Classifying Static X-ray Imager Diagnostic Using Convolutional Neural Networks

William Leach, James Henrikson

NIF, Lawrence Livermore National Laboratory

The Static X-ray Imager (SXI) diagnostic at the National Ignition Facility (NIF) produces a wide variety of image data, not all of which should be run through the automated signal processing algorithms. Currently all data is analyzed automatically which leads to some erroneous results. One approach to solve this issue is to use convolutional neural networks (CNNs) to filter out or flag the SXI images before sending them to the processing algorithms.

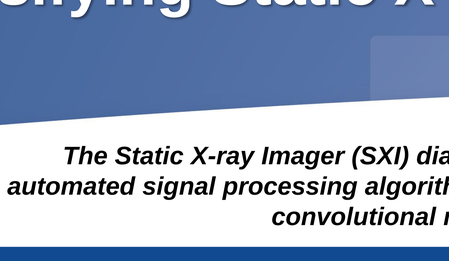
Introduction:

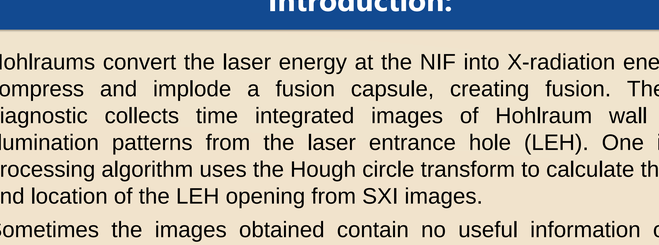
Hohlraums convert the laser energy at the NIF into X-radiation energy to compress and implode a fusion capsule, creating fusion. The SXI diagnostic collects time integrated images of Hohlraum wall X-ray illumination patterns from the laser entrance hole (LEH). One image processing algorithm uses the Hough circle transform to calculate the size and location of the LEH opening from SXI images.

Sometimes the images obtained contain no useful information or are noisy. Current approaches try to account for noise conditions on a case by case basis. However, the diagnostic data is abundant in its variety of examples. In addition the diagnostic data can, at the same time, be sparse (missing segments or pieces). CNNs have been shown to work well under these conditions and the strategy is to apply them for identifying useful data.

Hardware Setup and Preprocessing:

The dataset comes from the SXI-upper diagnostic:

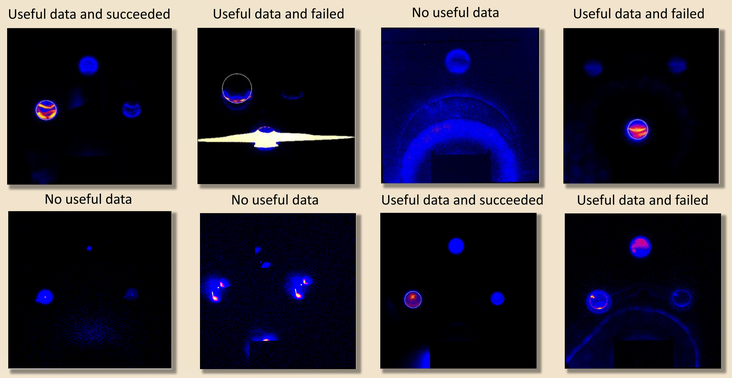




CCD Plate Image

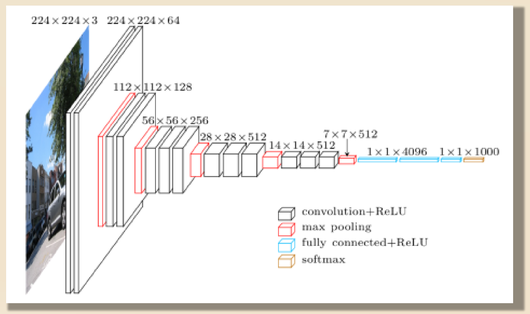
The images were collected on an X-ray image plate and digitally normalized to zero mean and unit variance. Keeping the image intensity histogram stationary stabilizes the inputs to the CNN and helps it converge faster.

Problem and Examples:



A white circle in the image indicates the estimated LEH circle (no circle shows the algorithm found nothing and didn't complete). The dataset contains about 1500 images and a NIF physicist was consulted to help determine the quality of the instances. The objective of this work is to determine if the image contains useful data and decide if the data is of high enough quality for the processing algorithm to be successful.

VGG-16 CNN Model:

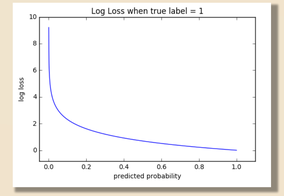


Transfer learning was used by fine tuning the VGG-16 network pretrained on the ImageNet database [3]. The first layers are already well pretrained from ImageNet to find basic building blocks that are in all images, for example lines and polygons. Only the last few layers are fine tuned to be receptive to SXI data. The network outputs three classes: useful data, useful with high quality, and not useful.

Training:

Categorical cross-entropy loss gives high penalty to predictions that are confident and wrong. In binary classification cross-entropy is calculated as [4]:

$$-(y\log(p)+(1-y)\log(1-p))$$



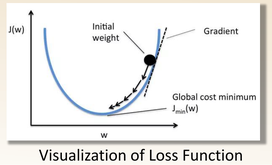
Where p is the probability of the observation coming from class 1 and y is the true label. For the 3 class output the loss can be calculated for each class for each observation and then summed [5]:

$$J = -\sum_{c=1}^{M} y_{o,c}\log(p_{o,c})$$

- M - 3 (classes)
- O - the observation
- C – class

The network is trained in Keras by updating its weights in the direction that decreases the loss function (the error):

$$\Delta w_j = -\eta\frac{\partial J}{\partial w_j}, \qquad w_j := w + \Delta w_j$$



Visualization of Loss Function

Preliminary Results and Future Work:

Overall Accuracy	Class 0: No Useful Data	Class 1: Useful data but Algorithm Failed	Class 2: Useful Data and Succeeded
~80 %	~91%	~55 %	~72%

Training on current SXI data has just begun and therefore results shown here are from initial network and hyperparameter setup. Class 0 & 2 had good performance whereas Class 1 had a sparse training set (few examples) which could explain the poor classification.
The next steps to improve performance are:

1. Augment the training data: images can be flipped, rotated, translated to increase the number of examples and image set variance
2. Test out different CNN architectures such as Inception or ResNet
3. Balance the classes by augmenting class 1 more aggressively

Acknowledgements and References:

I want to thank Robert Hatarik, Judy Liebman, Nathan Palmer, and Matt Reever for all of their assistance on this project.

1. https://www.researchgate.net/figure/VGG16-architecture-16_fig2_321829624
2. https://aip.scitation.org/doi/10.1063/1.3491316
3. https://keras.io/applications/#vgg16
4. http://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html
5. Bottou, Léon (1998). "Online Algorithms and Stochastic Approximations". Online Learning and Neural Networks. Cambridge University Press. ISBN 978-0-521-65263-6

LLNL-POST-755711

Lawrence Livermore National Laboratory

National Ignition Facility • Lawrence Livermore National Laboratory • Operated by the US Department of Energy

This work performed under the auspices of the U.S. Department of Energy at the Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

NIF&PS