Deep Learning Project: Instrument recognition in music

Presented by:

Ofek Dahan & Ori Hefetz

Agenda

Our goals

Theory

Settled on model

Results

Our Goals

In this project, we aimed to classify instruments present in a music sample.

- Initially, we attempted to classify multiple instruments per datapoint

(Using a one hot vector, a result of "1" if the instrument is present, "0"
if it's not)

- We wanted to reach high accuracy using digital signal processing methods, and a multi-branch CNN.
- Later, we opted to classifying the most prominent instrument, instead of multiple ones.

Deep learning tools

We focused our energies on the pre-processing part of the deep learning methodology.
We tried using many methods to provide the most discernable images for the CNN:

1. Different signal transformations.
2. Normalization, data mixing, and random noise.
3. Additive samples, portraying multiple instruments.
4. Transfer learning.

The IRMAS dataset

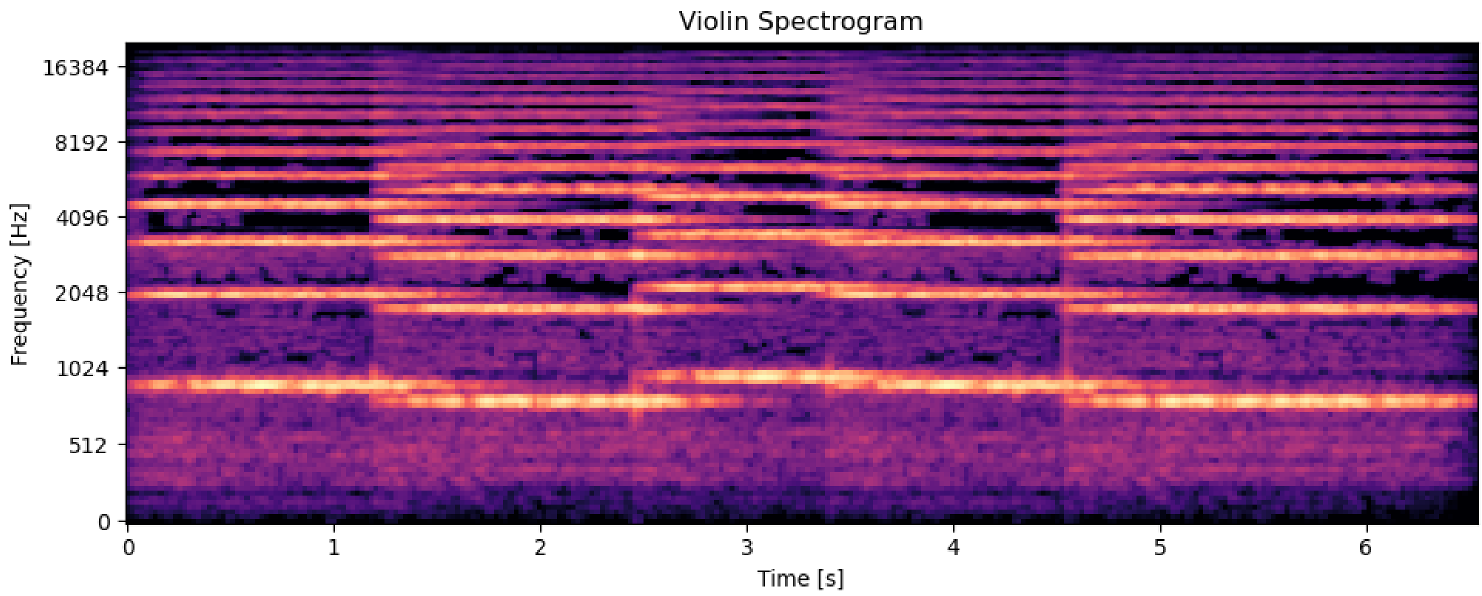
- Training set: Multi instrument, single label. 6700 samples
- Test set: Multi instrument, multi label. 2900 samples.

Theory: The basics

- Our output: 11x1 binary vector. '1' for each instrument identified.
- Loss: Binary cross entropy
- Optimizer: Adam. We chose batch size 16 to include a large diversity of instruments in each run, and keep the model light

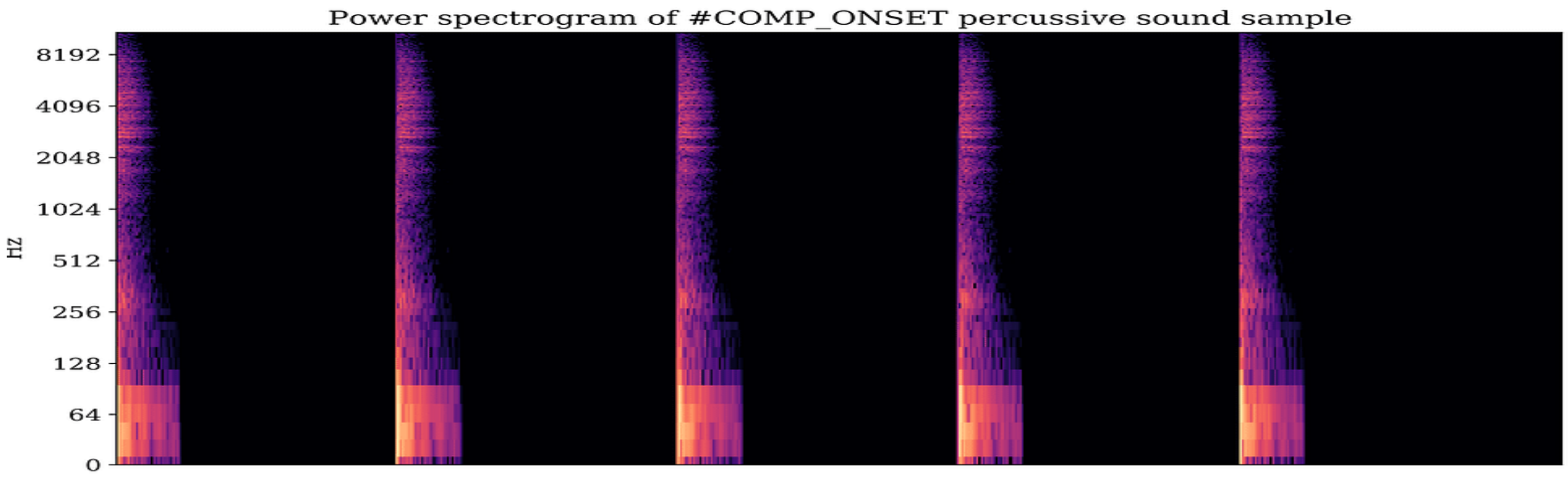
Theory: The STFT- Short time Fourier transform

1. Choose time window
2. Move time window over signal
3. Calculate Fourier transform for each window position.
4. Result: timeline of frequencies in the signal

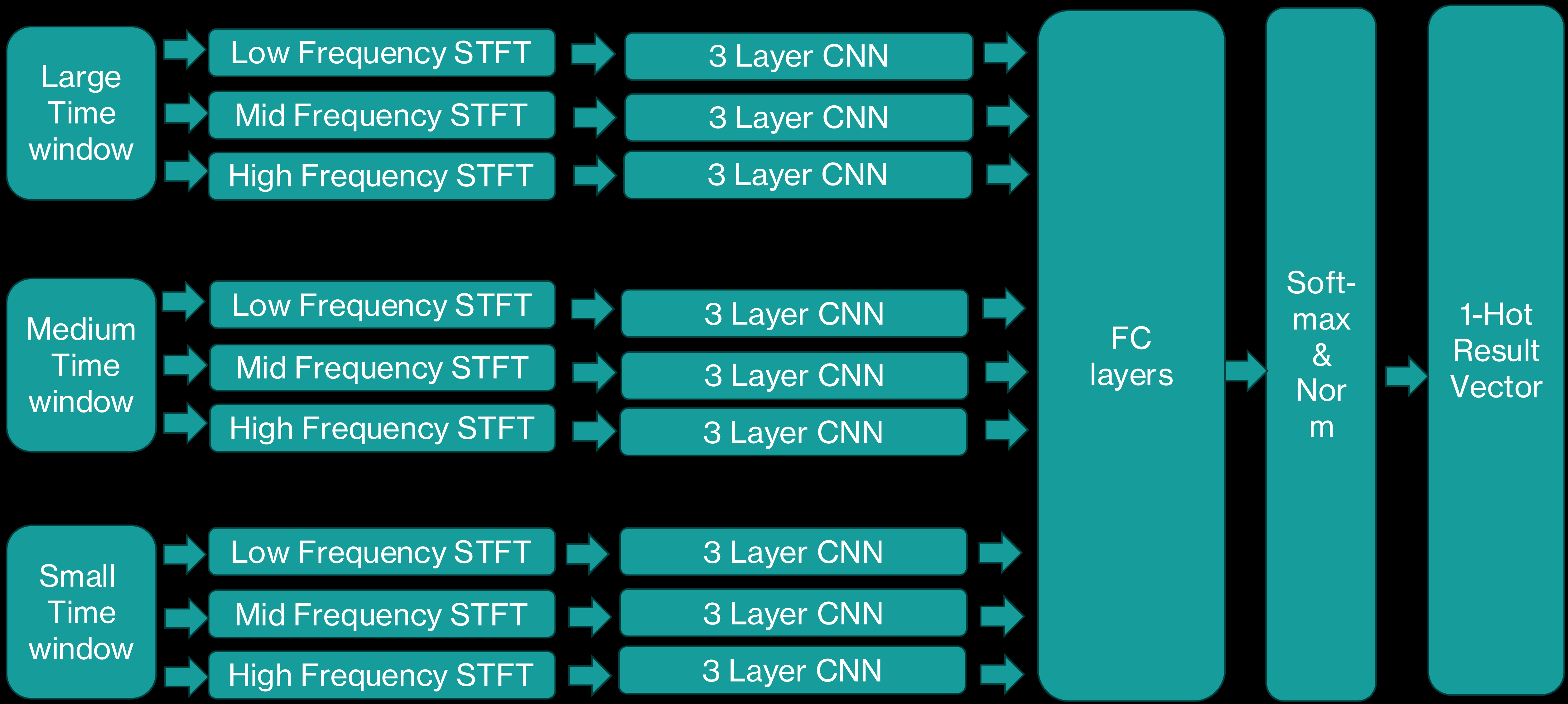


On the right, violin STFT (up)

Vs drum STFT (down)



Model 1.0: the 9-spectrogram model

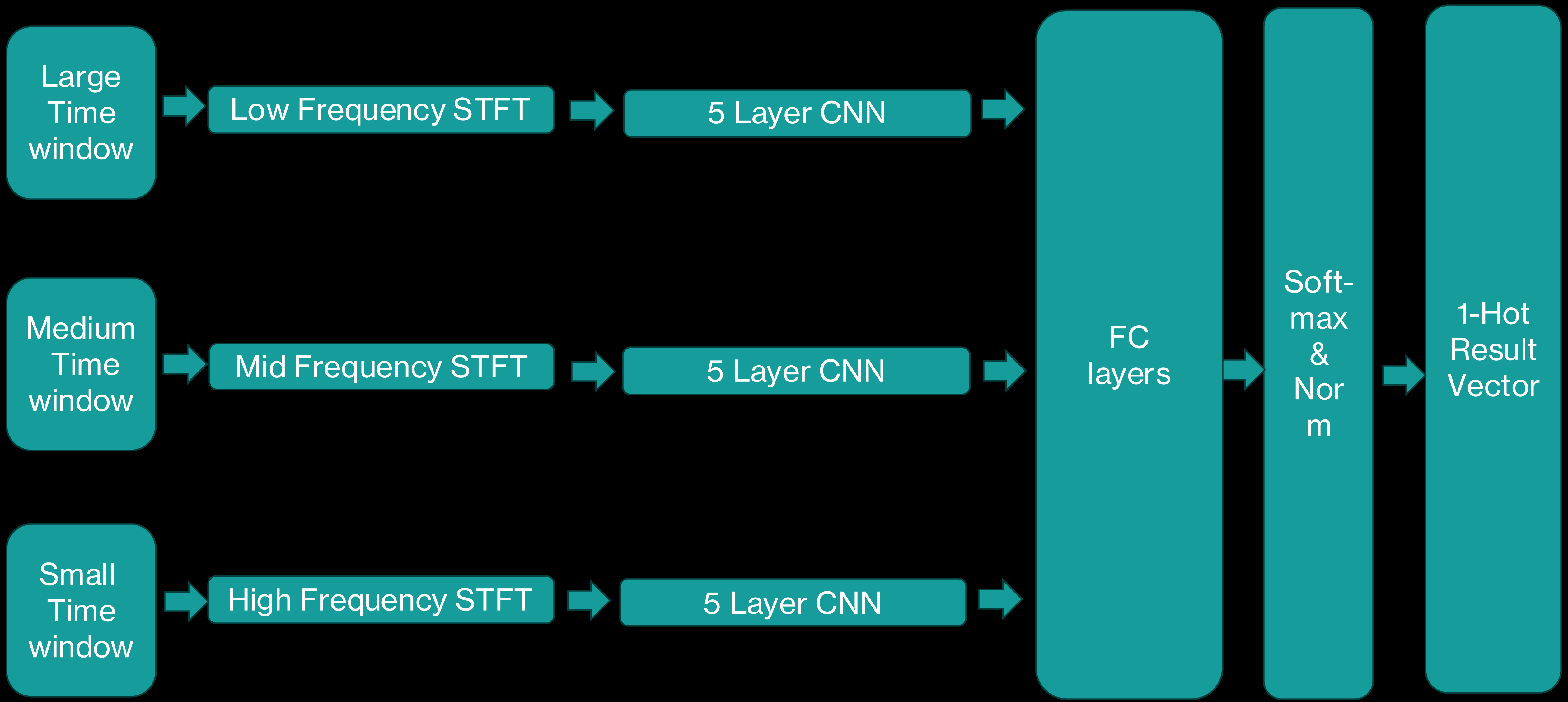


Lessons from model 1.X

- Each CNN must be wide and deep enough to reach results independently.
- Too many CNNs- data too sparse over a large network
- Repetitive data.
- The model becomes too large before it is useful.

- Best accuracy acquired: 15%

Model 2.0: the 3-spectrogram tight model



Model 2.3: the 3-spectrogram tight model, with transfer learning

```
Model Summary:
==========================================================================================================
Layer (type:depth-idx)                                     Output Shape              Param #
==========================================================================================================
ModelWrapper                                               [1, 11]                   --
├─MultiSTFTCNN_WithPANNs: 1-1                              [1, 11]                   --
│    └─ModuleList: 2-1                                     --                        --
│    │    └─PANNsFeatureExtractor: 3-1                     [1, 512]                  1,550,784
│    │    └─PANNsFeatureExtractor: 3-2                     [1, 512]                  1,550,784
│    │    └─PANNsFeatureExtractor: 3-3                     [1, 512]                  1,550,784
│    └─Sequential: 2-2                                     [1, 512]                  --
│    │    └─Linear: 3-4                                    [1, 1536]                 2,360,832
│    │    └─BatchNorm1d: 3-5                               [1, 1536]                 3,072
│    │    └─ReLU: 3-6                                      [1, 1536]                 --
│    │    └─Dropout: 3-7                                   [1, 1536]                 --
│    │    └─Linear: 3-8                                    [1, 768]                  1,180,416
│    │    └─BatchNorm1d: 3-9                               [1, 768]                  1,536
│    │    └─ReLU: 3-10                                     [1, 768]                  --
│    │    └─Dropout: 3-11                                  [1, 768]                  --
│    │    └─Linear: 3-12                                   [1, 512]                  393,728
│    │    └─BatchNorm1d: 3-13                              [1, 512]                  1,024
│    │    └─ReLU: 3-14                                     [1, 512]                  --
│    └─Linear: 2-3                                         [1, 11]                   5,643
==========================================================================================================
Total params: 8,598,603
Trainable params: 8,598,603
Non-trainable params: 0
Total mult-adds (Units.GIGABYTES): 2.12
==========================================================================================================
Input size (MB): 0.15
Forward/backward pass size (MB): 73.58
Params size (MB): 34.39
Estimated Total Size (MB): 108.12
==========================================================================================================
```

Model 2.3: the 3-spectrogram tight model, with transfer learning

```
Epoch 3: 100%|██████████| 246/246 [01:15<00:00,  3.27it/s, v_num=0, train/loss=1.510, train/Accuracy=0.487, val/loss=1.530, val/Accuracy=0.520]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation:   0%|          | 0/62 [00:00<?, ?it/s]
Validation DataLoader 0:   0%|          | 0/62 [00:00<?, ?it/s]
Validation DataLoader 0:  32%|███       | 20/62 [00:05<00:11,  3.56it/s]
Validation DataLoader 0:  65%|██████    | 40/62 [00:10<00:05,  3.67it/s]
Validation DataLoader 0:  97%|█████████▋| 60/62 [00:16<00:00,  3.69it/s]
Validation DataLoader 0: 100%|██████████| 62/62 [00:16<00:00,  3.72it/s]
Epoch 4:   0%|          | 0/246 [00:00<?, ?it/s, v_num=0, train/loss=1.510, train/Accuracy=0.487, val/loss=1.910, val/Accuracy=0.420]🔓 PANNs backbone layers unfrozen for fine-tuning

================================================================================
EPOCH 4: UNFREEZING BACKBONE FOR FINE-TUNING
================================================================================

Epoch 4: 100%|██████████| 246/246 [03:15<00:00,  1.26it/s, v_num=0, train/loss=1.240, train/Accuracy=0.692, val/loss=1.910, val/Accuracy=0.420]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation:   0%|          | 0/62 [00:00<?, ?it/s]
Validation DataLoader 0:   0%|          | 0/62 [00:00<?, ?it/s]
Validation DataLoader 0:  32%|███       | 20/62 [00:05<00:11,  3.58it/s]
Validation DataLoader 0:  65%|██████    | 40/62 [00:10<00:05,  3.69it/s]
Validation DataLoader 0:  97%|█████████▋| 60/62 [00:15<00:00,  3.76it/s]
Validation DataLoader 0: 100%|██████████| 62/62 [00:16<00:00,  3.79it/s]
```

Results- Model 2.3

```
🎵 2871/2874 Depeche Mode - Personal Jesus-15.wav
   🎯 Ground truth: acoustic_guitar, voice
   🎺 Top prediction: piano (0.2611)
   📊 Evaluation: ❌ INCORRECT
   📈 Running accuracy: 1462/2871 = 50.9%
--------------------------------------------------

🎵 2872/2874 Iron Maiden - 11 - The Evil That Men Do-8.wav
   🎯 Ground truth: acoustic_guitar, voice
   🎺 Top prediction: voice (0.8681)
   📊 Evaluation: ✅ CORRECT
   📈 Running accuracy: 1463/2872 = 50.9%
--------------------------------------------------

🎵 2873/2874 Debussy - Arabesque-7.wav
   🎯 Ground truth: piano
   🎺 Top prediction: clarinet (0.8762)
   📊 Evaluation: ❌ INCORRECT
   📈 Running accuracy: 1463/2873 = 50.9%
--------------------------------------------------

🎵 2874/2874 15_more than a feeling - boston-4.wav
   🎯 Ground truth: acoustic_guitar
   🎺 Top prediction: voice (0.8689)
   📊 Evaluation: ❌ INCORRECT
   📈 Running accuracy: 1463/2874 = 50.9%
--------------------------------------------------

✅ Inference completed!
📊 Final accuracy: 1463/2874 = 50.9%

============================================================
🎯 FINAL SUMMARY
============================================================
Total files processed: 2874
Files with ground truth: 2874
Correct predictions: 1463
Overall accuracy: 50.9%
============================================================
```

Lessons from model 2.X

1. Smaller network, with little loss in data.
2. Still, the CNNs were not deep enough to reach consistent decisions for all instruments
3. Some instruments did get good classifications- Idea for next project: Specialized networks for specific cases?
4. Multiple instrument training is too noisy. We got better results identifying the most prominent instrument here.
5. The network is still very large, not efficient enough

6. Best accuracy acquired: 51%

Conclusion

Our design was not good enough to justify it's complexity.

Choosing a better dataset could improve results, for some tasks.

We believe this field has potential, but as students we did not find the means to discover it.

Thank you!