Implementation of Compressive Sensing of Signals/possibly an image [Just the compression part]

A manifold amount of video data gets generated every minute as we read this document, ranging from surveillance to broadcasting purposes. Two roadblocks restrain us from using this data as such, the first being the storage which restricts us from only storing the information based on hardware constraints. Secondly, the computation required to process this data is costly, which makes it infeasible to work on them. Compressive sensing(CS) is a signal process technique; through optimization, the sparsity of a signal can be exploited to recover it from far fewer samples than required by the Shannon-Nyquist sampling theorem. There are two conditions under which recovery is possible. The first one is sparsity which requires the signal to be sparse in some domains. The second one is incoherence which is applied through the isometric property sufficient for sparse signals. To sustain these characteristics, preserving all attributes in the uncompressed domain would help any kind in this field. However, the existing dataset falls back in terms of continuous tracking of all the objects in the scene. Very few video datasets have comprehensive continuous monitoring of things. To address these problems collectively, in this work, we propose a real-time implementation of Compressive Sensing, where the data is compressed using pixel-wise coded exposure that resolves various other impediments.

A python implementation created by me can be found here: Github Link for Compressive Sensing
Papers related to it: https://ieeexplore.ieee.org/document/9506769

Possibly may need to use the PYNQ board.

OR

One Layer implementation of Deep Learning(DL) framework (either DNN, CNN if possible (fingers crossed)) [NOTE: Not the complete model]

Develop a custom DL layer that isn't already existing in the PYNQ compiler, be it a simple PReLU (a simple extension of ReLU) activation function or a custom sparse CNN layer.

Docs: PYNZ DL Inference Docs