TRANSFORMER COMPRESSED SENSING VIA GLOBAL IMAGE TOKENS

Marlon Bran Lorenzana, Craig Engstrom and Shekhar S. Chandra

The University of Queensland, Brisbane, Australia

ABSTRACT

This is the supplementary material document.

***Index Terms*—** Kaleidoscope, ViT, CS, MRI

1. EXPERIMENT CONFIGURATION

Axial Attention Axial attention is performed in a manner similar to that proposed in [6], where columns and rows are independently treated as input tokens for a Transformer neural networks (TNN) encoder layer. For horizontal h attention, we found it sufficient to:

1. $h \leftarrow$ Encode an image horizontally (rows)
2. $h \leftarrow$ LinearEmbed(h) + PositionEmbeddings
3. $h \leftarrow$ TransformerEncoderBlock(h)
4. $h \leftarrow$ FeedforwardBlock(h)
5. $h \leftarrow$ InverseLinearEmbed(h)
6. Repeat for vertical v image encoding (columns).

Kaleidoscope Transform and Kaleidoscope Tokens: Fig. 1 illustrates the process followed to produce Kaleidoscope tokens (KD) from the Kaleidoscope transform (KT). In the figure, the $(2, 1)$-KT decomposes the image into 4 separate under-sampled copies and concatenates them into a single representation. Each copy corresponds to unique pixel locations. The KD are then extracted by considering the under-sampled copies independently and used as inputs for a Vision Transformer (ViT). For an input image x, the process for utilising the Kaleidoscope Transform (KT) was as follows (KT and iKT are the forward and inverse Kaleidoscope transforms respectively):

1. $x \leftarrow KT(x)$
2. $x \leftarrow$ LinearEmbed(x) + PositionEmbeddings
3. $x \leftarrow$ TransformerEncoderBlock(x)
4. $x \leftarrow$ FeedforwardBlock(x)
5. $x \leftarrow$ InverseLinearEmbed(x)
6. $x \leftarrow iKT(x)$

Training Information: All Deep cascade of Transformer Neural Networks (DcTNN) employ $n_t = 2$, where patch and KD tokens are 16×16 with a model dimension of $d_{model} = 256$. The axial model dimension is $d_{model} = 320$. The feed-forward layer is set to $\lfloor d_{model}^{1.5} \rfloor$. The D5C5 Deep cascade of Convolutional Neural Networks (DcCNN) features 5 convolutional neural network (CNN) blocks with $n_c = 5$ and $n_f = 32$. D7C7 instead features 7 CNN blocks with $n_c = 7$ and $n_f = 64$.

We used a subset of the NYU fastMRI DICOM brain database to train and test all models [15]. In total there were $64,180$ T1-w slices. Training, validation and testing consisted of 80%, 10% and 10% of these images. For this study, we simulate single-coil magnitude images and cropped each to 320×320 resolution. Discrete Fourier space was sampled using a 1D Gaussian random mask (see Fig. 2). We use peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) to evaluate closeness to the original image. All experiments were conducted on an NVIDIA SXM-2T V100 graphics processing unit (GPU) with 32GB of vRAM. All networks were implemented in PyTorch and trained using the Adam optimiser. All DcTNN were trained with a learning rate of 10^{-4} and a batch size of 75. All DcCNN were trained following the original implementation [9], with batch sizes of 75 and 20 for D5C5 and D7C7 respectively.

DcTNN models were trained with mean-absolute-error (MAE) and an additional total-variation (TV) constraint. We found including the TV constraint helped to produce smoother reconstructions and limit high frequency artefacts. The resulting loss function was,

$$L_{MAE+TV} = \eta L_{MAE} + \gamma L_{TV}. \tag{1}$$

For our experiments, $\eta = 10^0$ and $\gamma = 10^{-7}$. DcCNN was instead trained using mean-squared-error (MSE) loss as-per the original implementation. Transformer networks trained for 400 epochs and CNN networks trained for 75. The network with the lowest validation loss was then chosen for testing.

2. ADDITIONAL RESULTS

Fig. 3 illustrates the relative performance between DcCNN and DcTNN for various under-sampling factors. In this comparison, we see that while PSNR and SSIM scores are higher for DcCNN at R4 (Fig. 3a), image textures and noise-like

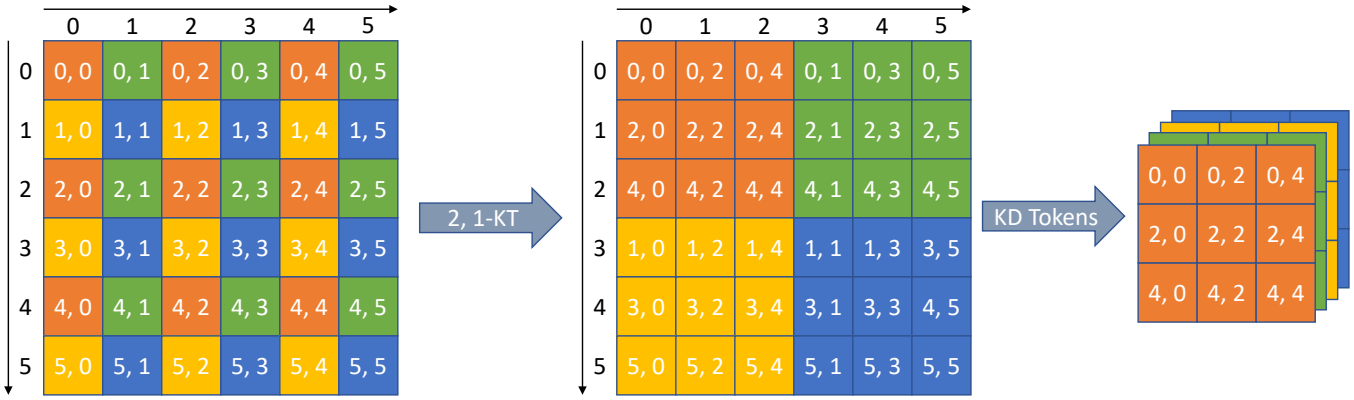
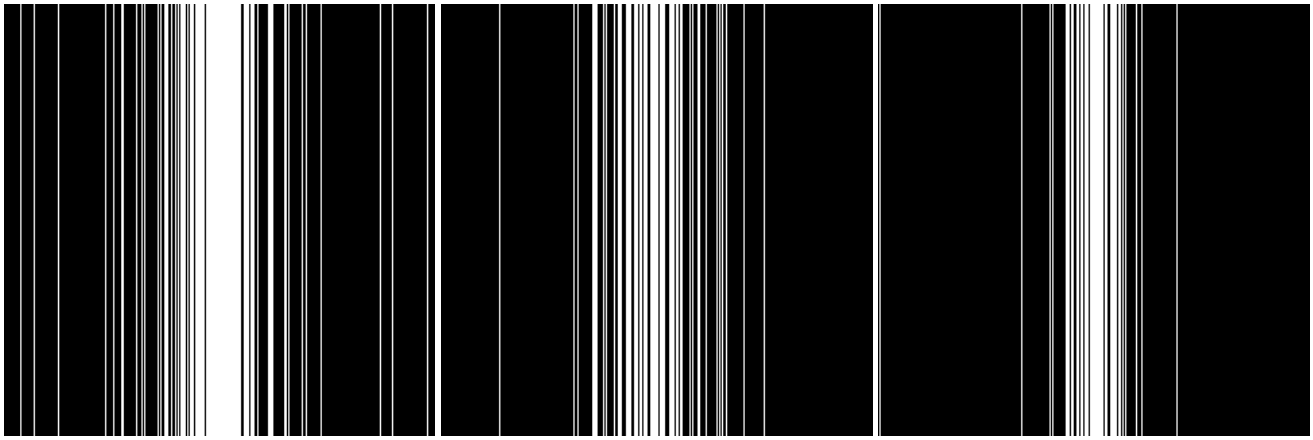


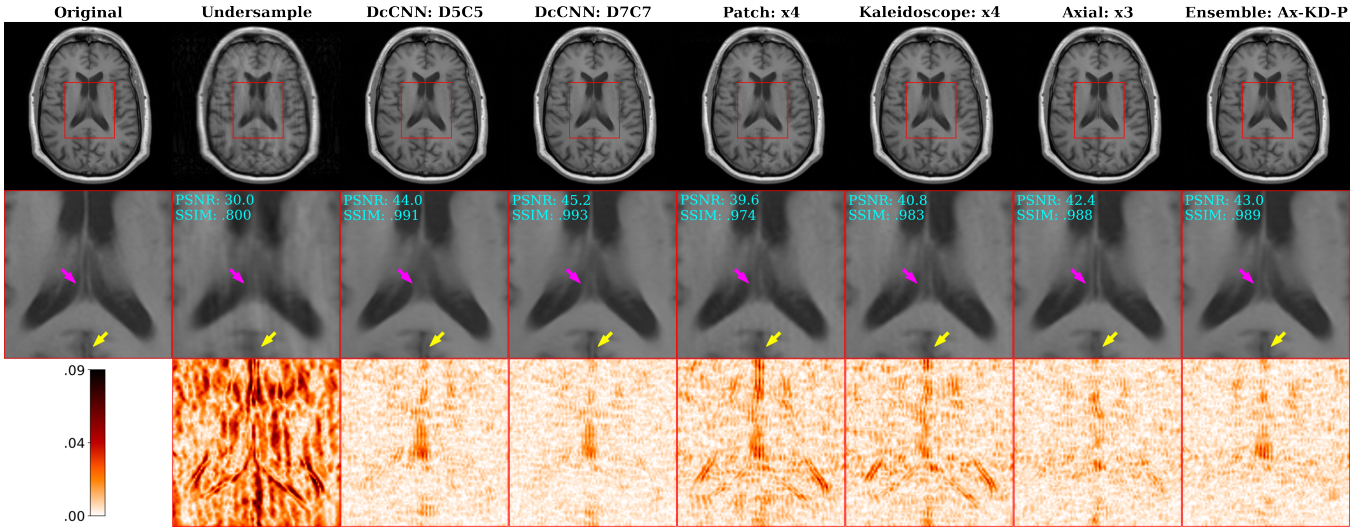
Fig. 1: Example of utilising the Kaleidoscope Transform (KT) to produce kaleidoscope tokens (KD). Colours indicate which Kaleidoscope tokens (KD) embedding each "pixel" corresponds to and are presented on a regular grid. On the left, each pixel is in its original location. Middle, has a $2, 1$-KT applied, meaning that pixels corresponding to the 4 under-sampled KD images are grouped together. Finally resulting in 4 concatenated down-sampled versions of the original image. These are considered separately as inputs.



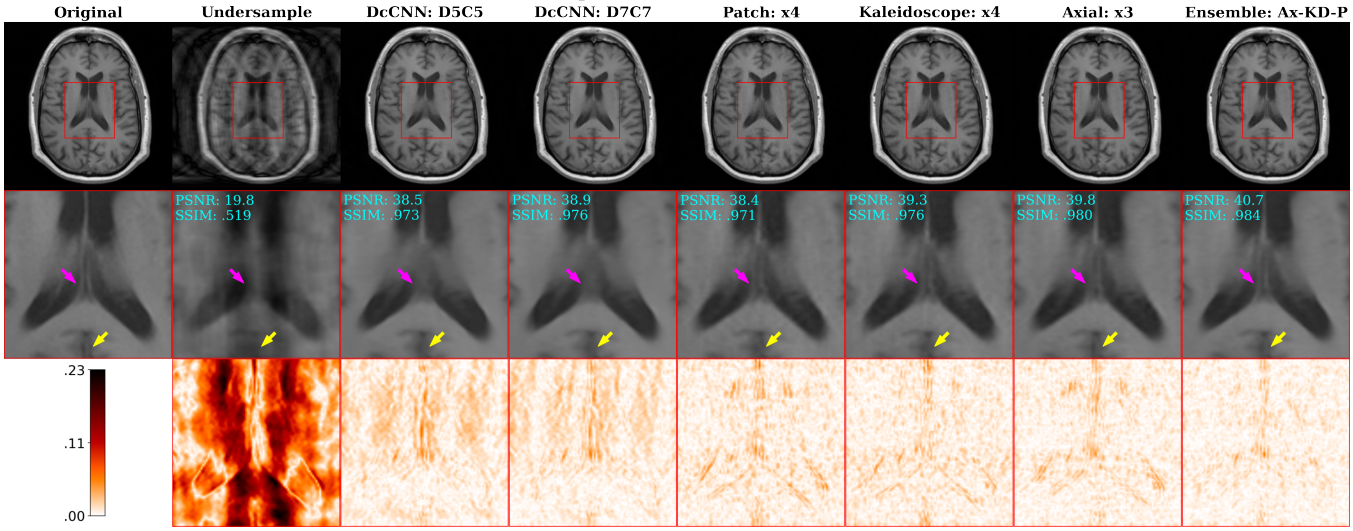
(a) Reduction Factor: 4 (b) Reduction Factor: 6 (c) Reduction Factor: 8

Fig. 2: Discrete Fourier space sampling masks used in our experiments. Each is a one-dimensional (1D) Gaussian random mask.

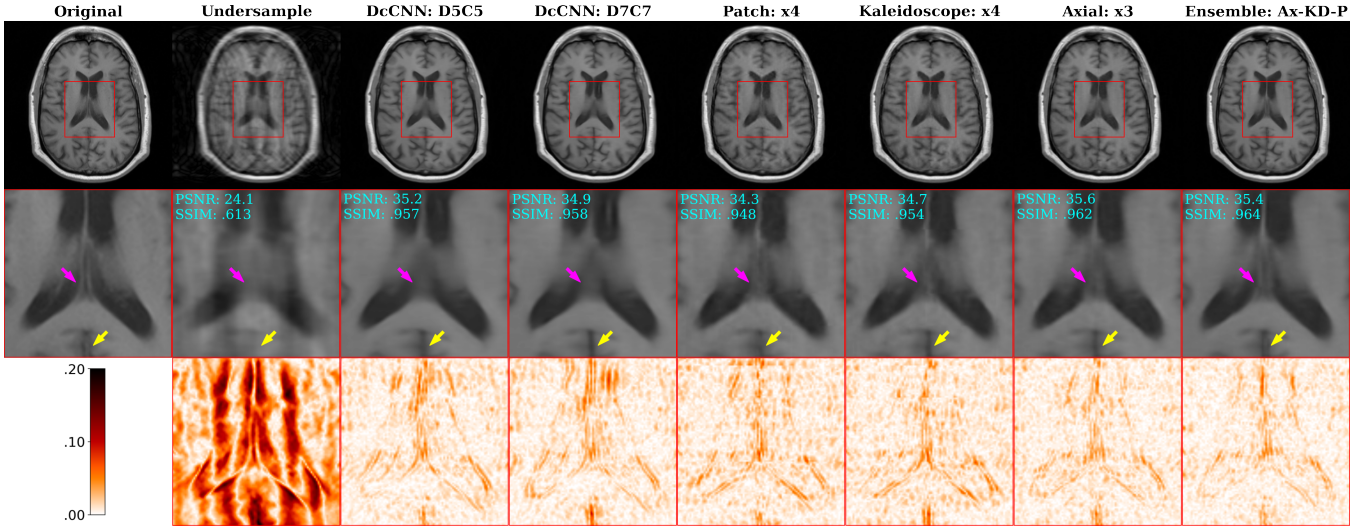
image features, such as the left and right ventricle, are better preserved by DcTNN methods. Further, Fig. 3c demonstrates that the ensemble DcTNN is capable of recovering significantly degraded image features at a high reduction factor (R8).



(a) Comparison at reduction factor 4.



(b) Comparison at reduction factor 6.



(c) Comparison at reduction factor 8.

Fig. 3: Reconstruction performance demonstrated at R4, R6 and R8 reduction factors for a cross-sectional brain MR image. The zoomed area (red rectangle) includes the region of the right and left lateral ventricle. Patch and Kaleidoscope DcTNN are comprised of 4 TNN layers. Axial and Ensemble: Ax-KD-P each have 3. Finally, DcCNN: D5C5 and D7C7 are the reference.