Jennings

P4: Celia Motion Analysis

Raun, Ailing, Prajay, Chris

Our Approach

- U-Net, negative dice loss
- Keras as the neural net framework
- Ad-hoc model development
- Later packaged up for reproducibility

Keras vs PyTorch

Keras

- Build networks imperatively
- Single top-level abstraction (model)
- Serialization is best
- Static backprop graph
  - Fixed input size

PyTorch

- Build networks as a class
- Three top-level abstractions (module, optimizer, loss fn)
- Serialization is good
- Dynamic backprop graph
  - Arbitrary spatial dimensions

Crop and Stich vs Padding

- Image inputs are different sizes; Keras wants fixed size.
- Option 1: Crop to common size, segment, stich results together
  - Very annoying to implement
  - Reduces the field of view
    - Only matters when the network is deep
- Option 2: Pad to maximum size, segment, crop results
  - Much easier to implement
  - Increases memory usage
- PyTorch doesn’t care; conv inputs can have arbitrary spatial dims.

Biggest problem

BURNOUT