Faces in the Wild (LFW) dataset [29] is employed to quantitatively evaluate the performance of the proposed methods. This dataset contains over 13,000 images representing 1,680 different individuals.

Figure 3 (Appendix) presents twelve sample images from a single subject. The experiments involve 86 subjects, each represented by 11 color images acquired under varying illumination, facial expressions, and partial occlusions. All images were converted to grayscale, cropped, and resized to 32 × 32 pixels. Principal Component Analysis (PCA) was subsequently applied to reduce dimensionality, retaining 200 principal features that preserve the dominant variance and ensure efficient and robust face recognition.

Empirical evaluation determined that the kernel parameter $r$is set to 1.5, while the threshold parameter is $t_s = 3$.

Recognition performance, reported as mean accuracy ± standard deviation (%) along with the corresponding computational time, is summarized in Table 1 (Appendix) and illustrated in Figure 6 (Appendix) for two training configurations: $N_c =$ 7and $N_c = 9$, where $N_c$denotes the number of training samples per class.

As shown in Table 1 (Appendix), KSRC_FRSMP achieves recognition rates comparable to the KCDVD method. KCDVD relies on a virtual dictionary framework to generate sparse coefficients and employs a coordinate descent approach to solve the kernel sparse representation problem in the feature space. Compared to the other algorithms, KSRC_FRSMP consistently delivers robust and reliable classification performance.

The reported computational time represents the average execution over ten trials. Although KSRC_FRSMP requires slightly more time than FKESRC, this is expected because FKESRC bypasses direct l_1-norm minimization. FKESRC uses a coordinate descent method in the kernel space, providing an indirect but efficient solution to the sparse optimization problem, which significantly accelerates processing. Conversely, SRC, ESRC, KESRC, and KSRC_FRSMP perform explicit l_1-norm minimization.

Although KSRC_FRSMP incurs a slightly higher computation time in some cases, it consistently outperforms FKESRC in terms of recognition accuracy. Moreover, it surpasses KSRC_l1ls in both classification performance and processing speed. For example, when $N_c = 7$, KSRC_FRSMP executes approximately 1.7 times faster than KSRC_l1ls, while for $N_c = 9$, the speedup remains substantial at around 1.5 times. These improvements are primarily attributed to the FRSMP strategy, which adaptively selects an optimal number of representative elements via a double-thresholding mechanism, effectively retaining only the most informative components. In addition, the iterative least-squares estimation embedded in FRSMP further reduces computational overhead, enhancing overall efficiency. Table 1 (Appendix) demonstrates that KSRC_FRSMP achieves the fastest execution among SRC, ESRC, KESRC, KCDVD, and KSRC_l1ls, highlighting its balance of accuracy and speed.

Under the KSRC_FRSMP framework, the average time required to represent and classify a single test sample is roughly 0.2 seconds for $N_c = 7$ and remains below 0.3 seconds for $N_c = 9$. In comparison, KCDVD requires nearly 1 second for the same task.

Overall, KSRC_FRSMP demonstrates significantly higher computational efficiency while maintaining competitive recognition accuracy compared with methods that rely on explicit l_1-norm optimization.

4.2.2 Evaluation of Face Recognition Methods on the GT face database

The Georgia Tech (GT) face database consists of 750 facial images from 50 individuals, with 15 color images per subject acquired under variations in pose, facial expression, and illumination.

In the experimental setup, all images were converted to grayscale, cropped, and resized to 60×5060 pixels, with representative samples shown in Figure 4 (Appendix). For each subject, Nc=3,4,N_c = 3, 4,Nc=3,4, and 555 images were randomly selected for training, and the remaining images were used for testing. Each configuration was repeated ten times, and the average recognition accuracy and computational cost were reported to ensure reliable performance evaluation.

Principal Component Analysis (PCA) was employed to reduce feature dimensionality, projecting each face image onto 50 dimensions. Based on empirical tuning, the kernel parameter was set to $r = 0.07$ and the threshold to $t_s = 3$. This configuration balances computational efficiency with recognition robustness and reflects the algorithm's practical performance.

The classification performance, reported as mean recognition rate ± standard deviation (in percentage), along with the corresponding execution time, is summarized in Table 2 (Appendix) for the three training scenarios. The variation in computational time across methods on the GT dataset is further illustrated in Figure 7 (Appendix). According to the results in Table 2 (Appendix), when only three training samples per class are available, the proposed KSRC_FRSMP approach achieves the highest recognition accuracy among all evaluated methods.

When the number of training samples increases to $N_c = 4$and $N_c = 5$, KSRC_FRSMP maintains recognition performance comparable to KCDVD while consistently outperforming the remaining competing algorithms.

In terms of efficiency, KSRC_FRSMP demonstrates substantially lower computational cost than most methods, except for FKESRC. As expected, the execution time of the proposed method remains significantly smaller than that of alternative approaches. For instance, when $N_c = 5$, KSRC_FRSMP operates approximately three times faster than KCDVD and nearly six times faster than KSRC_l1ls. Under this setup, representing and classifying a single test sample takes about 0.03 seconds, compared to 0.2 seconds for KSRC_l1ls. Table 2 (Appendix) shows that the proposed method achieves a strong balance between recognition accuracy and computational efficiency, making it suitable for practical, real-time applications.

While FKESRC maintains the highest computational speed among the compared methods, KSRC_FRSMP achieves noticeably superior recognition accuracy, demonstrating a more favorable trade-off between processing efficiency and classification reliability.

4.2.3 Evaluation of Face Recognition Methods on the AR face database

The AR face dataset contains over 4,000 images from 126 subjects (70 males and 56 females).

In this study, a subset of 120 subjects is employed, with each individual represented by 26 frontal-face images captured under various conditions, including illumination changes, facial expressions, and partial occlusions caused by accessories such as sunglasses and scarves. Figure 5 (Appendix) shows sample images of a single subject, converted to grayscale and resized to 20 × 25 pixels. For training, $N_c = 3,4,5,6$ images per subject were randomly selected, with the rest used for testing. Each setup was repeated ten times, and the average recognition accuracy and execution time were reported.

PCA was applied to extract a 300-dimensional feature vector from each image. KSRC_FRSMP was configured with $r = 1.5$ and threshold $t_s = 3$ chosen empirically, ensuring efficient computation while preserving discriminative features for robust face recognition.

Recognition results and computation times for different training configurations are shown in Figure 8 (Appendix). When only three training samples per class are available, KSRC_FRSMP achieves recognition performance comparable to KSRC_l1ls. For larger training sets ($N_c = 4$ and $N_c = 5$), the proposed method performs similarly to KESRC.

KSRC_FRSMP achieves similar recognition rates while offering superior computational efficiency. As shown in Table 3 (Appendix), it reduces execution time compared to KSRC_l1ls and KESRC. For instance, with $N_c = 3$, KSRC_FRSMP completes classification in 297.37 s versus 551.16 s for KSRC_l1ls, and with $N_c = 5$, it takes 443.51 s compared to 4403.5 s for KESRC. The per-sample processing time for $N_c = 4$ is 0.12 s, nearly twice as fast as KSRC_l1ls (0.2 s) and faster than KESRC (0.27 s).

KSRC_FRSMP's efficiency arises from the FRSMP strategy, which iteratively estimates the sparse signal via least-squares, updates the matrix inverse incrementally, and uses double-thresholding to select informative elements while pruning

incorrect atoms. This combination ensures high recognition accuracy with significantly reduced computation time, outperforming other sparse representation-based methods. Although FKESRC achieves the shortest execution time on the LFW dataset, KSRC_FRSMP consistently delivers higher recognition accuracy, illustrating the common trade-off between speed and classification reliability. Across the LFW, GT, and AR datasets, KSRC_FRSMP consistently attains the highest recognition performance, while FKESRC remains the most time-efficient method. FKESRC's speed advantage is largely due to its avoidance of l_1-norm optimization and use of a coordinate-descent approximation in the feature space, which significantly reduces computational cost.

In contrast, KSRC_FRSMP leverages iterative least-squares updates based on the Sherman–Morrison–Woodbury (SMW) identity, which significantly speeds up $l_1 - norm$ based sparse recovery compared to traditional solvers like KSRC_l1ls. KSRC_FRSMP has higher computational cost than FKESRC, which avoids explicit sparse reconstruction. KSRC_FRSMP is preferable when recognition accuracy is critical, such as in biometric verification or secure access, while FKESRC suits applications prioritizing speed over slight accuracy reductions, like real-time or embedded systems.

4.2.4 Statistical Validation of Experimental Results

Recognition performance was evaluated on the LFW, GT, and AR datasets to ensure statistical reliability and demonstrate that the improvements with KSRC_FRSMP are consistent and significant.
Table 4 (Appendix) reports the statistical verification results. Pairwise t-tests between KSRC_FRSMP and benchmark algorithms (SRC [28], [34], [35], KSRC [29], [36], KESRC [30], FKESRC [31], KCDVD [37]) show significant improvements in recognition accuracy across all datasets 0.05, confirming the robustness of KSRC_FRSMP's performance gains.

A Chi-Square (χ^2) test was performed to examine dataset dependence, with values from 15.8 to 24.6 exceeding the 0.05 significance threshold. Combined with the pairwise t-tests in Table 4 (Appendix), this confirms that KSRC_FRSMP's performance improvements are consistent across datasets.

A one-way ANOVA was conducted to assess recognition rate differences across algorithms. The F-statistic of 9.42 $(p < 0.01)$ indicates a significant variation, highlighting the superior performance of KSRC_FRSMP over competing methods.

Overall, the statistical analyses including t-tests, Chi-Square tests, and ANOVA confirm that KSRC_FRSMP consistently delivers robust and reliable improvements in recognition accuracy, reflecting the algorithm's inherent strengths rather than chance.

5 Conclusion

In this study, we introduced a novel face recognition approach leveraging the Fast Reduced Sampling Matching Pursuit (FRSMP) algorithm. The proposed method utilizes a double thresholding mechanism to identify the most informative components and estimates the sparse representation via least-squares minimization, efficiently updating the matrix inverse using the Sherman–Morrison–Woodbury formula. This design considerably reduces computational overhead compared to traditional greedy pursuit algorithms, which often suffer from unregulated element selection and higher reconstruction errors.

Extensive experiments under both noisy and noise-free conditions demonstrate that FRSMP consistently outperforms conventional greedy recovery methods in terms of both reconstruction accuracy and computational efficiency. Integrated into the KSRC framework, KSRC_FRSMP achieved recognition rates of 97.41% (LFW), 98.67% (GT), and 95.37% (AR), while reducing per-sample classification time to 0.2 s, 0.12 s, and 0.03 s, respectively. Statistical analyses (t-tests, Chi-Square, ANOVA) confirm these gains are significant, consistent, and robust across datasets.

While FKESRC is slightly faster, KSRC_FRSMP consistently achieves higher recognition accuracy, making it suitable for applications requiring reliable face recognition, such as biometric verification, whereas FKESRC is preferable when runtime efficiency is critical.

Future research could improve robustness under challenging conditions such as occlusions, varying illumination, and low-resolution images through advanced preprocessing or multi-scale feature extraction. Integrating sparse representation with deep learning models, like CNNs or transformer-based embeddings, may further enhance accuracy and generalization. Additionally, optimizing KSRC_FRSMP for real-time execution on embedded or resource-constrained platforms, and extending it to multimodal biometric systems, can increase reliability and expand practical applications.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, the author(s) utilized artificial intelligence (AI) technologies solely for the purpose of linguistic refinement and grammatical correction. Following the application of these tools, the author(s) comprehensively reviewed, verified, and edited the resulting text. The author(s) maintain full accountability and responsibility for the accuracy, integrity, and final content of the publication.

References:

[1] Taghouti, M. Compressed sensing. Computing in Communication Networks, Chapter 10 - Compressed sensing, Academic Press, ISBN. 978-0-12-820488-7, pp. 197–215, 2020. https://doi.org/10.1016/B978-0-12-820488-7.00023-2.

[2] Wang, B., Guo, J., Zhang, Y. Hierarchical feature concatenation-based kernel sparse representations for image categorization. Visual Computer, 33, pp. 647–663, 2017. https://doi.org/10.1007/s00371-016-1215-2.

[3] Jun Yin, Zhonghua Liu, Zhong Jin, Wankou Yang. Kernel sparse representation-based classification. Neurocomputing, 77(1), pp. 120–128, 2012. https://doi.org/10.1016/j.neucom.2011.08.018.

[4] Kshirsagar, V. P., Baviskar, M. R., Gaikwad, M. E. Face recognition using Eigenfaces. 3rd International Conference on Computer Research and Development, Shanghai, China, pp. 302–306, 2011. 10.1109/ICCRD.2011.5764137.

[5] Belhumeur, P. N., Hespanha, J. P., Kriegman, D. J. Eigenfaces vs. Fisherfaces: recognition using class-specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7), pp. 711–720, July 1997. https://ieeexplore.ieee.org/document/598228.

[6] Candès, E. J., Tao, T. Near-optimal signal recovery from random projections: universal encoding strategies? IEEE Transactions on Information Theory, 52(12), pp. 5406–5425, December 2006. https://ieeexplore.ieee.org/document/4016283.

[7] Tibshirani, R. Regression shrinkage and selection via the lasso: a retrospective. Journal of the Royal Statistical Society: Series B, 73(3), pp. 273–282, April 2011. https://doi.org/10.1111/j.1467-9868.2011.00771.x.

[8] Donoho, D. L., Tsaig, Y. Fast solution of ℓ1-norm minimization problems when the solution may be sparse. IEEE Transactions on Information Theory, 54(11), pp. 4789–4812, November 2008. https://ieeexplore.ieee.org/document/4655448.

[9] Blumensath, T., Davies, M. E. Gradient pursuits. IEEE Transactions on Signal Processing, 56(6), pp. 2370–2382, June 2008. https://ieeexplore.ieee.org/abstract/document/4480155.

[10] Tropp, J. A., Gilbert, A. C. Signal recovery from random measurements via orthogonal matching pursuit. IEEE Transactions on Information Theory, 53(12), pp. 4655–4666, December 2007. https://ieeexplore.ieee.org/abstract/document/4385788.

[11] Donoho, D. L., Tsaig, Y., Drori, I., Starck, J. L. Sparse solution of underdetermined linear systems by stagewise orthogonal matching pursuit. IEEE Transactions on Information Theory, 58(2), pp. 1094–1121, February 2012. https://ieeexplore.ieee.org/abstract/document/6145475.

[12] Blumensath, T., Davies, M. E. Stagewise weak gradient pursuits. IEEE Transactions on Signal Processing, 57(11), pp. 4333–4346, June 2009. https://ieeexplore.ieee.org/abstract/document/5109633.

[13] Zha, Z., Liu, X., Zhang, X. Compressed sensing image reconstruction via adaptive sparse nonlocal regularization. Visual Computer, 34, pp. 117–137, 2018. https://doi.org/10.1007/s00371-016-1318-9.

[14] Needell, D., Tropp, J. A. CoSaMP: iterative signal recovery from incomplete and inaccurate samples. Applied and Computational Harmonic Analysis, 26(3), pp. 301–321, May 2009. https://doi.org/10.1016/j.acha.2008.07.002.

[15] Dai, W., Milenkovic, O. Subspace pursuit for compressive sensing signal reconstruction. IEEE Transactions on Information Theory, 55(5), pp. 2230–2249, April 2009. https://ieeexplore.ieee.org/document/4839056.

[16] Abdel-Sayed, M. M., Khattab, A., Abu-Elyazeed, M. F. RMP: Reduced-set matching pursuit for efficient compressed sensing reconstruction. Journal of Advanced Research, 7(6), pp. 851–861, November 2016. https://doi.org/10.1016/j.jare.2016.08.005.

[17] Rosário, F., Monteiro, F. A., Rodrigues, A. Fast matrix inversion updates for massive MIMO detection and precoding. IEEE Signal Processing Letters, 23(1), pp. 75–79, Article

23897, January. 2016. https://doi.org/10.1109/LSP.2015.2500682.

[18] Hager, W. W. Updating the inverse of a matrix. SIAM Review, 31(2), pp. 221–239, 1989. https://doi.org/10.1137/1031049.

[19] Bannour, Z. L., Seddik, H. A new greedy sparse recovery algorithm for fast sparse representation. The Visual Computer, 38(7), pp. 2431–2445, 2022. https://doi.org/10.1007/s00371-021-02121-6.

[20] Bushnag, A., Chaabane, S. B., Harrabi, R., Smart agriculture: IoT-Based smart irrigation with advanced fuzzy logic control. Expert Systems with Applications, 299(D), 130168. March 2026. https://doi.org/10.1016/j.eswa.2025.130168.

[21] Chaabane, S. B., Harrabi, R., Bushnag, A., Seddik, H. Development of Surveillance Robots Based on Face Recognition Using High-Order Statistical Features and Evidence Theory. Journal of Imaging, 12(3), 107. February 2026. https://doi.org/10.3390/jimaging12030107.

[22] Beal, M. J. Variational algorithms for approximate Bayesian inference. PhD thesis,University of London, University College London (United Kingdom) ProQuest Dissertations & Theses, 10016139, May 2003, [Online]. https://www.proquest.com/dissertations-theses/variational-algorithms-approximate-bayesian/docview/1775215626/se-2 (Accessed Date: January 18, 2026).

[23] Xu, Y., Zhang, Z., Lu, G., Yang, J. Approximately symmetrical face images for preprocessing in face recognition and sparse representation-based classification. Pattern Recognition, 54, pp. 68–82, June 2016. https://doi.org/10.1016/j.patcog.2015.12.017.

[24] Dong, Z., Pei, M., Jia, Y. Orthonormal dictionary learning and application to face recognition. Image and Vision Computing, 51, pp. 13–21, 2016. https://doi.org/10.1109/TSP.2018.8441452.

[25] Martinez, A. M. The AR face database. CVC Technical Report, 24, Universitat Autonoma de Barcelona, 1998, [Online]. https://portalrecerca.uab.cat/en/publications/the-ar-face-database-cvc-technical-report-24/ (Accessed Date: January 18, 2026).

[26] Nassef, L., Alghamdi, M. I., Chaabane, S. B. Lightweight and Energy-Aware Intrusion Detection for Industrial IoT Using TinyML and Edge AI. Scientific Reports, 16(1), 18524. June 2026. https://doi.org/10.1038/s41598-026-50690-0.

[27] Chaabane, S. B., Harrabi, R., Seddik, H. Home security system with hybrid face recognition approach using convolutional neural Networks, fuzzy Logic, and SVM classifier. Multimedia Tools and Applications, 85(2), 125. February 2026. https://doi.org/10.1007/s11042-026-21261-2.

[28] Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., Ma, Y. Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(2), pp. 210–227, 19110489, February 2009. https://ieeexplore.ieee.org/abstract/document/4483511.

[29] Deng, W., Hu, J., Guo, J. Extended SRC: undersampled face recognition via intraclass variant dictionary. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(9), pp. 1864–1870, September 2012. https://ieeexplore.ieee.org/abstract/document/6133293.

[30] Fan, Z., Wei, C. Fast kernel sparse representation-based classification for undersampling problems in face recognition. Multimedia Tools and Applications, 79, pp. 7319–7337, 2020. https://doi.org/10.1007/s11042-019-08211-x.

[31] Fan, Z., Zhang, D., Wang, X., Zhu, Q., Wang, Y. Virtual dictionary-based kernel sparse representation for face recognition. Pattern Recognition, 76, pp. 1–13, April 2018. https://doi.org/10.1016/j.patcog.2017.10.001.

[32] Kim, S. J., Koh, K., Lustig, M., Boyd, S., Gorinevsky, D. An interior-point method for large-scale l1-regularized least squares. IEEE Journal of Selected Topics in Signal Processing, 1(4), pp. 606–617, 2007. https://ieeexplore.ieee.org/abstract/document/4407767.

[33] Fan, Z., Ni, M., Zhu, Q., Liu, E. Weighted sparse representation for face recognition. Neurocomputing, 151, pp. 304–309, March 2015. https://doi.org/10.1016/j.neucom.2014.09.035.

[34] Yang, X., Wang, Z., Wu, H., Jiao, L., Xu, Y., Chen, H. Stable and compact face recognition via unlabeled data-driven sparse representation-based classification. Signal Processing: Image Communication, 111, 116889, February 2023. https://doi.org/10.1016/j.image.2022.116889.

[35] Harrabi, R., Chaabane, S. B., Seddik, H. Securing Restricted Zones with a Novel Face

Recognition Approach Using Face Feature Descriptors and Evidence Theory. Computers, Materials & Continua, 87(2), 1. March 2026. https://doi.org/10.32604/cmc.2026.072054.

[36] Kang, C., Liao, S., Xiang, S., Pan, C. Kernel sparse representation with local patterns for face recognition. 18th IEEE International Conference on Image Processing, pp. 3009–3012, December 2011. https://ieeexplore.ieee.org/abstract/document/6116296.

[37] Fan, Z., Zhang, D., Wang, X., Zhu, Q., Wang, Y. Virtual dictionary-based kernel sparse representation for face recognition. Pattern Recognition, 76, pp. 1–13, April 2018. https://doi.org/10.1016/j.patcog.2017.10.001.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

Conceptualization, methodology design, writing original draft, project coordination, Software development, implementation of algorithms, data preprocessing, and manuscript editing, Formal analysis, validation, results evaluation, and writing—review and editing.

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for this research.

Conflict of Interest

The author confirms that there are no conflicts of interest associated with this manuscript. No financial, personal, or professional relationships exist that could have influenced the research findings or interpretations presented in this work.

Ethical Approval Statement:

This research did not involve any experiments on human participants or animals by the authors. All facial image data utilized in this study were sourced from publicly accessible and fully anonymized datasets (GT, LFW, and AR face databases) and were employed solely for academic and research purposes in compliance with their respective usage licenses. Consequently, an ethical approval statement is not required.

Data Availability Statement:

Available from the author upon request.

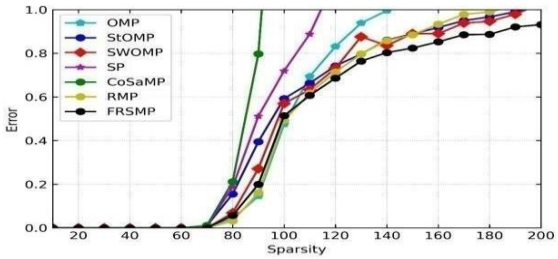
Patient Consent Statement:

As this study relies exclusively on publicly available datasets, no patient consent was required. No identifiable personal data were used or accessed in this research.

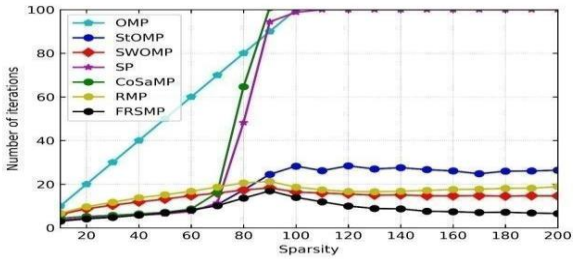
Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US

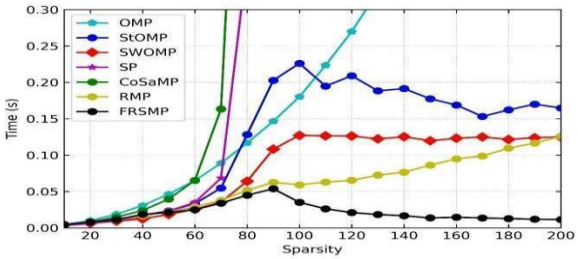
APPENDIX



(a) Reconstruction error as a function of the sparsity level

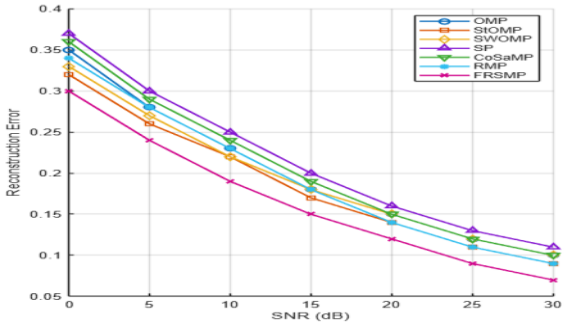


(b) Number of iterations as a function of the sparsity level

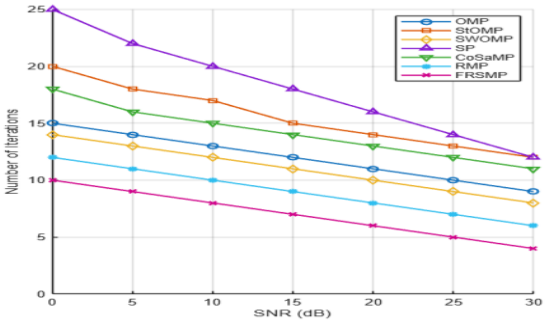


(c) Reconstruction time as a function of the sparsity level

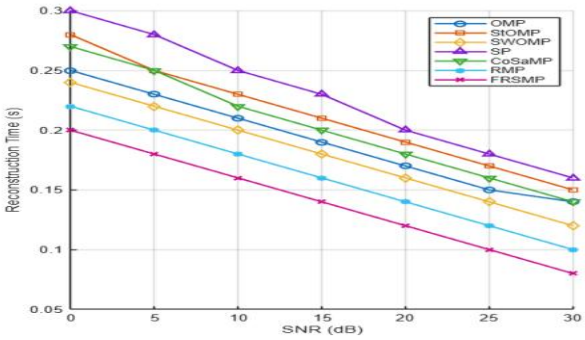
Fig. 1: Performance of different greedy pursuit algorithms for the noiseless case
Source: Created by the author.



(a) Reconstruction error as a function of Signal-to-Noise Ratio (SNR),



(b) Number of iterations as a function of SNR,



(c) Reconstruction time as a function of SNR.

Fig. 2: Performance comparison of different greedy pursuit algorithms and the proposed FRSMP method under noisy conditions (y = Ax + e)
Source: Created by the author.



Fig. 3: Some images of one person inLabeled Faces in the Wild (LFW) database [Reference Number]. face database

Source: Sample images selected from the Wild (LFW) database[24] [29].



Fig. 4: Some images of one person in GT face database

Source: Sample images selected from the GT face database [23].



Fig. 5: Some images of one person in AR face database

Source: Sample images selected from the AR face database, [25].

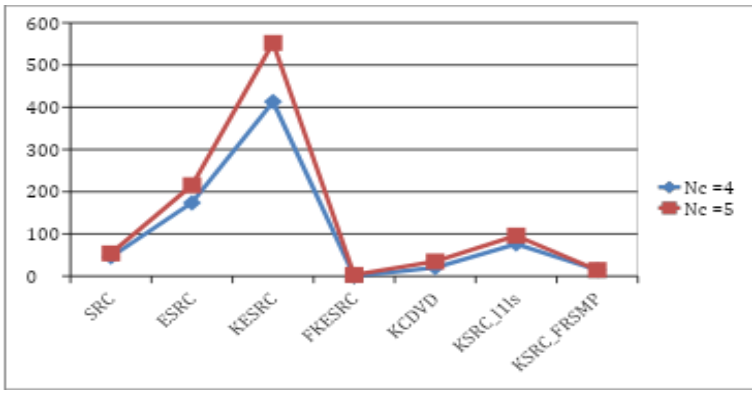


Fig. 6: Computational time on the LFW database

Source: Created by the author.

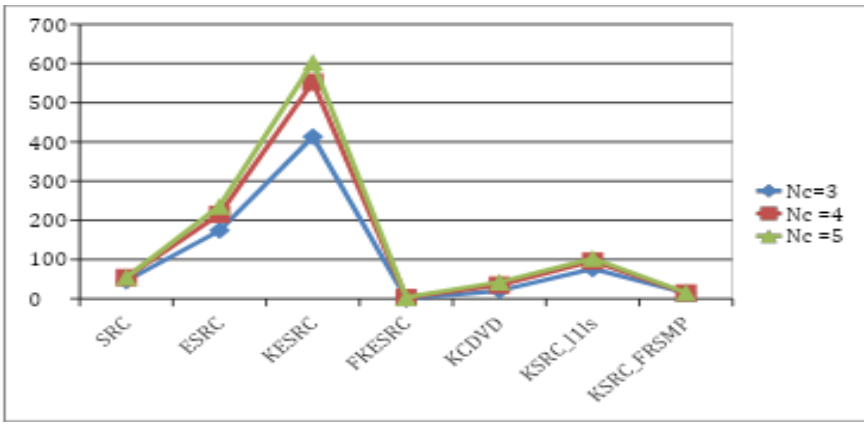


Fig. 7: Computational time on the GT database
Source: Created by the author.

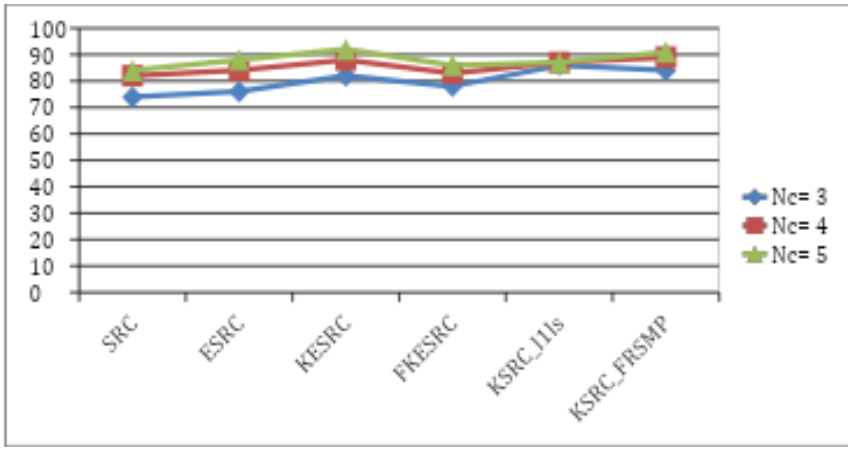


Fig. 8: Recognition rates on the AR database
Source: Created by the author.

Table 1. Recognition rates and Computational time on the LFW database

Techniques	Nc =7		Nc =9	
	Recog. %	Time(s)	Recog. %	Time(s)
SRC	37.56±1.64	312.41	42.33±2.68	188.01
ESRC	39.74±1.68	659.78	46.22±2.92	533.46
KESRC	40.73±1.89	561.45	45.29±3.60	461.31
FKESRC	39.74±3.01	9.16	45.06±2.92	9.12
KCDVD	46.66±2.76	182.60	51.63±2.60	172.69
KSRC_l1ls	40.52±1.76	118.58	45.93±2.71	71.93
KSRC_FRSMP	45.63±1.57	67.49	51.98±1.66	48.70

Source: Experimental results obtained by the author.

Table 2. Recognition rates and Computational time on the GT database

Algorithms	Nc= 3		Nc= 4		Nc= 5	
	Recog. %	Time(s)	Recog. %	Time(s)	Recog. %	Time(s)
SRC	54.33±3.07	45.59	58.11±2.54	53.82	62.06±1.96	54.33
ESRC	57.52±3.44	173.76	62.78±2.55	215.32	66.72±1.95	236.40
KESRC	62.17±2.10	413.09	66.13±2.15	552.35	70.28±1.87	603.16
FKESRC	58.03±3.58	2.35	62.80±1.42	2.90	67.44±2.60	3.49
KCDVD	61.98±2.47	20.26	68.42±1.42	34.70	72.14±2.09	42.45
KSRC_lIls	58.90±2.12	76.10	62.71±2.38	96.05	66.90±2.24	102.46
KSRC_FRSMP	65.42±0.86	14.15	69.85±1.22	14.56	72.18±1.18	15.71

Source: Experimental results obtained by the author.

Table 3. Computational time on the AR database

Techniques used	**Nc= 3**	**Nc= 4**	**Nc= 5**
SRC	1014.48	1355.49	1806.94
ESRC	2327.71	4111.81	4377.65
KESRC	2435.25	3378.61	4403.5
FKESRC	36.92	39.85	58.08
KSRC_lIls	551.16	702.07	865.57
KSRC_FRSMP	297.37	318.06	443.51

Source: Experimental results obtained by the author.

Table 4. Statistical Validation of Recognition Results for the Proposed KSRC_FRSMP Model

Statistical Test	**Compared Algorithms / Datasets**	**Test Statistic**	**p-Value**	**Significance Level (α)**	**Interpretation**
t-Test (Pairwise)	KSRC_FRSMP vs. SRC [28][34][35], KSRC [29][36], KESRC [30], FKESRC [31], KCDVD [37] on LFW, GT, AR datasets	–	$p < 0.05$	0.05	The mean recognition rates of KSRC_FRSMP are significantly higher than those of all compared methods, confirming the reliability of the observed improvements.
Chi-Square (χ^2) Test	Recognition outcomes vs. dataset type	$\chi^2 = 15.8 – 24.6$	$p < 0.05$	0.05	A strong and statistically significant dependence exists between the recognition method and dataset type, indicating consistent performance across datasets.
One-Way ANOVA	Recognition rates across all tested algorithms	$F = 9.42$	$p < 0.01$	0.01	Significant variance among algorithms, demonstrating that KSRC_FRSMP outperforms others with statistically superior mean performance.

Source: Experimental results obtained by the author.

ANNEX

Algorithm 1: kernel sparse representation-based classifier (KSRC)

Input: Training data matrix $A$and a test sample y
Output: Predicted class label of y, denoted as $label(y) = arg\,min_i\, r_i(y)$
Steps:

1. Apply a kernel transformation to both the training matrix $A$and the test sample y.
2. Construct the corresponding pseudo-transformation matrix B.
3. Normalize the columns of $B^T K_x$and $B^T k(A, y)$so that each column has unit l_2-norm.
4. Solve the l_1-norm minimization problem to obtain the coefficient vector $\hat{x}$using the FRSMP method (see Algorithm 2).
5. For each class $i = 1, \ldots, C$, calculate the residuals:

$$r_i(y) = \| B^T k(A, y) - B^T K \delta_i(\hat{x}) \|_2^2$$

Assign $y$to the class corresponding to the smallest residual:

$$label(y) = arg\,min_i\, r_i(y)$$

Algorithm 2: Updating the Inverse of a Matrix When adding the Columns

Input:

- Sensing matrix $\boldsymbol{\Phi}_{i-1}$
- Inverse matrix B_{i-1}^{-1}
- New columns matrix U
- Support set T_i

Steps:

1. Compute $f_1 = \Phi_{i-1}^T U$
2. Compute $f_2 = B_{i-1}^{-1} f_1$
3. Compute $D = (U^T U - f_1^T f_2)^{-1}$
4. Compute $f_3 = f_2 D$
5. Update $F_{11}^{-1} = B_{i-1}^{-1} + f_2 D f_2^T$
6. Form the updated inverse:

$$B_i^{-1} = [F_{11}^{-1} \ -f_3 \ -f_3^T \ D \ ]$$

(corresponds to the update provided in Eq. (20))

7. Rearrange each column $u \in U$and the corresponding row of B_i^{-1}according to the support set $T_i$to ensure correct positioning.

Output:

- Updated inverse matrix B_i^{-1}after adding new columns

Algorithm 3: Updating the Inverse of a Matrix When Removing the Columns

Input

- Current sensing matrix at iteration $i-1$: $\boldsymbol{\Phi}_{i-1}$
- Inverse Gram matrix: $B_{i-1}^{-1} = (\Phi_{i-1}^T \Phi_{i-1})^{-1}$
- Set of columns to be added or removed

Output

- Updated inverse Gram matrix B_i^{-1}

Procedure

Case 1: Adding new columns

1. Let Udenote the matrix formed by the newly selected columns.
2. Augment the sensing matrix:

$$\boldsymbol{\Phi}_i = [\Phi_{i-1} \ \ U]$$

Construct the block Gram matrix associated with $\boldsymbol{\Phi}_i$.

3. Compute the intermediate matrix:

$$D = U^T U - U^T \Phi_{i-1} B_{i-1}^{-1} \Phi_{i-1}^T U$$

4. Update the inverse Gram matrix using a blockwise inversion derived from the Sherman–Morrison–Woodbury identity:

$$B_i^{-1} = [B_{i-1}^{-1} + B_{i-1}^{-1} \Phi_{i-1}^T U D^{-1} U^T \Phi_{i-1} B_{i-1}^{-1} \ -B_{i-1}^{-1} \Phi_{i-1}^T U D^{-1} \ -D^{-1} U^T \Phi_{i-1} B_{i-1}^{-1} \ D^{-1} \ ]$$

Case 2: Removing columns

1. Partition the sensing matrix as:

$$\boldsymbol{\Phi}_i = [\Phi_{i+1} \ \ P]$$

where Pcontains the columns to be discarded.

2. Extract the corresponding blocks from the current inverse Gram matrix.
3. Define:

$$Z = B_{i+1}^{-1} \Phi_{i+1}^T P$$

4. Update the inverse Gram matrix by eliminating the contribution of P:

$$B_{i+1}^{-1} = B_i^{-1} - Z D^{-1} Z^T$$

5. The resulting matrix $B_{i+1}^{-1}$is used in the next iteration.

Algorithm 4: The proposed Fast Reduced Sampling Matching Pursuit (FRSMP) Algorithm**Input**

- Measurement vector y
- Sensing matrix Φ
- Sparsity parameter s
- Threshold parameter t_s
- Stopping tolerance ε
- Maximum number of iterations I_{max}

Output

- Sparse coefficient vector $\hat{x}$

Procedure

1. **Initialization**
   - ✓ Set residual $r^0 = y$
   - ✓ Initialize support set $T^0 = \emptyset$
   - ✓ Set iteration index $i = 0$
2. **While** stopping criterion is not satisfied **do**
   - **a. Correlation Computation**
     Compute the correlation vector:
     $$g = \Phi^T r^i$$
   - **b. Adaptive Support Selection**
     - Identify a preliminary candidate set corresponding to the largest correlations.
     - Apply a thresholding rule:
       $$\Omega = \{j \ \ | \ | \ g_j \ | \geq t_s \cdot \frac{\| \ r_J \ \|_2}{\sqrt{d}}\}$$
     - Merge the selected indices with the previous support:
       $$T^i = T^{i-1} \cup \Omega$$
   - **c. Coefficient Estimation**
     - Form the submatrix Φ_{T^i}
     - Estimate the sparse coefficients via least squares:
       $$\hat{x}_{T^i} = (\Phi_{T^i}^T \Phi_{T^i})^{-1} \Phi_{T^i}^T y$$
     - Update the inverse using Algorithm 3 instead of recomputing it.
   - **d. Pruning Step**
     - Retain only the $s$largest-magnitude coefficients in $\hat{x}_{T^i}$
     - Remove the remaining indices from the support set
     - Update the inverse matrix accordingly.
   - **e. Residual Update**
     $$r^{i+1} = y - \Phi \hat{x}$$
   - **f. Increment iteration counter:** $i = i + 1$
2. **End While**
3. **Return** final sparse estimate $\hat{x}$