**SPROJ\_EECE\_7398\_MXX\_V04\_Milestone\_4**

Welcome back to our fidip tutorial series. In the previous lesson, we downloaded the pre-trained models and prepared our data set. Now we're going to test those pre-trained models to make sure everything is working correctly, and then start the training process. Before we start training, let's test the pre-trained model to ensure everything is set up correctly.

Now, in our main directory, let's test the pre-trained model. This command runs the evaluation script and our validation data set using the pre-trained fidip model. They use GTB box as in use ground truth bounding box flag tells the model to use ground truth bounding boxes, rather than the predicted ones, which help isolate pose estimation performance from detection performance.

Now, let's examine the output to understand the model's performance on the validation set. You should see metrics like average precision, average recall, for different IOU thresholds. Now we're ready to start the training process. Let's look at the configuration file to understand what parameters we're using. This YML file contains all hyper-parameters for our training including learning rate, batch size, and model architecture details. It's important to understand these settings when working with small data sets, as they can significantly impact performance.

Let's go through understanding the training process in small data set regimes. While our model is training, let's discuss some key considerations for training models with small data sets. Overfitting risks. Small data sets make models prone to overfitting. Watch validation metrics closely for signs of overfitting. Training loss decreases while validation loss increases. Learning rate sensitivity. Models trained on small data can be more sensitive to learning rate. Starting with a smaller learning rate and using careful scheduling can help.

Regularization importance. Techniques like weight decay, dropout, and early stopping become even more important with small data sets. Transfer learning benefits. Starting with pre-trained weights, as fidip does, can significantly improve performance when data is limited. The model has already learned useful features from larger data sets. Domain adaptation efficient strategies. The core innovation in fidip is its domain adaptation approach, which helps transfer knowledge from data-rich audio post domain to data-scarce infant post domain.

Batch size considerations. While small data sets, smaller batch sizes can sometimes work better as they introduce more noise into gradient updates, which can help generalization monitoring training progress. Keep an eye on both training and validation metrics. With small data, validation performance can be noisy, so don't make decision based on a single epoch.

In this lesson, we've tested our pre-trained model and started the training process. The model will continue training for several hours depending on your hardware. In the next lesson, we'll monitor training progress, discuss strategies for ensuring reproducibility in machine learning experiment. See you there.