# DEEP LEARNING OF COMPRESSED SENSING OPERATORS

WITH STRUCTURAL SIMILARITY (SSIM) LOSS

# ABSTRACT

Compressed sensing (CS) is a signal processing framework for efficiently reconstructing a signal from a small number of measurements, obtained by linear projections of the signal. In this paper we present an end-to-end deep learning approach for CS, in which a fully-connected network performs both the linear sensing and non-linear reconstruction stages. During the training phase, the sensing matrix and the non-linear reconstruction operator are **jointly** optimized using *Structural similarity index (SSIM)* as loss rather than the standard Mean Squared Error (MSE) loss. We compare the proposed approach with state-of-the-art in terms of reconstruction quality under both losses, i.e. SSIM score and MSE score.

# INTRODUCTION

- Compressed sensing (CS) is a mathematical framework that defines the conditions and tools for the recovery of a signal from a small number of its linear projections (i.e. measurements).
- In the CS framework, the measurement device acquires the signal in the linear projections domain, and the full signal is reconstructed by convex optimization techniques.

# INTRODUCTION

- In this project we address the CS problem by using a novel loss function, the SSIM loss. Our approach is based on a deep neural network, which simultaneously learns the linear sensing matrix and the non-linear reconstruction operator under the SSIM loss.
- During training, the proposed network **jointly** optimizes both the linear sensing matrix and the non-linear reconstruction operator.

# COMPRESSES SENSING

- Given a signal $x \in \mathbb{R}^N$, an $M \times N$ sensing matrix $\Phi$ (such that $M \ll N$) and a measurements vector $y = \Phi x$, the goal of CS is to recover the signal from its measurements.
- The sensing rate is defined by $R = \frac{M}{N}$, and since $R \ll 1$ the recovery of $x$ is not possible in the general case.

# STRUCTURAL SIMILARITY INDEX (SSIM)

The Structural SIMilarity (SSIM) index is a method for measuring the similarity between two images. The SSIM index can be viewed as a quality measure of one of the images being compared, provided the other image is regarded as of perfect quality. The difference with respect to other techniques such as Mean Squared Error (MSE) or Peak Signal-to-Noise Ratio (PSNR) is that these approaches estimate absolute errors; on the other hand, SSIM is a perception-based model that considers image degradation as perceived change in structural information, while also incorporating important perceptual phenomena, including both luminance masking and contrast masking terms.

# THE PROPOSED APPROACH

We propose an end-to-end deep learning solution for CS, which jointly optimizes the sensing matrix $\Phi$ and the non-linear reconstruction operator, which is parameterized by a coefficients matrix $W$.

The proposed method provides a solution to the following joint optimization problem:

$$\{\widetilde{\Phi}, \widetilde{W}\} = \underset{\Phi, W}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} L(N_W(\Phi x_i), x_i)$$

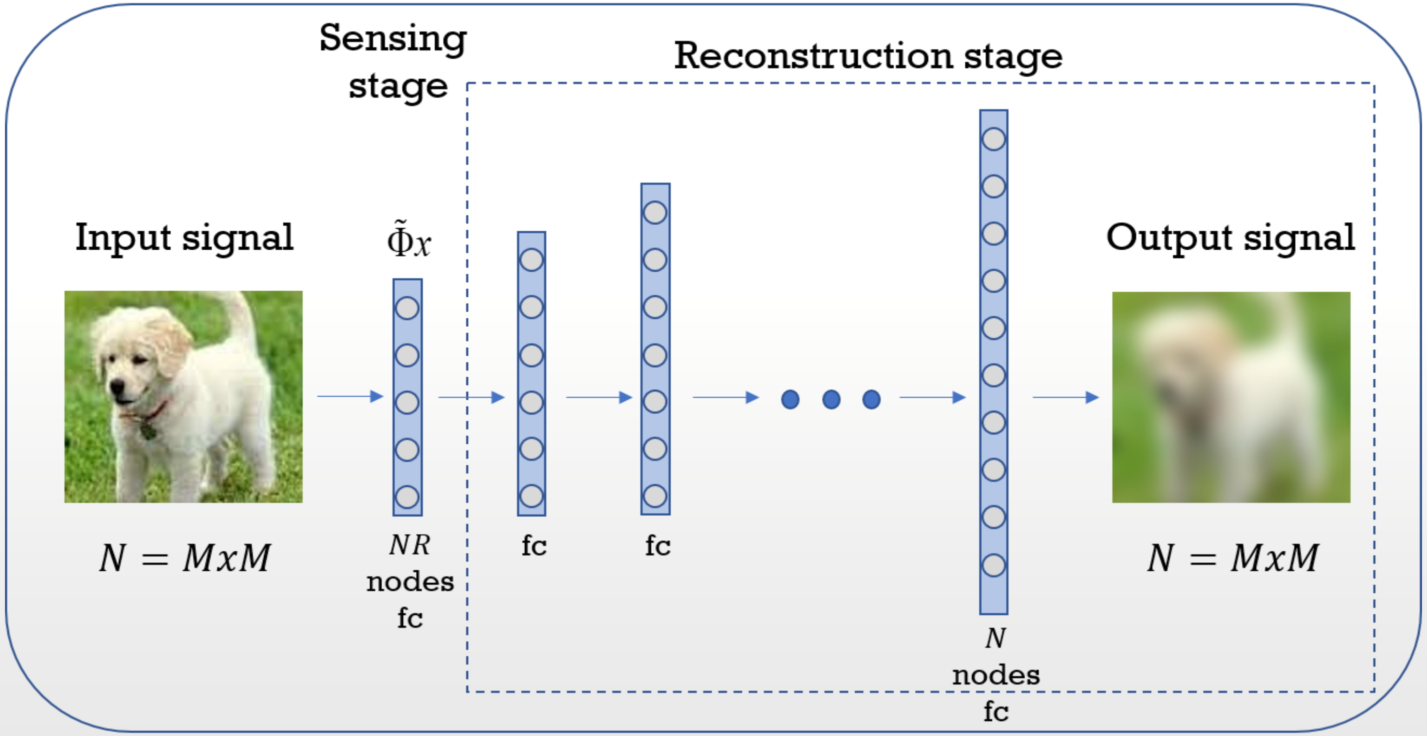
where $\{x_i\}_{i=1}^{N}$ is the collection of $N$ signals. The loss function $L(\cdot,\cdot)$ measures the distance between the input signal and the reconstructed one, provided by the reconstruction operator $N_W(\cdot)$, whose input is the compressed samples, denoted by $\Phi x_i$.

# THE PROPOSED APPROACH

The fully-connected network includes the following layers:

- An input layer with $N$ nodes.
- A compressed sensing fully-connected layer with $NR$ nodes, $R \ll 1$ (its weights form the sensing matrix).
- $K \geq 1$ reconstruction layers with $NB$ nodes in each layer, where $B \in \{1,2\}$. Each layer is followed by a sigmoid activation unit.
- An output layer with $N$ nodes.

# FULLY-CONNECTED NETWORK SCHEME



# DATASET

We trained the proposed architecture on CIFAR10.

The CIFAR10 dataset contains 60,000 color images of $32 \times 32 = 1024$ pixels, drawn from 10 different classes.

This dataset is divided into training and test sets, containing 50,000 and 10,000 images respectively.

Since our proposed network expects grayscale images, for training on CIFAR10 we used only the $1^{st}$ channel for each dataset image.

Moreover, we enlarged the training set to 200,000 samples by rotating the original training set images by 90, 180 and 270 degrees.

# TRAINING WITH SSIM AS LOSS FUNCTION

The SSIM loss function was used with $8 \times 8$ window size.

We tested 2 different weighting functions for the SSIM loss.

The first function is the uniform weighting function, i.e.

$$W(x, y) \equiv 1$$

The second function is:

$$W(x, y) = \log\left(1 + \frac{\sigma_x^2}{C_2}\right)\left(1 + \frac{\sigma_y^2}{C_2}\right)$$

Please read in paper about $\sigma_x^2, \sigma_y^2, C_2$ definitions.

# PERFORMANCE EVALUATION

| Sensing Rate | No. of Measurements | B | K | | Training with SSIM loss $W(x,y) = \log\left(1+\frac{\sigma_x^2}{C_2}\right)\left(1+\frac{\sigma_y^2}{C_2}\right)$ | Training with MSE | Training with SSIM loss $W(x,y) \equiv 1$ |
|---|---|---|---|---|---|---|---|
| 0.125 | 128 | 2 | 1 | SSIM score | 0.885392 | **0.892905** | 0.868903 |
| | | | | MSE score | 0.005893 | **0.003517** | 0.005757 |
| 0.0625 | 64 | 1 | 2 | SSIM score | 0.760333 | **0.793642** | 0.719304 |
| | | | | MSE score | 0.010316 | **0.006469** | 0.011553 |

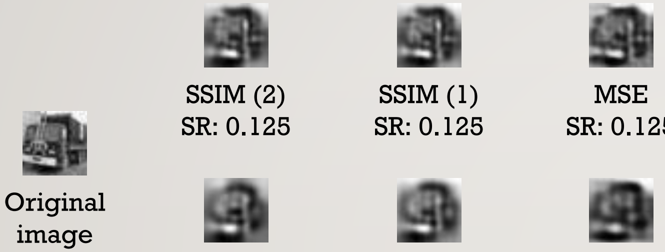
# RECONSTRUCTION RESULTS

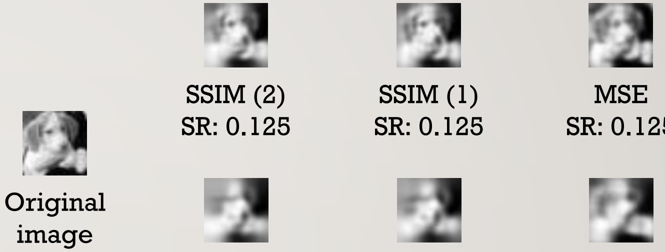


SSIM(1) stands for weight function $W(x,y) = \log\left(1+\frac{\sigma_x^2}{C_2}\right)\left(1+\frac{\sigma_y^2}{C_2}\right)$

SSIM(2) stands for weight function $W(x,y) \equiv 1$

# RECONSTRUCTION RESULTS




SSIM(1) stands for weight function $W(x, y) = \log\left(1 + \frac{\sigma_x^2}{C_2}\right)\left(1 + \frac{\sigma_y^2}{C_2}\right)$

SSIM(2) stands for weight function $W(x, y) \equiv 1$

# FUTURE WORK

- Since calculating SSIM score is more complicated than calculating MSE score for a given image, our approach can be further improved by combining the SSIM as loss function with block-based compressed sensing approach. Learning a deep neural network for blocks reconstruction under the SSIM loss would be significantly faster than learning a network for full-image reconstruction in cases where there are many patches per image.
- Another possible future work direction is to expand the SSIM loss function to multi-scale SSIM loss function.