*Journal of Applied Engineering and Technological Science*
Vol 6(1) 2024: 579-592

# *DCT AND SVD SPARSITY-BASED COMPRESSIVE LEARNING ON LETTUCES CLASSIFICATION*

**Lutvi Murdiansyah[1], Gelar Budiman[2], Indrarini Dyah Irawati[3*], Sugondo Hadiyoso[4], A. V. Senthil Kumar[5]**
School of Electrical Engineering, Telkom University, Bandung, Indonesia[1,2]
School of Applied Science, Telkom University, Bandung, Indonesia[3,4]
Hindustan College of Arts and Science, Coimbatore, India[5]
lutvi.murdiansyah@gmail.com[1], gelarbudiman@telkomuniversity.ac.id[2], indrarini@telkomuniversity.ac.id[3], sugondo@telkomuniversity.ac.id[4], hicasmcahod@hindusthan.net[5]

Received: 22 February 2024, Revised: 29 August 2024, Accepted: 06 September 2024
**Corresponding Author*

***ABSTRACT***
*Compressive Sensing (CS) technique in image compression represents efficient signal which offering solutions in image classification where the resources are constrained especially on a large image processing, storage resource, and computing performance. Compressive learning (CL) is a framework that integrates signal acquisition via compressed sensing (CS) and machine/deep learning for inference tasks directly on a small number of measurements, On the other hand, in real-world high-resolution (HR) data, where the image dataset is very limited CL, has the drawback of reduced accuracy under conditions of aggressive compression ratio. Here, a reconstruction method is necessary to maintain high levels of accuracy. To address this, we proposed a framework Deep Learning (DL) and Compressive Sensing that processing a small dataset of 92 images maintaining high accuracy. The framework developed in this paper employs processing sensing matrix A in compressive sensing with two transformation methods: DCT CL with Multi Neural Networks and the SVD method with GoogleNet framework. To maintain the same computation efficiency as DCT Compressive learning, SVD with GoogleNet framework provides a solution for object recognition, achieving accuracy values ranging from 89.47% to 63.15% for compression ratios of 3.97 to 31.75. This performance shows a linear tendency concerning the PSNR level, an index of signal reconstruction quality, and demonstrates an efficient process in the S matrix.*
***Keywords:*** *Compressive Sensing, Compressive Learning, Discrete Cosine Transform, Singular Value Decomposition, Accuracy.*

## 1. Introduction

The research background is pattern recognition as a methodology directed towards categorizing pre-processed images by assessing the likeness and resemblance of their features to identifying the class and location of objects in images also referred to as the problem of object detection (Sharif et al., 2022). However, issues arise when the available dataset does not match the scenario or requires manual labeling, making obtaining good training samples difficult. A large sample dataset size will require heavy computational processes during training and classification (inference). Compressive Sensing (CS) is employed for image compression of objects within the input data. CS has advantages over other compression algorithms because it detects sparse signals in the input and then performs the acquisition and compression processes in one step until the optimal size of the input signal is obtained (Z. Li et al., 2021). Image processing research in references (Dyah Irawati et al., 2023; Hamidon & Ahamed, 2023; Z. Li et al., 2021; Pratondo et al., 2023; Utami et al., 2021) enhances Deep Learning performance in vegetables image detection and classification for non-hydroponic farming on large sample datasets with simple background color, while we use datasets with complex background. In references (Chen & Pi, 2022; Mangia et al., 2020; Wen et al., 2022; Yu et al., 2023), authors proposed employed SVD image transformation and trained the dataset using GoogleNet but they did not use compression before learning process. We propose Deep Learning for vegetables image classification in real situation with complex background in the images. The vegetables are put into hydroponic pipes which standing on steel frames; thus, the backgrounds are not simple compared to previous paper datasets. We also propose the image compression before

deep learning process using CS. This process will reduce the size of the images before transmitting to the server for learning. We decompose the images using SVD, then we truncate the images before image transmission process from the detector to the server and receiver. This would result in reducing image sizes, reducing the transmission time, making the computational processes more efficient and reducing the storage needs for saving the images, but we still can maintain the accuracy.

## 2. Literature Review

Object detection is performed by utilizing methodologies like Convolutional Neural Networks (CNN) based on deep learning techniques (K B & J, 2020; Safitri et al., 2023). CNNs perform very well in image classification, especially when there is a large dataset containing hundreds of thousands of images. In compressive sensing and reconstruction, the best approach for the reconstruction of signals conditioned on $l_0$-norm and $l_1$-norm system will be met if $\delta_{2k} + \delta_{3k} < 1$ (X. Li et al., 2020; Z. Shen et al., 2023). The sampling matrix meeting this condition maintains the signal size, and then the measured signal becomes undistorted; thus, the reconstruction will be accurate. The process of reconstructing compressively sensed signals can be simplified by solving an under-determined system of equations conditioned on the $l_1$-norm through convex optimization. Over the past four decades, neural networks have been utilized to tackle various optimization problems, including convex optimization (Liu et al., 2023). Different neural network models have emerged for this purpose. One group of methods for compressive sensing signal reconstruction involves neural networks designed to replicate iterative CS algorithms using specialized architectures (Cui et al., 2019). These methods often employ a technique called algorithm unrolling (or unfolding), where each iteration is mapped to a network layer, and a fixed number of layers are stacked together, transitioning from sparse coding to the task of CS reconstruction (Mou & Zhang, 2022). In the proposed deep network, all parameters (including the transformation matrix) are adapted to repeated trained data, thereby optimizing the overall reconstruction process. The approach presented in (Sultana et al., 2019) also leverages the concept of unfolding the traditional iterative algorithm. ReconNet (Tran et al., 2021) marks the initial application of Convolutional Neural Networks (CNNs) in compressive sensing. Building on the achievements of CNN-based methods in image processing, Recurrent Neural Networks (RNNs) were introduced for CS signal reconstruction in (Bartan & Pilanci, 2021; Tran et al., 2022). However, training extremely deep networks poses challenges such as the vanishing gradient issue, overfitting, and accuracy plateauing. These challenges were addressed with the introduction of residual learning, as demonstrated by architectures like the residual network utilized in (Singh et al., 2020; Tran et al., 2022; Yao et al., 2019). A deep Residual Reconstruction Network, referred to as DR2-Net, incorporates a fully connected layer for obtaining an initial reconstruction and multiple residual learning blocks to deduce the difference between the ground truth image and the initial reconstruction.

It is not always necessary to reconstruct the entire original object data after the compressive sensing process; rather, an inference or conclusion can be drawn from the original data, which could be in the form of object image recognition. Terminologically, the process of CS without the approach of reconstructing the original signal is called Compressive Learning (CL). Direct high-level inference from compressed sensing (CS) data using deep learning (DL) circumvents the need for signal reconstruction. It allows us to operate directly in the compressed domain, utilizing the structure of Convolutional Neural Networks (CNNs) to extract discriminative non-linear features. In (S. Wang et al., 2023), a deep learning approach for image classification compression learning methodology was proposed, which utilizes CNNs and random Gaussian sensing matrices. In certain contexts, the reconstruction of signals is essential even when the main aim of compressed sensing (CS) systems is high-level inference (Machidon & Pejović, 2023). A combined reconstruction and inference framework was introduced by (Ahsan et al., 2021) which refines a deep learning (DL) pipeline. After the sensing matrix is learned, the pipeline splits into two distinct paths: one for image reconstruction and another for image labeling. This approach stands out for its efficiency compared to techniques that rely on

separate pipelines for the reconstruction and labeling tasks. Similarly, the research by (Song et al., 2020) merges these two steps into a single stage, facilitating the classification of compressed pathological signals directly at the sensor node.

Compressive Sensing combined with Deep Learning (CS-DL) brings the advantages of efficient sampling, and a powerful representation of learning still enhances reconstruction performance significantly (Hua et al., 2022) compared to Traditional Image Processing Techniques such as Fourier Transform-Based Methods, Wavelet Transform and Principal Component Analysis (PCA). Using DL, End-to-end training can eliminate the dependence on sparse representations in traditional CS methods, thereby simplifying the process. However, performance recovery at a low sensing rate remains an issue. The integration of compressive sensing (CS) with deep learning (DL) has led to significant advancements across various fields such as Magnetic Resonance Imaging (MRI), Computed Tomography (CT), Positron Emission Tomography (PET), Biomedical Signal Processing in Medical Imaging domain. Satellite Imaging, Hyperspectral Imaging, and Synthetic Aperture Radar (SAR) in the Remote Sensing domain. Wireless Sensor Networks, Smart Grid, Environmental Monitoring, Image Video Compression, Security, and Surveillance in the Internet of Things (IoT) domain.

The fields of compressed sensing (CS) and deep learning (DL) have experienced significant advancements, particularly at their intersection. In the ability to improve image recovery without the need for training data and very few measurements, Deep Image Prior (DIP), can recover high-quality images from incomplete or corrupted measurements, and has become popular in inverse problems in image restoration and medical imaging, including magnetic resonance imaging (MRI) (Liang et al., 2024). Deep Image Prior (DIP) recovers information from under-sampled imaging measurements by leveraging the inherent structure of convolutional neural networks (CNNs) focusing on the underlying training dynamics and kernel regime. While (Jo et al., 2021) analyze the DIP by the notion of effective degrees of freedom (DF) to monitor the optimization performance for denoising.

## 3. Research Methods

In this research, as shown in the flow diagram in Figure 3.1, a total of 92 images retrieved from the Hydroponic Laboratory of Applied Faculty divided into 4 classifications: 13 Butterhead, 29 Green Oakleaf & Green Curly Lettuce, 24 Romaine Lettuce & Other Types, and 29 Red Oakleaf & Lolorosa. The image processing follows the flow in diagram 3.1, which consists of two pathways of image recognition flow: Compressive Learning DCT, which will become of baseline, and Compressive Sensing SVD, which utilizes the optimal *L* singular value in order to reach good accuracy within smaller size data of image transmission. At first, the original image is read and resized to 256x256 pixels. Images block $X_{(m,n)}$ in the CL compression ratio, parameters are set to identical values to the SVD compression ratio. Compressive Learning Flow will consist process of Compressive Sensing 8x8 pixel blocks Random Normal DCT; Machine Learning CNN Multi Neural Network Model. While in Compressive Sensing SVD flow, the process will be CS SVD Random Feature Extraction, ISVD Image reconstruction, and Machine Learning CNN GoogleNet Model. In both pathways, the results are evaluated using confusion matrices and accuracy scores.

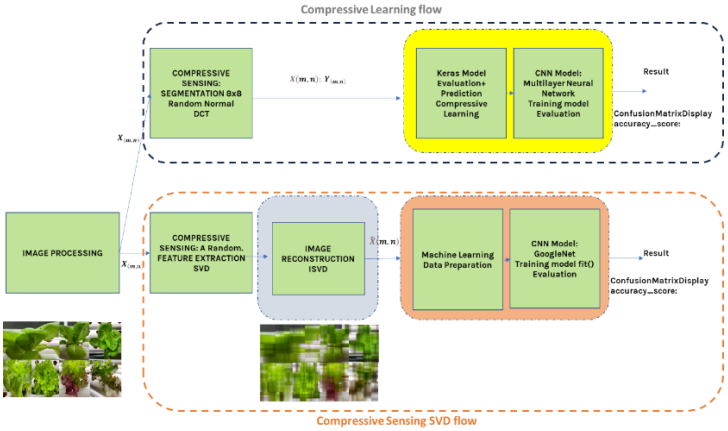


Fig. 3.1 Diagram of Research Simulation

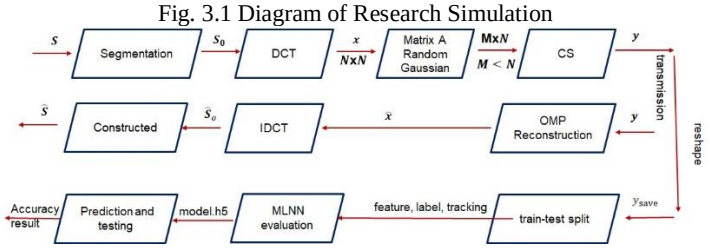


Fig. 3.2 Compressive Learning Simulation

Compressive learning simulation was designed to start by segmentation and resizing the original images $S$ RGB dataset 256 x 256 pixels and transforming it into a matrix 3 dimension $S_o$. Images are saved to a .mat file containing the image, class label, and resolution. For each image block, changing image blocks using DCT, each segment $S_o$ undergoes a DCT to convert the spatial domain data into frequency domain data $x$. Uses a Random Gaussian measurement matrix $A$ to obtain DCT coefficients $x$ into a lower-dimensional space, resulting in a compressed block $y$. This step involves compressive sensing (CS), where $M < N$ (Allwinnaldo et al., 2019). For each compressed block $y$, OMP is being used to reconstruct image blocks, producing $\hat{x}$ and calculates SNR for each block. Using IDCT to return image blocks to the spatial domain $\widehat{S_o}$.

Inspection and Saving of Results: Certain conditions (based on SNR values) are checked to determine whether the reconstruction was successful, and the reconstruction results are saved into an image file.

$$MSE = \frac{1}{M-N}\sum_{X=1}^{M}\sum_{y=1}^{N}[l(x,y) - l^1(x,y)]^2$$

$$PSNR = \ 10\ \log_{10}\frac{255^2}{MSE}$$

In the simulation of the Compressive Learning (CL) model employing a Multilayer Neural Network (MLNN), which is an artificial neural network comprising multiple neuron layers, the architectural design comprises an input layer, multiple hidden layers, and an output layer.

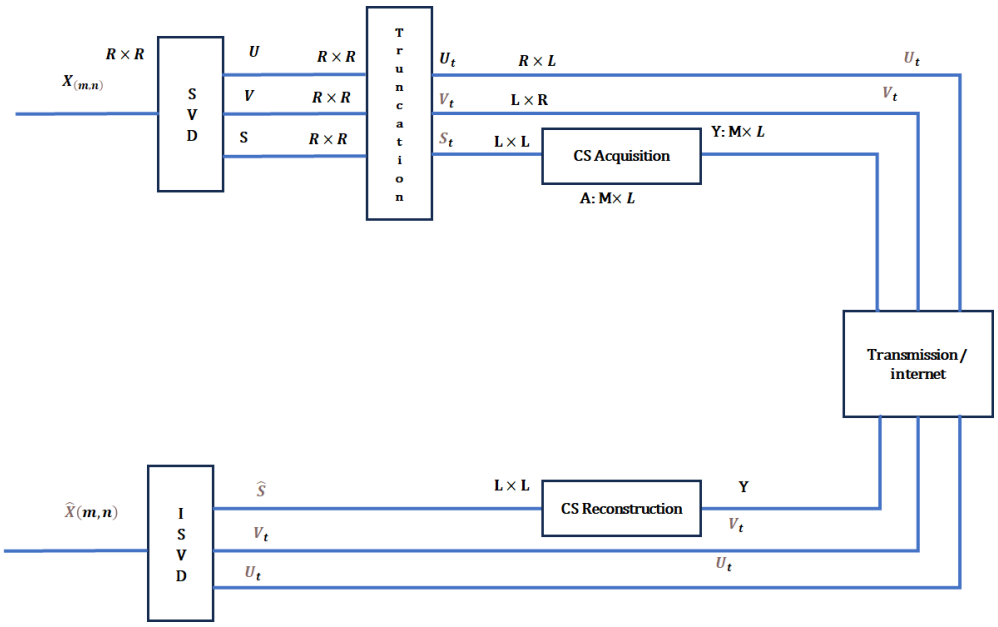


Fig 3.3 CS-SVD Simulation

In Fig 3.3 CS-SVD was designed by transforming the image matrix using the Sparse Singular Value and CS Technique. A specific process is involved where a truncated singular matrix, $S_r$ , converted into a more compact or compressed representation, $Y$ . This transformation is performed using an 'over-complete dictionary' or an excessive dictionary (Budiman et al., 2020).

The truncated singular matrix, denoted as $\hat{S}$, arises from a Singular Value Decomposition (SVD) process. In this method, only a subset of the most important singular values is preserved, aiming to efficiently encode the data within the framework of Compressive Sensing (CS). Inverse Singular Value Decomposition (ISVD) in the context of image reconstruction involves combining the matrices $\hat{S}$, $U_t$, and $V_t$ to form the signal $\hat{S}_{(m,n)}$, which approximates the original image $S_{(m,n)}$. This process allows large images to be compressed into smaller sizes without significant loss of information and then reconstructed with superior quality. Processing images in 8x8 blocks for image compression and feature extraction tasks. $Y_{(m,n)}$ initialise an array to store the processed data, related to the compressed representation or features of the image block. $Y_{(m,n)}$ is stored in a 3-dimensional array as processed data form, and each block is then compared with the 64 blocks of the original image array (x test, y test) evaluates the model on test data. It returns loss values and metrics, accuracy for the test data, and beneficial for gaining insight into the model's overall performance. Predictions are made using the trained model on test data. The result is a 3-dimensional array, where each prediction is a probability vector across all output classes. Next, compare the actual label with the predicted label. In the CS SVD Block, $L$ or the singular Matrix values from 1 to 8 are considered and define the number of leading singular values and vectors used in a Singular Value Decomposition (SVD) operation on image blocks. Each image is partitioned into blocks measuring 64 units in size. The original image is transformed into Matrices $S$, $U$ , and $V$ using the Singular Value Decomposition (SVD) technique. U and V are the matrices of left and right singular vectors., and $S_{LxL}$ is the matrix of singular values. The compression process is carried out in image blocks. For each block, SVD is performed to obtain matrices $S$, $U$, and $V$ . Matrices $U$ and $V$ are truncated according to the selected number of singular values $L$. The matrix $A$ of size $M \times L$, filled with random values, is used to acquire singular values from the matrix. This process results in matrix $Y$ which is the product of $A$ and S. The vector $S_r$ is calculated from the product of the inverse of $A$ with .

Matrix $\hat{S}$ , which is the diagonal of $S_r$ , is used along with $U_t$ (a truncated version of $U$ ) and $V_t$ (a truncated version of V) to produce a compressed version of the original image block. The compressed images are stored in bmp format. PSNR serves as a metric for evaluating image compression quality. The compression ratio is determined by comparing the number of elements in the original block to the number of elements in the compressed representation.

Analysis in mathematical will be a signal input $X \in \mathbb{R}^{MxN}$ $t$ ransformed into a two−dimensional matrix $X \in \mathbb{R}^{RxR}$ where $R$ = 256. The formulation of the two-dimensional matrix $X$ is as follows:

$$X = \begin{bmatrix} \mathrm{x}_1 & \mathrm{X}_{R+1} & \cdots & \mathrm{X}_{R(R-1)+1} \\ \mathrm{x}_2 & \mathrm{X}_{R+2} & \cdots & \mathrm{X}_{R(R-1)+2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{x}_R & \mathrm{X}_{2R} & \cdots & \mathrm{X}_{R^2} \end{bmatrix}$$

Process of SVD on $X$ obtains its orthogonal matrices $U \in \mathbb{R}^{RxR}$ , $S \in \mathbb{R}^{RxR}$ , and $V \in \mathbb{R}^{RxR}$ , where the relevant shown as:

$$X = USV^T$$

$S$ is a sparse diagonal matrix having $M$ non-zero elements in the diagonal of the matrix as $M$ singular values. In the compression senses, $U$ , S, and $V$ truncated to $U_t = U_{[1,\cdots R:1,\cdots L]} \in \mathbb{R}^{RxL}, S_t = S_{[1,\cdots L:1,\cdots L]} \in \mathbb{R}^{LxL}, V_t = V_{[1,\cdots R:1,\cdots L]} \in \mathbb{R}^{RxL}$ with L < R.
The CS acquisition $S_r$ applied as (Budiman et al., 2020):

$$Y = AS_t$$

$A \in \mathbb{R}^{MxL}$ the matrix $A$ of size $M \times L$, filled with random values, is used to acquire singular values from the matrix $Y \in \mathbb{R}^{MxL}$as the output of CS acquisition. The truncated matrix $S_t$ formulated as:

$$= \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_L \end{bmatrix}$$

The matrix Y is a diagonal matrix with dimensions L×L, where L is a positive integer. The diagonal elements of this matrix are represented by $\sigma_1, \sigma_2, \ldots, \sigma_L$, which are known as the singular values. Each $\sigma_i$ (for i = 1, 2, . . . , L) is a singular value element placed on the diagonal of the matrix. All off-diagonal elements of the matrix are zeros. This structure of Y indicates that it is a sparse matrix, predominantly filled with zeros except for its diagonal elements. We have three matrices to be transmitted, that is $U_t, V_t$, and $Y$ . From this result, we can calculate the Compression Ratio (CR) as the comparison between the original signal length and the transmitted signal length. The calculation of the reconstructed images with size as X in the matrix as :

$$X_t = U_t \hat{S}_t {V_t}^T = \begin{bmatrix} \hat{x}_1 & \hat{x}_{R+1} & \cdots & \hat{x}_{R(R-1)+1} \\ \hat{x}_2 & \hat{x}_{R+2} & \cdots & \hat{x}_{R(R-1)+2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x}_R & \hat{x}_{2R} & \cdots & \hat{x}_{R^2} \end{bmatrix}$$

The transformed matrix $X_t$ is calculated as the product of three matrices: $U_t$, the transformed singular value matrix $\hat{S}_t$, and the transpose of the transformed matrix $V_t$. The matrix $X_t$ is represented as a two-dimensional array with specific element positions. Each element in this array is denoted as $\hat{x}_{ij}$ , where i and j are the row and column indices, respectively. Where$X_t \in \mathbb{R}^{RxR}$, but its element values are different than the original signal of $X$. The $L$ value controls the signal quality and compression ratio. Finally, we can get $\hat{x}$ = [$\hat{x}_1$, $\hat{x}_2$, . . . , $\hat{x}_{R^2}$ ] as a reconstructed or decompressed version of the signal by converting a two-dimensional matrix $X_t$ back to the original dimensional signal $\hat{X}$; thus, we can calculate the signal quality by comparing $X$ and $\hat{X}$.

The training process for this model is conducted on Google Collaboratory, with the dataset divided into two parts: 75% for training and 25% for testing. The training model in the CS reconstruction simulation with SVD uses the GoogleNet architecture, A Convolutional Neural Network (CNN) structured according to the Inception architecture. This network employs Inception modules, allowing selection from various convolutional filter sizes (1x1, 3x3, 5x5) at each block. The input layer is defined based on the shape of the training images. Several convolutional, max pooling, and inception blocks are applied with the GoogleNet architecture, although specific parameters might differ slightly. Auxiliary classifiers (auxiliary functions) are a feature of GoogleNet to combat the vanishing gradient problem and to provide additional regularization. The model adds flattening, batch normalization, and a final dense layer for classification, compiled with the Adagrad optimizer and categorical cross-entropy loss, suitable for a multi-class classification task, trained using the fit method, with callbacks for model checkpointing and logging, which is good practice. The saved model is evaluated on the test set, and the loss and accuracy are printed out.

## 4. Results and Discussions

### Result of Images Reconstruction

The overall purpose of the Image Processing and compression routine involves reading images, resizing them, processing blocks of the image using SVD, and then reconstructing the image blocks. The variable L determines the level of data used in these reconstructions impacting the quality and size of the processed image. By selecting the first L singular values/vectors (U, S, V) (1 to 8), the simulation is essentially performing a form of data reduction or compression. The larger the L, the more singular values are used, leading to a higher fidelity reconstruction of the original image block. Conversely, a smaller L results in less data being used, potentially speeding up processing and reducing data size at the expense of image quality. The block or size of Image Blocks for Processing is set to 64, and the block parameter specifies the size of the sub-regions (blocks) of the image that are processed independently. Resolution (Image Resolution) refers to the dimensions 256x256 (width and height) to which the input image is resized before processing. Computation of PSNR and Compression Ratio: after processing each image block, the code computes the Peak Signal-to-Noise Ratio (PSNR) and a compression ratio (denoted as rasio), which are standard measures for evaluating image processing quality and efficiency. Reconstructed images $\hat{X}$ as a reconstructed or decompressed version of the signal by converting a two-dimensional matrix $X_t$.

### Result of Computational Cost

Comparing the computational cost, compressive Learning computation of Truncated SVD-based methods delivers a smaller burden of Computational complexity because it doesn't have the cost logarithmic associated with OMP iteration, and $L$ singular rank has not increased computation cost relatively to a lower rank.

Table 4.1 – CS SVD Computational Cost

| L | CS Ratio | Computation Cost |
|---|---|---|
| 1 | 31.7519 | 1633ms/step |
| 2 | 15.876 | 1578ms/step |
| 3 | 10.584 | 1539ms/step |
| 4 | 7.938 | 1562ms/step |
| 5 | 6.3504 | 1576ms/step |
| 6 | 5.292 | 1560ms/step |
| 7 | 4.536 | 1605ms/step |
| 8 | 3.969 | 1577ms/step |
| Original | 1 | 2000ms/step |

**Accuracy Performance of model SVD and GoogleNet**

Simulation was conducted to process images using a block-based approach for compression denoising, and then calculate the Peak Signal-to-Noise Ratio (PSNR) and compression ratio (ratio) for each processed image. The GoogleNet model uses a deep learning library such as TensorFlow or PyTorch with MATLAB's Deep Learning Toolbox and then passes the processed images through the model to get the accuracy vector values.

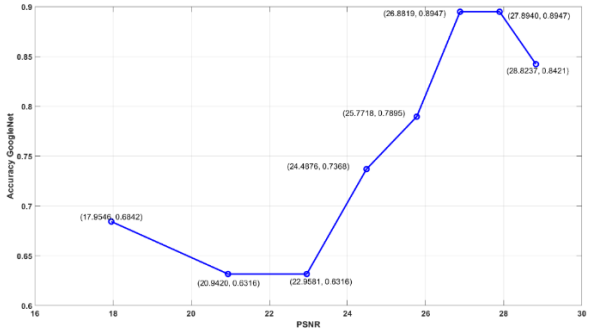


Fig 4.1 GoogleNet accuracy to PSNR

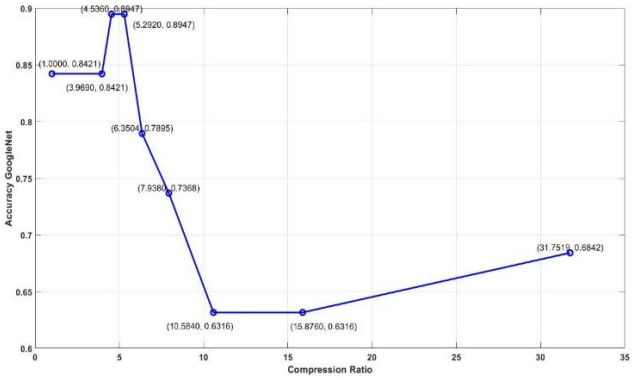


Fig 4.2 GoogleNet Accuracy to Compression Rate

From the chart of results in figures 4.1 and 4.2, we infer that different levels of processing or compression have been applied to some data, and the impact of these levels on the quality (as measured by PSNR) and the performance of a GoogleNet model (as measured by accuracy) has been recorded. As the PSNR decreases, which indicates a reduction in quality, the accuracy of the GoogleNet model also varies. The accuracy does not decrease linearly with PSNR; levels 6 and 7 have higher accuracies than the original data, which could suggest that a certain amount of processing a defect on a pixel could be beneficial.

**Accuracy Performance of Compressive Learning Model CNN**

The variable $\boldsymbol{Ysave}$ represents a three-dimensional matrix of compressed images to be a placeholder for the compressed form of the original images, processed using compressive sensing techniques. The $\boldsymbol{Ysave}$ matrix shape is determined by the number of images, the number of measurements M, and the total number of blocks across all images. Essentially, $\boldsymbol{Ysave}$ is a collection of compressed representations of the image blocks. Compression is achieved by reducing the number of coefficients that represent each block. This reduction is accomplished by multiplying the full set of Discrete Cosine Transform (DCT) coefficients by a smaller measurement matrix A, thereby capturing the essential information in fewer coefficients. The process operates under the assumption that the image blocks are sparse in the DCT domain, a common premise in image processing.

The measurements stored in $\boldsymbol{Ysave}$ are utilized in an attempt to reconstruct the original image blocks using the Orthogonal Matching Pursuit (OMP) algorithm. In summary, $\boldsymbol{Ysave}$ is crucial for both the compression step—storing the reduced set of measurements that represent the image blocks—and the reconstruction step—providing the data from which the original

blocks are reconstructed. This entire process is an application of compressive sensing, a technique that exploits sparsity to capture and reconstruct signals with fewer samples than traditionally required by the Nyquist-Shannon sampling theorem. Moreover, ***Ysave*** serves as the dataset for both training and testing the model. The accuracy of the model is assessed based on its performance on data in predicting.

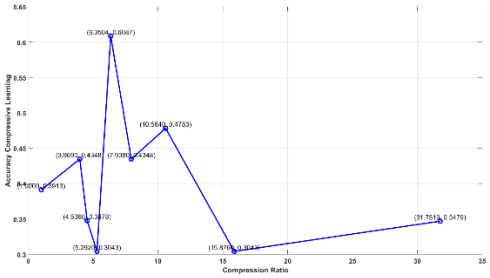


Fig 4.3 Accuracy CL to Compression Ratio

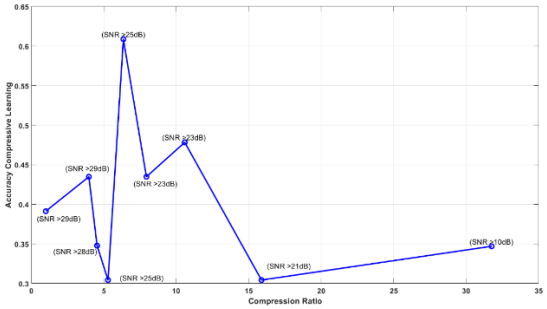


Fig 4.4 Accuracy CL on the Threshold of Minimum SNR

**Analysis of the Data**

From the PSNR measurements of the reconstructed images against noise, for each value of L in the Singular matrix and trained with GoogleNet framework models, and then compared with the results of Compressive Learning, the following accuracy table was obtained:

Table 4.2 - Performance metrics of the chosen model

| L | PSNR | Ratio | GoogleNet accuracy | CS Direct-Inference Accuracy |
|---|---|---|---|---|
| 1 | 17.9546 | 31.7519 | 0.684210539 | 0.3478 |
| 2 | 20.942 | 15.876 | 0.631578922 | 0.3043 |
| 3 | 22.9581 | 10.584 | 0.631578922 | 0.4783 |
| 4 | 24.4876 | 7.938 | 0.736842096 | 0.4348 |
| 5 | 25.7718 | 6.3504 | 0.789473712 | 0.6087 |
| 6 | 26.8819 | 5.292 | 0.894736826 | 0.3043 |
| 7 | 27.894 | 4.536 | 0.894736826 | 0.3478 |
| 8 | 28.8237 | 3.969 | 0.842105269 | 0.4348 |
| Origin | infinite | 1 | 0.842105269 | 0.3913 |

The degree of accuracy of the GoogleNet model framework achieves results that are nearly linear concerning the compression ratio of the original image matrix size dimensions and the truncated CS image.

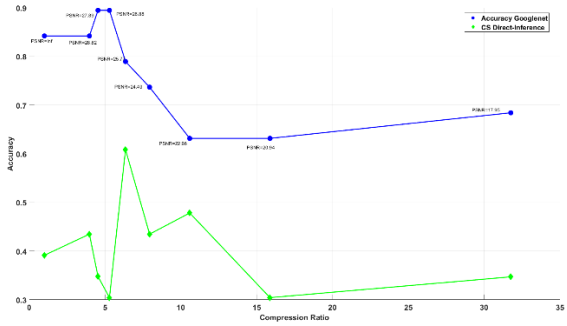


Fig 4.5 - Accuracy of SVD GoogleNet and DCT CS Direct Inference

## Discussion

We propose that the complex datasets referenced in the table refer to images containing multiple objects, including a non-simplified background. In our dataset, each image features a vegetable along with a pipe and a steel frame. Understandably, deep learning algorithms may find it more challenging to classify these images compared to those in a simplified dataset.

Table 4.3 - Comparison of the chosen model and current research

| References | Complex Datasets | CS | Deep Learning | PSNR | Accuracy (%) | Compression Ratio (´) |
|---|---|---|---|---|---|---|
| Proposed | √ | √ | √ | 27.894 | 89.2 | 0.220458554 |
| Block-based compressed sensing (Haq et al., 2021) | × | √ | √ | 37.75 | Not reported | 0.3 |
| Complex-Valued Gaussian Measurement Matrix (Yue Wang et al., 2023) | × | √ | × | 33.95 | Not reported | 0.2 |
| Multi-scale Dilated CNN (Zeng et al., 2021) | × | √ | √ | 31.75 | Not reported | 0.1 |
| Invertible Privacy-Preserving Adversarial Reconstruction (IPPARNet) (Xiao et al., 2023) | × | √ | √ | 29.19 | Not reported | 0.2 |
| M33C1313-reg Adaptive Modular CNN (Wu & Pan, 2022) | × | × | √ | Not reported | 99.26 | 1 |
| Lightweight Attended Multi-Scale Residual Network (LAMRN) (Yan et al., 2021) | × | × | √ | 33.78 | Not reported | 1 |
| Lightweight Image Super-Resolution with Adaptive Weighted Learning Network (AWSRN) (C. Wang et al., 2019) | × | × | √ | 33.78 | Not reported | 1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Hybrid NonLocal Sparsity Regularization (HNLSR) (L. Li et al., 2020) | × | √ | √ | 33.69 | Not reported | 0.2 |
| Sparsity Averaging with Reweighted Analysis (SARA) (Satrya et al., 2023) | × | √ | √ | 30.86 | Not reported | 0.2 |
| Compressed ultrafast photography (CUP) PnP-FFDNet (Q. Shen et al., 2022) | √ | √ | √ | 28.37 | Not reported | 0.5 |
| YOLO-EfficientNet (Yidong Wang et al., 2024) | √ | × | √ | Not reported | 95.78 | 1 |
| SVM-RGB &HSV identification (Deng et al., 2022) | √ | × | × | Not reported | 93 | Not reported |
| Lettuce stage classification K-Nearest Neighbor (KNN) (Lauguico et al., 2020) | √ | × | × | Not reported | 91.67 | Not reported |
| Support Vector Machines (SVM) using histograms of oriented gradients (HOG) (Osorio et al., 2020) | √ | × | √ | Not reported | 94 | Not reported |

## 5. Conclusion

The accuracy of compressive learning (CL) Direct inference using DCT (Discrete Cosine Transform) and Random Gaussian measurements against varying compression ratios as the compression ratio increases, the ability to accurately reconstruct the original signal may not decrease uniformly. This is typical in compressive sensing, where the signal recovery performance strongly depends on the sparsity of the signal in the basis, the measurement matrix A properties. The model accuracy of three different models - GoogleNet and an Accuracy CL (a comparative model), where GoogleNet exhibits a range of accuracy from approximately 0.642 to 0.842. There is a trend where the accuracy generally increases as the PSNR increases, which suggests that better image quality correlates with higher accuracy in GoogleNet's performance. Accuracy CL shows a range from approximately 0.3043 to 0.6522. The optimal balance between compression and accuracy appears to be model-dependent. For GoogleNet, higher PSNR correlates with higher accuracy, indicating a preference for lower compression ratios. In contrast, Accuracy CL do not exhibit a consistent correlation, suggesting that they may be more robust to compression but with overall lower accuracy than GoogleNet.

The optimized compression ratio to balance is between the image quality (as indicated by PSNR) and the model accuracy. The sweet spot appears to be around a compression ratio of 5 to 10, where GoogleNet still maintains relatively high accuracy and high compression ratios. (above 10), as they significantly degrade the performance of all models. And PSNR value above 25 dB is considered good for various image processing applications. Train models directly on compressed image data to potentially improve their robustness and accuracy at higher

compression ratios. The CL model considers further training or fine-tuning with a focus on compressed datasets, as its accuracy lags the other models.

**Acknowledgement**
This work is financially supported by the Indonesia Endowment Fund for Education (LPDP), as well as the Deputy for Research Facilitation and Innovation, National Research and Innovation Agency (BRIN), under grant number 37/II.7/HK/2023

**References**

Ahsan, M., Eshkabilov, S., Cemek, B., Küçüktopcu, E., Lee, C. W., & Simsek, H. (2021). Deep Learning Models to Determine Nutrient Concentration in Hydroponically Grown Lettuce Cultivars (Lactuca sativa L.). *Sustainability*, *14*(1), 416. https://doi.org/10.3390/su14010416

Allwinnaldo, Budiman, G., Novamizanti, L., Alief, R. N., & Ansori, M. R. R. (2019). QIM-based Audio Watermarking using Polar-based Singular Value in DCT Domain. *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 216–221. https://doi.org/10.1109/ICITISEE48480.2019.9003921

Bartan, B., & Pilanci, M. (2021). *Training Quantized Neural Networks to Global Optimality via Semidefinite Programming*. http://arxiv.org/abs/2105.01420

Budiman, G., Suksmono, A. B., & Danudirdjo, D. (2020). Compressive Sampling with Multiple Bit Spread Spectrum-Based Data Hiding. *Applied Sciences*, *10*(12), 4338. https://doi.org/10.3390/app10124338

Chen, J. I.-Z., & Pi, C.-S. (2022). Assessment for Different Neural Networks with FeatureSelection in Classification Issue. *Sensors*, *22*(8), 3099. https://doi.org/10.3390/s22083099

Cui, Y., Wang, J., Qi, J., Zhang, Z., & Zhu, J. (2019). Underdetermined DOA Estimation of Wideband LFM Signals Based on Gridless Sparse Reconstruction in the FRF Domain. *Sensors*, *19*(10), 2383. https://doi.org/10.3390/s19102383

Deng, W., Zhou, F., Gong, Z., Cui, Y., Liu, L., & Chi, Q. (2022). Disease Feature Recognition of Hydroponic Lettuce Images Based on Support Vector Machine. *Traitement Du Signal*, *39*(2), 617–625. https://doi.org/10.18280/ts.390224

Dyah Irawati, I., Budiman, G., Saidah, S., Rahmadiani, S., & Latip, R. (2023). Block-based compressive sensing in deep learning using AlexNet for vegetable classification. *PeerJ Computer Science*, *9*, e1551. https://doi.org/10.7717/peerj-cs.1551

Hamidon, M. H., & Ahamed, T. (2023). Detection of Defective Lettuce Seedlings Grown in an Indoor Environment under Different Lighting Conditions Using Deep Learning Algorithms. *Sensors*, *23*(13), 5790. https://doi.org/10.3390/s23135790

Haq, E. U., Jianjun, H., Huarong, X., & Li, K. (2021). Block-based compressed sensing of MR images using multi-rate deep learning approach. *Complex & Intelligent Systems*, *7*(5), 2437–2451. https://doi.org/10.1007/s40747-021-00426-6

Hua, J., Rao, J., Peng, Y., Liu, J., & Tang, J. (2022). Deep Compressive Sensing on ECG Signals with Modified Inception Block and LSTM. *Entropy*, *24*(8), 1024. https://doi.org/10.3390/e24081024

Jo, Y., Chun, S. Y., & Choi, J. (2021). *Rethinking Deep Image Prior for Denoising*. http://arxiv.org/abs/2108.12841

K B, P., & J, M. (2020). Design and Evaluation of a Real-Time Face Recognition System using Convolutional Neural Networks. *Procedia Computer Science*, *171*, 1651–1659. https://doi.org/10.1016/j.procs.2020.04.177

Lauguico, S. C., Concepcion II, R. S., Alejandrino, J. D., Tobias, R. R., Macasaet, D. D., & Dadios, E. P. (2020). A Comparative Analysis of Machine Learning Algorithms Modeled from Machine Vision-Based Lettuce Growth Stage Classification in Smart Aquaponics. *International Journal of Environmental Science and Development*, *11*(9), 442–449. https://doi.org/10.18178/ijesd.2020.11.9.1288

Li, L., Xiao, S., & Zhao, Y. (2020). Image Compressive Sensing via Hybrid Nonlocal Sparsity

Regularization. *Sensors*, *20*(19), 5666. https://doi.org/10.3390/s20195666

Li, X., Feng, G., & Zhu, J. (2020). An Algorithm of l 1 -Norm and l 0 -Norm Regularization Algorithm for CT Image Reconstruction from Limited Projection. *International Journal of Biomedical Imaging*, *2020*, 1–6. https://doi.org/10.1155/2020/8873865

Li, Z., Li, Y., Yang, Y., Guo, R., Yang, J., Yue, J., & Wang, Y. (2021). A high-precision detection method of hydroponic lettuce seedlings status based on improved Faster RCNN. *Computers and Electronics in Agriculture*, *182*, 106054. https://doi.org/10.1016/j.compag.2021.106054

Liang, S., Bell, E., Qu, Q., Wang, R., & Ravishankar, S. (2024). *Analysis of Deep Image Prior and Exploiting Self-Guidance for Image Reconstruction*. http://arxiv.org/abs/2402.04097

Liu, J., Wang, Q.-G., & Yu, J. (2023). Convex Optimization-Based Adaptive Fuzzy Control for Uncertain Nonlinear Systems With Input Saturation Using Command Filtered Backstepping. *IEEE Transactions on Fuzzy Systems*, *31*(6), 2086–2091. https://doi.org/10.1109/TFUZZ.2022.3216103

Machidon, A. L., & Pejović, V. (2023). Deep learning for compressive sensing: a ubiquitous systems perspective. *Artificial Intelligence Review*, *56*(4), 3619–3658. https://doi.org/10.1007/s10462-022-10259-5

Mangia, M., Prono, L., Marchioni, A., Pareschi, F., Rovatti, R., & Setti, G. (2020). Deep Neural Oracles for Short-window Optimized Compressed Sensing of Biosignals. *IEEE Transactions on Biomedical Circuits and Systems*, 1–1. https://doi.org/10.1109/TBCAS.2020.2982824

Mou, C., & Zhang, J. (2022). TransCL: Transformer Makes Strong and Flexible Compressive Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–16. https://doi.org/10.1109/TPAMI.2022.3194001

Osorio, K., Puerto, A., Pedraza, C., Jamaica, D., & Rodríguez, L. (2020). A Deep Learning Approach for Weed Detection in Lettuce Crops Using Multispectral Images. *AgriEngineering*, *2*(3), 471–488. https://doi.org/10.3390/agriengineering2030032

Pratondo, A., Ismail, Sujana, A. P., Susanti, F., Roedavan, R., & Budianto, A. (2023). Classification of Sweet Potato Leaf Variants using Transfer Learning. *2023 9th International Conference on Wireless and Telematics (ICWT)*, 1–6. https://doi.org/10.1109/ICWT58823.2023.10335271

Safitri, I., Suksmono, A. B., Danudirdjo, D., & Usman, K. (2023). Super-Resolution in Sparse Representation Fusion for Medical Images. *2023 International Conference on Electrical Engineering and Informatics (ICEEI)*, 1–6. https://doi.org/10.1109/ICEEI59426.2023.10346824

Satrya, G. B., Ramatryana, I. N. A., & Shin, S. Y. (2023). Compressive Sensing of Medical Images Based on HSV Color Space. *Sensors*, *23*(5), 2616. https://doi.org/10.3390/s23052616

Sharif, M. S., Afolabi, M. O., Zorto, A., & Elmedany, W. (2022). Enhancement Techniques for Improving Facial Recognition Performance in Convolutional Neural Networks. *2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, 494–499. https://doi.org/10.1109/3ICT56508.2022.9990811

Shen, Q., Tian, J., & Pei, C. (2022). A Novel Reconstruction Algorithm with High Performance for Compressed Ultrafast Imaging. *Sensors*, *22*(19), 7372. https://doi.org/10.3390/s22197372

Shen, Z., Chen, Q., & Yang, F. (2023). A convex relaxation framework consisting of a primal–dual alternative algorithm for solving ℓ0 sparsity-induced optimization problems with application to signal recovery based image restoration. *Journal of Computational and Applied Mathematics*, *421*, 114878. https://doi.org/10.1016/j.cam.2022.114878

Singh, A., Rajan, P., & Bhavsar, A. (2020). SVD-based redundancy removal in 1-D CNNs for acoustic scene classification. *Pattern Recognition Letters*, *131*, 383–389. https://doi.org/10.1016/j.patrec.2020.02.004

Song, Y., Cao, Z., Wu, K., Yan, Z., & Zhang, C. (2020). *Learning Fast Approximations of Sparse Nonlinear Regression*. http://arxiv.org/abs/2010.13490

Sultana, F., Sufian, A., & Dutta, P. (2019). *A Review of Object Detection Models based on*

*Convolutional Neural Network*. https://doi.org/10.1007/978-981-15-4288-6_1

Tran, D. T., Gabbouj, M., & Iosifidis, A. (2022). Progressive and compressive learning. In *Deep Learning for Robot Perception and Cognition* (pp. 187–220). Elsevier. https://doi.org/10.1016/B978-0-32-385787-1.00014-2

Tran, D. T., Yamac, M., Degerli, A., Gabbouj, M., & Iosifidis, A. (2021). Multilinear Compressive Learning. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(4), 1512–1524. https://doi.org/10.1109/TNNLS.2020.2984831

Utami, N. S., Novamizanti, L., Saidah, S., & Apraz Ramatryana, I. N. (2021). SVD on a Robust Medical Image Watermarking based on SURF and DCT. *2021 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, 32–38. https://doi.org/10.1109/IAICT52856.2021.9532515

Wang, C., Li, Z., & Shi, J. (2019). *Lightweight Image Super-Resolution with Adaptive Weighted Learning Network*. http://arxiv.org/abs/1904.02358

Wang, S., Jiang, X., Liu, X., Dong, Z., Pei, R., & Wang, H. (2023). End-to-end reconstruction of multi-scale holograms based on CUE-NET. *Optics Communications*, *530*, 129079. https://doi.org/10.1016/j.optcom.2022.129079

Wang, Yidong, Wu, M., & Shen, Y. (2024). Identifying the Growth Status of Hydroponic Lettuce Based on YOLO-EfficientNet. *Plants*, *13*(3), 372. https://doi.org/10.3390/plants13030372

Wang, Yue, Xue, L., Yan, Y., & Wang, Z. (2023). A Novel Complex-Valued Gaussian Measurement Matrix for Image Compressed Sensing. *Entropy*, *25*(9), 1248. https://doi.org/10.3390/e25091248

Wen, H., He, X., & Huang, T. (2022). Sparse signal reconstruction via recurrent neural networks with hyperbolic tangent function. *Neural Networks*, *153*, 1–12. https://doi.org/10.1016/j.neunet.2022.05.022

Wu, W., & Pan, Y. (2022). Adaptive Modular Convolutional Neural Network for Image Recognition. *Sensors*, *22*(15), 5488. https://doi.org/10.3390/s22155488

Xiao, D., Li, Y., & Li, M. (2023). Invertible Privacy-Preserving Adversarial Reconstruction for Image Compressed Sensing. *Sensors*, *23*(7), 3575. https://doi.org/10.3390/s23073575

Yan, Y., Xu, X., Chen, W., & Peng, X. (2021). Lightweight Attended Multi-Scale Residual Network for Single Image Super-Resolution. *IEEE Access*, *9*, 52202–52212. https://doi.org/10.1109/ACCESS.2021.3069775

Yao, H., Dai, F., Zhang, S., Zhang, Y., Tian, Q., & Xu, C. (2019). DR2-Net: Deep Residual Reconstruction Network for image compressive sensing. *Neurocomputing*, *359*, 483–493. https://doi.org/10.1016/j.neucom.2019.05.006

Yu, J., Liu, K., Liu, W., Yao, W., & Xie, J. (2023). Recurrence plot image and GoogLeNet based historical abuse backtrace for li-ion batteries. *Journal of Energy Storage*, *74*, 109378. https://doi.org/10.1016/j.est.2023.109378

Zeng, C., Wang, Z., Wang, Z., Yan, K., & Yu, Y. (2021). Image Compressed Sensing and Reconstruction of Multi-Scale Residual Network Combined with Channel Attention Mechanism. *Journal of Physics: Conference Series*, *2010*(1), 012134. https://doi.org/10.1088/1742-6596/2010/1/012134