DeepMerge: Studying Distant Merging Galaxies with Deep Neural Networks

Ćiprijanović, A.[1] Snyder, G. F.[4] Nord, B.[1,2,3], Peek, J. E.

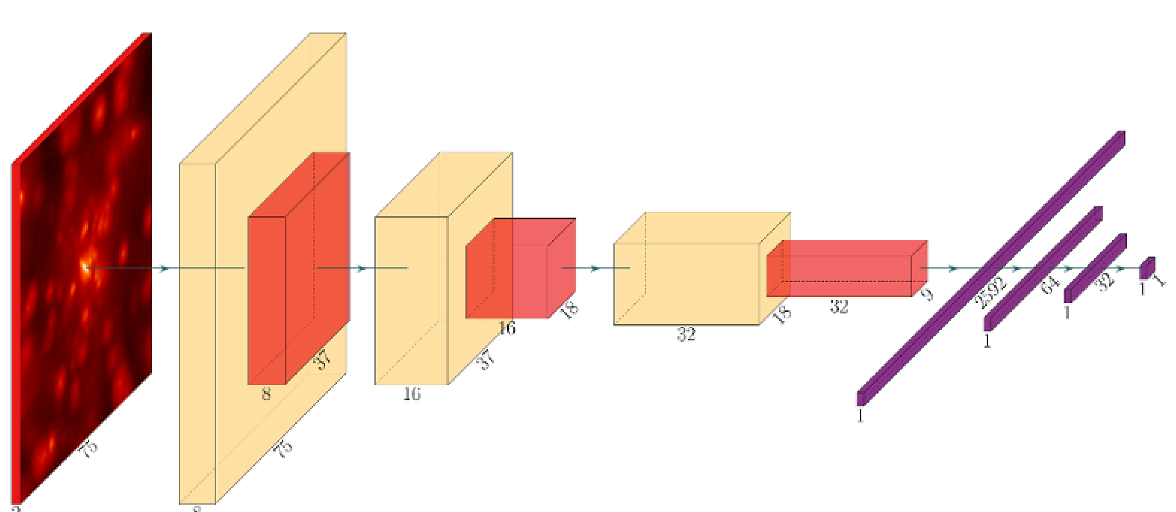
[1]Fermi National Accelerator Laboratory, Batavia IL; [2]Kavli Institute for Cosmological Physics, University of Chicago, Chicago IL; [3]Department of Astronomy and Astrophysics, University of Chicago, Chicago IL; [4]Space Telescope Science Institute, Baltimore MD; [5]Department of Physics and Astronomy, Johns Hopkins University, Baltimore MD.

Introduction:

The hierarchical merging of galaxies is a probe of the cosmos to test the canonical ΛCDM cosmology paradigm. A particularly interesting period is "cosmic high noon" at redshifts z∼2−3, during which star formation rates were the highest, and significant amounts of stellar mass assembled into galaxy-scale bodies. Detecting galaxy mergers in observations by conventional automated methods (which use extracted parameters of galaxy structure - asymmetry, clumpiness, concentration etc.) or by visual inspection has proven to be quite time-consuming and prone to errors.

Convolutional Neural Network (CNNs) are a primary representative of deep learning algorithms which are used in computer vision tasks by training to detect features in images. CNNs were already used for classification of low-redshift merging galaxies [1, 2]. Here we will use CNN to to learn directly from images (without the need to extract morphology parameters) of distant merging galaxies in order to distinguish between merging and non-merging objects [3].

Method:

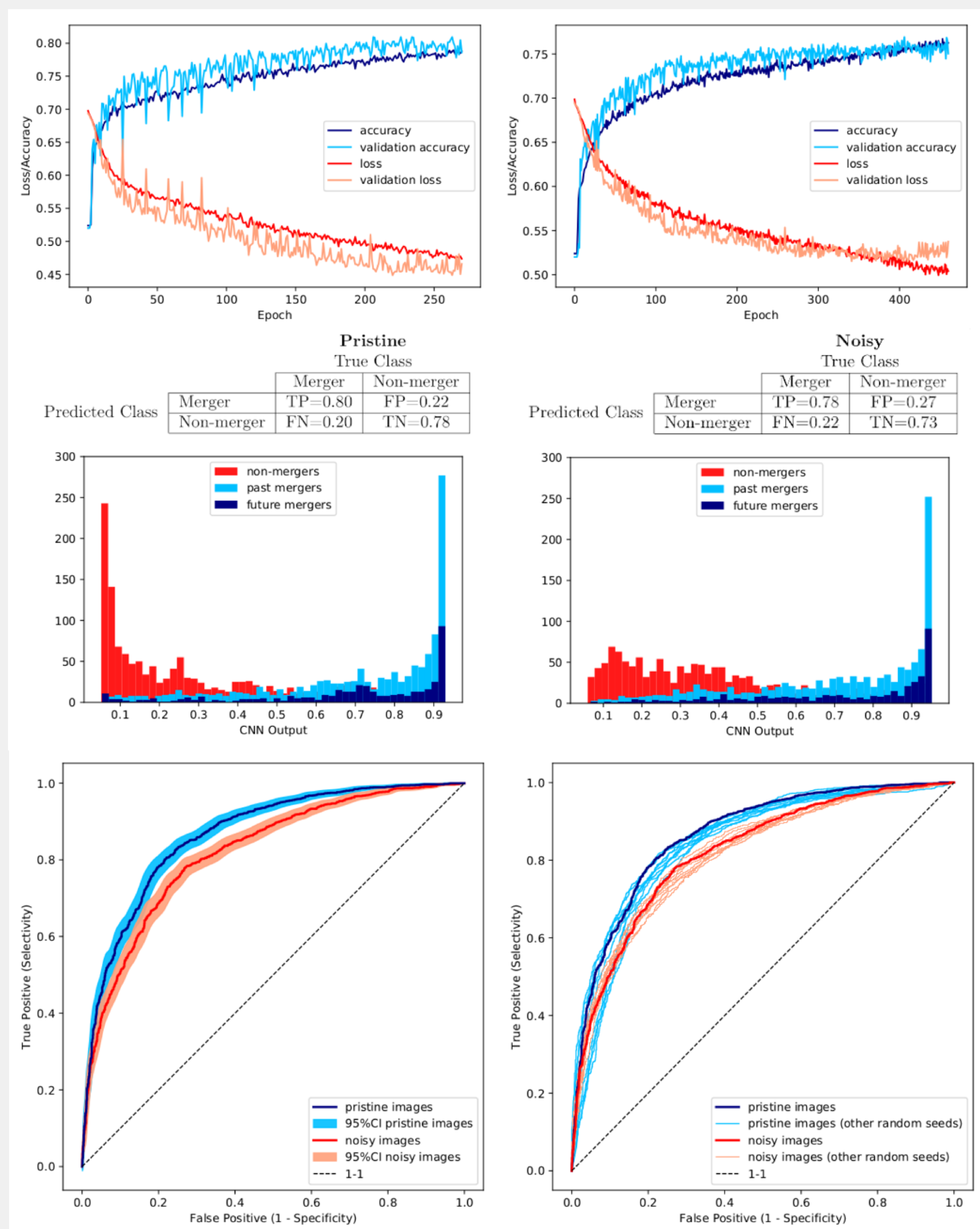


DeepMerge architecture (convolutional layers - orange; pooling - red; dense - violet)

We train DeepMerge CNN with images of merging galaxies at redshift z=2 from Illustris-1 simulation [4] (pristine set of images - simulated images with added PSF; noisy set - simulated images+PSF+random sky shot noise). We use the same set of hyper-parameters with both data sets. Over-fitting is treated with dropout, L2 regularization and early stopping. Optimization was done using Adam optimizer, with binary cross-entropy as our loss function.

Fermilab

Results:



	Noisy / Pristine
Accuracy	0.76 / 0.79
Precision	0.77 / 0.81
Recall	0.78 / 0.80
AUC	0.82 / 0.86
F1 Score	0.77 / 0.81
Brier Score	0.17 / 0.15

Top *- loss and accuracy during training.*

Middle *- confusion matrices and histograms of output values for past (objects that merged in 250Myr before the image was taken), future mergers (objects that will merge in the next 250Myr after the image was taken) and non-mergers (Pristine images - left; Noisy images - right).*

Bottom Right *- ROC curves with 95%CI generated by bootstrapping test set of images*

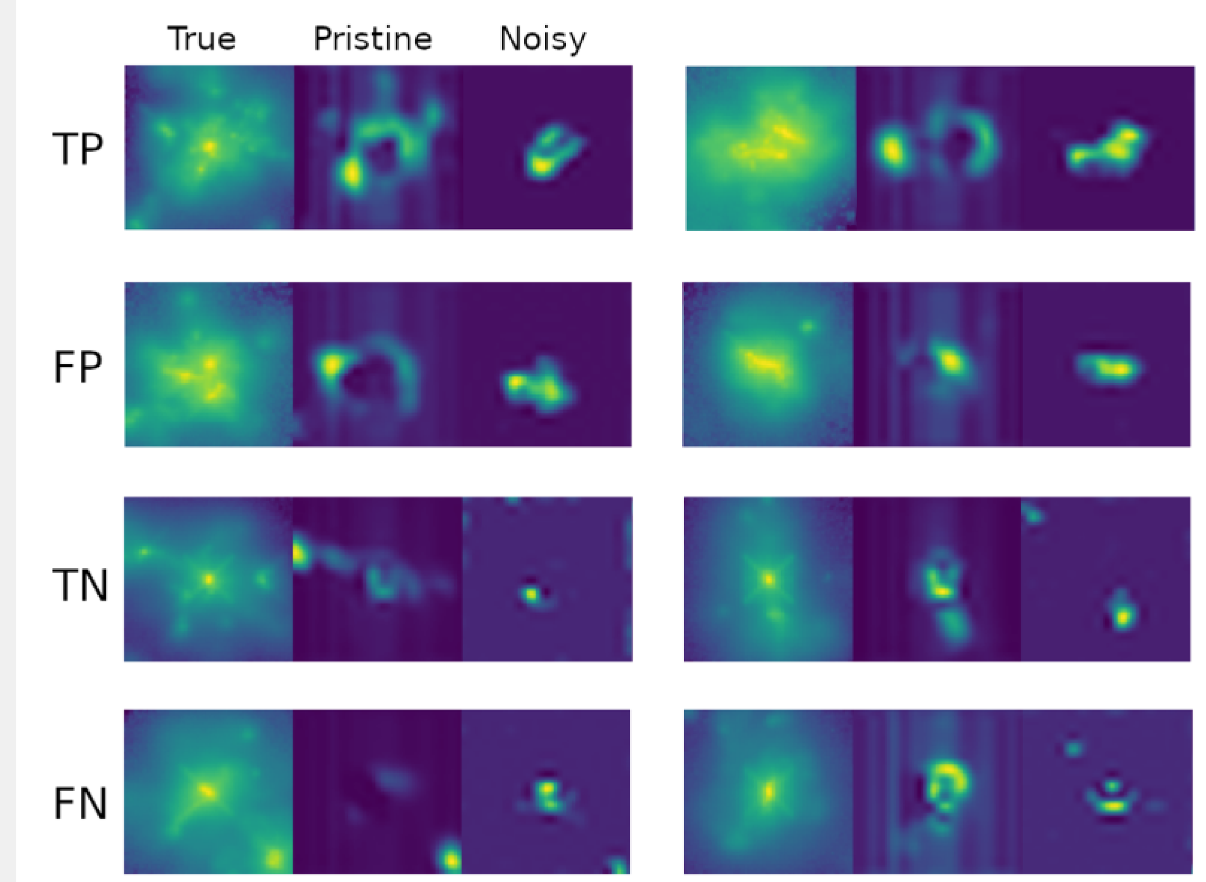
Bottom Left *- ROC curves after testing with different random seed used for image shuffling before making train, validation and test sets.*

SERBIAN ASTRONOMICAL CONFERENCE

Grad-CAMs:

Gradient-weighted Class Activation Mapping (Grad-CAM) - is used to show which pixels of the image were the most important for classification into a particular class [5].

In case of pristine images most important regions for mergers are faint region in galaxy periphery (ring like features on Grad-CAMs), while non-mergers are classified based on more central regions. In case of noisy images faint structures are not visible, and mergers are classified using information from the central regions of galaxies.



Grad-CAMs for images correctly /incorrectly classified as mergers (true positives -TP / false positives -FP) and correctly / incorrectly classified as non-mergers (true

Conclusion:

- DeepMerge performs better than standard classification methods and the Random Forest classifier from [6], in case of distant galaxy classification.
- Classification is based on images themselves, and the performance is equally good for both past and future mergers.
- Using Grad-CAMs we find which regions are important for pristine/noisy classification.

References:

[1] Ackermann, S., et al.: 2018. MNRAS, 479, 415–425.

[2] Pearson, W.J. et al.: 2019, A&A, 626, A49.

[3] Ćiprijanović, A. et al.:2020, Astron.Comput., 32, 100390.

[4] Selvaraju, R.R. et al.: 2016, arXiv:1610.02391.

[5] Vogelsberger, M. et al.: 2014 . MNRAS, 444, 1518–1547.

[6] Snyder, G.F. et al.: 2019, MNRAS, 486, 3702–3720.