Machine Learning based image segmentation for region of interest selection in PDV data

Thomas Matthews

Strategic Deterrence
Lawrence Livermore National Laboratory

Photonic Doppler Velocimetry Workshop 2024

October 17th, 2024

Prepared by LLNL under Contract DE-AC52-07NA27344.

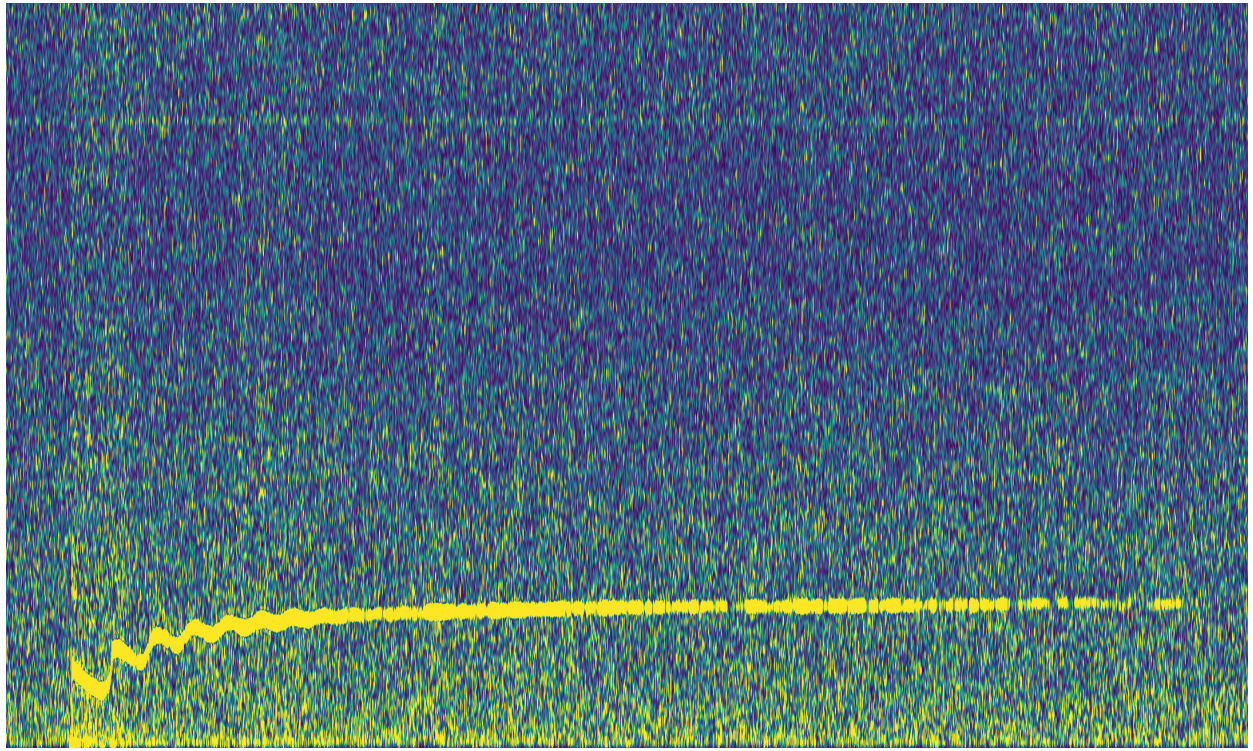
LLNL-PRES-869967

Outline

1. Introduction to Image segmentation
2. Machine Learning tools
3. U-Net Architecture
4. Creating Training Data
5. Results
6. Incorporating models into other tools

Variations in PDV data make a “catch all” algorithm difficult to create

- Traditional approaches of using statistics to extract velocity data from waveforms have mixed results.
- Often algorithms will work well for data generated by a particular type of experiment and perform poorly on others.
- For this reason, most tools require a human to select a region of interest(ROI) in the spectrogram.



Machine Learning can offer a robust region of interest extraction

- By reducing the problem to region of interest selection, the human intensive component of PDV analysis can be automated.
- The remaining steps of PDV analysis are already done using a variety of methods.
- By replacing just this step in the process, analysis tool developers can offer flexibility to the end users and allow them to the choose how the velocity value is computed.

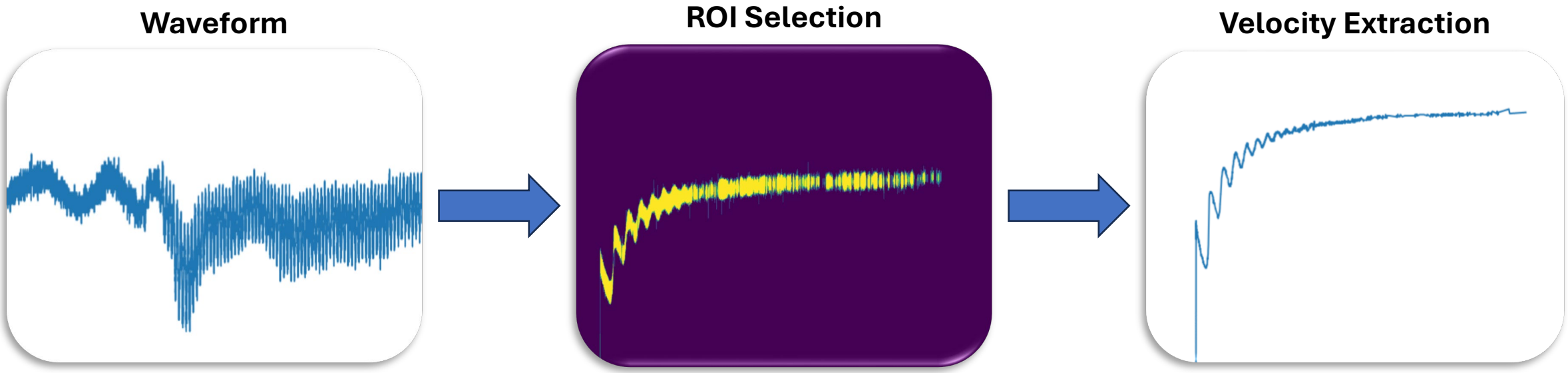
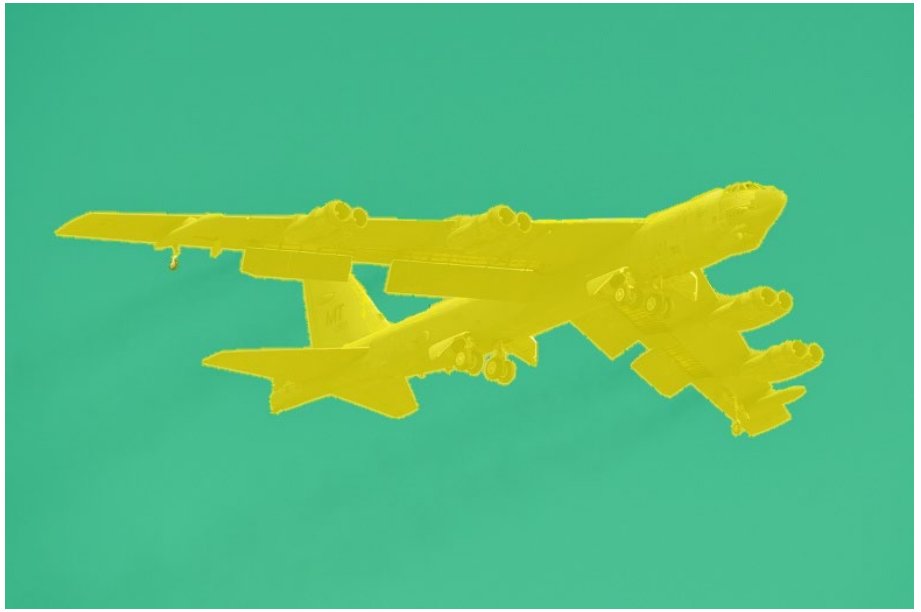


Image Segmentation

- Image segmentation is the process of assigning each of the pixels in an image a label.
- The labels represent different features of the image.
- The labels can be used as a mask for where in the image the signal is located.



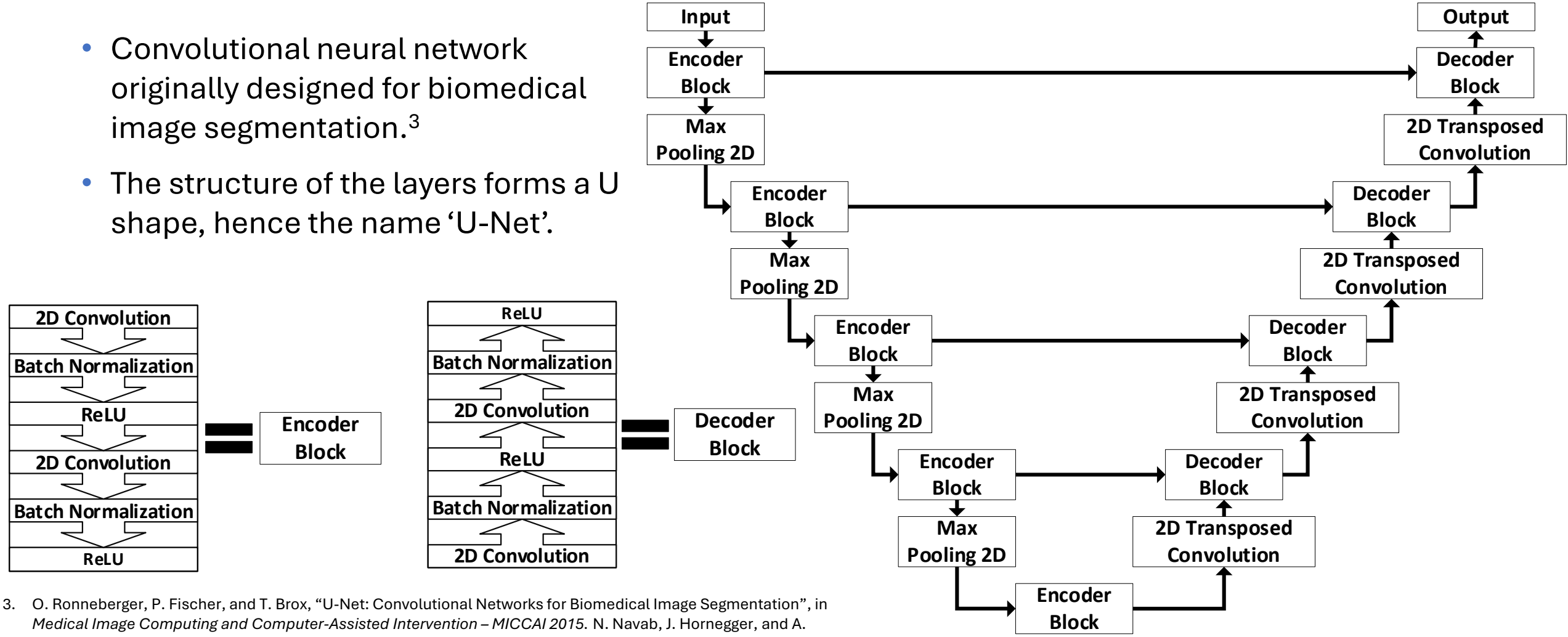
Machine Learning Tools used

- Many machine learning tools exist today.
  - Keras, Caffe, Amazon SageMaker, Apache MXNet, Scikit-learn, Azure ML Studio, MATLAB, etc.
- Python is very popular for machine learning.
- TensorFlow
  - Developed by Google
  - Open Source
  - Released Nov. 2015
  - Able to execute low-level operations across different platforms, making it good for developing on one platform and deploying to another.[1]
  - Better suited to large projects
- PyTorch
  - Originally developed by Meta, currently governed by the PyTorch Foundation (part of the Linux Foundation)
  - Open Source
  - Released in Sep. 2016
  - Trains models using reverse-mode automatic differentiation, which records completed operations and replays them backwards to compute gradients.[2]
  - Better suited to rapid prototyping.

1. "TensorFlow – What Is It and Why Does It Matter?". NVIDIA.com https://www.nvidia.com/en-us/glossary/tensorflow/ (accessed September 13, 2024)
2. "What is PyTorch? | Data Science | NVIDIA Glossary". NVIDIA.com. https://www.nvidia.com/en-us/glossary/pytorch/ (accessed September 13, 2024)

U-Net Architecture

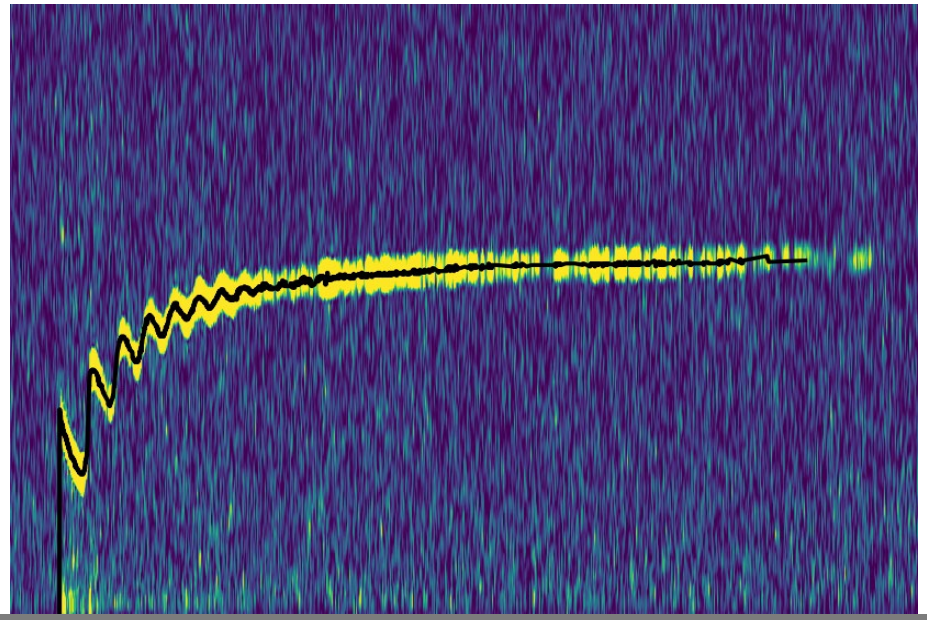
- Convolutional neural network originally designed for biomedical image segmentation.[3]
- The structure of the layers forms a U shape, hence the name ‘U-Net’.

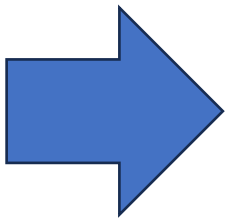


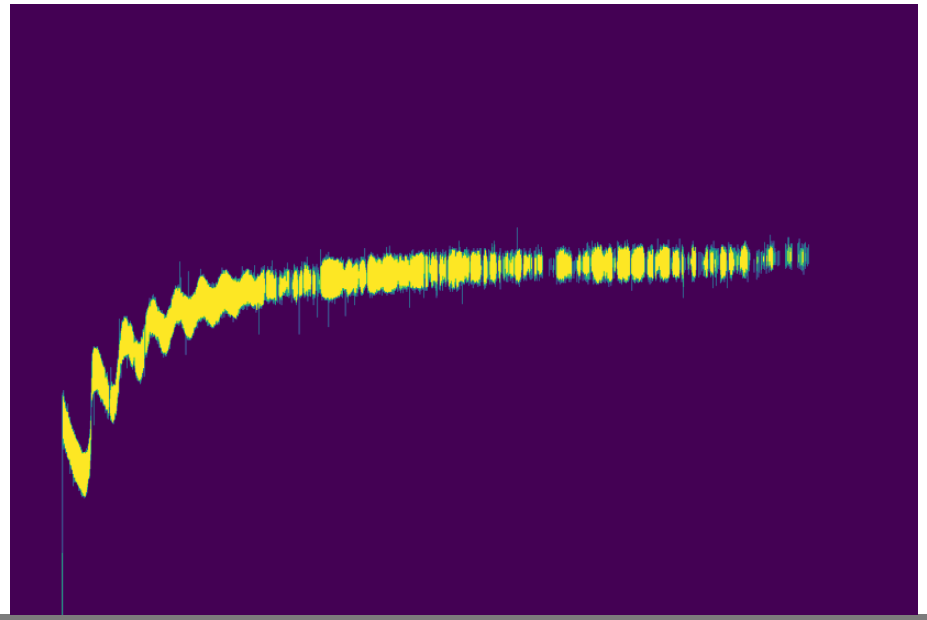
3. O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. N. Navab, J. Hornegger, and A. Frangi, Ed., Munich, Germany, 2015, pp.234-241, doi: 10.1007/978-3-319-24574-4_28

Where can I get a large dataset for training a model?

- An enormous amount of PDV data already exists.
- It is however not in a useful form for training an image segmentation model.
- Using the existing velocity traces and spectrograms, a mask of the ground truth can be created.
- Multiple pieces of training data can be generated from a single waveform by varying the spectrogram parameters and cropping the spectrogram in different locations.





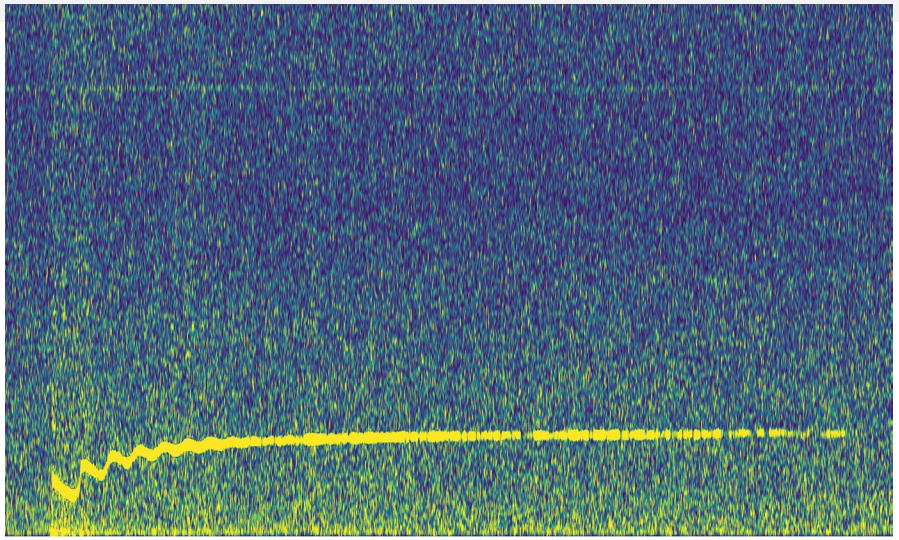


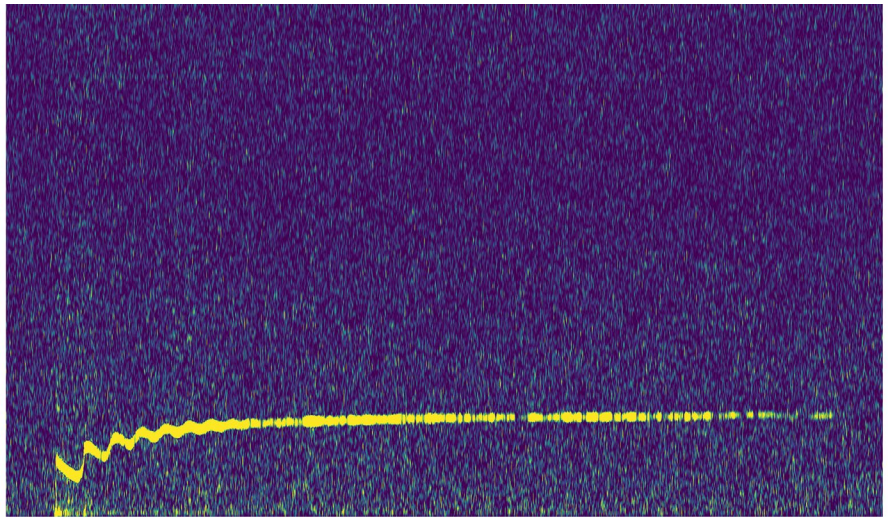
You can’t simply throw data at a model

- Data must be preprocessed to be input to the model.
  1. Start with spectrogram
  2. Scale the spectrogram
  3. Slice the spectrogram into pieces
  4. Turn array into tensor object
  5. Make prediction
  6. Convert prediction back into original spectrogram shape
- The ground truth values must be similarly converted into the same format as the model’s output.
- The data used to train the model has huge impact on the success of the model.
  - The majority of adjustments were made to the training data, not other aspects (model, optimizer, loss function, etc.).
  - Much more of the time was spend working making the data than anything else.

To make the intensity values uniform across different data the spectrogram values are turned into an image

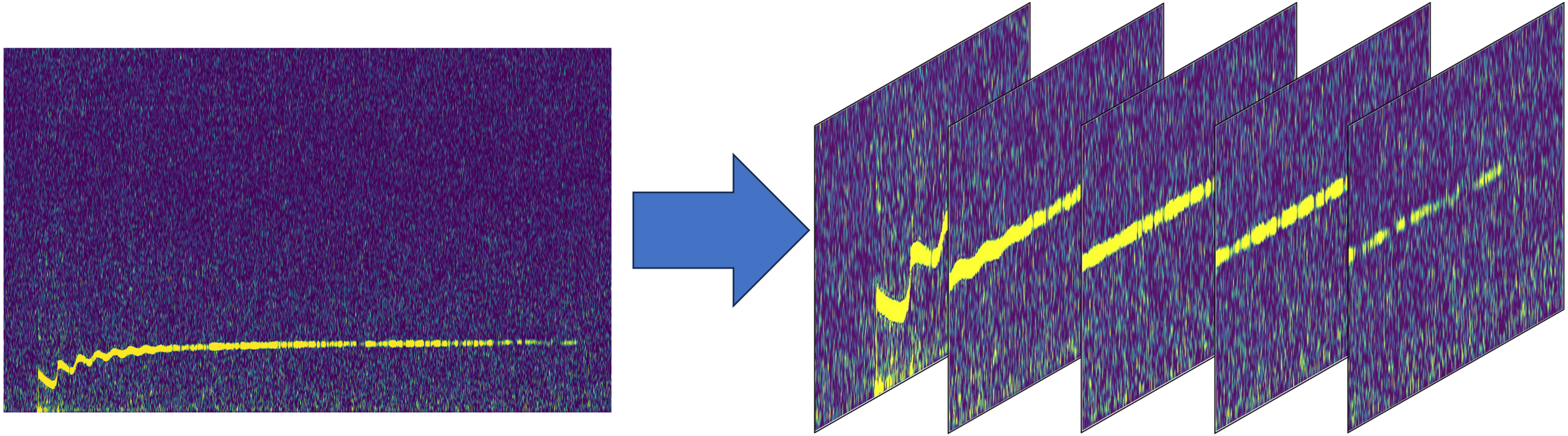
- This idea was created by Abel Diaz and was intended to be presented at the planned 2020 PDV workshop.
- The spectrogram is converted to an image by:
    1. The median of each row and column is subtracted from each row and column and values below zero are set to zero.
    2. The adjusted values are then scaled to integers between 0 and 255, with values above the 99% quantile being 255 and values below the 30% quantile being zero.





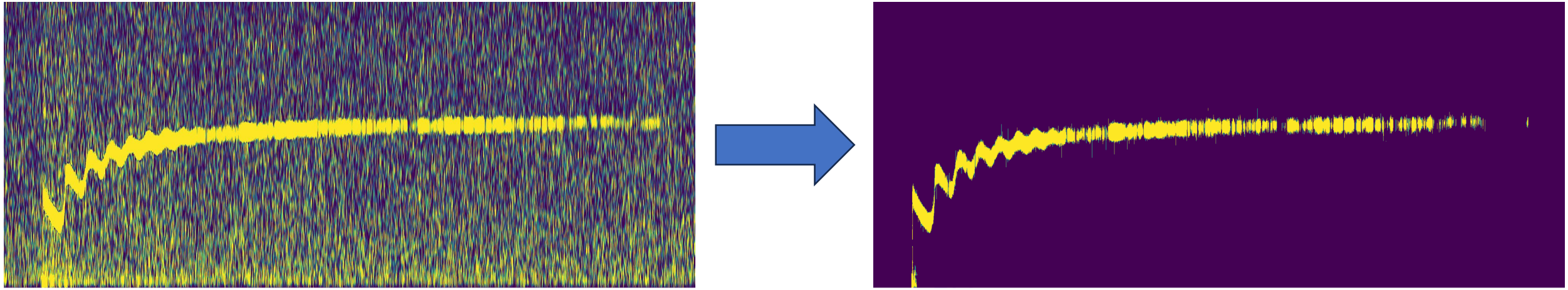
The model requires a fixed shape to the data

- The shape of the input and output is fixed when designing the model structure.
- By breaking the images into pieces of the correct size, data of any shape can be put through the model.



Results

- This model was trained for 4 iterations on a dataset of over 32,000 images.
- The model achieved an accuracy of 98.33%.
  - The error comes from the model selecting some of the other features, such as the probe breaking as signal.
- The model doesn't perform well on ejecta yet, more data is needed for training.

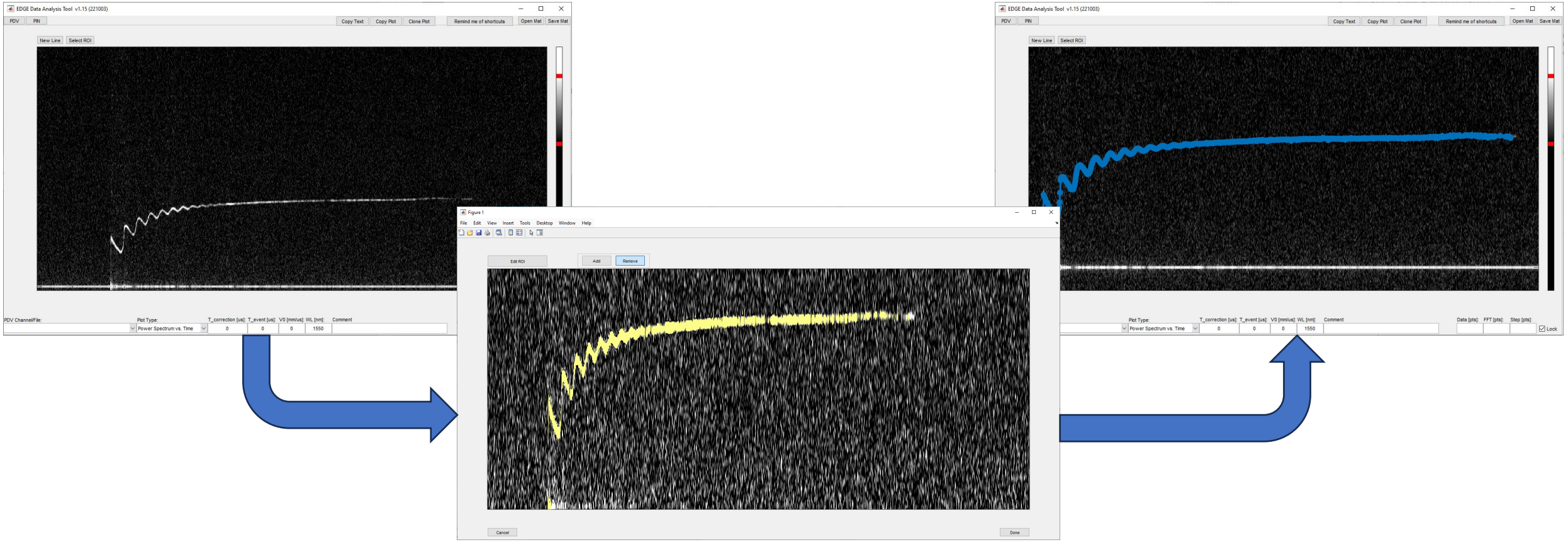


For smoother integration, models should be incorporated into existing tools.

- Development of models should focus on building them to have inputs and outputs that work with existing tools.
  - This may require tailoring the training data to the form used in the data analysis tool.
- There numerous options for integrating models with other tools:
  - The Open Neural Network Exchange (ONNX) provides standard format for machine learning models allowing them to used in different frameworks.
    - Models can be developed and trained using one framework and then converted to ONNX to be used in another framework.
  - Models can be used in their native language. Many programming languages offer a means to run scripts from other languages.
    - For example: MATLAB and LabVIEW can both call a python script, pass arguments to it, and receive outputs.

Example: The image segmentation model was incorporated into one of LLNL's existing data tools, EDGE

- Integrating the model into EDGE (which was written in MATLAB) took adjusting the data trained on, tailoring python scripts to accept MATLAB outputs, and bundling python into the EDGE distribution.



Summary

- Image segmentation can be utilized to create generalized region of interest extraction tools.
- Training a model requires a large volume of data.
  - Legacy data can be leveraged to create training data.
- How the data is preprocessed makes a big difference in how the model performs.
- For maximum usability, models should be incorporated in tools already in use in the field.
  - There are many tools available for machine learning, choosing the right one can help with integration.

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.