Big Data Computing
2022/2023

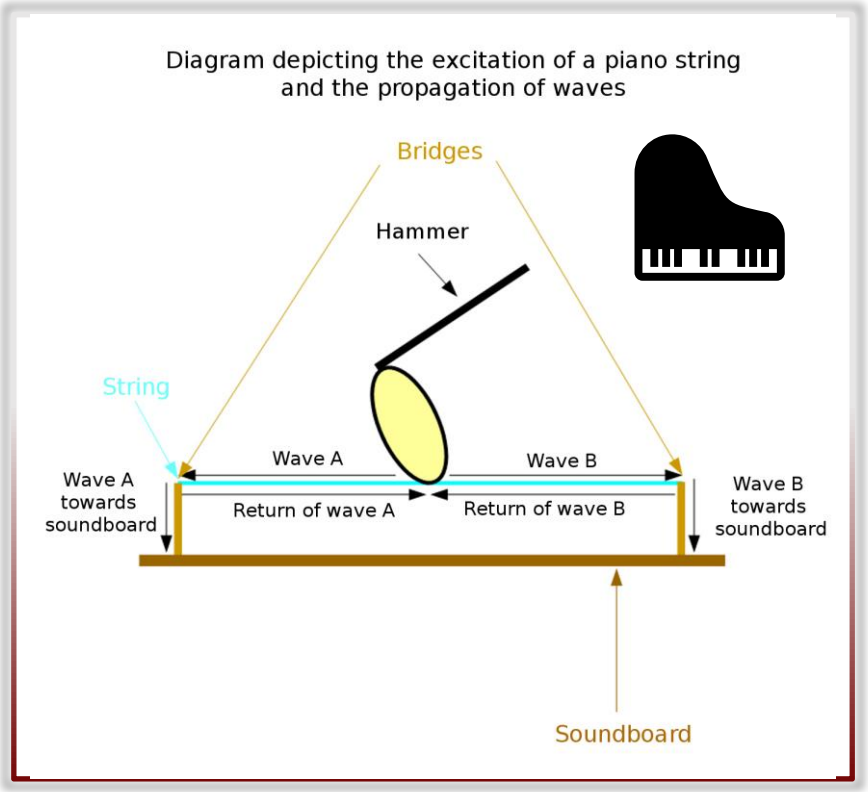
Deep Audio Modelling

Davide Gabrielli

1883616

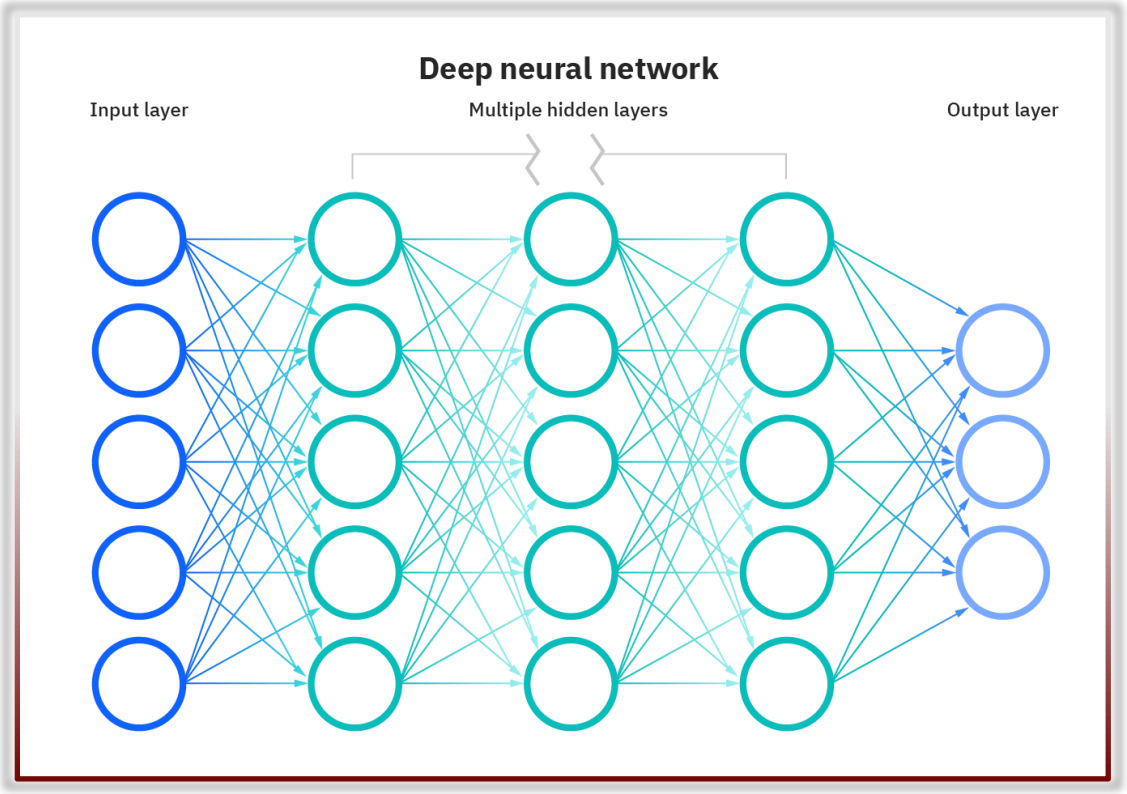
What does it mean "Audio Modelling"?

Physical Modelling Synthesis is a sound synthesis technique that employs **mathematical models to simulate** the behaviour of a **musical instrument**, recreating its unique sound characteristics.



Why using Deep Neural Networks?

Since Deep Neural Networks are able to **approximate non-linear functions**, they are really useful when dealing with the trivial problem of defining a mathematical function that characterizes a musical instrument.



Dataset

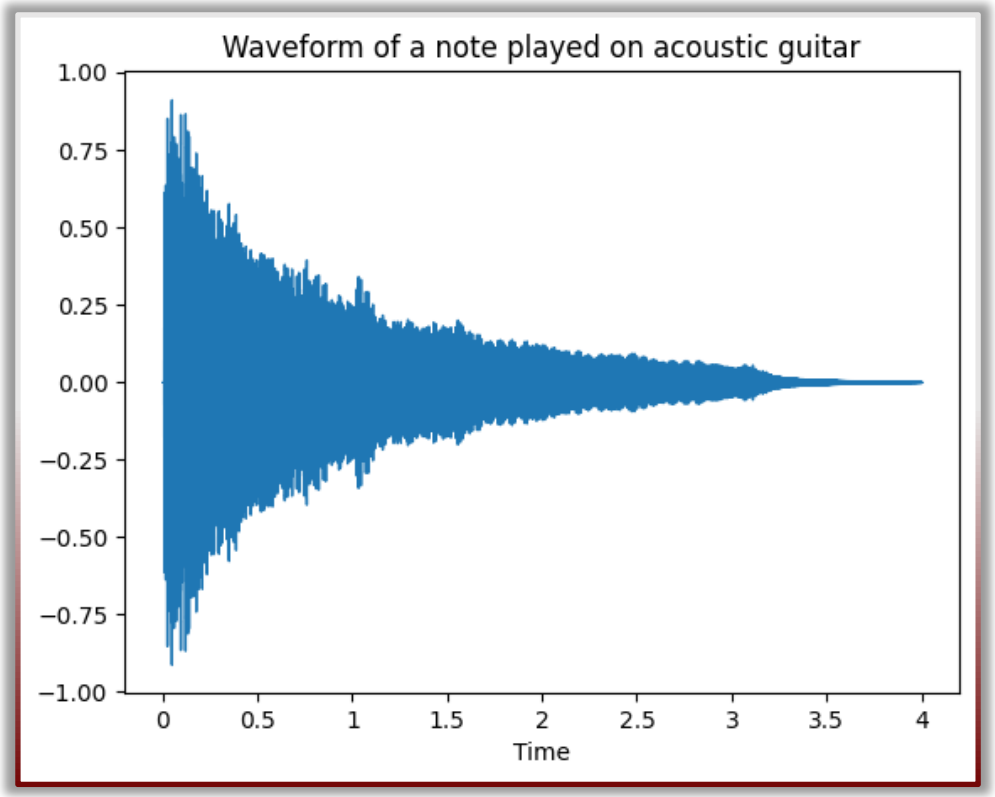
The **NSynth Dataset** from Google Magenta Project is a large-scale and high-quality dataset of annotated musical notes.

It contains 305,979 musical notes, each with a unique pitch, timbre, and envelope for **1,006 instruments** (≈ 78 GB).

Audio Signal

Sound is **digitally represented** through sampling, which involves recording samples of **sound pressure** over time at **regular intervals**.

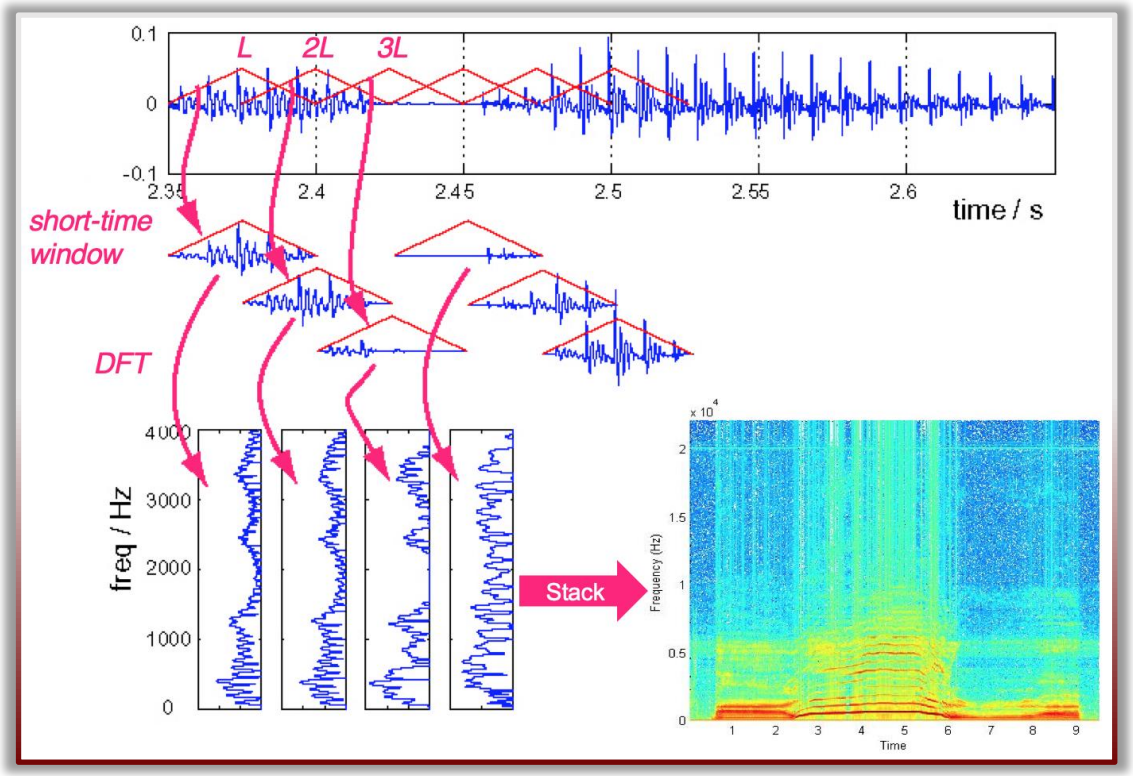
This type of representation does not easily yield the **features of interest** to our auditory system, such as sound frequency, which are typically obtained only by **analysing large windows** of data.



Spectrograms

A **better representation** of sound for our problem would be the **spectrogram**.

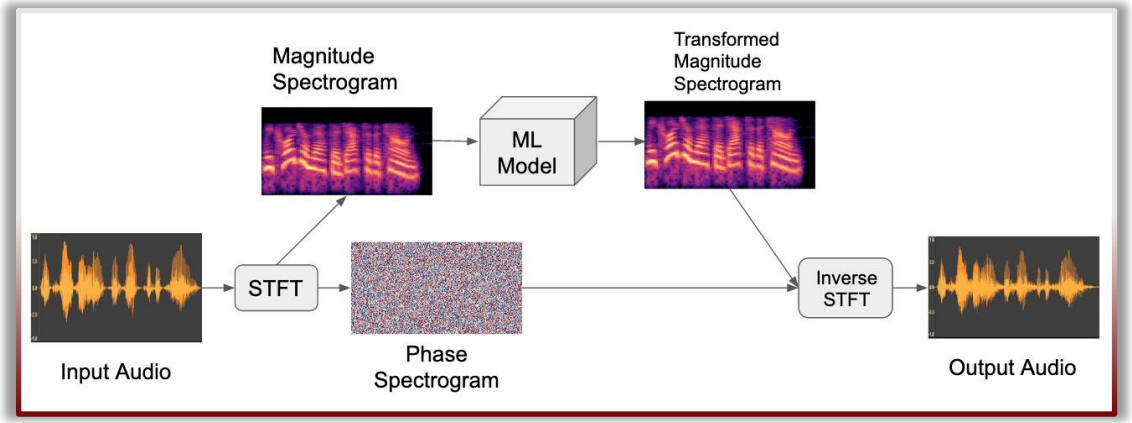
Spectrograms are generated by applying the **Short Time Fourier Transform** (STFT) to the audio signal.



Why Spectrograms?

Spectrograms are matrices **M** of float values where each **M[i][j]** value represents the magnitude of the **i-th frequency** at the **j-th** time step.

In this way we can handle audio in the same way we handle images but we have to consider that:



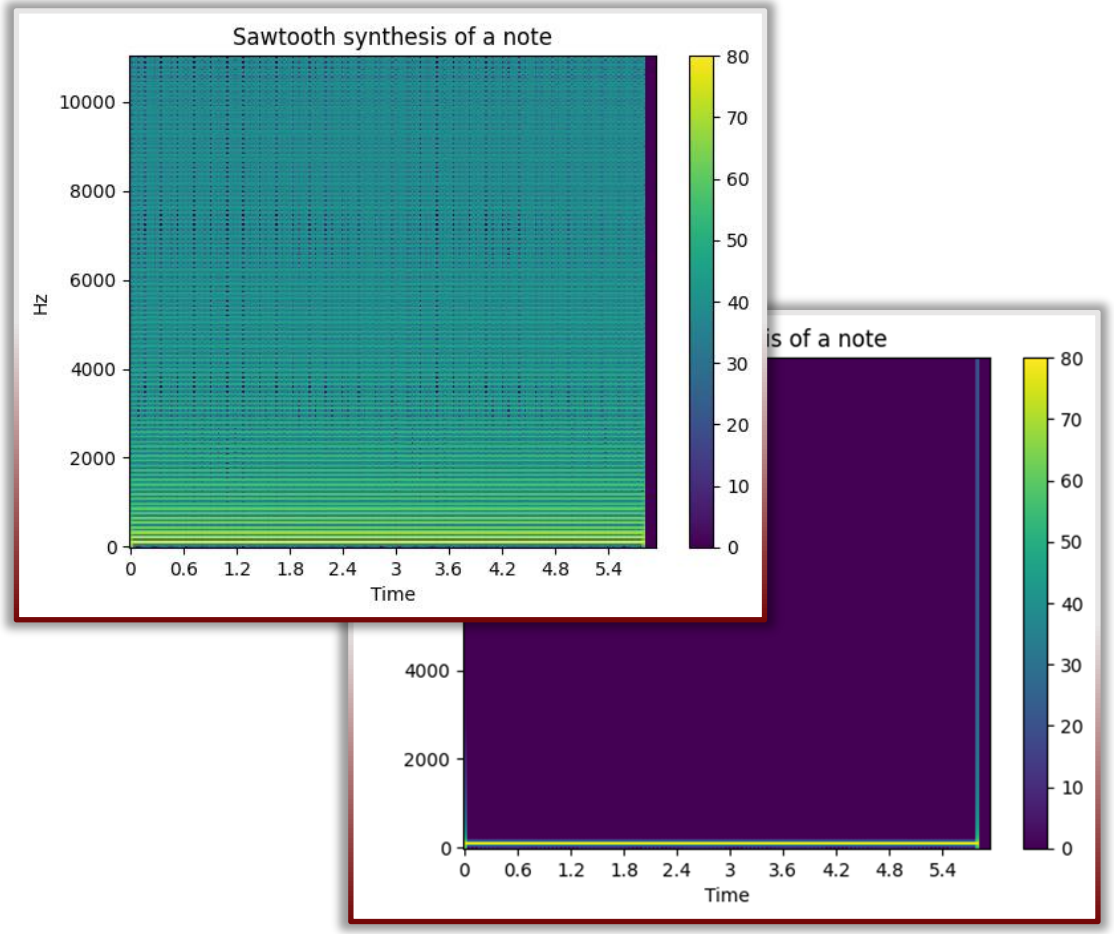
- The values are into the **range [0, +80]**.
- We have to ignore the **phase** of the signal

Employed Strategy

Midi information has really few data and learning a function able to reconstruct sound starting from those is really hard.

In order to make the model **learn faster** and keep **coherence on pitch** information some tricks have been used:

- **Midi to audio synthesis**
- **Conversion between spectrograms**

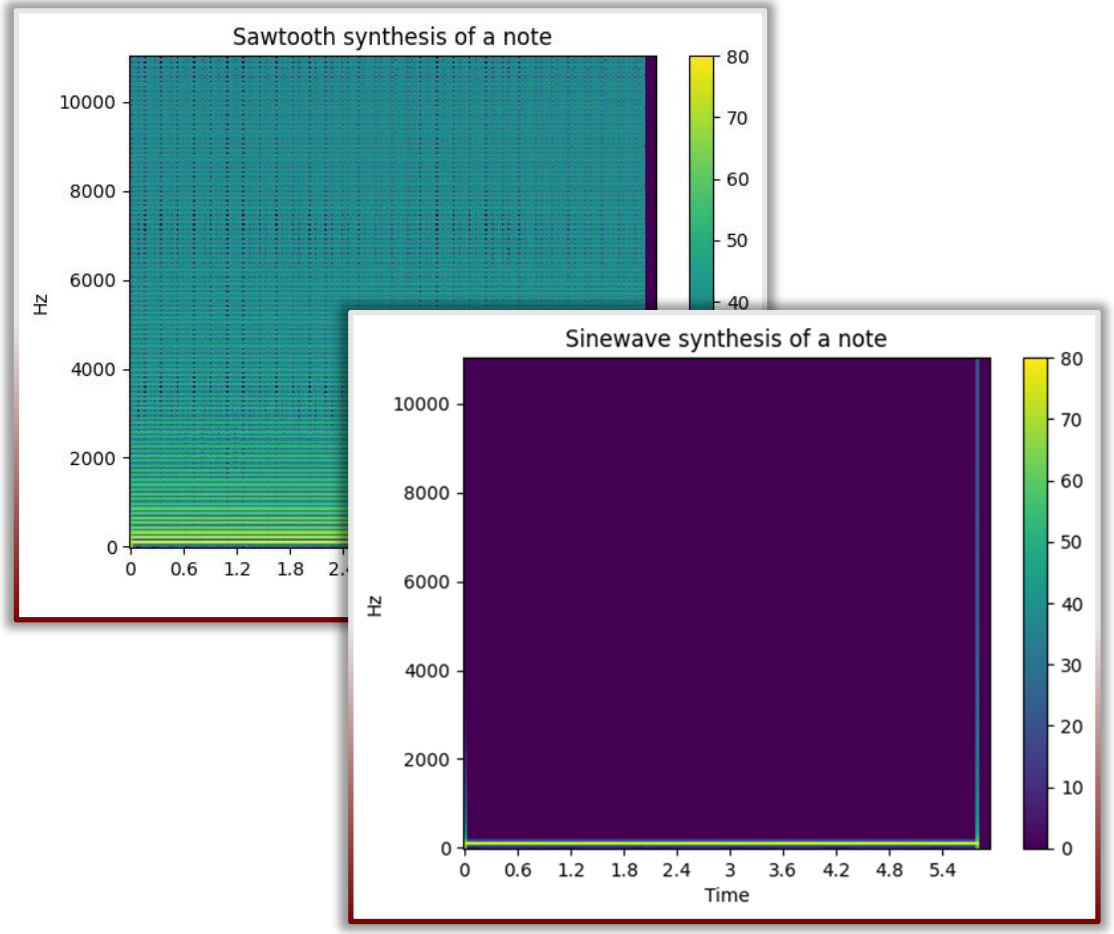


Employed Strategy

Midi information has really few data and learning a function able to reconstruct sound starting from those is really hard.

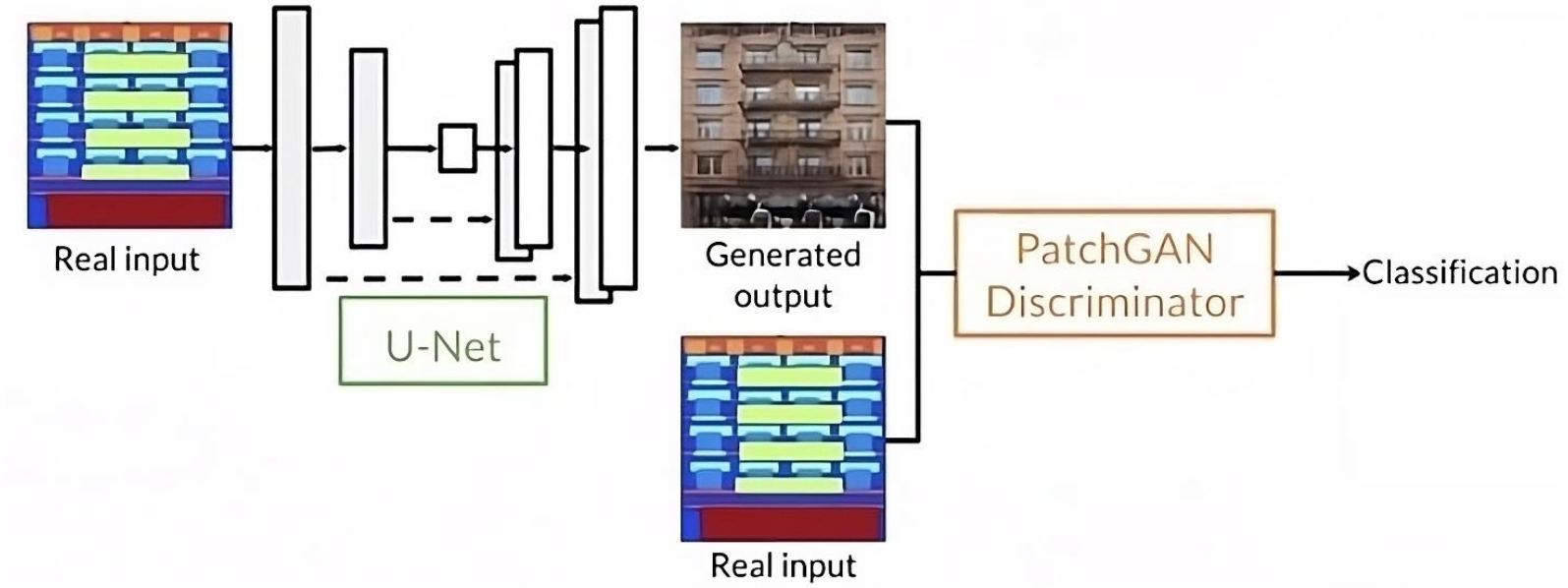
In order to make the model **learn faster** and keep **coherence on pitch** information some tricks have been used:

- **Midi to audio synthesis**
- **Conversion between spectrograms**



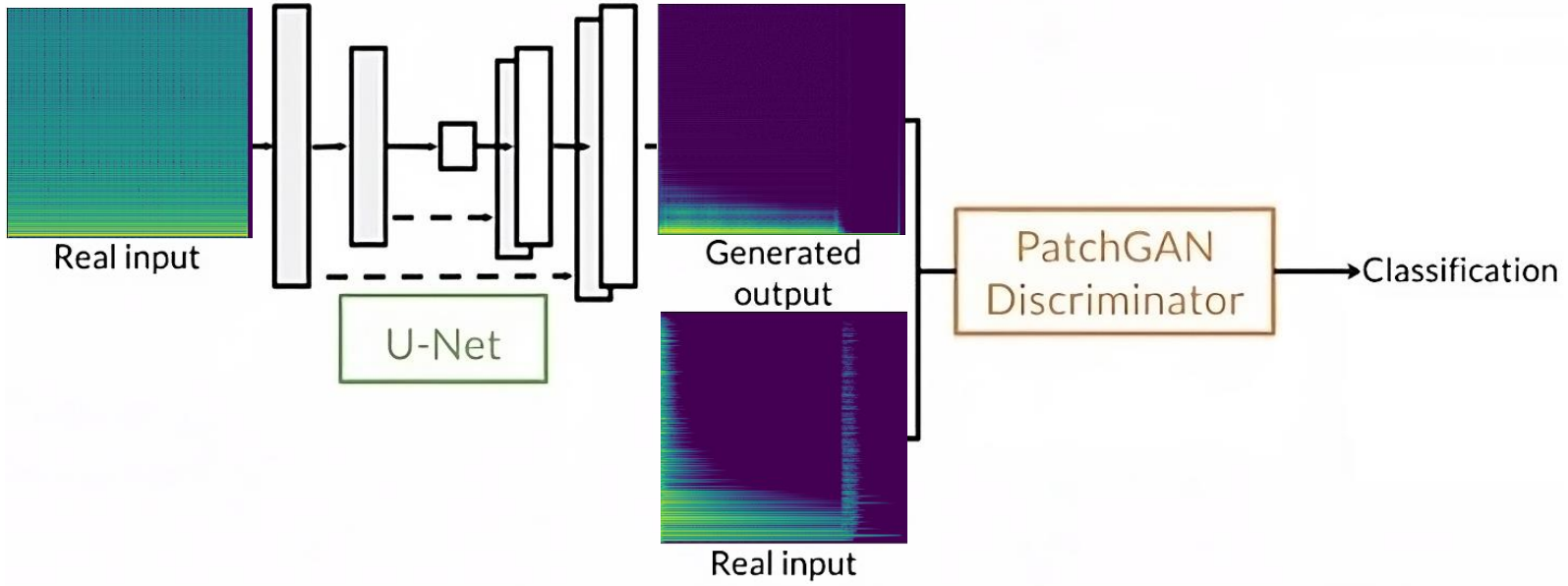
Pix2Pix: Image-to-Image Translation

Pix2Pix is a Generative Adversarial Network, or **GAN**, model designed for general purpose **image-to-image** translation.



Pix2Pix Architecture for Spectrograms

In our case instead of images we'll use spectrograms of **midi-synthesized as domain A** and **spectrograms of the instrument** we want to model **as domain B** for the translation.



Training Approaches

For this experiment combination of the following different methods have been developed and tested:

- Finding the best **midi representation**:
  - Sawtooth synthesis
  - Sinewave synthesis
- Finding the best **activation function**:
  - Sigmoid
  - Tanh

Spark + Petastorm + Pytorch

The model is written in **Pytorch** and **Petastorm** have been used to convert data from the **Spark Dataframe** to a Pytorch Dataloader.

Evaluation Results

In order to compare models, it have been used as evaluation metrics the **Fréchet Audio Distance**.

This method extracts some relevant feature from the generated audio and compute the distance from those of the original audio.

This metric have been computed using two different models **VGGISH** and **PANN**.

	Sawtooth + Tanh	Sawtooth + Sigmoid	Sinewave + Tanh	Sinewave + Sigmoid
FAD-VGGISH	**2.372**	3.346	5.211	8.518
FAD-PANN	$1219 * 10^{-6}$	$322 * 10^{-6}$	$379 * 10^{-6}$	$\mathbf{92 * 10^{-6}}$

Results Listening

Thanks for the attention!