DTU Compute
Department of Applied Mathematics and Computer Science

Brown Cricket Segmentation

Depei Wang[1,*], Kazi Ejajul[2]

[1] Technical University of Denmark, Department of Electrical and Photonics Engineering, Kgs. Lyngby, Denmark
[*] Corresponding author, Tel.: +45 50398475,, Email: s232909@dtu.dk

[2] Technical University of Denmark, Department of Compute, Kgs. Lyngby, Denmark

1 GOAL

This project aims to train and test a deep learning model to segment bugs from micro-CT scanned 3d images. We're concentrating on the Brown Cricket bug among 12 types bugs in Bug-NIST2024 dataset. Example image displayed in *ParaView* is shown in figure 1.

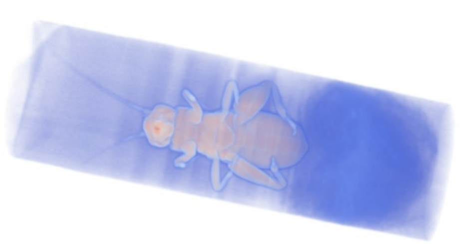

Figure 1: An example volume inspected in Paraview

Our work contributes to improved techniques for analyzing volumetric data and leverages deep learning methodologies to accurately identify and segment bugs within complex datasets.

2 METHODOLOGY

The data for this project consists of approximately 700 volumes of Brown Cricket bugs in 256x128x128 resolution, sourced from the DTU 3D Imaging Center repository. The volumes are in .tif format and require pre-processing, including labeling, reorientation and binarization. Labeling is done through a combination of automatically threshold segmentation and manual corrections. Example label slices are shown in Figure 2.

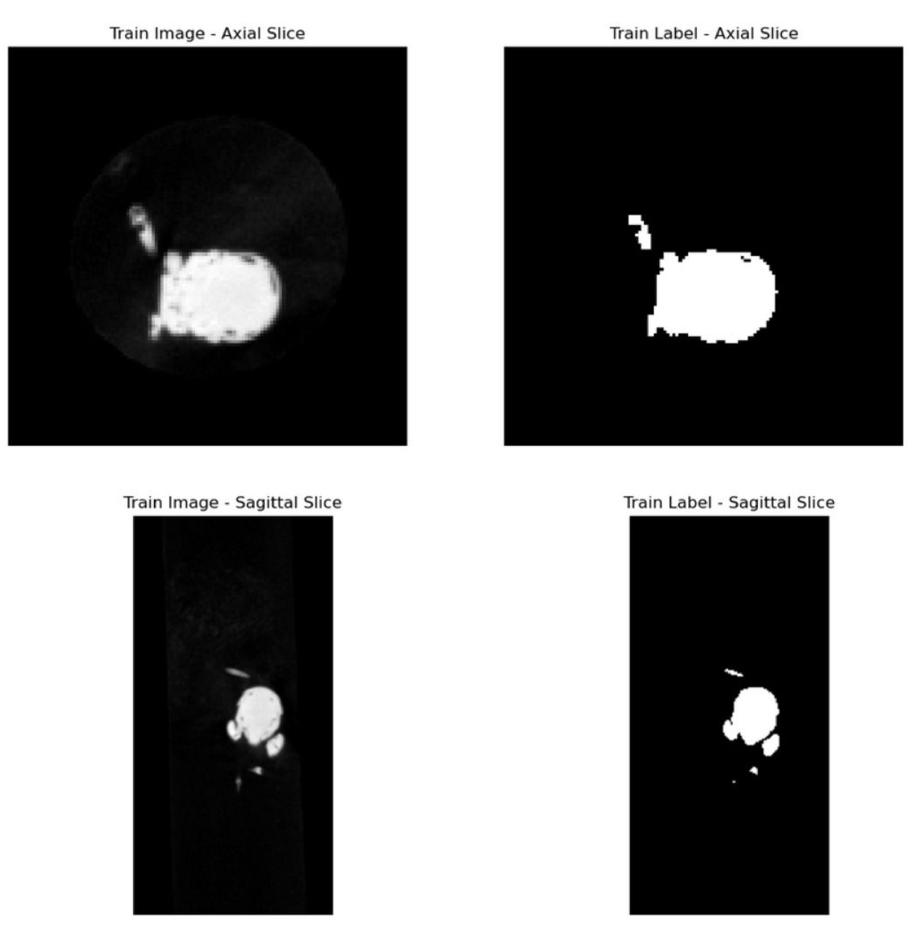

Figure 2: Example labels by automatically threshold segmentation, first row two images are axial slices from micro-CT scan and label, second row two images are sagittal slices from micro-CT scan and label.

We utilized 3D U-Net model as it was developed for biomedical image segmentation. The network training pipeline starts with data preparation and pre-processing, ensuring uniformity in channels and dimensions. The pipeline is shown in Figure 3.

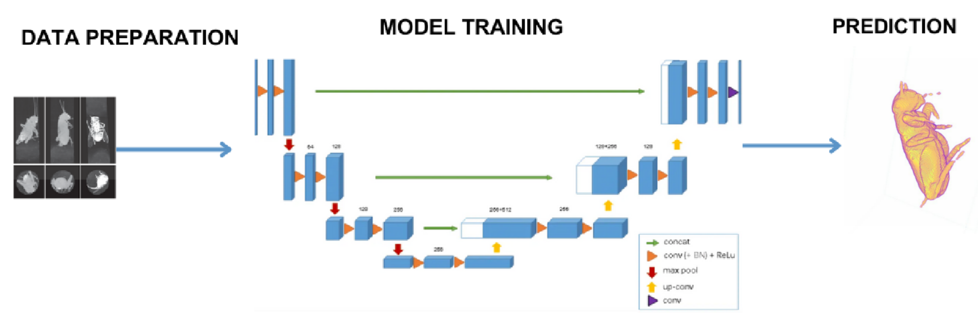

Figure 3: Training network pipeline

It uses techniques like intensity scaling and random rotations to create a consistent dataset for training and evaluation. The U-Net model is initialized with 3 spatial dimensions, one input channel, two output channels and 0.2 dropout rates. The training loop involves forward/backward propagation, Adam optimizer updates, and Dice loss computation over multiple epochs.

3 RESULTS

To let our model first running the pipeline, we trained it on the automatically labeled dataset by threshold method. Both images and labels were divided into train, validation and test sub-folders. Approximately 500 images and labels were trained into the network and around 100 images and labels were used for validation. Baseline model was trained in 25 epochs, and saved best model at 20^{th} epoch, training and validation curves is shown in Figure 4. The baseline model-threshold segmented data trained model-output example was shown in figure 5.

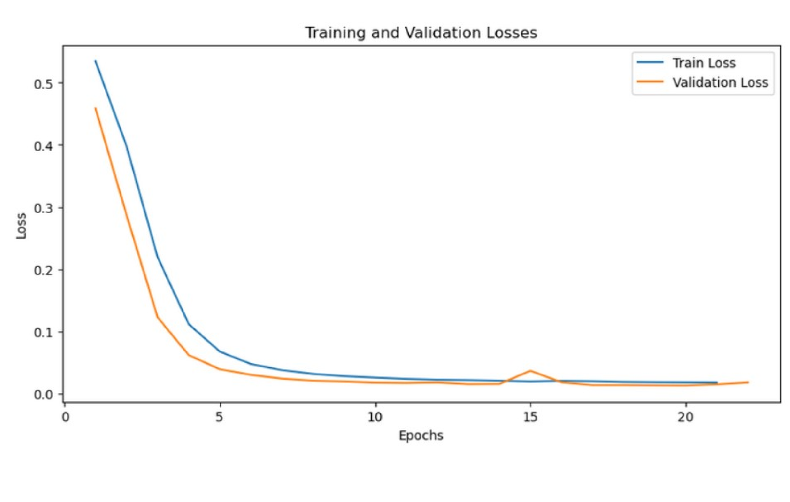

Figure 4: Training loss and validation loss calculated using Diceloss function.

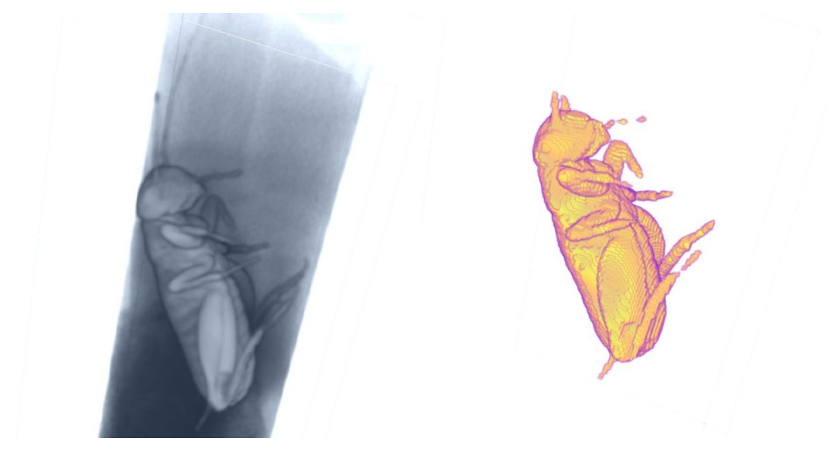

Figure 5: Test result from baseline model dislayed in Tomviz, left bug is from micro-CT scanned volume, right bug was segmented from this volume.

Then, we developed our model by training previous saved best model on the manually labeled dataset using ITK-snap software segment 17 selected images contain required details. We aim to segment small features e.g. antennas, tip of legs and breads. But manual data seems degraded the capability of the model, so we trained second model also based on previous checkpoint and on the same threshold label but with more epochs, result is shown in Figure 6, red circles indicates features not recognized.

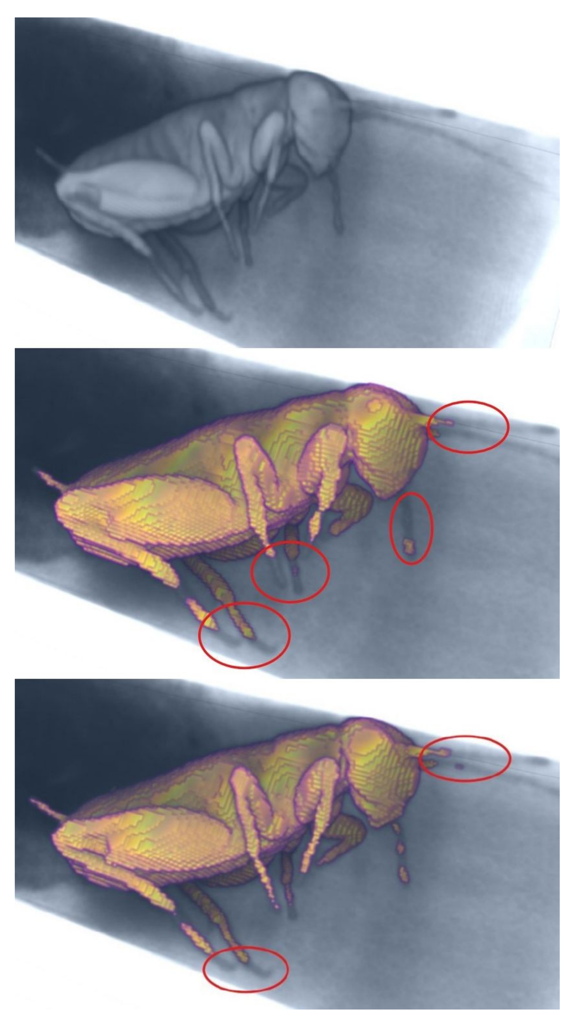

Figure 6: Two segmented bug labels based on same one micro-CT scanned image, bug in middle was segmented by ITK-snap label data trained model, bug at the bottom was segmented by threshold label trained model.

4 DISCUSSION

Our results demonstrate that a 3D U-Net model can effectively segment 3D volumetric data, but needs more well labeled data.