Cosmic Ray Detection

Atharv Ramesh Nair

January 1, 2024

Table of Content

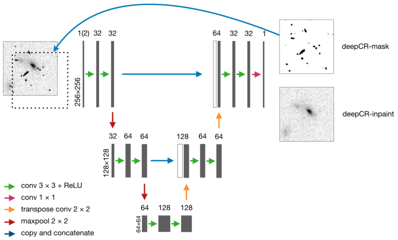
DeepCR

- Trained on HST ACS/WFC imaging data of 3 categories
  - extragalactic field
  - globular cluster
  - local group galaxies
- CR Pixels Labelling using ASTRO-DRIZZLE Pipeline
- **Sky Augmentation**

$$\textbf{Actual}: \; n = (f_{\text{star}} + f_{\text{sky}}) \cdot t_{\text{exp}} + n_{\text{CR}}$$

$$\textbf{Augmented}: \; n_0 = n + \alpha \cdot f_{\text{sky}} \cdot t_{\text{exp}} = f_{\text{sky}} \cdot (1+\alpha) \cdot t_{\text{exp}} + n_{\text{CR}}$$

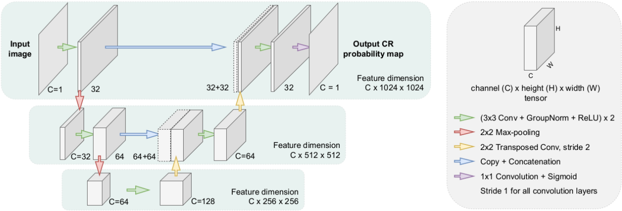
Different exposure times using different values of α



Cosmic CONN- DATASET

- Trained on data from LCO (*Las Cumbres Observatory*) global network of 23 telescopes
- 4500 scientific images : $4K \times 4K, 3K \times 2K, 2K \times 2K$ images
- **BANZAI** data reduction pipeline for instrumental signature removal
  - bad-pixel removing
  - bias and dark removal
  - flat-field correction
- Focus on distinguishing CR pixels from astronomical sources
- Near Earth Objects, Satellites contribute to small fraction of false positive labels

Cosmic CONN - Architecture



UNet based architecture

- Authors claim simple UNet is not able to handle higher dynamic range and extreme spatial variations

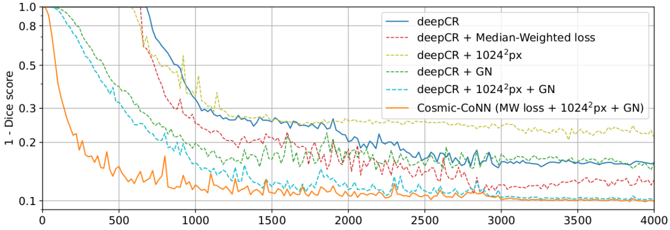
Cosmic CONN - Deep Learning Framework

- **Median Weighted Loss Function :**

$$L(P, Y, M) = -\sum_{i,j}(Y_{ij}log(P_{ij}) + M_{ij}(1 - Y_{ij})log(1 - P_{ij}))$$

  where P,Y, M are the predictions, labels and median weighted mask respectively
- Median Mask obtained from transformation on the median of consecutive exposures
  - sky subtraction
  - clipping 1-5 σ's
  - Gaussian Smoothing (5x5 Kernel with $\sigma = 2$)
  - Unit Normalisation and Clamping with lower bound (α)

Results

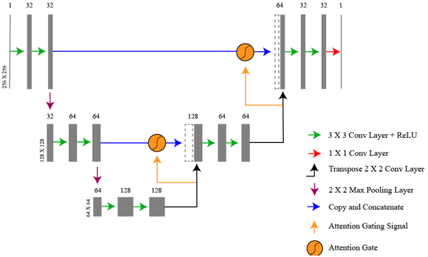


Dice Score vs Epochs for different types of models

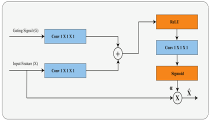
Method	Dice score > 0.85	LCO Precision	Gemini 1×1 Precision	Gemini 2×2 Precision
deepCR (baseline)	2980	89.19%	79.59%	84.88%
deepCR + Median-Weighted loss	2080	92.98%	78.76%	83.08%
deepCR + 1024^2px	n/a	89.35%	82.57%	86.55%
deepCR + GN	1420	90.82%	77.07%	89.30%
deepCR + 1024^2px + GN	1040	93.17%	84.54%	92.09%
Cosmic-CoNN (MW loss + 1024^2px + GN)	**380**	**93.40%**	**86.80%**	**94.37%**

Number of epochs to achieve Dice Score > 0.85 and Precision at 95 % Recall

Attention UNet

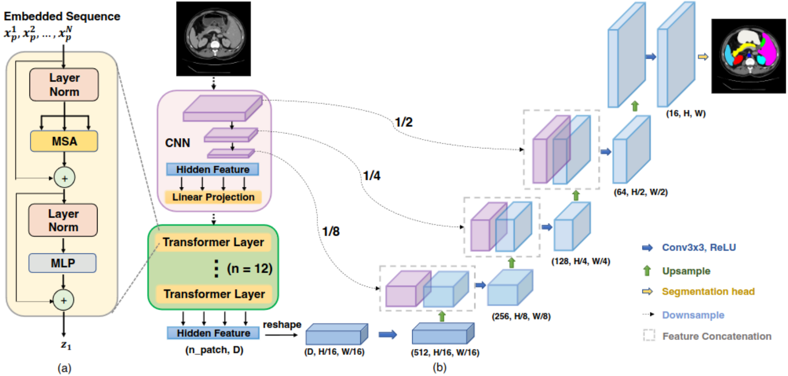


Attention UNet



Attention Gate

TransUNet - Architecture



TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation - Chen et al.

TransUNet Architecture

ResNet-50 + ViT Encoder

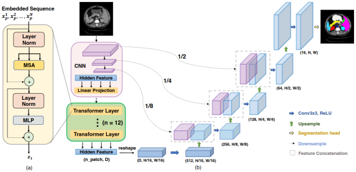
- Base ViT :Input resolution – (224,224)
- P = 16, D = 768
- MLP size = 3072
- No. of layers = 12
- No. of heads = 12

Cascading Upsampler (CUP) Decoder

- Bilinear Upsampling
- Concatenate features from ResNet encoder

Training

- SGD with learning rate 0.01, momentum 0.9
- Weight decay 1e-4



Architecture

Results from Original Paper

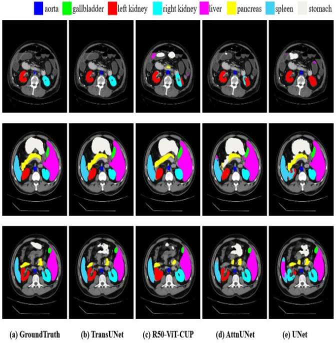


Table 1: Comparison on the Synapse multi-organ CT dataset (average dice score % and average hausdorff distance in mm, and dice score % for each organ).

Framework		Average		Aorta	Gallbladder	Kidney (L)	Kidney (R)	Liver	Pancreas	Spleen	Stomach
Encoder	Decoder	DSC ↑	HD ↓								
V-Net [9]		68.81	-	75.34	51.87	77.10	80.75	87.84	40.05	80.56	56.98
DARR [5]		69.77	-	74.74	53.77	72.31	73.24	94.08	54.18	89.90	45.96
R50	U-Net [12]	74.68	36.87	84.18	62.84	79.19	71.29	93.35	48.23	84.41	73.92
R50	AttnUNet [13]	75.57	36.97	55.92	63.91	79.20	72.71	93.56	49.37	87.19	74.95
ViT [4]	None	61.50	39.61	44.38	39.59	67.46	62.94	89.21	43.14	75.45	69.78
ViT [4]	CUP	67.86	36.11	70.19	45.10	74.70	67.40	91.32	42.00	81.75	70.44
R50-ViT [4]	CUP	71.29	32.87	73.73	55.13	75.80	72.20	91.51	45.99	81.99	73.95
TransUNet		**77.48**	**31.69**	87.23	63.13	81.87	77.02	94.08	55.86	85.08	75.62

Synapse multi-organ segmentation dataset.

- 30 abdominal CT scans with 3779 axial clinical CT images of 512 $\times$ 512 pixels
- 18 training cases (2212 axial slices) and 12 cases for validation

Results from Original Paper (Continued)

Framework	Average	RV	Myo	LV
R50-U-Net	87.55	87.10	80.63	94.92
R50-AttnUNet	86.75	87.58	79.20	93.47
ViT-CUP	81.45	81.46	70.71	92.18
R50-ViT-CUP	87.57	86.07	81.88	94.75
TransUNet	89.71	88.86	84.53	95.73

Comparison of Dice Score

Automated cardiac diagnosis challenge

- Segment into left ventricle (LV), right ventricle (RV) and myocardium (MYO)
- 70 training cases (1930 axial slices), 10 cases for validation and 20 for testing

Results from Ablation Studies

- 6.88 % improvement when trained on (512,512) patch at the expense of much larger compute cost
- Consistent improvement in performance while increasing the skip connections [0,1,3]
- Marginal Improvement (0.35 %) while using (8,8) patches

DeCAM Dataset

- 56 raw $4k \times 2k$ images from four photometric bands (g, r, i, z) with 90s exposure time
- Synthetically generated CR Hits
  - CR Identification using Astro-SCRAPPY and replacement using median filter to get uncontaminated image
  - Dark Frame Extraction: obtained when no light is incident on sensor
  - Mask M of affected pixels is obtained as follows:

$$M_p = \begin{cases} 1 & \text{if } D_p > m_D + 3\sigma_D \\ 0 & \text{otherwise} \end{cases}$$

  - M is dilated with a 3×3 pixel kernel to create the final $M^{(D)}$ mask. This mask serves as ground truth.
- Divided into 256×256 crops to facilitate batch training
- Split: Training (90%) Validation (10%)
- Same Data Augmentation (simulating Exposure Time) used in DeepCR

Hyperparameters

Following is the best TransUNet configuration from all tuning performed

Vision Transformer

- Patch Size : 16×16
- Hidden Size : 384
- MLP DIM : 768
- No. of heads : 6
- No of encoder layers : 6

CNN

- No. of Convolutional Layers : [2,2,2]
- Loss Function : $0.5(BCE\_Loss + 0.5(1 - Dice\_Score))$

Number of Parameters : **16,657,809**

Results

Fixed LR at **0.001** for 100 epochs

Table: Comparison of different configurations of TransUNet

Configuration	No. of parameters	TPR @ 0.01% FPR
Default	16,657,809	0.972736
Loss = 0.9*BCE + 0.1*Dice	16,657,809	0.970416
Dropout = 0.1	16,657,809	0.923
Loss = BCE	16,657,809	0.966
no. of heads = 3]	16,657,809	0.9645
Conv layers: [3,3,3]	18,125,457	0.972775

With Learning Rate Scheduler

Configuration	TPR @ 0.01% FPR
ReduceLROnPlateau	0.9735
Multi Step LR Scheduler	0.9729

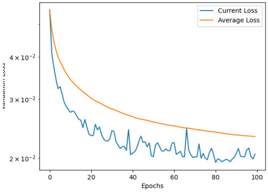
Results

Starting LR $= 0.001$ and reduced by 0.1 every time validation loss was flat

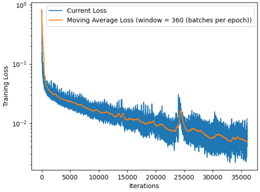
Table: Comparison of Different Architectures

Configuration	TPR @ 0.01% FPR	TPR @ 0.1% FPR	Dice Score
TransUNet	0.9762	0.9890	0.965825
Att-UNet	0.9753	0.9964	0.965606
Cosmic CoNN	0.9726	0.9964	0/96328

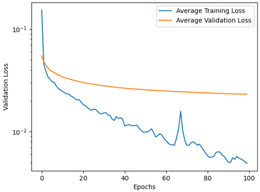
Training Plots - Fixed LR = 0.001 (Default)



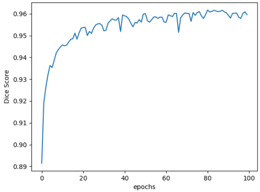
Validation Loss



Training Loss

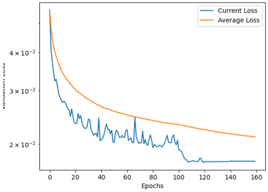


Training vs Validation

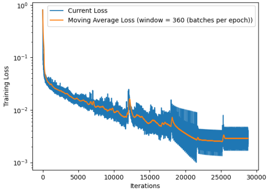


Dice Score

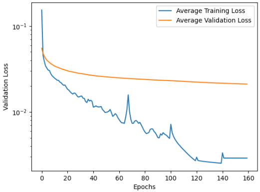
Training Plots Reducing LR manually



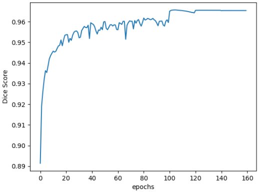
Validation Loss



Training Loss



Training vs Validation



Dice Score

Dictionary Learning Approach

- Image is divided into 11×11 patches
- A sparse representation (256×1) y from 11×11 (121) patches x using a learned dictionary
- Random Forest Classifier to distinguish between sparse representations y for CR and Non-CR patches

Learning the Dictionary

$$\hat{y} = \text{argmin} \|x - Dy\|_2^2 \quad \text{s.t} \quad \|y\|_0 \leq K$$

where D is the dictionary Done using Orthogonal Matching Pursuit (OMP) and Approximate K-SVD [Rubinstein et.al. Efficient Implementation of the K-SVD Algorithm Using Batch Orthogonal Matching Pursuit. CS Technion. 40.]

Orthogonal Matching Pursuit

Algorithm 5 APPROXIMATE K-SVD

Input: Signal set $\mathbf{X}$, initial dictionary $\mathbf{D}_0$, target sparsity K, number of iterations k.

Output: Dictionary $\mathbf{D}$ and sparse matrix $\mathbf{\Gamma}$ such that $\mathbf{X} \approx \mathbf{D\Gamma}$

Init: Set $\mathbf{D} := \mathbf{D}_0$

for $n = 1 \ldots k$ **do**

$\quad \forall i: \quad \mathbf{\Gamma}_i := \underset{\underline{\gamma}}{\text{Argmin}} \, \|\underline{x}_i - \mathbf{D}\underline{\gamma}\|_2^2 \quad \text{Subject To} \quad \|\underline{\gamma}\|_0 \leq K$

$\quad$ **for** $j = 1 \ldots L$ **do**

$\qquad \mathbf{D}_j := \underline{0}$

$\qquad I := \{$*indices of the signals in* $\boldsymbol{X}$ *whose representations use* $\underline{d}_j\}$

$\qquad \underline{g} := \mathbf{\Gamma}_{j,I}^T$

$\qquad \underline{d} := \mathbf{X}_I \underline{g} - \mathbf{D\Gamma}_I \underline{g}$

$\qquad \underline{d} := \underline{d} / \|\underline{d}\|_2$

$\qquad \underline{g} := \mathbf{X}_I^T \underline{d} - (\mathbf{D\Gamma}_I)^T \underline{d}$

$\qquad \mathbf{D}_j := \underline{d}$

$\qquad \mathbf{\Gamma}_{j,I} := \underline{g}^T$

$\quad$ **end for**

end for

Efficient K-SVD

Algorithm 5 Approximate K-SVD

Input: Signal set $\mathbf{X}$, initial dictionary $\mathbf{D}_0$, target sparsity K, number of iterations k.

Output: Dictionary $\mathbf{D}$ and sparse matrix $\mathbf{\Gamma}$ such that $\mathbf{X} \approx \mathbf{D\Gamma}$

Init: Set $\mathbf{D} := \mathbf{D}_0$

for $n = 1 \dots k$ **do**

    $\forall i : \quad \mathbf{\Gamma}_i := \underset{\underline{\gamma}}{\text{Argmin}} \, \|\underline{x}_i - \mathbf{D}\underline{\gamma}\|_2^2 \quad \text{Subject To} \quad \|\underline{\gamma}\|_0 \leq K$

    for $j = 1 \dots L$ **do**

        $\mathbf{D}_j := \underline{0}$

        $I := \{\textit{indices of the signals in } \mathbf{X} \textit{ whose representations use } \underline{d}_j\}$

        $\underline{g} := \mathbf{\Gamma}_{j,I}^T$

        $\underline{d} := \mathbf{X}_I\underline{g} - \mathbf{D\Gamma}_I\underline{g}$

        $\underline{d} := \underline{d}/\|\underline{d}\|_2$

        $\underline{g} := \mathbf{X}_I^T\underline{d} - (\mathbf{D\Gamma}_I)^T\underline{d}$

        $\mathbf{D}_j := \underline{d}$

        $\mathbf{\Gamma}_{j,I} := \underline{g}^T$

    end for

end for

Results

Dictionary Learning

Configuration	TPR @ 0.01% FPR	Dice Score
Att-UNet	0.9727	0.9800
TransUNet	0.96	0.972

Without Dictionary Learning

Configuration	TPR @ 0.01% FPR	Dice Score
Att-UNet	0.971	0.979
TransUNet	0.965	0.972