applied sciences

MDPI

Article

Orthogonalization of the Sensing Matrix Through Dominant Columns in Compressive Sensing for Speech Enhancement

Vasundhara Shukla * and Preety D. Swami

Department of Electronics and Communication Engineering, University Institute of Technology RGPV, Bhopal 462033, India; preetydswami@gmail.com
* Correspondence: shuklav2015@gmail.com

Abstract: This paper introduces a novel speech enhancement approach called dominant columns group orthogonalization of the sensing matrix (DCGOSM) in compressive sensing (CS). DCGOSM optimizes the sensing matrix using particle swarm optimization (PSO), ensuring separate basis vectors for speech and noise signals. By utilizing an orthogonal matching pursuit (OMP) based CS signal reconstruction with this optimized matrix, noise components are effectively avoided, resulting in lower noise in the reconstructed signal. The reconstruction process is accelerated by iterating only through the known speech-contributing columns. DCGOSM is evaluated against various noise types using speech quality measures such as SNR, SSNR, STOI, and PESQ. Compared to other OMP-based CS algorithms and deep neural network (DNN)-based speech enhancement techniques, DCGOSM demonstrates significant improvements, with maximum enhancements of 42.54%, 62.97%, 27.48%, and 8.72% for SNR, SSNR, PESQ, and STOI, respectively. Additionally, DCGOSM outperforms DNN-based techniques by 20.32% for PESQ and 8.29% for STOI. Furthermore, it reduces recovery time by at least 13.2% compared to other OMP-based CS algorithms.

Keywords: compressive sensing (CS); orthogonal matching pursuit (OMP); sensing matrix optimization; voice activity detection (VAD); speech enhancement; particle swarm optimization (PSO)

Citation: Shukla, V.; Swami, P.D. Orthogonalization of the Sensing Matrix Through Dominant Columns in Compressive Sensing for Speech Enhancement. *Appl. Sci.* **2023**, *13*, 8954. https://doi.org/10.3390/app13158954

Academic Editors: Valentino Santucci, Douglas O'Shaughnessy, Ying Shen, Cunhang Fan and Ya Li

Received: 21 April 2023
Revised: 20 June 2023
Accepted: 22 July 2023
Published: 4 August 2023

1. Introduction

In recent years, the domains of signal processing and information theory have shown significant interest in compressive sensing (CS) [1,2]. Compressive sensing, alternatively referred to as compressed sensing or compressive sampling, has emerged as a potent technique for signal processing, with extensive applications across multiple fields. By leveraging the innate sparsity or compressibility of signals, compressive sensing enables the acquisition and reconstruction of signals using considerably fewer measurements than conventional methods demand. This pioneering concept has transformed data acquisition and processing across diverse domains, including image and video processing, audio processing, medical imaging, wireless communication, etc. [3–7].

Conventional approaches often depend on uniformly sampling signals at the Nyquist rate, which can be computationally burdensome and require significant storage capacity. In contrast, compressive sensing enables sub-Nyquist sampling, thereby lowering the demands for data acquisition and storage while preserving the fidelity of the signal.

The CS suggests that, compared to the number of samples required by conventional Nyquist-based approaches, a signal can be reconstructed with fewer samples (observations) [8]. However, for CS to work properly, the input signal must be highly compressible, or more specifically, sparse. A sparse signal has few active (nonzero) components in relation to its length. The signals can exhibit this property either in their sampling domain or any other underlying transform domain such as Fourier, wavelet, curvelet, etc. In CS, there are two major components: encoding (sampling) and decoding (recovery). Let vector $x \in R^{N\times 1}$ denote N successive samples coming from a transducer. The vector x can be transformed

into a sparse vector $X = \psi x$ through a $N \times N$ sparse transforming matrix ψ. If X contains k non-zero elements, it is said to be k sparse. The reconstruction of the original data x from $y \in R^{M\times 1}$ measurements, where $y = \phi X = \phi\psi x$, requires $k << M << N$. Generating a suitable sensing matrix to improve the reconstruction process is one of the most challenging tasks in CS. The restricted isometry property (RIP) states that the performance of compressed sensing algorithms improves when the mutual coherence between the sensing matrix ϕ and the transforming matrix ψ decreases [9,10].

The rest of the paper is organized as follows: Section 2 briefly reviews the related work. Section 3 provides a detailed explanation of the proposed technique along with the theoretical background of the different processes used in this work. In Section 4, the evaluation results for the proposed technique with detailed analysis are presented. This section also includes a comparison of various methods. Finally, Section 5 concludes the paper.

2. Related Work

Finding an optimal CS matrix for compression and reconstruction is one of the most difficult problems in CS. Randomly constructing CS matrix elements that meet the RIP requirement does not produce the optimal solution. Because of this, random sensing matrices have been abandoned in favor of deterministic matrices in a number of research studies. The Ternary matrix [11], Walsh–Hadamard [12], and low-density parity checks [13] are examples of deterministic matrices that can be used in computations.

Another challenge is finding suitable algorithms for compressed speech enhancement and a domain of transformation that is suitable for sparse signals. Many basis functions are available for speech signals [14,15], like the Fourier Transform, Cosine Transform, and Wavelet Transforms [16].

There are several different types of sensing matrices; for example, the Gaussian random matrix, Bernoulli, Fourier, wavelets, etc. [14]. Several optimization techniques, such as l_1 minimization [17], orthogonal matching pursuit (OMP) [18], and compressive sampling matching pursuit (CoSaMP) [19], were also utilized to recover the compressively sampled speech signals. The study presented by Pilastri et al. [20] analyzed the performance of several CS reconstruction algorithms applied to image reconstruction and suggested that the l_1 minimization-based basis pursuit algorithm outperforms the other OMP methods.

As presented in [21], the basis pursuit algorithm also provides a slight edge over the CoSaMP for speech enhancement. Several CS-based methods for speech enhancement have also been proposed in [22–24]. Wu et al. [22,24] employed a random sampling matrix in their sensing process.

At present, the most accurate approach for determining the quality of speech is to conduct subjective listening tests. However, subjective evaluations of speech enhancement algorithms are costly and time-consuming, despite the fact that they are accurate, reliable, and performed under rigorous conditions [25–28]. Therefore, many efforts have been made to develop objective measures that are highly correlated with speech quality.

Several attempts have been made over the years to predict subjective speech quality using objective speech quality metrics [25]. Objective measures based on LPC, such as cepstrum distance measures (CEP) [29] and frequency-weighted segmental SNR (fwsegSNR) [25], are among the objective speech quality measures examined in [28], along with others, such as segmental SNR (segSNR) [30], weighted-slope spectral distance (WSS) [31], and PESQ [32,33].

The preceding discussion has highlighted that the basic compressive sensing idea has been modified in different ways to achieve better speech enhancement performance. Many ideas have been proposed for efficient reconstruction of the compressed signal [34], as shown in Figure 1. However, this work is primarily concerned with the efficient reconstruction of compressed signals while achieving better enhancement using greedy algorithms. Greedy algorithms recover the signal iteratively by choosing a local optimal solution at each iteration with the intention of eventually discovering the global optimal solution. In

general, greedy algorithms take the steps depicted in Figure 2. In each repetition, the quantity of the column selected varies slightly. Only one column is selected for each iteration of the matching pursuit (MP) [35], orthogonal matching pursuit (OMP), and matching pursuit based on least squares (MPLS) [36] algorithms.

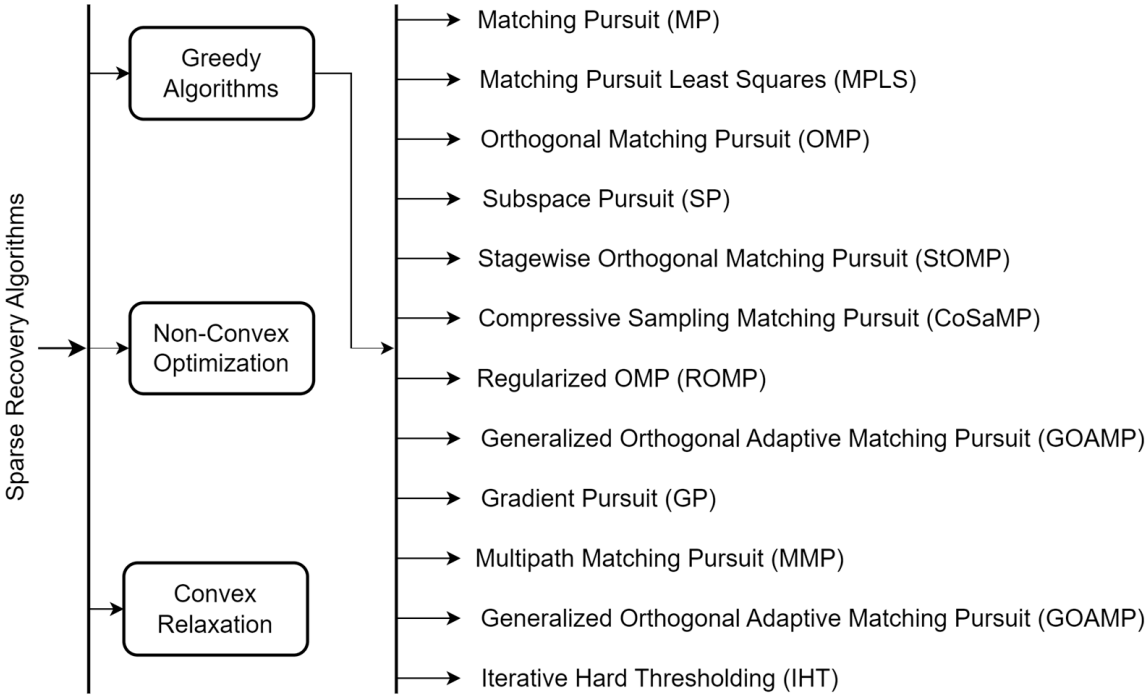

Figure 1. Sparse recovery algorithms.

In contrast, the subspace pursuit (SP) [37], stagewise-OMP (StOMP) [38], compressive sampling matching pursuit (CoSaMP) [19], regularized-OMP (ROMP) [39], generalized-OMP (GOMP) [40], generalized orthogonal adaptive matching pursuit (GOAMP) [41], constrained backtracking matching pursuit (CBMP) [42], generalized backtracking regularized adaptive matching pursuit (GBRAMP) [43], OMP algorithm based on singular value decomposition [44], enhanced block-based OMP pursuit [45], gradient pursuit (GP) [46], and multipath matching pursuit (MMP) [47], involve columns with projected values that exceed the thresholds selected by the algorithm. Other differences between the algorithms include the way the residual vector γ_{iter} is calculated and how the non-zero values of x_{est} are estimated. For instance, the MPLS and the subspace pursuit (SP) algorithms do not estimate x_{est} until the last step of their respective processes.

As the aforementioned discussion has highlighted, all the existing greedy algorithms use either single or groups of column selection. For every frame of the signal, iterations are required to find the dominating columns set ($\phi_{iter}^{dominant}$), as described in Figure 2. This makes the algorithm complex and time-consuming. To overcome this problem, we proposed an optimized measurement matrix generation in such a way that all dominating columns consisting of the original signal components are known in advance. Furthermore, the optimization minimizes the presence of noise components in these columns to achieve better signal enhancement.

Based on these investigations, this paper proposes a CS-based speech enhancement technique called the dominant columns group orthogonalization of the sensing matrix (DCGOSM).

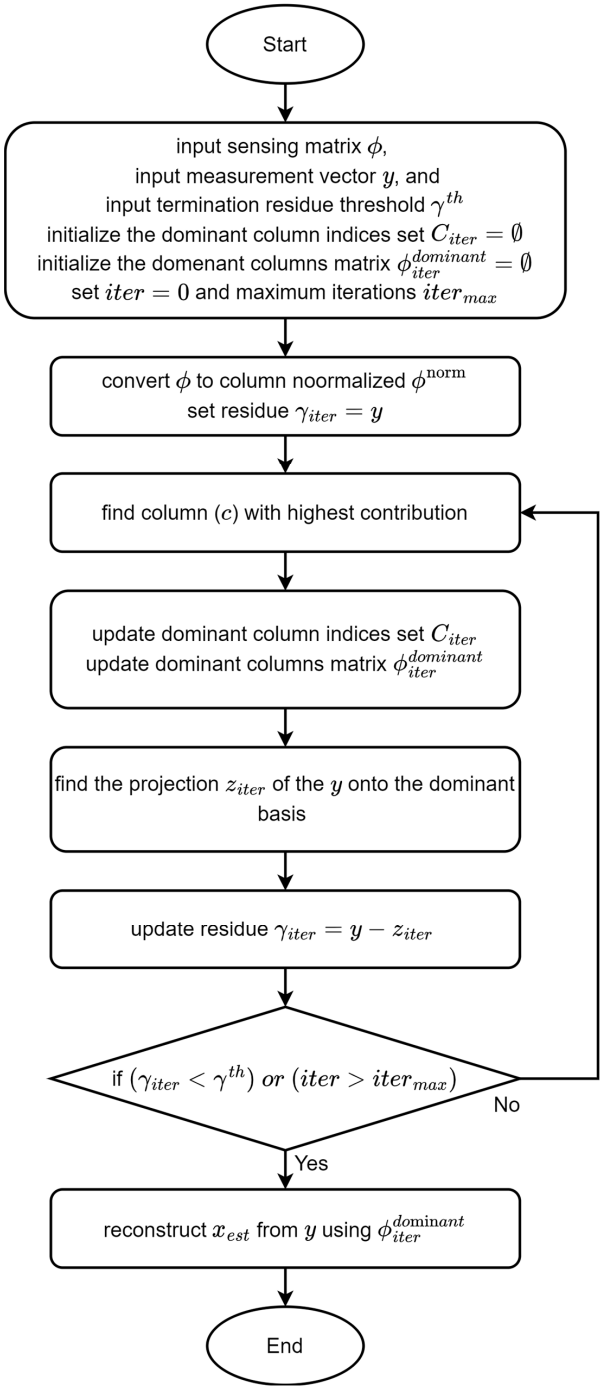

Figure 2. Flow chart for greedy algorithms.

3. Theoretical Background

3.1. Preprocessing

In signal processing, preprocessing plays an important role. Once experimental results are obtained, they are modeled to extract useful information. Generally, the data output is either too large, too small, or fragmented. As a part of preprocessing, the data are classified and processed accordingly [48]. In this work, we use the following normalization procedure to translate the signal into the range of $[-1,\ 1]$ [49]:

$$S_{norm}^{i} = 2 \times \left(\frac{S_{org}^{i} - S_{min}}{S_{max} - S_{min}} \right) - 1 \tag{1}$$

where S^i_{org}, S^i_{norm} are the i^{th} samples of the original and normalized (scaled) signal, respectively, and S_{max} and S_{min} are the maximum and minimum values achieved by the signal S_{org}, respectively.

3.2. Discrete Cosine Transform (DCT) and Inverse DCT (IDCT)

The DCT transforms a signal into the frequency domain, similar to the discrete Fourier transform (DFT). However, unlike to the DFT, the DCT relies only on real-valued cosine functions; therefore, it is simpler to compute [50]. Another property that makes the DCT superior to the DFT is its high spectral compaction. Therefore, a signal's DCT transformation includes more energy in fewer coefficients than other transformations, such as the DFT.

For sparse encoding of a signal, where a small number of nonzero samples are required, this is preferable. Due to the small number of DCT coefficients that hold substantial energy in a speech signal, all remaining coefficients can be reduced to zero without losing any significant information [51]. Therefore, the voice signal will be sparsely represented in this way.

Figure 3 illustrates the process of generating a sparse signal representation using the discrete cosine transform (DCT). In Figure 3a, a speech signal frame consisting of 32 samples is depicted. The corresponding DCT transform with 32 coefficients is shown in Figure 3b. The DCT coefficients are then arranged in descending order and presented in Figure 3c. Furthermore, Figure 3d shows the plot of energy in the largest K coefficients, where K takes values of 5, 10, 15, 20, 25, and 30. The plot in Figure 3d demonstrates that the top five DCT coefficients encompass 75.65% of the total energy of the 32 samples in the speech signal frame, while the top ten coefficients capture 95.36% of the total energy. These findings provide empirical evidence that only a small number (five to ten) of coefficients are sufficient to extract the most pertinent information from the complete frame. Therefore, zeroing the non-significant coefficients will produce a sparse signal with minimal information loss.

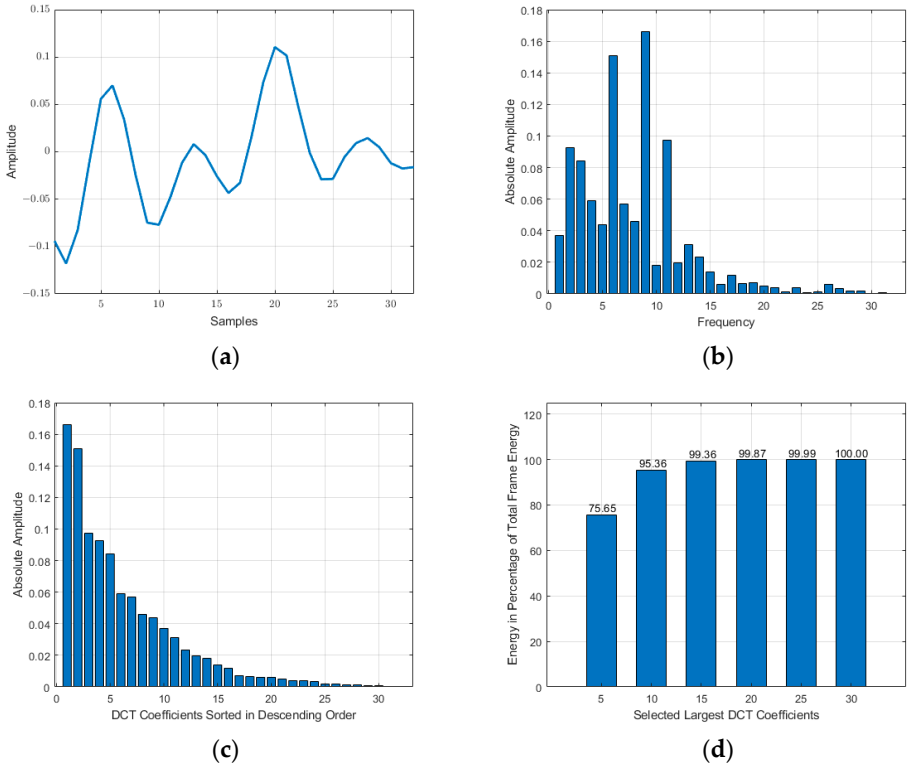

Figure 3. Illustration of the process of generating a sparse signal representation using the DCT. Where plot (**a**) shows the sampled speech signal, plot (**b**) shows the amplitudes of its DCT coefficients, plot (**c**) shows DCT coefficients arranged in descending order, and plot (**d**) shows the percentage of total energy contained by the largest K coefficients, where K takes values of 5, 10, 15, 20, 25, and 30.

There are several different iterations of DCT available, typically referred to as DCT-I to DCT-IV. They all have very subtle distinctions amongst one another [52]. Among these four variants, DCT-II is by far the most common and has thus been chosen for this work. The following equation can be used as a general formula for computing the DCT of a one-dimensional signal $f[n]$ with N number of samples:

$$g[k] = \begin{cases} \Lambda[k]\sqrt{\frac{2}{N}}\sum_{n=0}^{N-1} f[n]\cos\left[\frac{\pi k(2n+1)}{2N}\right], & 0 \le k < N \\ 0, & Otherwise \end{cases} \tag{2}$$

where $g[k]$ denotes the DCT coefficients vector of length N.

The IDCT is calculated as:

$$f[n] = \begin{cases} \sqrt{\frac{2}{N}}\sum_{k=0}^{N-1} \Lambda[k]g[k]\cos\left[\frac{\pi k(2n+1)}{2N}\right], & 0 \le k < N \\ 0, & Otherwise \end{cases} \tag{3}$$

where $\Lambda[k]$ is defined as:

$$\Lambda[k] = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0 \\ 1, & 1 \le k \le N \end{cases} \tag{4}$$

3.3. Framing and Deframing

In signal processing, framing refers to the process of splitting samples into small groups (blocks) known as "frames". Real-time block-based signal processing systems often use framing because it minimizes the processing and memory requirements by dividing one large chunk of samples into several small frames [53]. It also helps in extracting features from non-stationary signals (i.e., speech) by dividing them into small blocks during which they are assumed to be stationary. Figure 4 demonstrates how the frames are designed to overlap with one another to prevent data loss between successive frames [54]. In this figure F_a, F_b, and F_c represent the frames a, b, and c, respectively, and OL_{ab} and OL_{bc} are the overlapped frames.

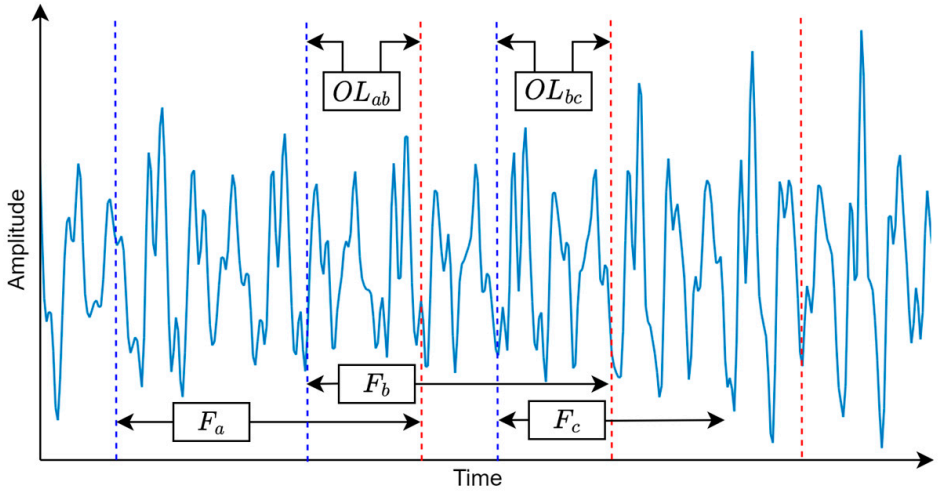

Figure 4. Framing process and frame overlapping, where blue and red dashed lines represent the starting and ending of the frames, respectively.

To avoid spectral distortions at the frame edges and to achieve seamless continuity among consecutive frames of speech, the Hamming window ($w(n)$) is multiplied with each frame [55]:

$$w(n) = 0.54 - 0.46 \cdot \cos\left(\frac{2\pi n}{N-1}\right), \; 0 \le n \le N \tag{5}$$

where N denotes the samples in the speech frame. The windowed signal $y(n)$ can be calculated as:

$$\mathrm{y(n)} = \mathrm{x(n)} \times \mathrm{w(n)},\ 0 \leq n \leq N \tag{6}$$

Windowing acts as a low-pass filter, enhancing the signal at the center and smoothing it at the edges.

3.4. Voice Activity Detector

Voice activity detection (VAD) is a process that detects the presence of a voice in a noisy signal. It is an essential component in speech processing applications like voice and speech detection, speech recognition, speech enhancement, and speaker recognition [56].

In this work, a VAD detector based on the work presented in [56] is utilized. Figure 5 shows an overview of the VAD. According to this model, the four different feature streams are used as follows:

1. Spectral Shape: The short-term power spectrum envelope of the speech signal.
2. Spectro–temporal modulations: The spectral scales and temporal rates of speech.
3. Voicing: The vibrations of the vocal cords (folds) that appear in speech as fundamental and harmonic components of the pitch (frequency).
4. Long-term variability: The variations in speech caused by successive phone generation.

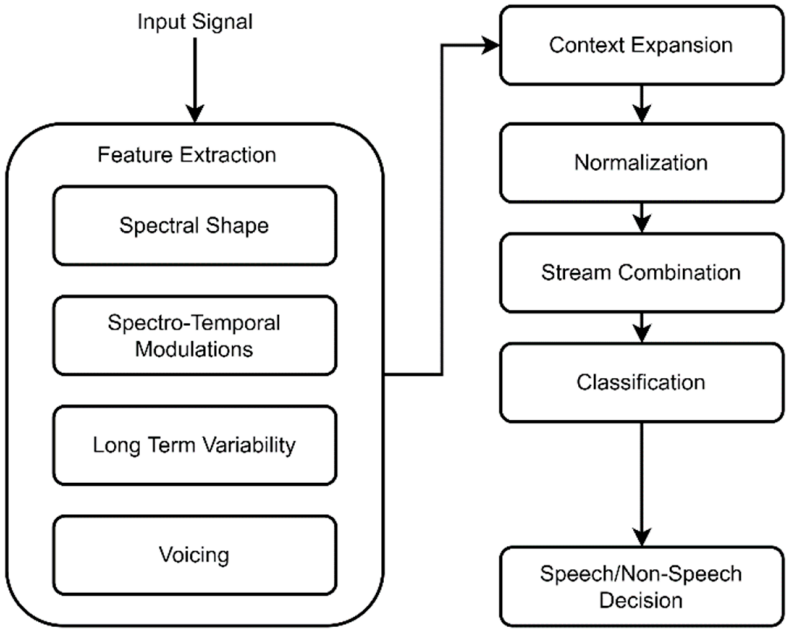

Figure 5. Overview of preprocessing steps used in VAD [24].

These feature streams are extracted from the input audio stream. These streams are divided into small frames, then the DCT of each feature stream is taken. The DCT contains a large number of coefficients. However, the first five DCT components are sufficient to extract the most relevant context information from those frames [56].

The resulting five-dimensional vectors from each of the four feature streams are normalized and combined into a single twenty-dimensional frame feature vector.

Finally, these frame feature vectors are used to train a multilayer perceptron (MLP) classifier to detect the presence of speech. Speech segments are detected by thresholding the ratio of speech and non-speech outputs of the MLP.

3.5. Particle Swarm Optimization (PSO)

Particle swarm optimization is a meta-heuristic searching algorithm that is used to obtain the optimal solution to numerical problems. PSO was originally proposed by Kennedy

and Elberhartin in 1995, [57] influenced by the social behavior of bird flocking or fish schooling. Currently, the algorithm has been well-tested on many complex mathematical problems and found to be efficient and fast. The pseudocode for PSO is presented in Algorithm 1. PSO initially populates particles (points in the solution space), and these particles are considered a probable solution to the optimization problem, which is given by a fitness function. In every iteration, a new position for the particles is calculated using several parameters such as the velocity vector, global best solution, local best solution, etc.

Figure 6 shows the velocity and position update procedure for i^{th} particle at time step t using Equations (7) and (8), respectively. p^i_{best} represents the particle's information gained from its past movements. Similarly, g_{best} represents the collective information gained by all the particles.

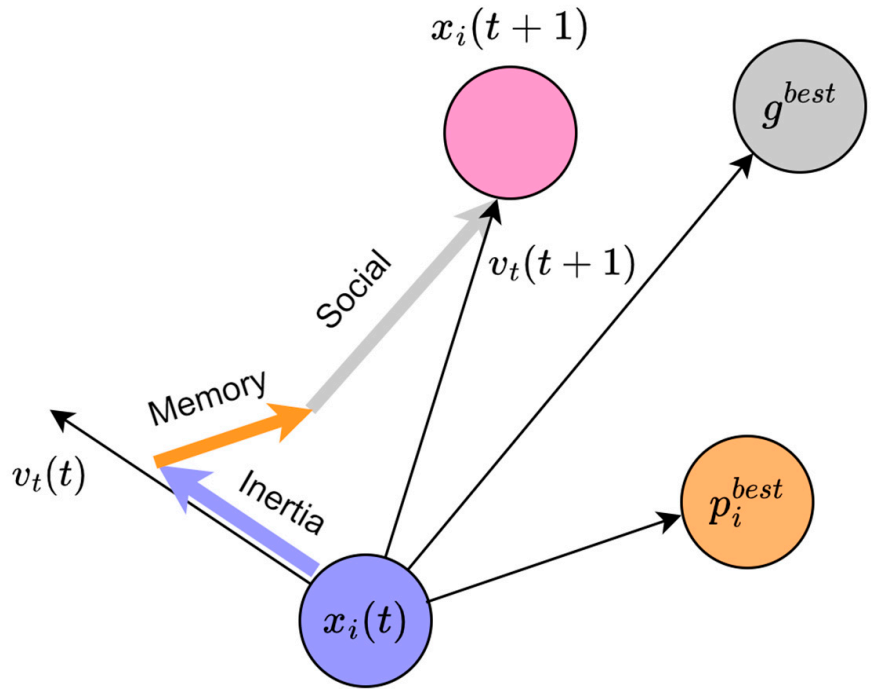

Figure 6. Particle velocity update procedure in PSO.

The velocity update of each particle is performed using the following equation:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot \left(p^i_{best} - x_i(t)\right) + c_2 \cdot \left(g_{best} - x_i(t)\right) \tag{7}$$

where c_1, and c_2 are priority factors used to configure the dominance of p^i_{best} and g_{best}. The new position $x_i(t+1)$ of the i^{th} particle is calculated as:

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{8}$$

In Equation (7), the first term $(w \cdot v_i(t))$ contains the variable w, which represents the inertia weight. Therefore, the whole term is known as the inertia component, which is responsible for maintaining the particle moment it initially possessed.

The value of w is usually maintained between 0.8 and 1.2. Setting up a smaller value for w reduces the algorithm's search space and converges quickly to the optima (sometimes trapping in local optima), while higher values force the algorithm to search over a wider space (which sometimes skips the global optima).

The second term in Equation (7), $c_1 \cdot \left(p^i_{best} - x_i(t)\right)$, serves as the particle's memory and is known as cognitive component. It forces the process to search in the area where the best solution can be found based on the particle's perspective. This coefficient (c_1) is usually set to 2.

The third term in Equation (7), $c_2 \cdot (g_{best} - x_i(t))$, is known as the social component. It forces the particle movements towards the area belonging to the best of each particle. This component is also set to 2.

3.6. Compressive Sensing (CS)

Compressive sensing (CS) is also referred as compressive sampling, sparse recovery, or compressed sensing. Conventional Shannon theorem-based sampled signal recovery algorithms have the following basic constraints [58]:

- The Shannon theorem states that the sampling rate needs to be more than or equal to twice the signal's bandwidth [59].
- Linear algebra stipulates that the length of the observed samples must be equal to or greater than the length of original signal [38].

Algorithm 1: PSO Algorithm

Initialize:
  Set initial positions $x_i(0) | \, i \in \{1, 2, 3, \ldots, N\}$ and velocities $v_i(0) | \, i \in \{1, 2, 3 \ldots, N\}$ of each of the N particles randomly within a specified range.
  Set intertia weight w.
  Set constant c_1 and c_2.
  Set p^i_{best} for each particle as the initial position.
  Set g_{best} as the best position among all particles.
Main:
  While termination criterion is not met do:
    For each particle do:
      Evaluate fitness of particle through objective function presented in eq. (18)
        If current position is better than p^i_{best}:
          Update p^i_{best} with current position
        If current position is better than g_{best}:
          Update g_{best} with current position
      For each particle do:
  Update particle velocity using:
    $v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (p_{best} - x_i(t)) + c_2 \cdot r_2 \cdot (g_{best} - x_i(t))$
  Update positions of particles using:
    $x_i(t+1) = x_i(t) + v_i(t+1)$
End while
Return g_{best} as the best solution found

Compressive sensing aims to overcome these limitations by stating that sparse signals can be reconstructed from incomplete samples. It enables the reconstruction of a compressed signal with only a few linear and non-adaptive measurements [1]. Additionally, unlike conventional signal compression methods, which use a sample-then-compress procedure, the CS achieves compression while sampling. The fundamental equation for recovering the CS signal is as follows:

$$[y]_{M \times 1} = [\phi]_{M \times N} \cdot [x]_{N \times 1} + [v]_{M \times 1} \tag{9}$$

where x, y, and v denote the sparse signal, observation vector, and noise, respectively, while ϕ denotes the sensing (recovery) matrix. To achieve proper compression and recovery through CS, the following conditions must be satisfied:

1. $M \ll N$, to ensure the compression [60,61].
2. x is required to be a k-sparse vector that satisfies $k << N$, where N denotes the length of x.
3. ϕ required to be a matrix having full rank.

The reconstruction of x from y using Equation (9) requires the estimation of ϕ-inverse:

$$x \approx \phi^{-1} y, \text{ neglecting the } \nu \tag{10}$$

Since ϕ is a non-square matrix ($M \neq N$), instead of inverse, the pseudo-inverse is used. Pseudo-inverses are defined for matrices with real or complex entries, and it is unique for these matrices [62]. The estimation of pseudo-inverse requires the decomposition of the phi matrix into three matrices using singular value decomposition (SVD) factorization as follows:

$$\phi = USV^T \tag{11}$$

where ϕ and U are an orthogonal matrix and S is a diagonal matrix containing singular values. Therefore, the pseudo-inverse of matrix ϕ through singular values is calculated as:

$$\phi^+ = VS^+U^T \tag{12}$$

where S^+ is obtained by replacing each non-zero singular value in S with its reciprocal. Using the above solution, the reconstruction of x can be achieved using following equation:

$$x = \left(VS^+U^T\right)y \tag{13}$$

Equation (13) may have an infinite number of solutions, but given the sparsity in x, the best solution could be obtained through l_0 norm minimization, as shown below:

$$\min\|x\|_0, \textit{ subject to } y = \phi x \tag{14}$$

where $\|\cdot\|_0$ denotes the l_0 norm operation. In the case of partial reconstruction, the above equation can be rewritten as:

$$\min\|x\|_1, \textit{ subject to } \|y = \phi x\|_2 \leq \gamma^{th} \tag{15}$$

where $\|\cdot\|_1$, $\|\cdot\|_2$ denotes the l_1, l_2 norm operations, respectively, and γ^{th} denotes the termination residue threshold.

3.7. Orthogonal Matching Pursuit (OMP)

Solving the Equations (14) and (15) using the l_0 norm is an NP-hard problem; therefore, an alternative approach known as the orthogonal matching pursuit (OMP) algorithm [18] could be used. However, OMP has the following prerequisites:

1. The signal x needs to be a k-sparse, where $k << N$.
2. The matrix ϕ is required to fulfill the condition $M_\phi < \left(\frac{1}{2k-1}\right)$, where M_ϕ denotes the mutual coherence of column vectors of matrix ϕ and is calculated as follows:

$$M_\phi = \max_{i \neq j} \frac{\left|\phi_i \cdot \phi_j\right|}{\left(\|x_i\|_2 \|x_j\|_2\right)} \tag{16}$$

where ϕ_i and ϕ_j denote the i^{th} and j^{th} columns of the ϕ matrix, respectively.

OMP is a greedy method that seeks to discover the solution for elements of x in one-by-one fashion through an iterative process [40]. The pseudocode for the OMP algorithm is presented in Algorithm 2.

Algorithm 2: OMP Algorithm

Inputs:

Input signal vector $\boldsymbol{x} \in \boldsymbol{R}^{N}$ must be k-sparse. /* sparse representation of speech frame using DCT */

Sensing matrix $\boldsymbol{\phi} \in \boldsymbol{R}^{M \times N}$, where $\boldsymbol{M_{\phi}} < \left(\frac{1}{2k-1}\right)$.

Termination residue threshold γ^{th}. /* unwanted or noise components */

Maximum number of iterations $\boldsymbol{iter_{max}}$.

Initialize:

$\boldsymbol{iter} = \mathbf{0}$ /* iteration counts */

$\boldsymbol{y} = \boldsymbol{\phi x}$ /* measurement vector creation */

$\gamma_{iter} = \boldsymbol{y}$ /* current residue. */

$\boldsymbol{C_{iter}} = \emptyset$ /*indices of dominant columns.*/

$\boldsymbol{\lambda_{iter}} = \varnothing$ /*contributions of dominant columns.*/

$\boldsymbol{\phi_{iter}^{dominant}} = \varnothing$ /*dominent columns matrix. */

$\boldsymbol{x_{est}} = \mathbf{0}_{\mathbf{1} \times N}$ /* estimated filtered $\boldsymbol{x}$. */

Main:

$\boldsymbol{\phi_i^{norm}} = \frac{\phi_i}{\|\phi_i\|}$, where $\boldsymbol{\phi_i} := \boldsymbol{col_i}(\boldsymbol{\phi})$ /* columns normalization of sensing matrix */

While $((\gamma_{iter} > \gamma^{th})$ ***or*** $(\boldsymbol{iter} < \boldsymbol{iter_{max}}))$ **do:**

$\lambda_c^{tmp} = \underset{c}{\mathbf{argmax}} \left| \gamma_{iter} \cdot \boldsymbol{\phi_i^{norm}} \right|$ /* finds the column (c) with highest contribution (λ_c^{tmp}). */

If $(c \notin \boldsymbol{C_{iter}})$ **then:** /* verifying that the column indices in not the dominant column list */

$\boldsymbol{C_{iter}} = \{\boldsymbol{C_{iter-1}}, \boldsymbol{c}\}$ /* adding column indices in the list */

$\boldsymbol{\lambda_{iter}} = \left\{\boldsymbol{\lambda_{iter-1}}, \lambda_c^{tmp}\right\}$ /* adding the contribution of the column */

$\boldsymbol{\phi_{iter}^{dominant}} = \left\{\boldsymbol{\phi_{iter-1}^{dominant}}, \boldsymbol{\phi_c^{norm}}\right\}$ /* update dominating column matrix */

Else

$\boldsymbol{\lambda_{iter,c}} = \boldsymbol{\lambda_{iter-1,c}} + \lambda_c^{tmp}$ /* update dominating columns contributions only as column indices already exists */

End If

$\boldsymbol{z_{iter}} = \left(\boldsymbol{\phi_{iter}^{dominant}} \cdot \left(\boldsymbol{\phi_{iter}^{dominant^T}} - \boldsymbol{\phi_{iter}^{dominant}}\right)^{+} \cdot \boldsymbol{\phi_{iter}^{dominant^T}}\right) \cdot \boldsymbol{y}$ /* finding the projection of the $\boldsymbol{y}$ onto the dominant basis */

$\gamma_{iter} = \boldsymbol{y} - \boldsymbol{z_{iter}}$ /* remaining residue. */

$\boldsymbol{iter} = \boldsymbol{iter} + \mathbf{1}$

End While

For $(\boldsymbol{i} = \mathbf{1}; \boldsymbol{i} \leq \boldsymbol{card}(\boldsymbol{C_{iter}});\ \boldsymbol{i} = \boldsymbol{i} + \mathbf{1})$ **do:**

For $(\boldsymbol{j} = \mathbf{1}; \boldsymbol{j} \leq \boldsymbol{card}(\boldsymbol{C_{iter}});\ \boldsymbol{j} = \boldsymbol{j} + \mathbf{1})$ **do:**

$\boldsymbol{\phi^{reduced}}(\boldsymbol{i}, \boldsymbol{j}) = \boldsymbol{\phi_{iter}^{dominent}}(:, \boldsymbol{i})^{T} \cdot \boldsymbol{\phi_{iter}^{dominent}}(:, \boldsymbol{j})$ /* generating reduced dimension sensing matrix through dominent columns */

End For

End For

$\boldsymbol{y^{reduced}} = \left(\boldsymbol{\phi^{dominant}}\right)^{T} \cdot \boldsymbol{y}^{T}$ /* generating reduced dimension $\boldsymbol{y}$ */

$\boldsymbol{x_{est}^{reduced}} = \left(\boldsymbol{\phi^{reduced}}\right)^{+} \cdot \boldsymbol{y^{reduced}}$ /* estimation of reduced dimension $\boldsymbol{x}$ from $\boldsymbol{y^{reduced}}$ */

For $(\boldsymbol{i} = \mathbf{1}; \boldsymbol{i} <= \boldsymbol{N}; \boldsymbol{i} = \boldsymbol{i} + \mathbf{1})$ **do:**

$\boldsymbol{x_{est}}(\boldsymbol{C_{iter}}(\boldsymbol{i})) = \boldsymbol{x_{est}^{reduced}}(\boldsymbol{i})$ /* filling up the full dimension $\boldsymbol{x_{est}}$ from $\boldsymbol{x_{est}^{reduced}}$ */

End For

Outputs:

$\boldsymbol{C_{iter}}$

$\boldsymbol{\lambda_{iter}}$

$\boldsymbol{\phi_{iter}^{dominant}}$

$\boldsymbol{x_{est}}$

3.8. Dominant Columns Group Orthogonalization of Sensing Matrix (DCGOSM)

As described in Algorithm 2, the OMP algorithm collects the coefficients of the sensing matrix's dominant columns while excluding the columns containing residue components.

Considering these characteristics of OMP, the orthogonalization of the dominant columns group of the sensing matrix for speech and noise signals can be achieved as follows:

1. Let x_s and x_n be the sparse representation of speech and noise signals, respectively.
2. If the sensing matrix is denoted by ϕ, then the compressed speech and noise signals can be obtained by $y_s = \phi x_s$ and $y_n = \phi x_n$, respectively.
3. The OMP algorithm can be used to recover the enhanced speech signal $x_s^{'}$ from $y_s^{'}$. The OMP finds the dominant columns, their contributions, and the reduced sensing matrix with only dominant columns (denoted by C_{iter}, λ_{iter}, and $\phi_{iter}^{dominent}$, respectively, in Algorithm 2) to estimate the enhanced speech signal while leaving the residue noise component columns intact.
4. If V_s and V_n are the reduced sensing matrix columns' contributions to the speech and noise signals obtained through OMP, respectively, then the orthogonalization of dominant columns group of the sensing matrix can be achieved by optimizing the columns of the sensing matrix, such that:

$$f_{obj} = minimize\left(\sum_{i=1}^{N_s}\sum_{j=1}^{N_n} V_s(:,i)\cdot V_n(:,j)\right) \tag{17}$$

where N_s and N_n are the number of columns in matrix V_s and V_n, respectively. The optimization is performed to minimize the f_{obj} value to its maximum extent or up to a certain termination threshold.

3.9. Proposed DCGOSM-Based Speech Enhancement Technique

Figure 7 shows an illustrative block diagram of the proposed approach. Referring to this diagram, the proposed method can be divided into two parts. The first part is used to obtain the required sensing matrix, and the second part is used to enhance noisy speech.

3.9.1. Process of Obtaining the Required Sensing Matrix

The process of obtaining the required sensing matrix involves obtaining an optimized sensing matrix with orthogonalized dominant columns for clean speech and noise signals. The speech enhancement performed using greedy algorithms assumes that speech signals exhibit a structured temporal pattern and a relatively narrowband spectral content, whereas the noise signals lack these properties. Therefore, when reconstruction is performed, the noise components equally spread over all columns of the sensing matrix, whereas the speech components become concentrated over a small number of columns (dominant columns, see Section 3.8). Therefore, when reconstruction is performed through dominant columns, the amount of the noise significantly reduces in the reconstructed speech. However, such approaches have following limitations:

A. They assume that the noise components spread equally over all columns or noise must be white noise. However, for sounds other than white noise, this distribution of components will not be possible. Therefore, for sounds other than white noise, the performance of these approaches will naturally decrease.
B. Since the dominant columns are not known initially, they need to be searched every time from all columns, which increases processing time.
C. The algorithms have no awareness of speech and noise signals, and the selection of dominant columns is done solely on the basis of contribution. The algorithms always assume that higher contribution is due to speech components. Therefore, during higher noise conditions, the algorithms may select columns with dominating noise components. The condition may worsen in non-white noise cases.

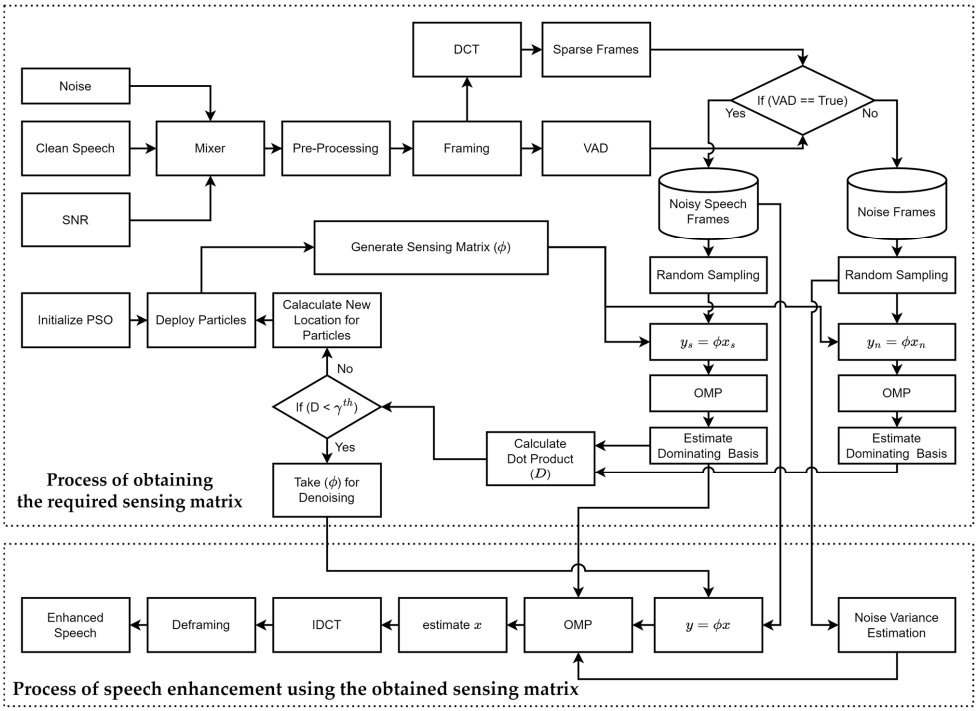

Figure 7. Block diagram of the proposed algorithm.

To overcome these limitations, the proposed DCGOSM approach obtains the sensing matrix based on the type of noise present in the noisy speech. The obtained sensing matrix guarantees separate dominating columns for clean speech and noise components. During the generation of the required sensing matrix, the dominant columns for the speech components are also stored; hence, during the enhancement process, the iterations will be performed only through these columns. The algorithm identifies speech and non-speech frames and obtains the dominant columns separately for speech and non-speech frames. Therefore, the dominant columns selected for the speech components are guaranteed to contain only speech components. The effect of this can also be seen in the results presented in Tables 1–3, where the proposed DCGOSM gives better relative improvement at higher noise conditions (lower SNRs).

Table 1. Performance comparison for white noise using the NOIZEUS dataset sample sp05.wav.

	Techniques							
	OMP		**CoSaMP**		**StOMP**		**Proposed**	
SNR (dB)	**SNR (dB)**	**SSNR (dB)**	**SNR (dB)**	**SSNR (dB)**	**SNR (dB)**	**SSNR (dB)**	**SNR (dB)**	**SSNR (dB)**
0	5.74	4.631	3.113	2.814	4.417	3.581	**6.380**	**5.874**
5	9.124	8.049	7.481	6.328	8.392	7.164	**9.627**	**9.157**
10	12.339	11.385	11.542	10.221	**11.916**	10.773	**12.639**	**12.298**
15	14.862	14.101	14.658	13.527	14.696	13.807	**14.993**	**14.745**
20	16.266	15.941	16.267	15.765	16.299	15.944	**16.390**	**16.386**
SNR (dB)	PESQ	STOI	PESQ	STOI	PESQ	STOI	PESQ	STOI
0	1.801	0.688	1.759	0.634	1.796	0.608	**2.296**	**0.748**
5	2.319	**0.764**	2.069	0.689	2.173	0.699	**2.795**	0.731
10	2.570	0.793	2.403	0.734	2.509	0.768	**3.092**	**0.809**
15	2.759	**0.837**	2.611	0.776	2.708	0.817	**3.201**	0.825
20	2.957	**0.877**	2.760	0.824	2.874	0.869	**3.220**	0.862

The bold font indicates the best obtained result for the selected measure in that row.

Table 2. Performance comparison for babble noise using the NOIZEUS dataset sample sp05.wav.

	Techniques									
	OMP		**CoSaMP**		**StOMP**		**K-SVDCS**		**Proposed**	
SNR (dB)	**SNR (dB)**	**SSNR (dB)**	**SNR (dB)**	**SSNR (dB)**	**SNR (dB)**	**SSNR (dB)**	**SNR (dB)**	**SSNR (dB)**	**SNR (dB)**	**SSNR (dB)**
0	2.560	2.385	0.758	1.638	1.388	1.853	--	3.80	**3.112**	3.072
5	6.201	5.835	5.193	4.602	5.681	4.965	--	3.50	**6.673**	**6.592**
10	10.119	9.345	9.349	8.471	9.532	8.708	--	3.06	**10.176**	**9.885**
15	13.118	12.438	13.119	12.195	13.074	12.168	--	1.60	**13.361**	**13.056**
20	15.383	14.873	15.453	14.890	15.156	14.685	--	-0.94	**15.516**	**15.325**
SNR (dB)	PESQ	STOI	PESQ	STOI	PESQ	STOI	PESQ	STOI	PESQ	STOI
0	1.780	0.652	1.917	**0.683**	1.969	0.644	1.96	0.66	**2.304**	0.615
5	2.216	**0.736**	2.251	0.726	2.308	0.715	2.28	0.72	**2.384**	0.689
10	2.434	**0.811**	2.513	0.759	2.513	0.771	2.52	0.79	**2.674**	0.748
15	2.606	**0.827**	2.646	0.774	2.709	0.805	2.69	0.81	**2.903**	0.785
20	2.667	**0.859**	2.772	0.816	2.853	0.847	2.85	0.83	**3.194**	0.824

The bold font indicates the best obtained result for the selected measure in that row.

Table 3. Performance comparison for f-16 noise using the NOIZEUS dataset sample sp05.wav.

	Techniques							
	OMP		**CoSaMP**		**StOMP**		**Proposed**	
SNR (dB)	**SNR (dB)**	**SSNR (dB)**	**SNR (dB)**	**SSNR (dB)**	**SNR (dB)**	**SSNR (dB)**	**SNR (dB)**	**SSNR (dB)**
0	3.288	2.490	1.292	1.744	2.056	2.036	**4.687**	**4.058**
5	7.220	6.101	5.639	4.656	6.266	5.129	**7.926**	**7.454**
10	10.689	9.631	9.844	8.445	10.264	8.901	**11.206**	**10.748**
15	13.772	12.803	13.599	12.358	13.517	12.366	**14.037**	**13.602**
20	15.686	15.095	15.711	14.933	15.629	14.949	**15.821**	**15.599**
SNR (dB)	PESQ	STOI	PESQ	STOI	PESQ	STOI	PESQ	STOI
0	1.684	0.659	1.887	0.629	1.942	0.591	**2.304**	**0.677**
5	2.101	0.751	2.183	0.682	2.268	0.680	**2.384**	**0.685**
10	2.509	**0.787**	2.496	0.745	2.577	0.756	**2.674**	0.772
15	2.654	**0.836**	2.632	0.756	2.747	0.808	**2.903**	0.791
20	2.838	**0.862**	2.747	0.796	2.878	0.839	**3.194**	0.808

The bold font indicates the best obtained result for the selected measure in that row.

The process of obtaining the required sensing matrix involves the following steps:

1. In the first step, the clean speech sample is mixed with the noise using the mixer block, which adjusts the amplitude of the noise signal according to the given SNR and then adds it to the clean speech to generate the noisy speech sample.
2. The noisy speech sample is passed through the preprocessing block, which normalizes the signal. Normalization helps to reduce the impact of variations in recording conditions, microphone characteristics, and speaker differences. By normalizing the speech signals, the relative differences in amplitude caused by these factors are minimized, making the subsequent processing algorithms more robust and less sensitive to such variations.
3. Subsequently, the signal is divided into frames using overlapping Hamming windows. Considering the block-based processing nature of the CS, these windows are required to divide the speech signal into frames. This process also reduces the computational complexity of subsequent processing steps, and the analysis can be performed on smaller segments, reducing the amount of data to be processed and

improving efficiency. The framing of the signal causes spectral leakage at the edges of the frame. The overlapping Hamming windows are used to mitigate spectral leakage by smoothly transitioning between adjacent frames, reducing the abrupt changes at the frame boundaries.

4. Signal sparsity is the fundamental condition for CS. Therefore, to make the frame sparse, each frame is converted into the frequency domain using the discrete cosine transform (DCT), then the insignificant DCT components within the transformed frame are set to zero. The term "significance" refers to the components that encompass the most energy, specifically those exceeding 90%. This objective can be accomplished by strategically selecting the highest 25% of the coefficients.
5. The non-transformed speech frames are sent to the VAD black for identification of the speech and non-speech frames. In addition, the preliminary noise variance (v_n) estimation is performed using the initial frame, which is considered to carry only noise.
6. The noisy speech and non-speech frames are then grouped into separate databases for DB_{speech} and $DB_{non-speech}$. This helps in obtaining dominant columns separately for noisy speech and non-speech frames.
7. To minimize the optimization time, only a small number of frames are chosen from these databases. The selected noise frames are further used for the noise variance estimation (v_n^{final}), which is used during the final enhancement process. Estimating the noise variance from the multiple selected frames reduces the error possibilities, especially in the cases where the noise is not uniformly distributed throughout the entire signal duration.
8. PSO is initialized according to the number of particles and maximum generations. Each particle p_i^j (here, the superscript j represents the generation, and the subscript i represents the particle) is defined as an $M \times N$ dimension row vector, where each element is bounded within the range of $[-1,\ 1]$.

$$p_i^j = \left[e_1^i,\ e_2^i,\ \ldots,\ e_{M\times N}^i\right],\ where\ \forall i,\ j : \left\{-1 \le e_j^i \le 1\right\} \tag{18}$$

9. Each of these particles (p_i^j) represents a sensing matrix (ϕ_i^j) when converted into an M rows and N columns matrix:

$$\phi^i = \begin{bmatrix} e_1^i & e_2^i & \cdots & e_N^i \\ e_{N+1}^i & e_{N+2}^i & \cdots & e_{2\times N}^i \\ \vdots & \vdots & \ddots & \cdots \\ e_{(M-1)\cdot N+1} & e_{(M-1)\cdot N+2} & \cdots & e_{M\times N}^i \end{bmatrix}, \tag{19}$$

where

$$\forall i, j : \left\{-1 \le e_j^i \le 1\right\}$$

10. However, before using this matrix as a sensing matrix, it is required to make ϕ_i^j a sample from the uniform spherical ensemble (USE). This is done by dividing each element of each column of the matrix by the column's l_2 norm:

$$\phi^i = \begin{bmatrix} \frac{e_1^i}{r_1} & \frac{e_2^i}{r_2} & \cdots & \frac{e_N^i}{r_N} \\ \frac{e_{N+1}^i}{r_1} & \frac{e_{N+2}^i}{r_2} & \cdots & \frac{e_{2\times N}^i}{r_N} \\ \vdots & \vdots & \ddots & \cdots \\ \frac{e_{(M-1)\cdot N+1}}{r_1} & \frac{e_{(M-1)\cdot N+2}}{r_2} & \cdots & \frac{e_{M\times N}^i}{r_N} \end{bmatrix}, \tag{20}$$

where

$$r_i = \left\| \left[e^i_{0\times N+j},\ e^i_{1\times N+j}, \ldots, e^i_{(M-1)\times N+j} \right] \right\|_2 = \sqrt{\sum_{k=0}^{M-1} \left| e^i_{k\times N+j} \right|^2}$$

11. Using the sensing matrix $\left(\phi^j_{i,USE}\right)$ formed by i^{th} particle p^j_i, the observation vector $y^i_s = \phi^j_{i,USE} x_s$ for speech, and $y^i_n = \phi^j_{i,USE} x_n$ for noise, frames are generated.
12. The OMP is applied to y^i_s and y^i_n independently to reconstruct x_s and x_n, respectively. However, our aim is not to find x_s and x_n, but to find the dominating basis vectors (columns) V^i_s and V^i_n in ϕ^i_{USE} for y^i_s and y^i_n. The OMP uses v_n as the termination threshold.
13. Our aim is to make $\phi^j_{i,USE}$ a dominant columns group orthogonal sensing matrix where the columns used to represent the speech signals and noise signals are different. Therefore, speech signals can be extracted from the noisy samples using only the speech dominant columns group.
14. To match the above-mentioned condition, the $\phi^j_{i,USE}$ matrix must satisfy the condition $f_{obj} = 0$ (Equation (17)).
15. However, the condition $f_{obj} = 0$ is only possible for the ideal cases. Therefore, the optimization is performed to minimize the f_{obj} value to its maximum extent or up to a certain termination threshold (γ^{th}) (Equation (18)).
16. $f^j_{i,obj}$ is calculated through $\phi j_{i,USE}$ for each particle p^j_i.
17. Once the $f^j_{i,obj}$ is calculated for each particle in the current generation, based on the $f^j_{i,obj}$ values obtained from all the generations (1 to j), the $p_{i,best}$ (particle's best),\ and g_{best} (global best) values are estimated.
18. Based on these estimations, the new positions of particles p^{j+1}_i are calculated as described in Equations (7) and (8).
19. PSO repeats the process until the proper $\phi^j_{i,USE}$ matrix is generated.
20. After completion of this process, we obtain the sensing matrix ($\phi^j_{i,USE}$), final noise variance (v^{final}_n), dominating columns for speech signal components (V^i_s), and noisy-speech frames database (DB_{speech}).

3.9.2. Process of Speech Enhancement Using the Obtained Sensing Matrix

1. Speech enhancement is performed only on the noisy speech frames separated by VAD during the generation of sensing matrix. Additionally, the enhancement is performed on a frame-by-frame basis using the obtained $\phi^j_{i,USE}$ matrix, the final noise variance (v^{final}_n), and the OMP algorithm. However, because the dominating columns for speech signal components (V^i_s) are already known, the OMP is only iterated through these columns instead of all the columns of ϕ^{norm}_i.
2. The enhanced speech frames are then converted into the time domain by taking IDCT and finally converted into a single enhanced speech stream through the de-framing operation.

4. Performance Evaluation and Result Analysis

4.1. Sensing Matrix Optimization

The idea behind the proposed approach is to isolate the components of speech and noise signals into different columns of a sensing matrix. The degree of isolation is reflected by the fitness function of the optimization algorithm, where a lower value reflects a higher level of isolation. To validate this concept, we have shown the column contributions of the sensing matrix for speech and noise frames before (Figure 8) and after (Figure 9) optimization. The convergence plot of the optimization process is also shown (Figure 10) to link the fitness function and the level of isolation.

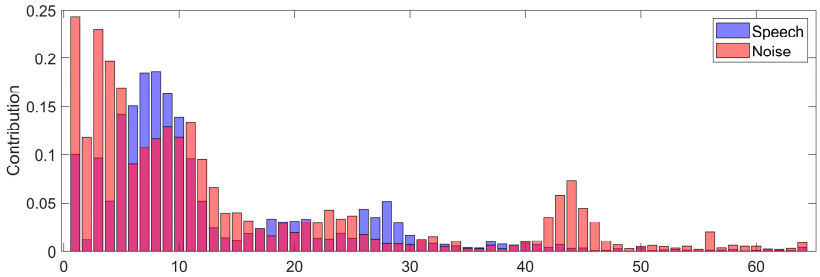

Figure 8. Column contributions for speech components (blue bars) and noise components (red bars) in the non-optimized sensing matrix.

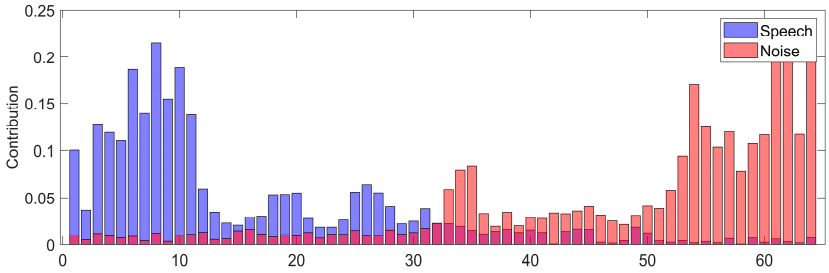

Figure 9. Column contributions for speech components (blue bars) and noise components (red bars) in the optimized sensing matrix.

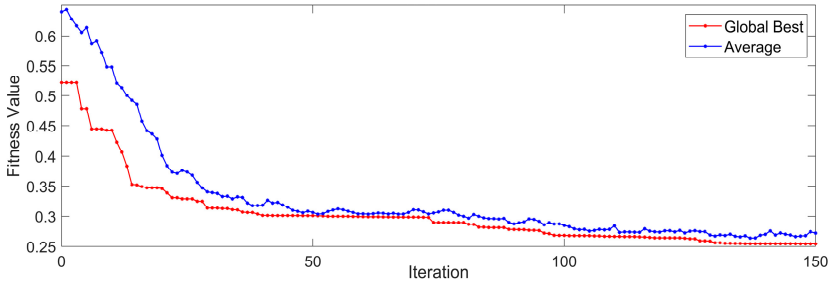

Figure 10. Sensing matrix optimization convergence plot showing global best (red) and average (blue) fitness value for each iteration.

The convergence plot shows that, when the optimization begins, the average and best fitness values were 0.64 and 0.52, respectively, which resulted in highly overlapping column contributions, whereas at the end of the optimization, the average and best fitness values were 0.27 and 0.25, respectively, which resulted in well-separated column contributions. Figure 9 also shows that after optimization, 95% of the total signal energy was concentrated in half the number of columns.

4.2. Speech Quality Analysis

The proposed DCGOSM-based speech enhancement technique was compared with four baseline methods: OMP [18], CoSaMP [19], StOMP [38], and the K-SVD-based CS technique (K-SVDCS) [63] (note: this technique was only compared for babble noise, as the paper only provided the results for this noise). Additionally, it was compared with some other methods based on human perception-related objective functions, such as DNN-MSE [64], DNN-PMSEQ [65], BLSTM-MSE [64], BLSTM-PMSQE [65], and DNN-Quality-Net [64].

To evaluate the performance of the proposed algorithm, the NOISEX-92 [66] and NOIZEUS [67] datasets were used. The clean speech samples were taken from the NOIZEUS dataset, which contains 30 clean speech files (sp01.wav to sp30.wav) with a single channel (mono), 16-bit PCM, sampled at 8 kHz, and encoded in the Windows "wav" format. The noise samples were taken from the NOISEX-92, which contains white noise, babble, and

f-16 noise files with a single channel sampled using a 16-bit analog-to-digital converter (A/D) at a sampling rate of 19.98 kHz. These samples were encoded in the MATLAB "mat" format. However, for the experimental analysis, we downsampled all signals to 8 kHz.

The clean speech signal was mixed with one of the noise signals at a time to evaluate the performance of the proposed technique. The amplitude of the noise was modified before mixing according to the required SNR. The noisy speech was then processed through the enhancement algorithm to obtain the enhanced speech. This process is depicted in Figure 11, which shows the time domain plots of the "sp05.wav" clean speech sample, the "babble" noise, and the noisy speech signal generated by corrupting the clean speech signal with babble noise, and their respective spectrogram plots (Figure 11b,d,f). The blue bars in the spectrogram of the enhanced speech (Figure 11f) depict the events where VAD detected non-speech components. These bars were mostly synchronized with the silence events in the clean speech (Figure 11a), which demonstrate that the VAD is capable of detecting non-speech signals even in the presence of noise (Figure 11c).

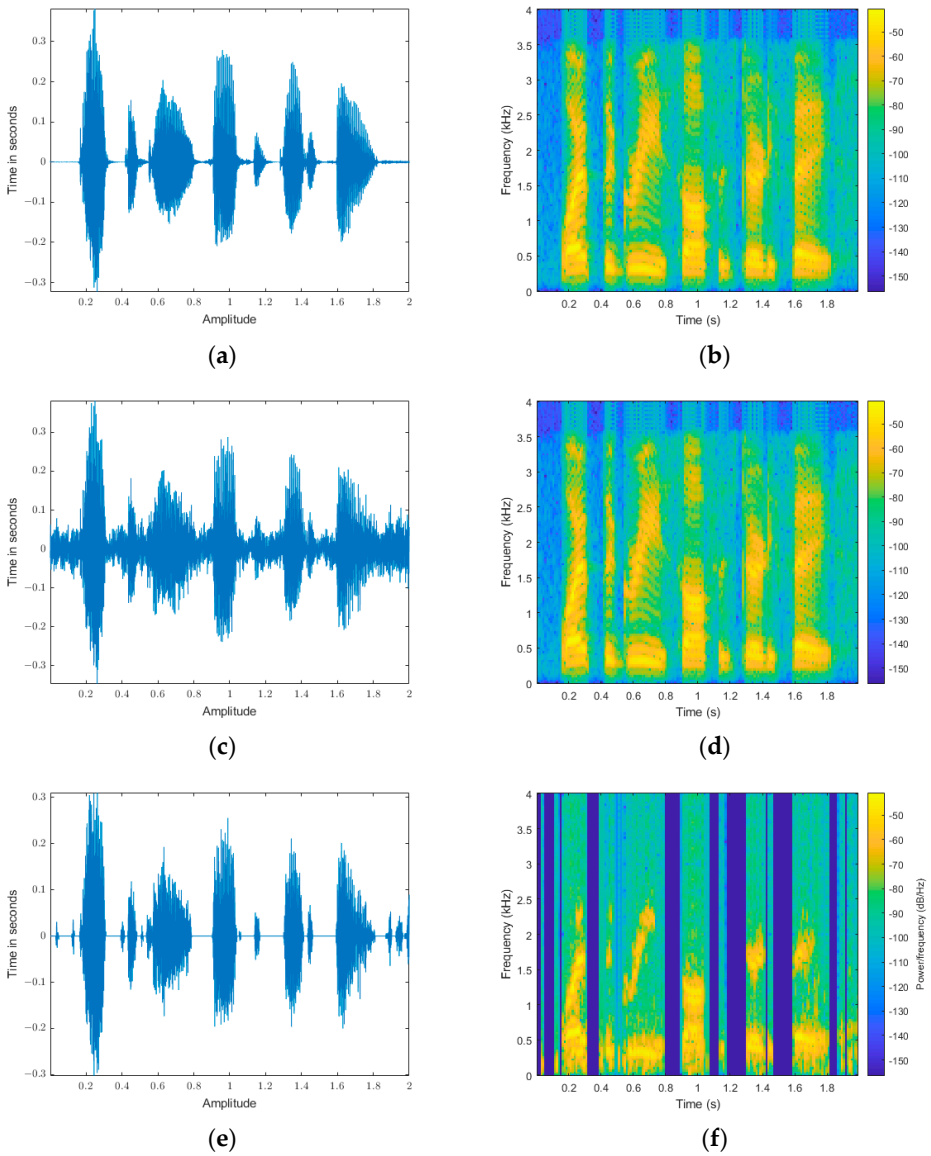

Figure 11. Plots for (**a**) clean speech, (**c**) noisy speech with 5 dB SNR corrupted by babble noise, (**e**) speech samples enhanced by the proposed algorithm with their respective spectrogram plots in (**b**,**d**,**f**).

Finally, to compare the performance of each algorithm, four objective measures, the signal-to-noise ratio (SNR), segmental SNR (SSNR), short-time objective intelligibility (STOI), and perceptual evaluation of speech quality (PESQ), were utilized.

The performance comparison of the proposed technique with different variants of the OMP algorithm for white, babble, and f-16 noises is presented in Tables 1–3, respectively. These results demonstrate the superior performance of the proposed technique over competitors for SNR, SSNR, and PESQ, irrespective of the noise type and magnitude. However, for STOI, the proposed algorithm performed better under low SNR conditions.

A further in-depth analysis of these tables showed that the proposed technique achieved the highest SNR improvement of 42.54% for f-16 noise, the second-highest improvement of 21.56% for babble noise, and the third-highest improvement of 11.03% for white noise at 0 dB of SNR. The minimum improvement of 0.41% was achieved for babble noise at 20 dB. Similar behavior was observed for SSNR; however, at 0 dB of SNR, the higher improvements were 62.97%, 28.80%, and 26.84% for f-16, babble, and white noise, respectively. White noise achieved the lowest improvement of 2.77% at 20 dB.

In comparison to SNR and SSNR, the proposed algorithm showed different behavior for PESQ and achieved the three highest improvements for white noise at 0 dB, 5 dB, and 10 dB SNR, which were 27.48%, 20.53%, and 20.31%, respectively. The fourth-highest improvement of 18.64% was achieved for f-16 noise, followed by 17.01% for babble noise, both at 0 dB SNR.

For STOI, the only significant improvement of 8.72% was achieved for white noise at 0 dB SNR, followed by 2.73% for f-16 at 0 dB for SNR.

It is also worth mentioning that for every noise, the proposed algorithm always achieved the best SNR, SSNR, and PESQ, except for the babble noise at 0 dB SNR, where the K-SVDCS achieved the best SSNR.

To further validate the performance of the proposed algorithm, a comparison with non-OMP (non-CS) algorithms for PESQ and STOI measures is presented in Tables 4 and 5, respectively. These results show that the proposed technique performed better at all SNRs except 18 dB, where DNN-Quality-Net improved by 3.96%. However, the proposed algorithm achieved the highest PESQ improvement of 20.32% at −6 dB. For STOI, the BLSTM (MSE) performed better for 18 dB, 12 dB, and 6 dB, while for 0 dB and −6 dB, the proposed technique performed better and achieved a maximum improvement of 8.29% at −6 dB SNR.

Table 4. PESQ comparison for white noise using the NOIZEUS dataset sample sp05.wav.

	Techniques					
SNR (dB)	**DNN (MSE) [64]**	**DNN-PMSQE [65]**	**BLSTM (MSE) [64]**	**BLSTM (PMSQE) [65]**	**DNN-Quality-Net [64]**	**Proposed**
18	2.810	3.082	3.287	2.899	**3.377**	3.243
12	2.576	2.819	2.908	2.777	3.010	**3.024**
6	2.275	2.497	2.504	2.578	2.614	**2.777**
0	1.912	2.111	2.065	2.261	2.171	**2.337**
−6	1.530	1.711	1.569	1.865	1.671	**2.244**

The bold font indicates the best obtained result for the selected measure in that row.

Table 5. STOI comparison for white noise using the NOIZEUS dataset sample sp05.wav.

	Techniques					
SNR (dB)	**DNN (MSE) [64]**	**DNN PMSQE [65]**	**BLSTM (MSE) [64]**	**BLSTM (PMSQE) [65]**	**DNN-Quality-Net [64]**	**Proposed**
18	0.855	0.886	**0.972**	0.882	0.966	0.842
12	0.831	0.865	**0.942**	0.867	0.937	0.821
6	0.788	0.822	**0.885**	0.836	0.882	0.815
0	0.715	0.741	0.796	0.773	0.794	**0.807**
−6	0.604	0.615	0.663	0.667	0.663	**0.718**

The bold font indicates the best obtained result for the selected measure in that row.

4.3. Recovery Time Analysis

A reconstruction time comparison of the proposed algorithm with the different OMP-based reconstruction algorithms is shown in Table 6. The results show that the proposed algorithm consistently outperformed the other three algorithms in terms of execution time across all the configurations. In Configuration (8, 32, 64), the proposed algorithm achieved an execution time of 0.3261, which was lower than the execution times of OMP (0.7956), CoSaMP (0.8071), and StOMP (0.3439). Similarly, in Configuration (16, 64, 128), the proposed algorithm (0.3527) demonstrated a lower reconstruction time compared to OMP (0.9434), CoSaMP (0.7652), and StOMP (0.3694). This trend continued in Configurations (32, 128, 256) and (64, 256, 512), where the proposed algorithm consistently exhibited the lowest reconstruction time among the four algorithms, followed by the StOMP as the second best, while the OMP took the longest time. In terms of percentage improvements, the proposed algorithm achieved a maximum reduction of 70% in reconstruction time for the single column selection algorithm (OMP), while the maximum improvements over the multiple column selection algorithms were 59% and 13% for CoSaMP and STOMP, respectively. The results also show that the reconstruction time increased with the sensing matrix size.

Table 6. Reconstruction time comparison (in seconds).

	Techniques			
Configuration (*k*,*m*,*n*)	**OMP**	**CoSaMP**	**StOMP**	**Proposed**
(8,32,64)	0.7956	0.8071	0.3439	**0.3261** *
(16,64,128)	0.9434	0.7652	0.3694	**0.3527**
(32,128,256)	1.3928	1.2097	0.4637	**0.4165**
(64,256,512)	2.8048	2.0487	0.9845	**0.8542**

The bold font indicates the best obtained result for the selected measure in that row. * k = sparsity, m = rows in sensing matrix, n = columns in sensing matrix.

5. Conclusions

This paper proposes a compressive sensing-based speech enhancement technique that uses the dominant columns group orthogonalization of the sensing matrix (DCGOSM). The proposed method attempts to find the sensing matrix in such a way that each column can either contain the noise component or the clean speech component, which is obtained by calculating the dot product of column contributions for the noise and speech samples. The proposed technique greatly reduces the speech recovery time by iterating only through the speech-dominating columns of the sensing matrix, as opposed to other OMP-based techniques, which require iterations through all columns of the sensing matrix. Another advantage of the proposed technique is that it provides uniform improvement for different quality measures, unlike the other sensing matrix optimization techniques that only improve the measures defined in the objective function. In this work, the PSO is adopted for the optimization of the sensing matrix. However, in the future, other optimization algorithms can also be tested. Furthermore, the OMP can be replaced with multiple column selection approaches such as CoSaMP and STOMP.

Author Contributions: Methodology, V.S.; Software, V.S.; Validation, V.S.; Formal analysis, P.D.S.; Supervision, P.D.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used are publicly available.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Donoho, D.L. Compressed Sensing. *IEEE Trans. Inf. Theory* **2006**, *52*, 1289–1306. [CrossRef]
2. Candes, E.J.; Romberg, J.; Tao, T. Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information. *IEEE Trans. Inf. Theory* **2006**, *52*, 489–509. [CrossRef]
3. Rani, M.; Dhok, S.B.; Deshmukh, R.B. A Systematic Review of Compressive Sensing: Concepts, Implementations and Applications. *IEEE Access* **2018**, *6*, 4875–4894. [CrossRef]
4. Xia, K.; Pan, Z.; Mao, P. Video Compressive Sensing Reconstruction Using Unfolded LSTM. *Sensors* **2022**, *22*, 7172. [CrossRef]
5. Vanjari, H.B.; Kolte, M.T. Comparative Analysis of Speech Enhancement Techniques in Perceptive of Hearing Aid Design. In Proceedings of the Third International Conference on Information Management and Machine Intelligence, Jaipur, India, 23–24 December 2021; Springer Nature: Singapore, 2023; pp. 117–125.
6. Calisesi, G.; Ghezzi, A.; Ancora, D.; D'Andrea, C.; Valentini, G.; Farina, A.; Bassi, A. Compressed Sensing in Fluorescence Microscopy. *Prog. Biophys. Mol. Biol.* **2022**, *168*, 66–80. [CrossRef]
7. Kwon, H.-M.; Hong, S.-P.; Kang, M.; Seo, J. Data Traffic Reduction with Compressed Sensing in an AIoT System. *Comput. Mater. Contin.* **2022**, *70*, 1769–1780. [CrossRef]
8. Shannon, C.E. Communication in the Presence of Noise. *Proc. IRE* **1949**, *37*, 10–21. [CrossRef]
9. Donoho, D.L.; Stark, P.B. Uncertainty Principles and Signal Recovery. *SIAM J. Appl. Math.* **1989**, *49*, 906–931. [CrossRef]
10. Candès, E.J.; Romberg, J.K.; Tao, T. Stable Signal Recovery from Incomplete and Inaccurate Measurements. *Commun. Pure Appl. Math.* **2006**, *59*, 1207–1223. [CrossRef]
11. Amini, A.; Marvasti, F. Deterministic Construction of Binary, Bipolar, and Ternary Compressed Sensing Matrices. *IEEE Trans. Inf. Theory* **2011**, *57*, 2360–2370. [CrossRef]
12. Ben-Haim, Z.; Eldar, Y.C.; Elad, M. Coherence-Based Performance Guarantees for Estimating a Sparse Vector Under Random Noise. *IEEE Trans. Signal Process.* **2010**, *58*, 5030–5043. [CrossRef]
13. Baraniuk, R.; Davenport, M.; DeVore, R.; Wakin, M. A Simple Proof of the Restricted Isometry Property for Random Matrices. *Constr. Approx.* **2008**, *28*, 253–263. [CrossRef]
14. Abrol, V.; Sharma, P.; Budhiraja, S. Evaluating Performance of Compressed Sensing for Speech Signals. In Proceedings of the 2013 3rd IEEE International Advance Computing Conference (IACC), Ghaziabad, India, 22–23 February 2013; pp. 1159–1164.
15. Xu, S.-F.; Chen, X.-B. Speech Signal Acquisition Methods Based on Compressive Sensing. In *Systems and Computer Technology*; CRC Press: Boca Raton, FL, USA, 2015; ISBN 978-0-429-22578-9.
16. Swami, P.D.; Sharma, R.; Jain, A.; Swami, D.K. Speech Enhancement by Noise Driven Adaptation of Perceptual Scales and Thresholds of Continuous Wavelet Transform Coefficients. *Speech Commun.* **2015**, *70*, 1–12. [CrossRef]
17. Donoho, D.L. For Most Large Underdetermined Systems of Linear Equations the Minimal ℓ1-Norm Solution Is Also the Sparsest Solution. *Commun. Pure Appl. Math.* **2006**, *59*, 797–829. [CrossRef]
18. Yang, H.; Hao, D.; Sun, H.; Liu, Y. Speech Enhancement Using Orthogonal Matching Pursuit Algorithm. In Proceedings of the 2014 International Conference on Orange Technologies, Xi'an, China, 20–23 September 2014; pp. 101–104.
19. Needell, D.; Tropp, J.A. CoSaMP: Iterative Signal Recovery from Incomplete and Inaccurate Samples. *Appl. Comput. Harmon. Anal.* **2009**, *26*, 301–321. [CrossRef]
20. Pilastri, A.L.; Tavares, J.M.R. Reconstruction Algorithms in Compressive Sensing: An Overview. In Proceedings of the 11th edition of the Doctoral Symposium in Informatics Engineering (DSIE-16), Porto, Portugal, 3 February 2016.
21. Firouzeh, F.F.; Ghorshi, S.; Salsabili, S. Compressed Sensing Based Speech Enhancement. In Proceedings of the 2014 8th International Conference on Signal Processing and Communication Systems (ICSPCS), Gold Coast, Australia, 15–17 December 2014; pp. 1–6.
22. Wu, D.; Zhu, W.-P.; Swamy, M.N.S. Compressive Sensing-Based Speech Enhancement in Non-Sparse Noisy Environments. *IET Signal Process.* **2013**, *7*, 450–457. [CrossRef]
23. Gemmeke, J.F.; Van Hamme, H.; Cranen, B.; Boves, L. Compressive Sensing for Missing Data Imputation in Noise Robust Speech Recognition. *IEEE J. Sel. Top. Signal Process.* **2010**, *4*, 272–287. [CrossRef]
24. Wu, D.; Zhu, W.-P.; Swamy, M.N.S. The Theory of Compressive Sensing Matching Pursuit Considering Time-Domain Noise with Application to Speech Enhancement. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2014**, *22*, 682–696. [CrossRef]
25. Quackenbush, S.R.; Barnwell, T.P.; Clements, M.A. *Objective Measures of Speech Quality*; Prentice Hall: Upper Saddle River, NJ, USA, 1988; ISBN 978-0-13-629056-8.
26. ITU-T Recommendation Database. Available online: https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=14949&lang=en (accessed on 7 June 2023).
27. *Speech Coding and Synthesis*; Kleijn, W.B.; Paliwal, K.K. (Eds.) Elsevier: Amsterdam, The Netherlands; New York, NY, USA, 1995; ISBN 978-0-444-82169-0.
28. Hu, Y.; Loizou, P.C. Evaluation of Objective Quality Measures for Speech Enhancement. *IEEE Trans. Audio Speech Lang. Process.* **2008**, *16*, 229–238. [CrossRef]
29. Tribolet, J.; Noll, P.; McDermott, B.; Crochiere, R. A Study of Complexity and Quality of Speech Waveform Coders. In Proceedings of the ICASSP'78. IEEE International Conference on Acoustics, Speech, and Signal Processing, Tulsa, OK, USA, 10–12 April 1978; Volume 3, pp. 586–590.

30. Hansen, J.H.L.; Pellom, B.L. An Effective Quality Evaluation Protocol for Speech Enhancement Algorithms. In Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP 1998), Sydney, Australia, 30 November 1998; ISCA; p. 0917-0.
31. Klatt, D. Prediction of Perceived Phonetic Distance from Critical-Band Spectra: A First Step. In Proceedings of the ICASSP'82. IEEE International Conference on Acoustics, Speech, and Signal Processing, Paris, France, 3–5 May 1982; Volume 7, pp. 1278–1281.
32. P.862: Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs. Available online: https://www.itu.int/rec/T-REC-P.862 (accessed on 7 June 2023).
33. P.862.3: Application Guide for Objective Quality Measurement Based on Recommendations P.862, P.862.1 and P.862.2. Available online: https://www.itu.int/rec/T-REC-P.862.3/_page.print (accessed on 7 June 2023).
34. Crespo Marques, E.; Maciel, N.; Naviner, L.; Cai, H.; Yang, J. A Review of Sparse Recovery Algorithms. *IEEE Access* **2019**, *7*, 1300–1322. [CrossRef]
35. Mallat, S.G.; Zhang, Z. Matching Pursuits with Time-Frequency Dictionaries. *IEEE Trans. Signal Process.* **1993**, *41*, 3397–3415. [CrossRef]
36. de Paiva, N.M.; Marques, E.C.; de Barros Naviner, L.A. Sparsity Analysis Using a Mixed Approach with Greedy and LS Algorithms on Channel Estimation. In Proceedings of the 2017 3rd International Conference on Frontiers of Signal Processing (ICFSP), Paris, France, 6–8 September 2017; pp. 91–95.
37. Dai, W.; Milenkovic, O. Subspace Pursuit for Compressive Sensing Signal Reconstruction. *IEEE Trans. Inf. Theory* **2009**, *55*, 2230–2249. [CrossRef]
38. Donoho, D.L.; Tsaig, Y.; Drori, I.; Starck, J.-L. Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit. *IEEE Trans. Inf. Theory* **2012**, *58*, 1094–1121. [CrossRef]
39. Needell, D.; Vershynin, R. Uniform Uncertainty Principle and Signal Recovery via Regularized Orthogonal Matching Pursuit. *Found. Comput. Math.* **2009**, *9*, 317–334. [CrossRef]
40. Wang, J.; Kwon, S.; Shim, B. Generalized Orthogonal Matching Pursuit. *IEEE Trans. Signal Process.* **2012**, *60*, 6202–6216. [CrossRef]
41. Sun, H.; Ni, L. Compressed Sensing Data Reconstruction Using Adaptive Generalized Orthogonal Matching Pursuit Algorithm. In Proceedings of the 2013 3rd International Conference on Computer Science and Network Technology, Dalian, China, 12–13 October 2013; pp. 1102–1106.
42. Bi, X.; Leng, L.; Kim, C.; Liu, X.; Du, Y.; Liu, F. Constrained Backtracking Matching Pursuit Algorithm for Image Reconstruction in Compressed Sensing. *Appl. Sci.* **2021**, *11*, 1435. [CrossRef]
43. GBRAMP: A Generalized Backtracking Regularized Adaptive Matching Pursuit Algorithm for Signal Reconstruction—ScienceDirect. Available online: https://www.sciencedirect.com/science/article/abs/pii/S0045790621001907 (accessed on 7 June 2023).
44. Zhang, C. An Orthogonal Matching Pursuit Algorithm Based on Singular Value Decomposition. *Circuits Syst. Signal Process.* **2020**, *39*, 492–501. [CrossRef]
45. Das, S.; Mandal, J.K. An Enhanced Block-Based Compressed Sensing Technique Using Orthogonal Matching Pursuit. *Signal Image Video Process.* **2021**, *15*, 563–570. [CrossRef]
46. Blumensath, T.; Davies, M.E. Gradient Pursuits. *IEEE Trans. Signal Process.* **2008**, *56*, 2370–2382. [CrossRef]
47. Kwon, S.; Wang, J.; Shim, B. Multipath Matching Pursuit. *IEEE Trans. Inf. Theory* **2014**, *60*, 2986–3001. [CrossRef]
48. Elad, M. Optimized Projections for Compressed Sensing. *IEEE Trans. Signal Process.* **2007**, *55*, 5695–5702. [CrossRef]
49. Singh, R.; Bhattacharjee, U.; Singh, A.K. Performance Evaluation of Normalization Techniques in Adverse Conditions. *Procedia Comput. Sci.* **2020**, *171*, 1581–1590. [CrossRef]
50. Blinn, J.F. What's That Deal with the DCT? *IEEE Comput. Graph. Appl.* **1993**, *13*, 78–83. [CrossRef]
51. Stanković, L.; Brajović, M. Analysis of the Reconstruction of Sparse Signals in the DCT Domain Applied to Audio Signals. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 1220–1235. [CrossRef]
52. Reznik, Y.A. Relationship between DCT-II, DCT-VI, and DST-VII Transforms. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 5642–5646.
53. Prabhu, K.M.M. *Window Functions and Their Applications in Signal Processing*; Taylor & Francis: Oxford, UK, 2014; ISBN 978-1-4665-1584-0.
54. Shukla, V.; Swami, P.D. Speech Enhancement Using VAD for Noise Estimation in Compressive Sensing. In *Proceedings of the Data, Engineering and Applications*; Sharma, S., Peng, S.-L., Agrawal, J., Shukla, R.K., Le, D.-N., Eds.; Springer Nature: Singapore, 2022; pp. 357–369.
55. Lokesh, S.; Devi, M.R. Speech Recognition System Using Enhanced Mel Frequency Cepstral Coefficient with Windowing and Framing Method. *Clust. Comput* **2019**, *22*, 11669–11679. [CrossRef]
56. Van Segbroeck, M.; Tsiartas, A.; Narayanan, S.S. A Robust Frontend for VAD: Exploiting Contextual, Discriminative and Spectral Cues of Human Voice. In Proceedings of the INTERSPEECH, Lyon, France, 25–29 August 2013; pp. 704–708.
57. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
58. Candes, E.J.; Wakin, M.B. An Introduction To Compressive Sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 21–30. [CrossRef]

59. Jerri, A.J. The Shannon Sampling Theorem—Its Various Extensions and Applications: A Tutorial Review. *Proc. IEEE* **1977**, *65*, 1565–1596. [CrossRef]
60. Yuan, X.; Haimi-Cohen, R. Image Compression Based on Compressive Sensing: End-to-End Comparison with JPEG 2020. *IEEE Trans. Multimed.* **2020**, *22*, 2889–2904. [CrossRef]
61. Fira, M.; Costin, H.-N.; Goraș, L. A Study on Dictionary Selection in Compressive Sensing for ECG Signals Compression and Classification. *Biosensors* **2022**, *12*, 146. [CrossRef]
62. Golub, G.; Kahan, W. Calculating the Singular Values and Pseudo-Inverse of a Matrix. *J. Soc. Ind. Appl. Math. Ser. B Numer. Anal.* **1965**, *2*, 205–224. [CrossRef]
63. Haneche, H.; Boudraa, B.; Ouahabi, A. A New Way to Enhance Speech Signal Based on Compressed Sensing. *Measurement* **2020**, *151*, 107117. [CrossRef]
64. Fu, S.-W.; Liao, C.-F.; Tsao, Y. Learning With Learned Loss Function: Speech Enhancement With Quality-Net to Improve Perceptual Evaluation of Speech Quality. *IEEE Signal Process. Lett.* **2020**, *27*, 26–30. [CrossRef]
65. Martin-Doñas, J.M.; Gomez, A.M.; Gonzalez, J.A.; Peinado, A.M. A Deep Learning Loss Function Based on the Perceptual Evaluation of the Speech Quality. *IEEE Signal Process. Lett.* **2018**, *25*, 1680–1684. [CrossRef]
66. Varga, A.; Steeneken, H.J.M. Assessment for Automatic Speech Recognition: II. NOISEX-92: A Database and an Experiment to Study the Effect of Additive Noise on Speech Recognition Systems. *Speech Commun.* **1993**, *12*, 247–251. [CrossRef]
67. Hu, Y.; Loizou, P.C. Subjective Comparison and Evaluation of Speech Enhancement Algorithms. *Speech Commun.* **2007**, *49*, 588–601. [CrossRef]