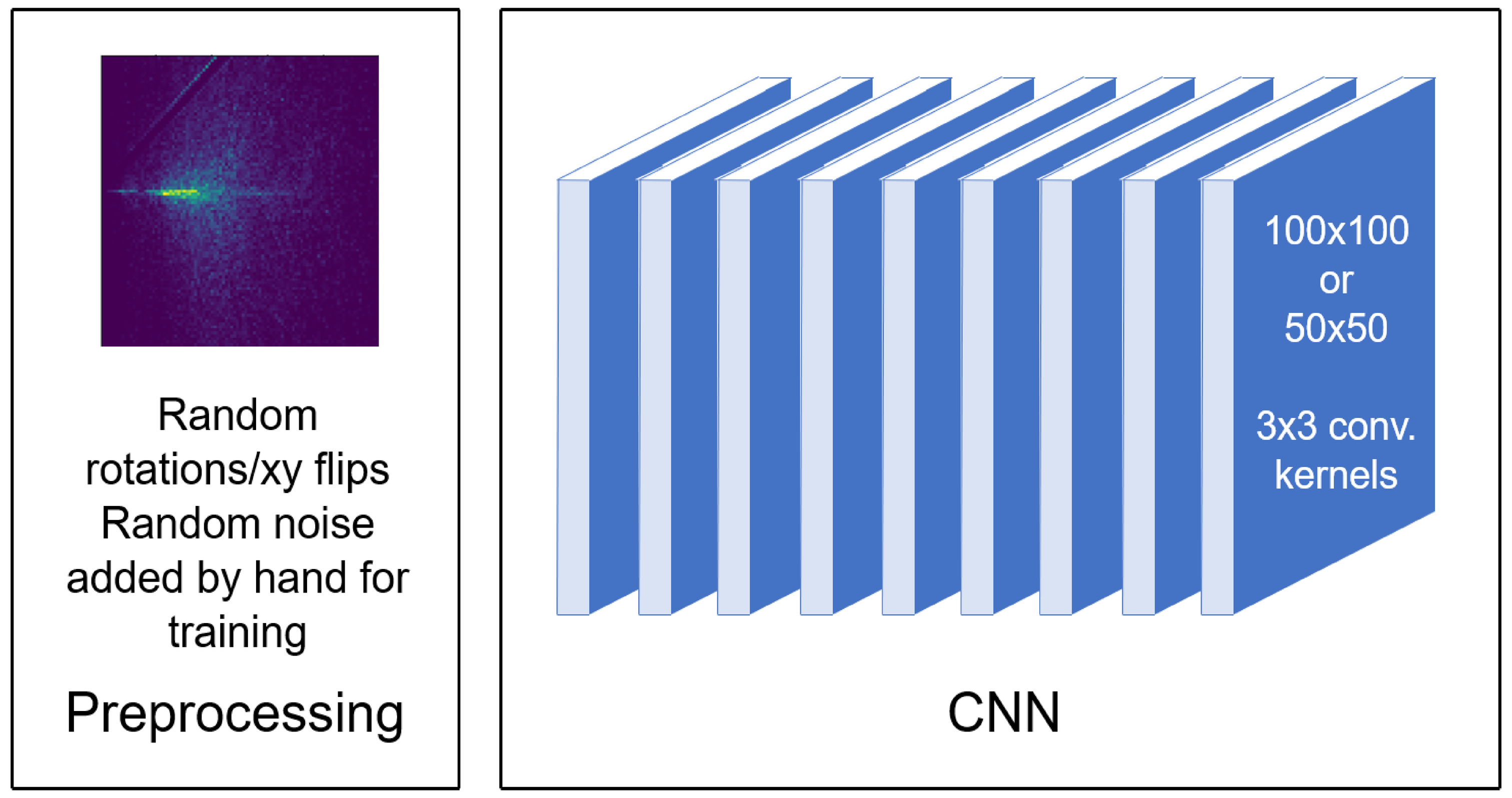
AI Denoising to Accelerate Detector Simulation

Lena Franklin, University of Maryland at College Park

Introduction

Detector simulation is critical to experimental HEP; however this simulation (commonly done through toolkits such as Geant4) is computationally intensive. Performance can be improved somewhat through technical optimization, but more is needed. Using machine learning (ML) to accelerate simulation is a promising field, however efforts to use generative adversarial networks (GANs) or optimized autoencoders have faced issues. Using convolutional neural networks (CNNs) for denoising has been successful in non-HEP applications such as image processing. This poster investigates the efficacy of using CNNs to denoise Geant4 simulations. This could increase the accuracy of simulations performed under settings designed to increase computational efficiency.

Network Architecture



We use a network with nine convolutional layers and 3x3 pixel kernels and 100 feature vectors at each layer. A rectified linear unit (ReLU) activation function is applied at each layer. At each convolutional layer, appropriate padding is used to preserve the size of the input tensor in order to produce a denoised output tensor with the same dimensions as the original noisy tensor. The network accepts 2-dimensional input images with a single channel which records the energy deposited in an x-y area of the simulated detector.

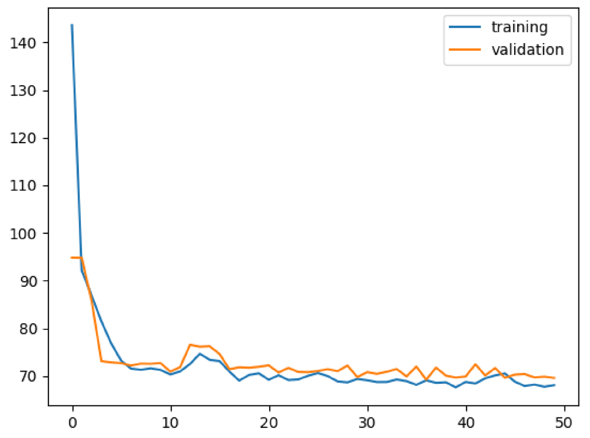
This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internships Program (SULI).

Loss and Training

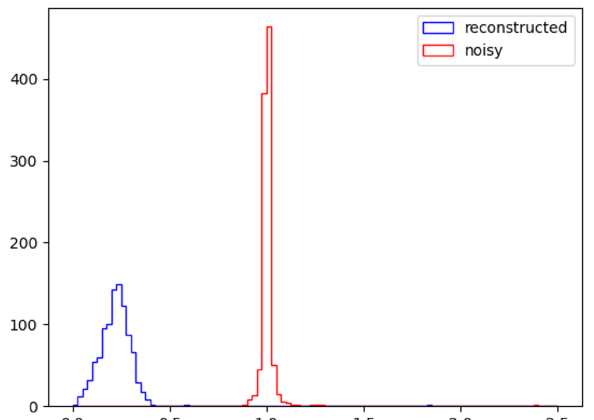
Our loss function prioritizes interesting regions of the simulation with high deposited energy by splitting the images into patches and returning the mean absolute error of the patch with the greatest loss, which corresponds to the patch with greater energy deposits.

Initial training was done with 100x100 pixel images generated from Geant4 simulations of single 100 GeV photons in the CMS electromagnetic calorimeter. These images were randomly rotated during preprocessing.

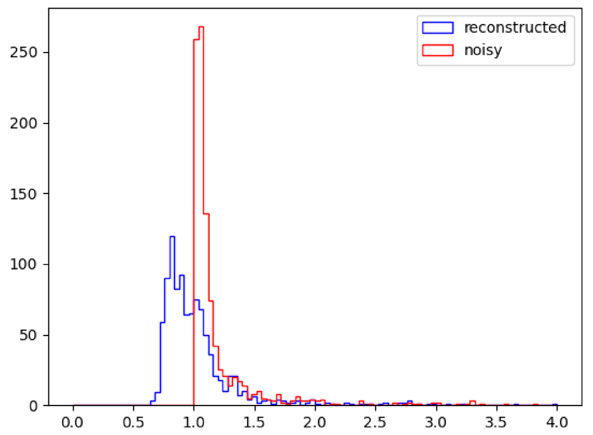
Loss during training as a function of epoch for a model trained for 100x100 pixel images. Loss values reflect the average per-pixel energy difference in the region of interest. 50x50

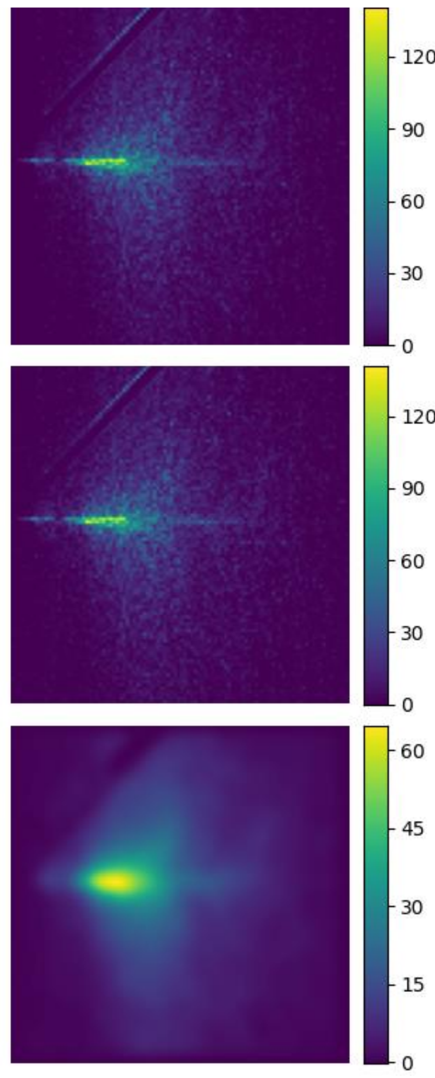


The ratio of the maximum energy value in each image in the reconstructed to ground truth images and the noisy and ground truth images for a network trained for 50 epochs on 100x100 pixel images.



The ratio of total energy present in the reconstructed vs ground truth images and the noisy and ground truth images for a network trained for 50 epochs on 100x100 pixel images.





Visualization of input (top), noisy (middle) and reconstructed (bottom) energy for 100x100 pixel images and 50x50 pixel patches in the loss function..

Optimizing Network Performance

Noise Level	Total Energy Ratio: Reconstructed/Truth	Total Energy Ratio: Noisy/Truth	Maximum Energy Value Ratio: Reconstructed/Truth	Maximum Energy Value Ratio: Noisy/Truth
σ = 1	0.92	1.09	0.25	1.00
σ = 3	0.54	1.23	0.18	1.00
σ = 5	0.51	1.42	0.16	1.00
σ = 10	0.51	1.93	0.15	0.99
σ = 15	0.52	2.46	0.15	1.03
σ = 20	0.50	3.02	0.12	1.08

Results of a 50 epoch training of images of 100x100 pixels each, patch size of 50x50.

Noise Level	Total Energy Ratio: Reconstructed/Truth	Total Energy Ratio: Noisy/Truth	Maximum Energy Value Ratio: Reconstructed/Truth	Maximum Energy Value Ratio: Noisy/Truth
σ = 4	0.99	1.03	1.01	1.00
σ = 12	0.97	1.13	1.00	1.01
σ = 20	0.96	1.26	1.00	1.03
σ = 40	0.95	1.62	0.93	1.05

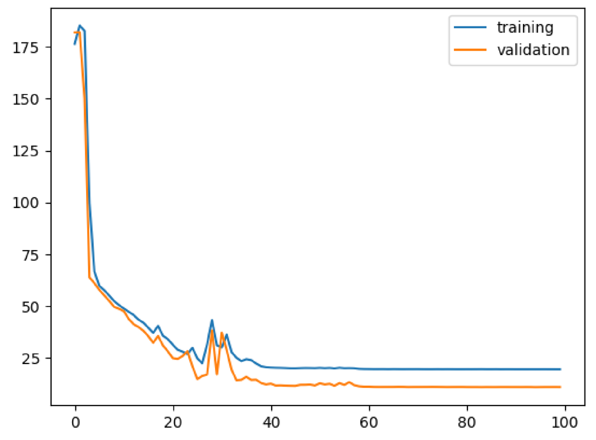
Results of a 50 epoch training of images of 50x50 pixels each, training on full image.

To improve the results achieved, we attempted to vary the parameters of the network and of training, including the input noise level of the training and validation data, the size and number of patches tested within the loss function, and the size of the convolutional kernels.

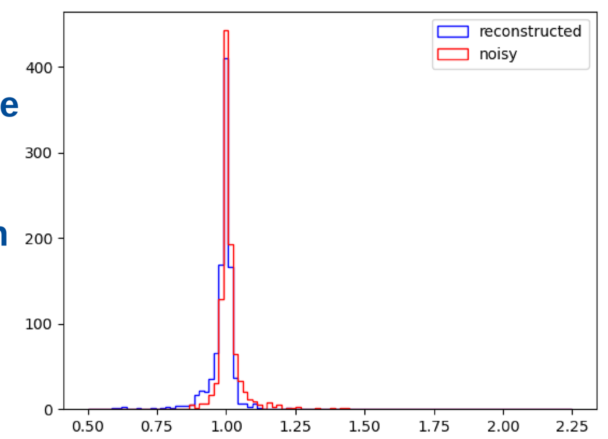
Optimizing Network Performance

Performance was considerably improved when the size of the input was reduced from 100x100 to 50x50 pixels. We hypothesize this allowed the network to learn how to preserve desired features of the images, rather than those created by noise.

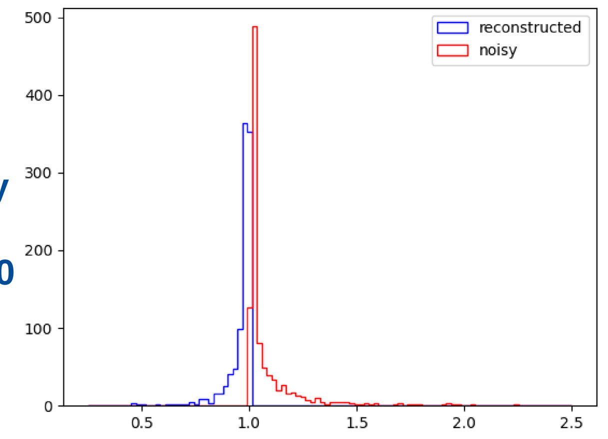
Loss during training as a function of epoch for a model trained for 50x50 pixel images. Loss values reflect the average per-pixel energy difference in the region of interest.

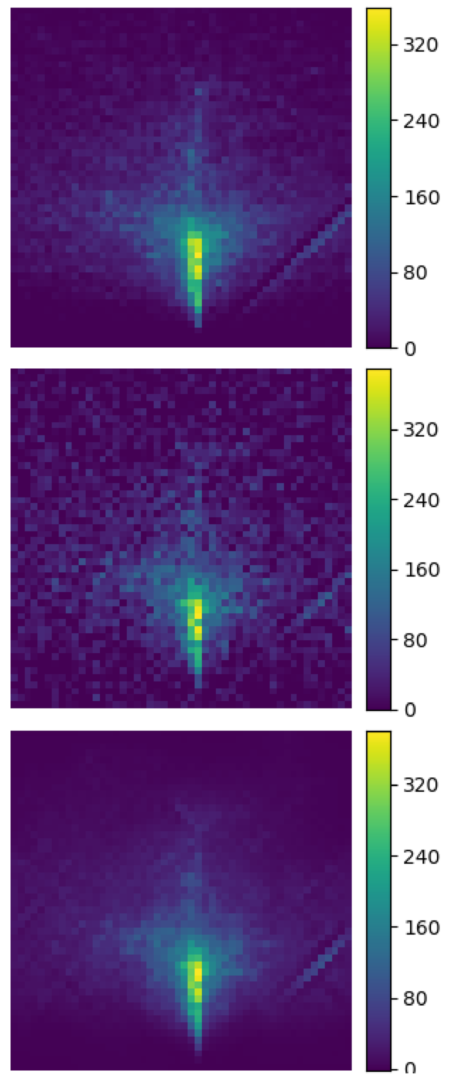


The ratio of the maximum energy value in each image in the reconstructed and ground truth images and the noisy and ground truth images for a network trained for 50 epochs on 50x50 pixel images.



The ratio of total energy present in the reconstructed vs ground truth images and the noisy and ground truth images for a network trained for 50 epochs on 50x50 pixel images.





Visualization of input (top), noisy (middle) and reconstructed (bottom) energy for 50x50 pixel images.

Future Work

- Try using a kernel-based prediction architectures in which the final network layer outputs a kernel of scalar weights to be applied to a noisy input area
- Use a 3-dimensional dataset which will more accurately reflect the detector for simulation purposes
- Analyze modified loss functions & preprocessing methods.
- Determine coarse settings to speed up Geant4 and generate noisy images instead of adding noise by hand.
- Add additional images channels to store additional information e.g. timing, PID, etc.

References

Steve Bako, Thijs Vogels, et al. "Kernel-predicting convolutional networks for denoising Monte Carlo renderings," ACM Trans. Graph. 36, 4, Article 97 (2017).

SimDenoising https://github.com/kpedro88/SimDenoising.git

SimDenoising_training https://github.com/lenafranklin/SimDenoising_training.git