ML4NR

Machine learning for noise reduction in old audio records

Bachelor thesis

Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

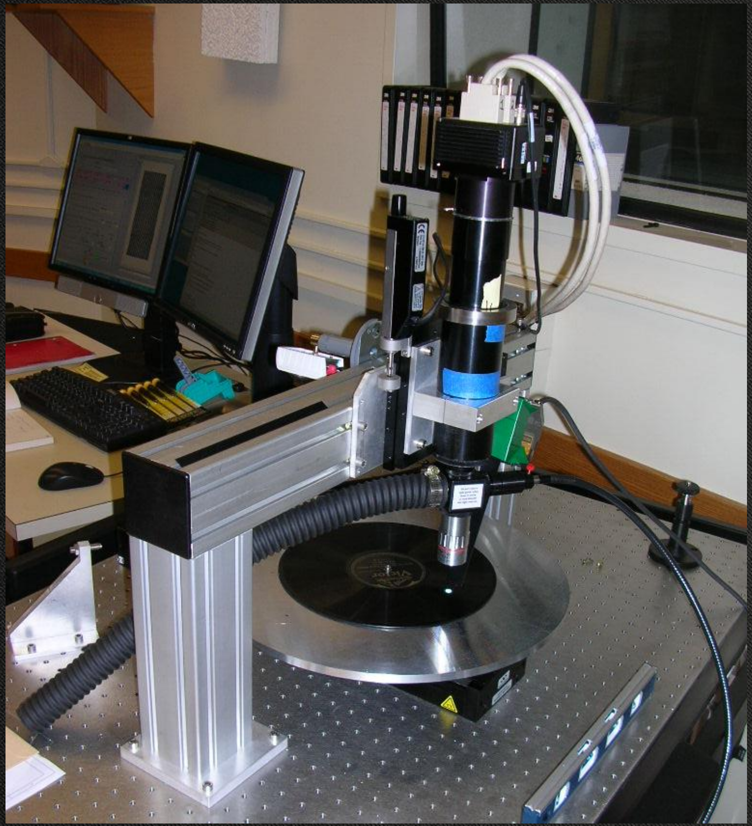
BERKELEY LAB
Lawrence Berkeley National Laboratory

Audio records

- Thomas Edison, 1877
- Impractical cylinders
- Brittle material, deterioration
- Here : 78 RPM Shellac discs

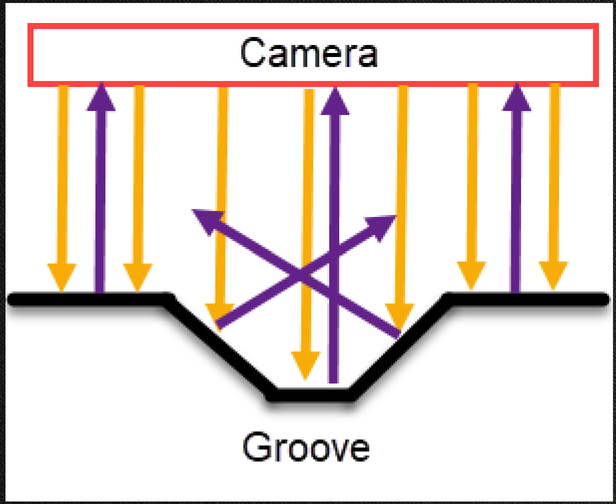
- Preservation of valuable data

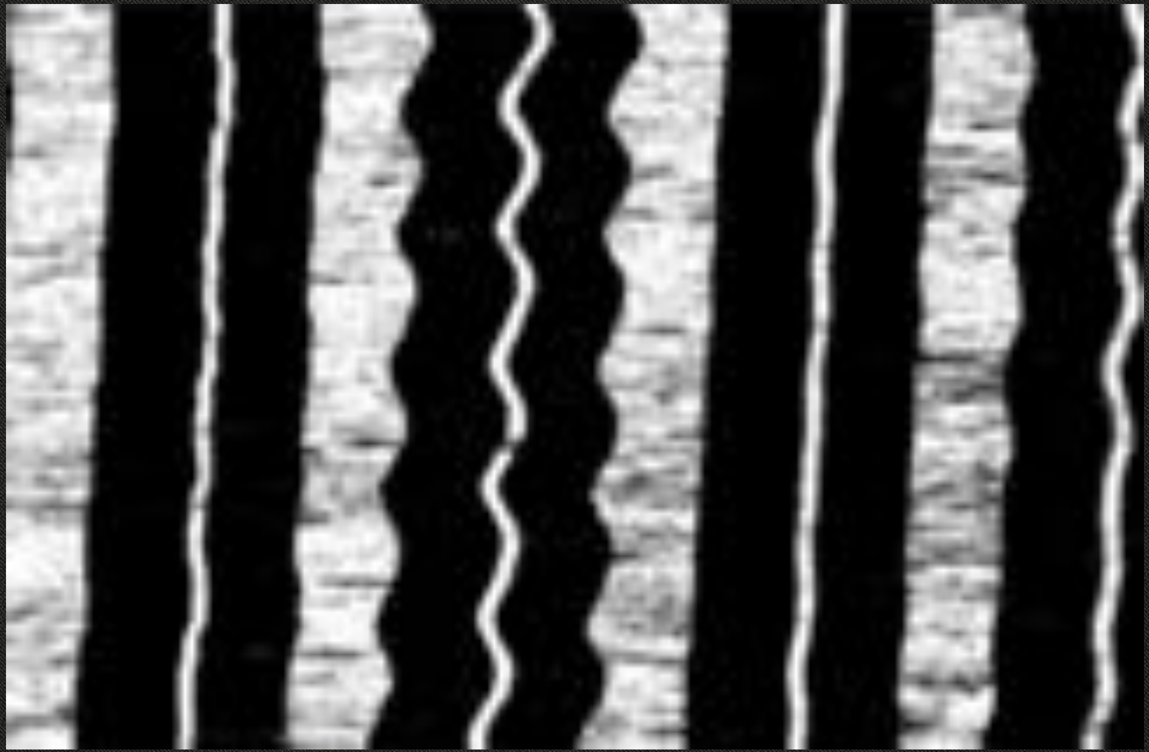
IRENE and Weaver

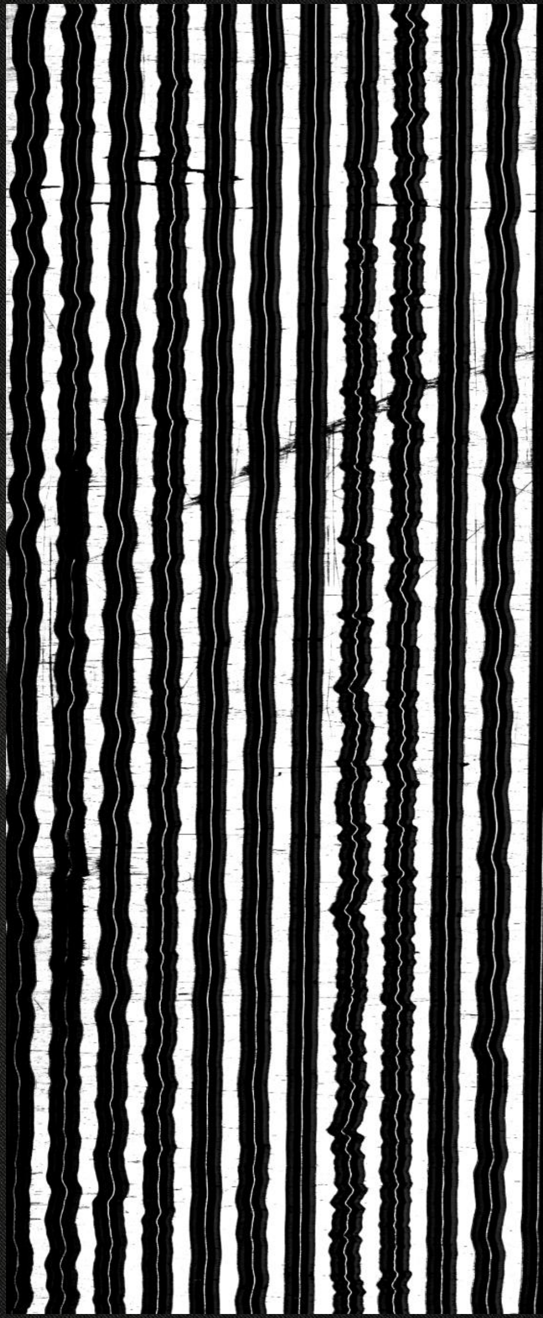


- Imaging machine
- Processing software
- Able to simulate play-back
- Preservation of records as pictures

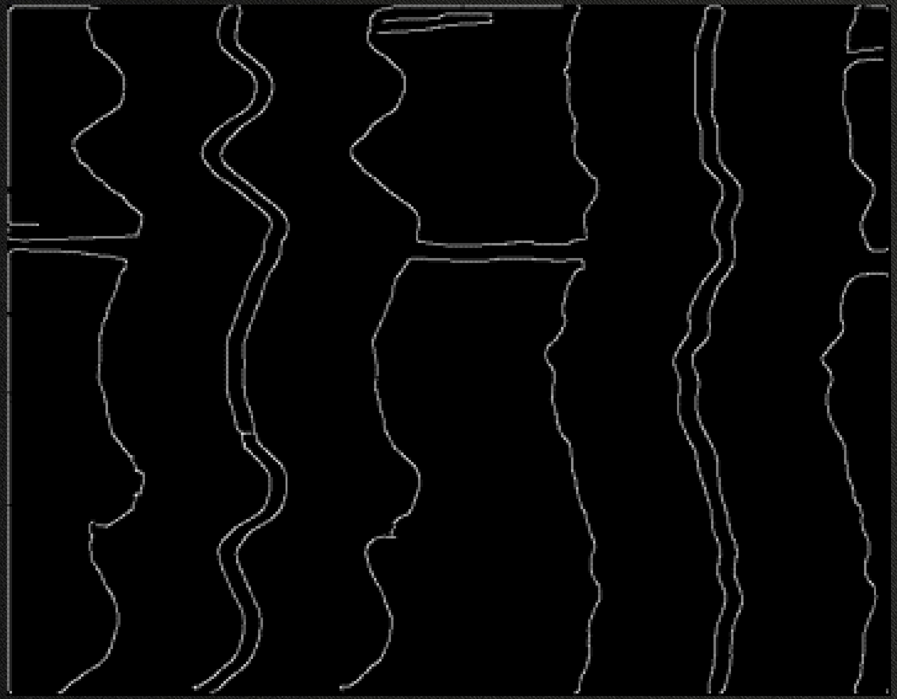
Taking pictures







Audio reconstruction



- Edge detection, middle of groove
- Perpendicular velocity, not position
- Affected by noise

Machine learning

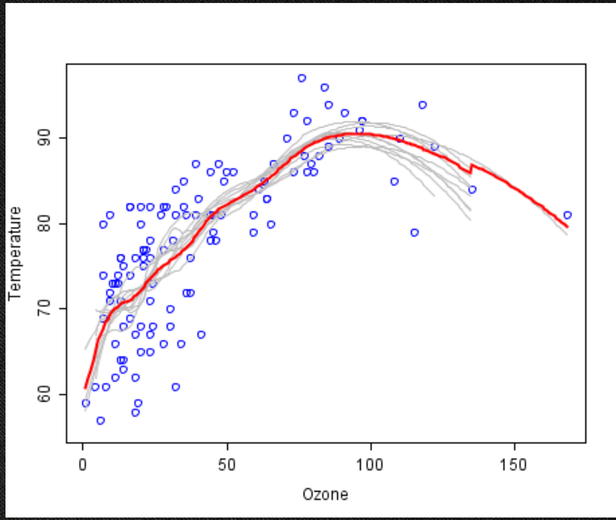
Classification : FNN, CNN, RNN, ResNet, …

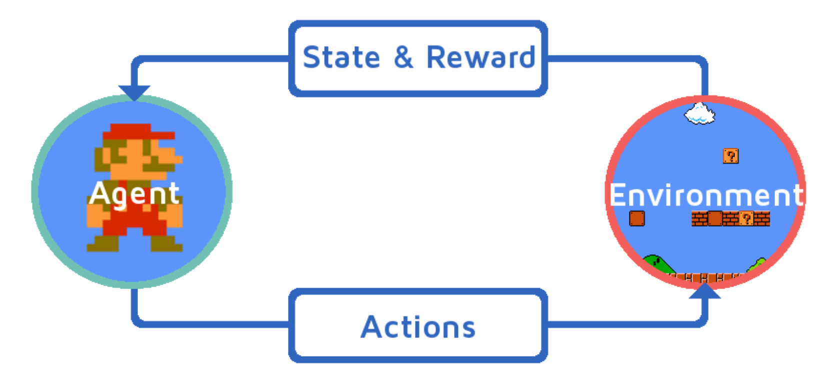
Regression : logistic, linear, LSTM, CNN …

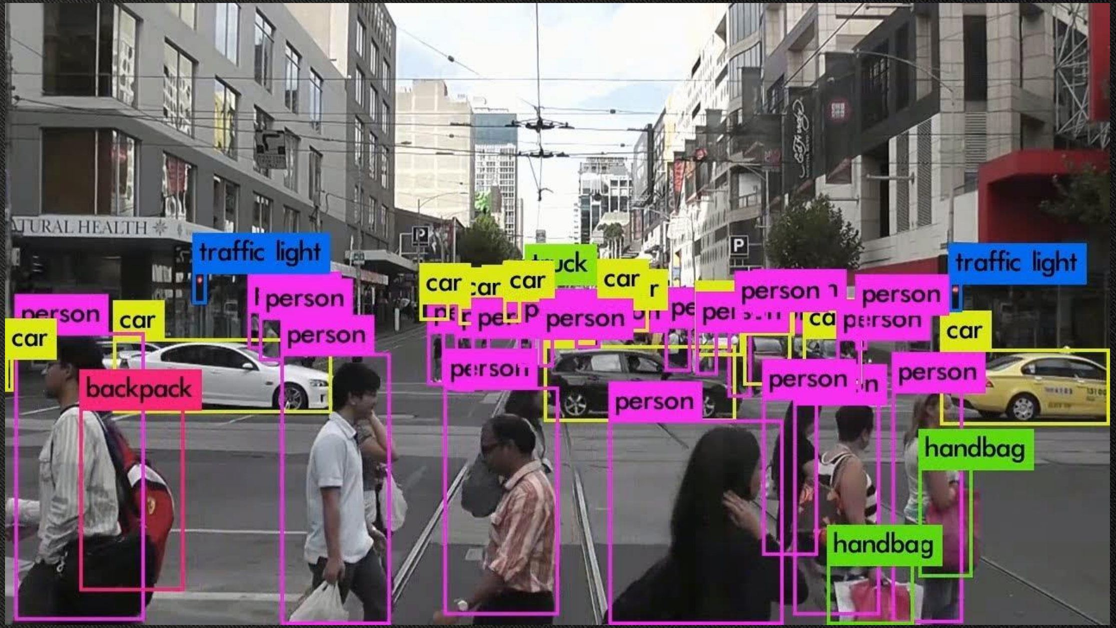
Reinforcement : Decision making

Generation : GAN, …

Denoising : Auto-encoder, …



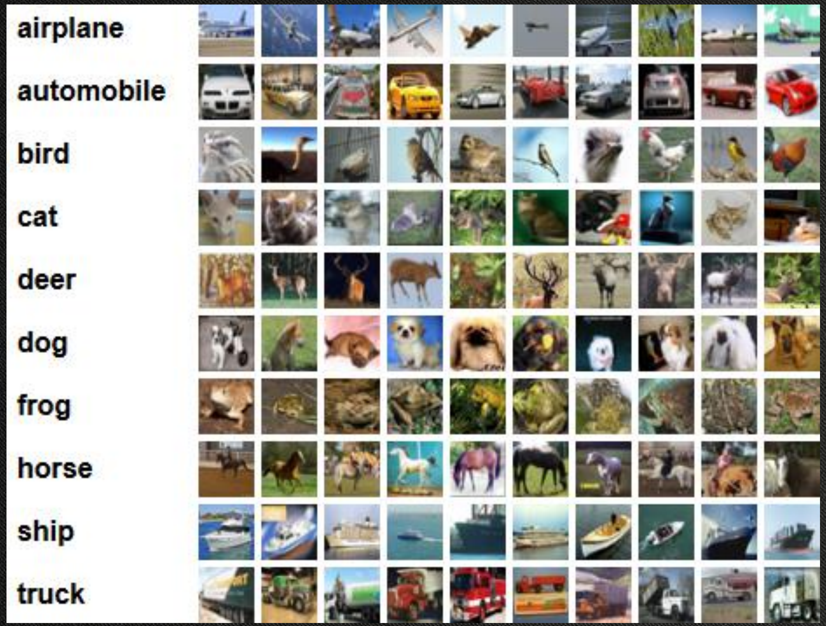




Objectives

- Improve the audio generated from images. Divided in prototypes for validation :
- Prototype 0 : Keras familiarization
- Prototype 1 : clean sound from clean sine grooves
- Prototype 2 : clean sound from noisy sine grooves
- Prototype 3 : noisy sound from disc images
- Prototype 4 : clean sound from disc images

Prototype 0



Keras :

- Makes implementation of machine learning modes simpler
- Lots of tools for data processing

CIFAR-10 :

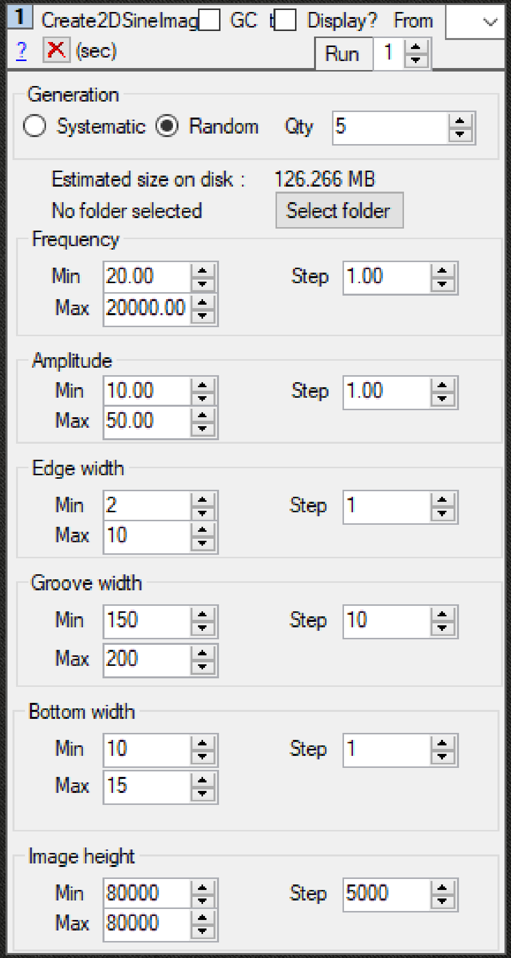
- 60’000 images in 10 classes

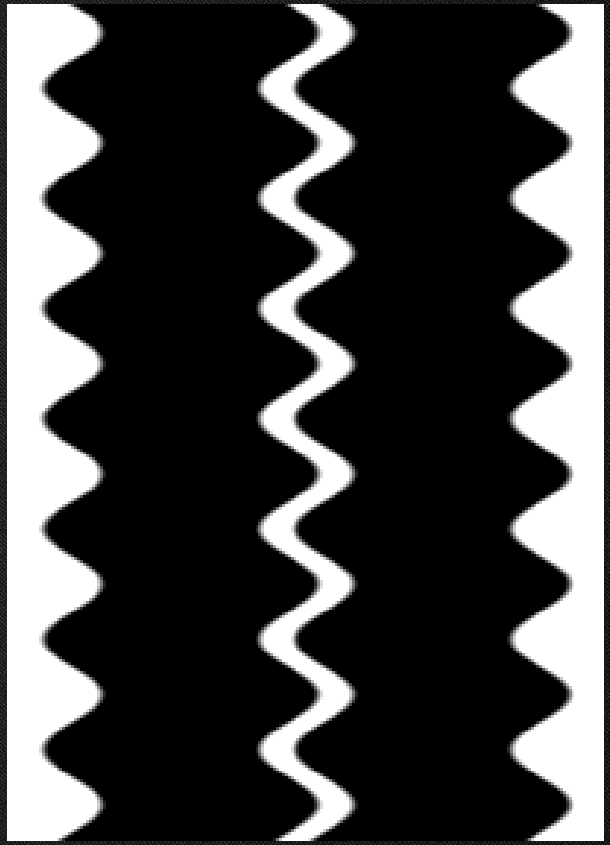
```
model = Sequential()
model.add(Conv2D(32, (3, 3), padding='same',
                 input_shape=x_train.shape[1:]))
model.add(Activation('relu'))
model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
```

```
model.fit(x_train, y_train,
          batch_size=batch_size,
          epochs=epochs,
          validation_data=(x_test, y_test),
          shuffle=True)
```

Prototype 1 - dataset

- Generation of pure sine groove images with Weaver (80’000 x 220)
- Randomized values for robustness of training
- Weaver’s audio reconstruction as target

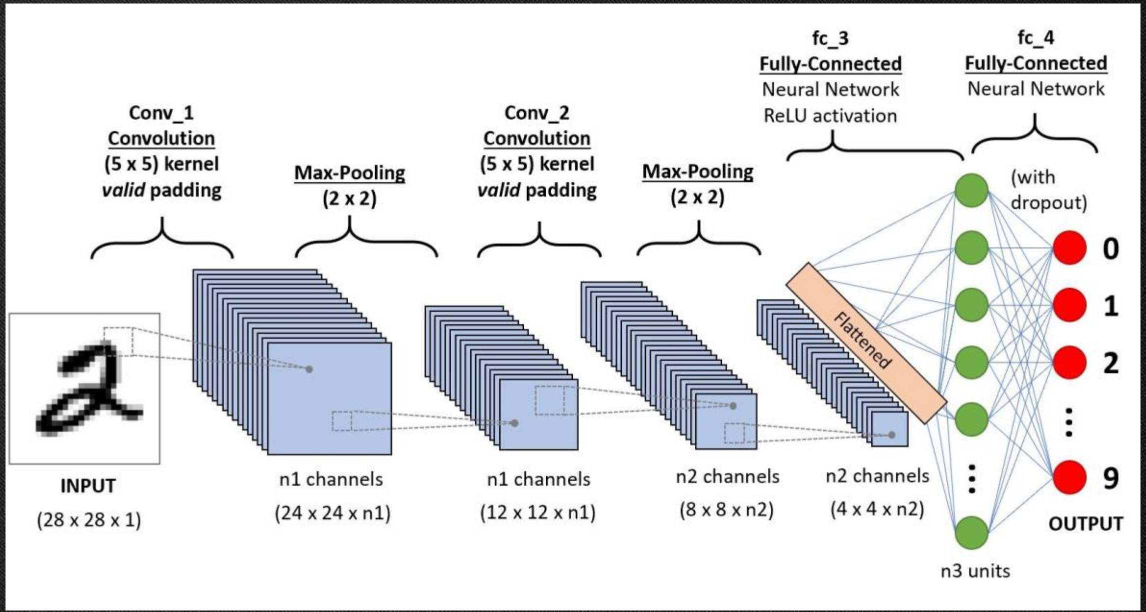




Prototype 1 – first version

- Convolutional neural network (same as prototype 0)
- Input : entire groove image - Target : entire sound wave

- Problems : Trillions of weights



Prototype 1 – second version

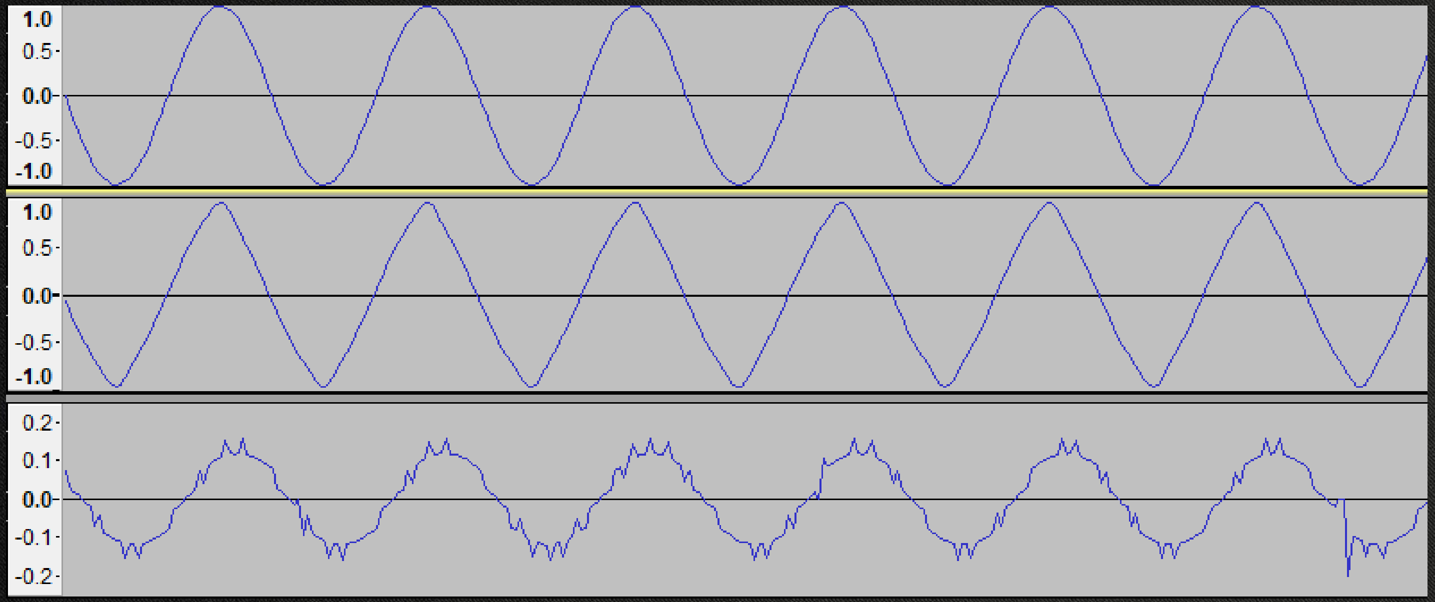
- Long Short-Term Memory network
- Input : block of rows
- Problem : no convergence
- Target : amplitude of sound (~= lateral velocity of stylus)

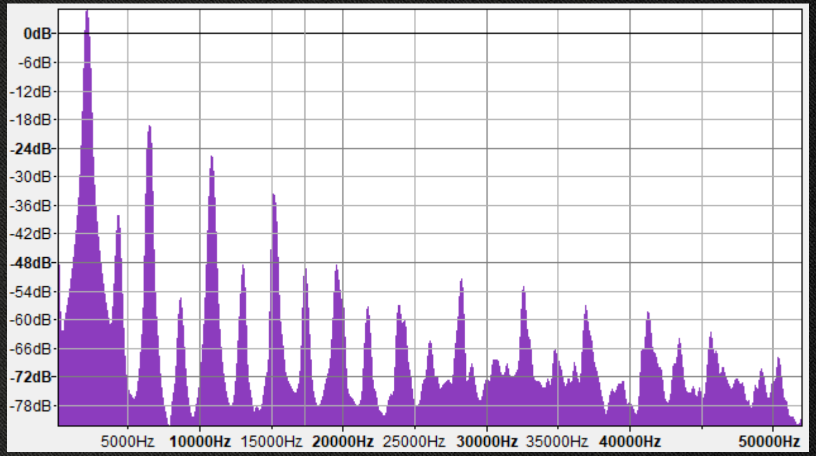
-> 0.8
-> 1.0
-> 0.6
-> 0.1
-> -0.3
-> -0.9
-> -1.0
-> -0.4
-> 0.0
-> 0.4

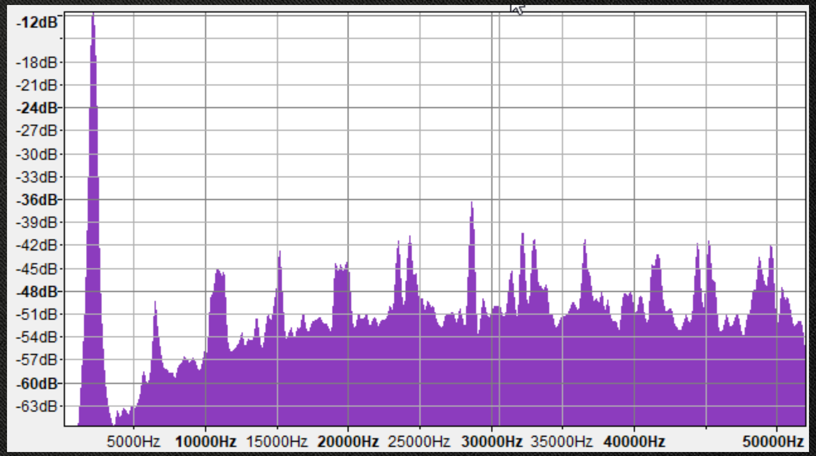
Prototype 1 – third version

- CNN again
- Input : block of rows
- Target : amplitude of sound (lateral velocity od stylus)
- Data preprocessing : Normalization of pixel values (0 to 1), generation of pure sine wave as target
- Promising results !

Prototype 1 – third version



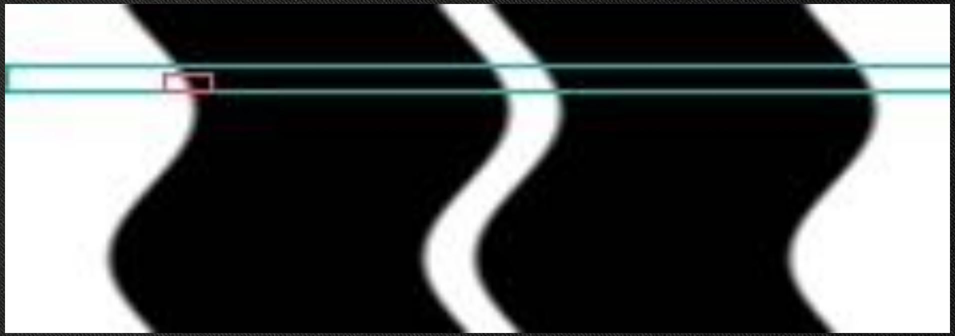




Prototype 1 – Fine tuning

Finding best values for all hyper-parameters :

- Number of rows in a block
- Number of predictions in a block
- Size of convolution filters
- Max pooling or not, what size
- Dropout or not, what rate
- Optimizer and its learning and decay rate
- Activation functions

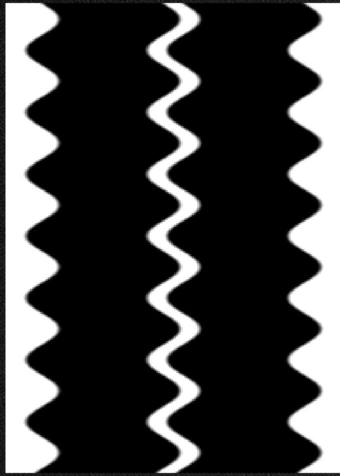


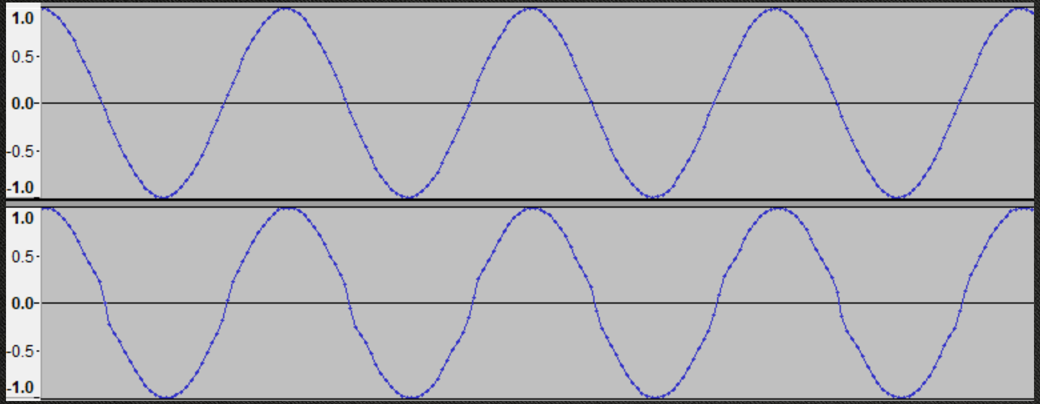
Prototype 1 – Final Results

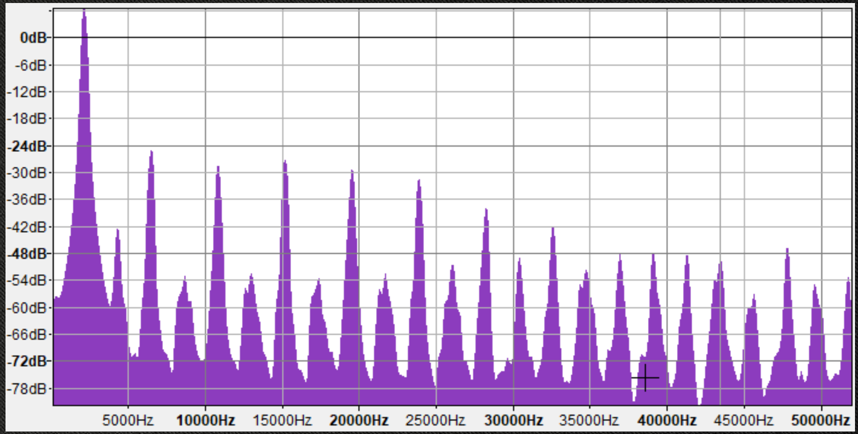
Best values = small values !

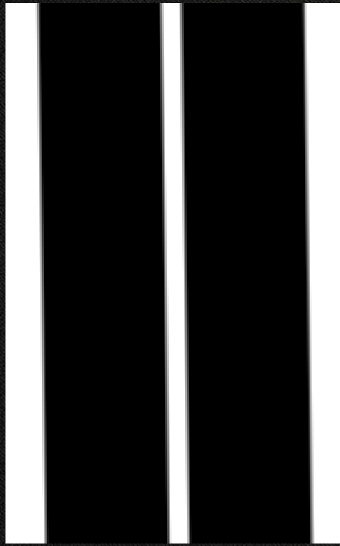
- Only 5 rows each time, with 1 prediction
- Convolutions of 3x3
- Max pooling of 2x2
- Dropout of 0.25 for first layers, 0.5 for last
- Close to default values for RMSprop optimizer
- ReLu for first layers, tanh for last

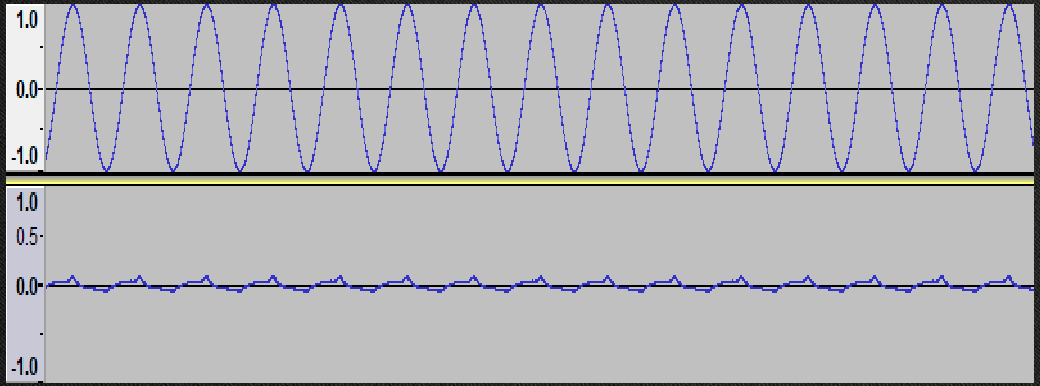
Prototype 1 – Final Results



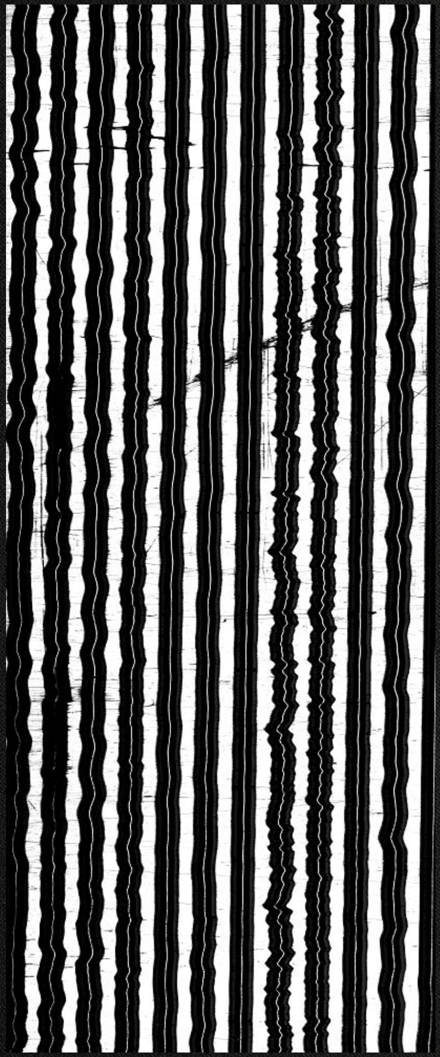


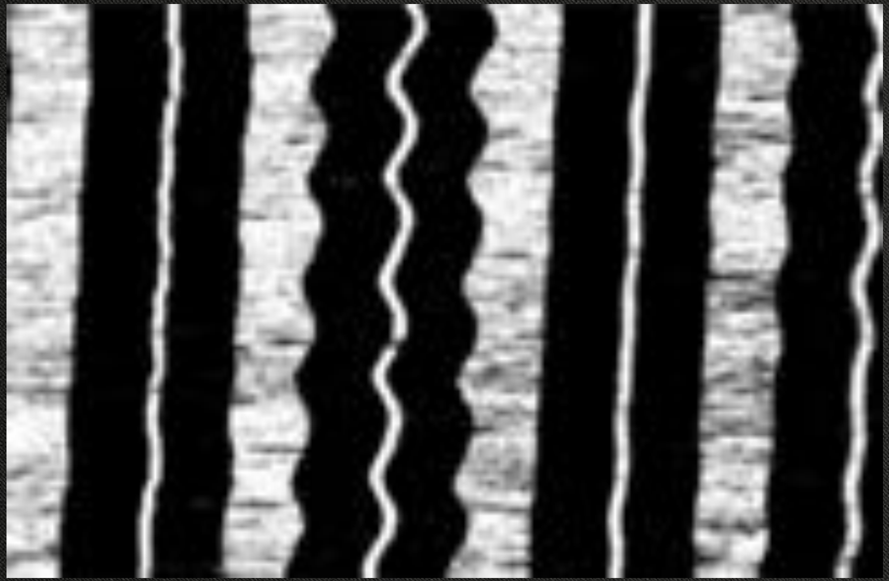


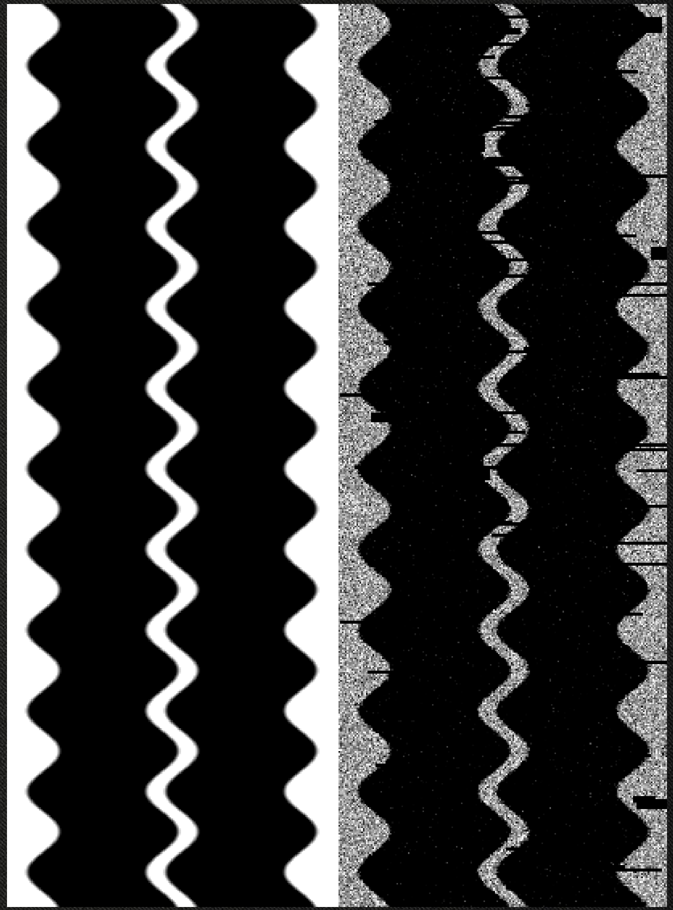




Prototype 2 – Noisy images







Prototype 2 – Fine tuning again

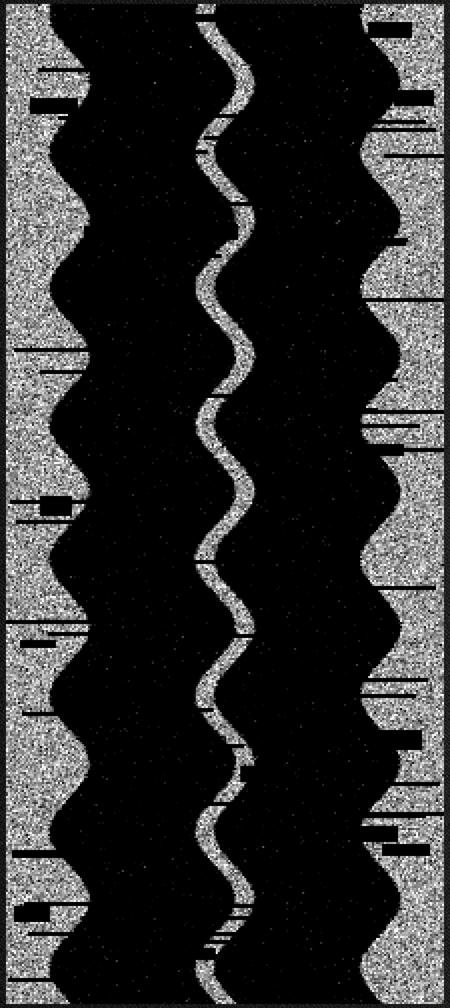
- Properties are different
- Maybe seeing more rows can help correct defaults
- The optimizer is bad : results are fast but not improving over time.

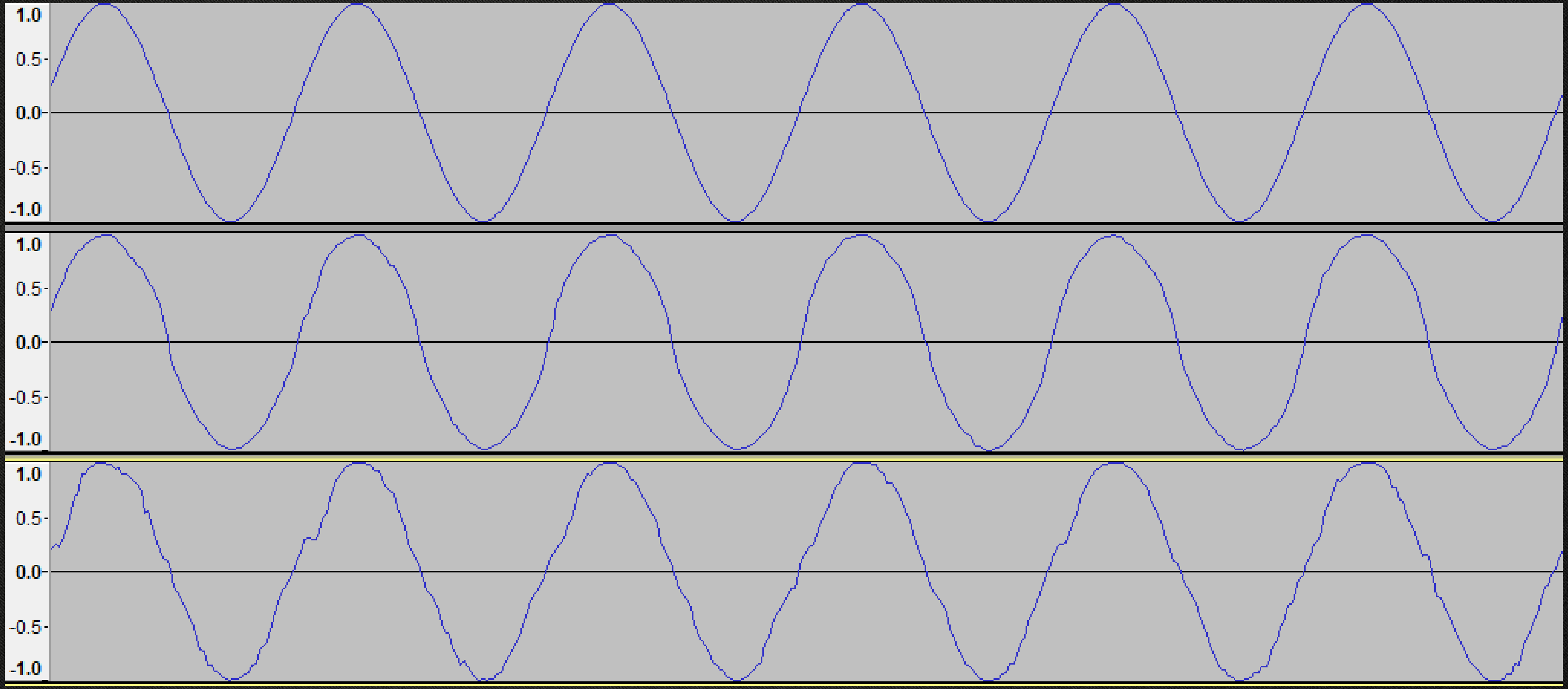
- General architecture stays the same

Prototype 2 – Results

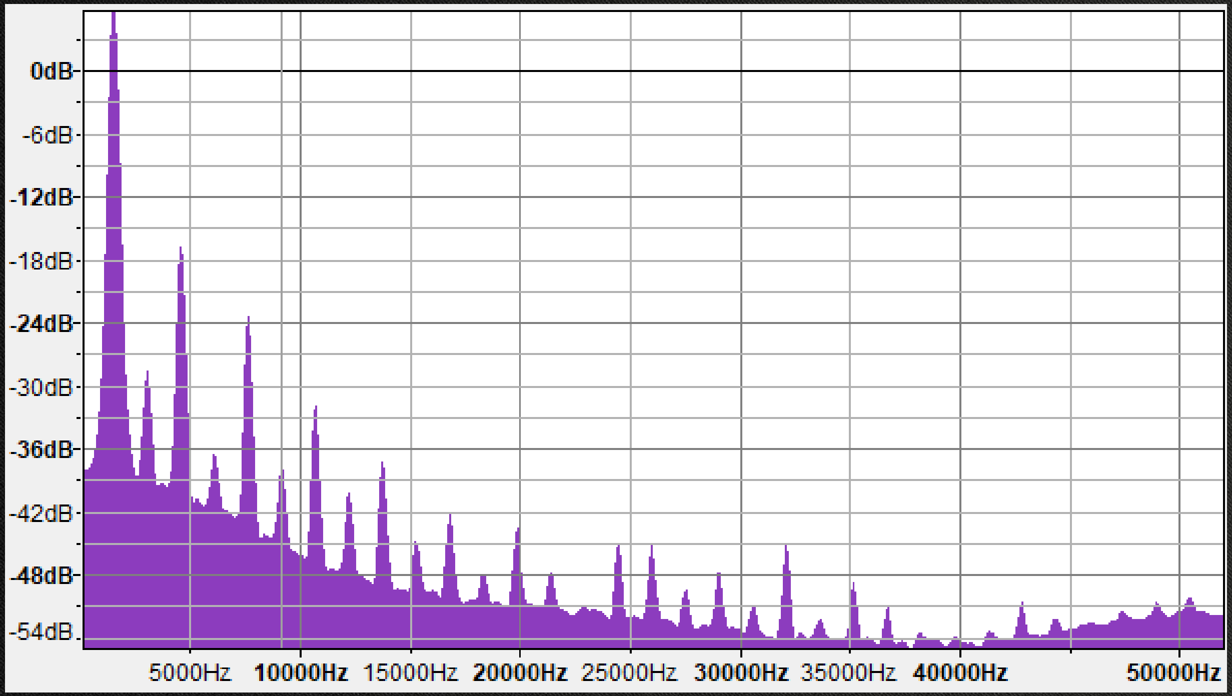
- Best results with 15 rows of pixels
- RMSProp and Adam optimizers have good but different results, slightly improving over time

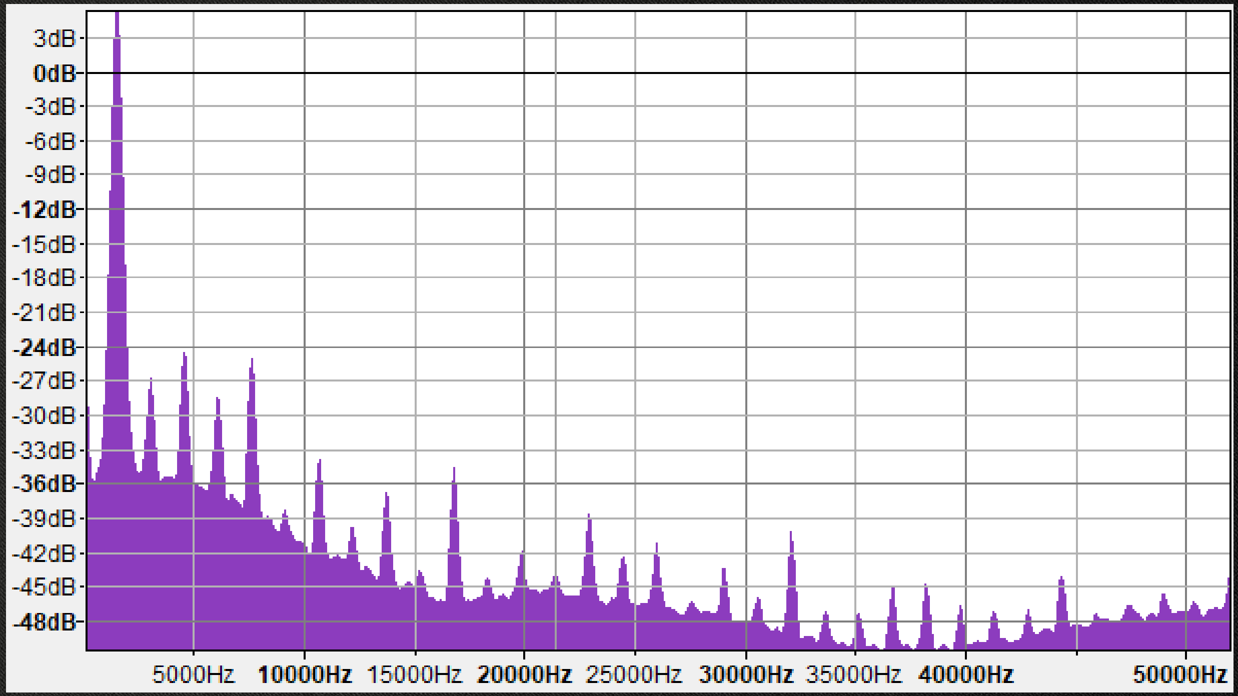
Prototype 2 – Results





Prototype 2 – Results





Next steps

- Using actual disc images
- Transfer learning or training from the start
- Finish report and evaluate project