# Project: Instrument Classification – Initial Steps

## How we started

- We decided to start with a smaller subset of the data to first verify that the entire pipeline was functional. Instead of the complete training set (289k files), we took the test split (~4k samples). The goal was to set up the entire pipeline—from data to the first model—and then scale up later.

## What we did

### 1. Data loading

- We downloaded and unpacked the NSynth dataset test split.
- Metadata was loaded from examples.json, and a DataFrame was created with paths to the WAV files and their corresponding labels.
- We balanced the mini-subset so that each instrument family had approximately the same number of examples.

### 2. Preprocessing

- WAV files were converted into log-mel spectrograms using the librosa library.
- We normalized the data to make it easier for the CNN to learn.
- Dimensions: 64 mel filters x time frames.

### 3. Dataset and DataLoader

- We created a PyTorch Dataset that returns a (spectrogram, label) pair.
- Data was split into training and validation sets (80/20).

### 4. Model

- We implemented a small CNN with several convolutional layers and one fully connected layer
- The output is a softmax over 10 instrument families.

### 5. Training

- We used the Adam optimizer and CrossEntropyLoss.
- We trained for 5 epochs on the mini-subset.

- We recorded the loss and accuracy per epoch.

## 6. Results

- Validation accuracy: approximately 67%.
- Some classes performed quite well (e.g., brass recall = 1.0), while keyboard and reed caused more problems.
- The confusion matrix clearly indicated where the model made errors.

## Next steps

We plan to:

- Try a larger dataset (valid/train),
- Train longer and on a stronger architecture