Feature Extraction: Image, Video, Audio, & Text Data

Saransh Goel	2019A7PS0988P
Ayush Sharma	2018A3PS0326P

Submitted to

Dr. Navneet Goyal

Department of Computer Science & Information Systems

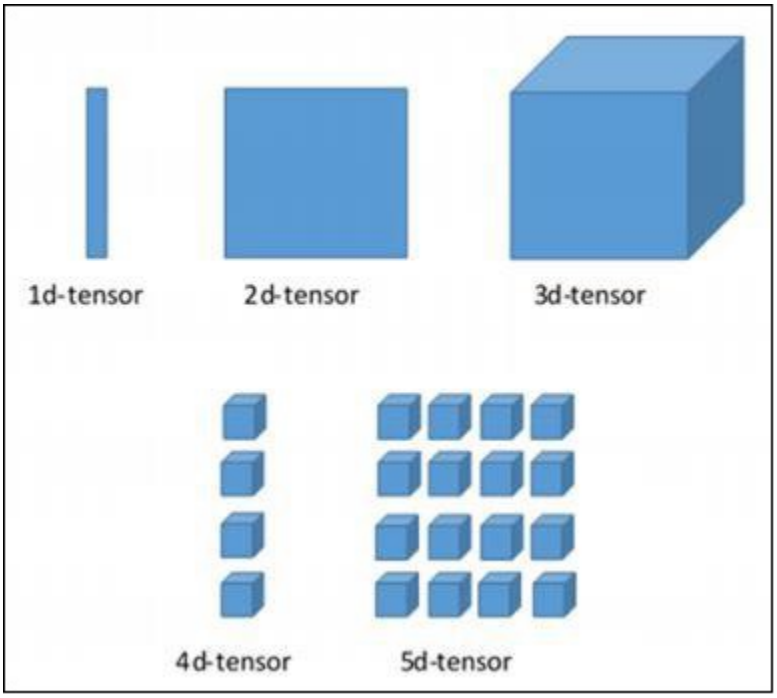
For the partial fulfillment of the course

CS F320 Foundations of Data Science

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

DATA REPRESENTATION

Before going further to learn about the methods of extracting information from text, image, audio and video, firstly let's talk about the data structure used to represent these elements. Images cannot be saved directly or acted upon directly because computers work in terms of numbers. There are various ways to store the data but another important thing is to retain its original internal relationships and should not be computation intensive. Nowadays one of the best solutions to the above problem is tensors. Tensors are n-dimensional data structures like scalar is zero dimensional, vector is one dimensional, matrix is two dimensional, cube is three dimensional.



Rank

Unit of dimensionality described within the tensor is called rank. It identifies the number of dimensions of the tensor. A rank of a tensor can be described as the order or n-dimensions of a tensor defined.

Shape - number of rows and columns

Type - type of data stored

There are many advantages of tensors.

1. *Tensors* retain the original internal relationships of the data, for example a 2-d tensor is used to represent the gray scale image. If we tried to store pixel values in linear array format some relations of pixels with pixels upward and downward will be lost but with 2-d tensors this problem is resolved. RGB images can be stored in a 3-d tensor containing three layers of red, green and blue.

2. *Tensor Processing Units(TPUs)* are google's custom - developed applications - specific integrated circuits (ASICs) specially made for computation of tensors in an efficient and fast way. TPU resources accelerate the algebraic computations which are usually very slow on normal hardware due to high dimensionality but TPU hardware is designed for tensors only hence is faster. This advantage of TPU helps tensors to grow with such a fast speed.

3. *TensorFlow* is an open source machine learning framework developed by Google to apply various machine and deep learning concepts. TensorFlow uses tensors for storing data.TensorFlow is designed in python programming language.

```
import tensorflow as tf
import numpy as np

matrix1 = np.array([(2,2,2),(2,2,2),(2,2,2)],dtype = 'int32')
matrix2 = np.array([(1,1,1),(1,1,1),(1,1,1)],dtype = 'int32')
print (matrix1)
print (matrix2)
matrix1 = tf.constant(matrix1)
matrix2 = tf.constant(matrix2)
matrix_product = tf.matmul(matrix1, matrix2)
matrix_sum = tf.add(matrix1,matrix2)
matrix_3 = np.array([(2,7,2),(1,4,2),(9,0,2)],dtype = 'float32')
print (matrix_3)
print ("matrix 1 = ",matrix1)
print ("matrix 2 = ",matrix2)
print ("matrix 1 + matrix 2 = ",matrix_sum)
print ("matrix 1 * matrix 2 = ",matrix_product)
```

```
matrix 1 =  tf.Tensor(
[[2 2 2]
 [2 2 2]
 [2 2 2]], shape=(3, 3), dtype=int32)
matrix 2 =  tf.Tensor(
[[1 1 1]
 [1 1 1]
 [1 1 1]], shape=(3, 3), dtype=int32)
matrix 1 + matrix 2 =  tf.Tensor(
[[3 3 3]
 [3 3 3]
 [3 3 3]], shape=(3, 3), dtype=int32)
matrix 1 * matrix 2 =  tf.Tensor(
[[6 6 6]
 [6 6 6]
 [6 6 6]], shape=(3, 3), dtype=int32)
```

IMAGE

Image processing in machine learning models

Image processing is very common nowadays, many big companies use image processing for securing things like face id, for providing information like google lens etc.

Use of image processing in real world

1. Security

   Used in mobile phones as face identification in place of password or fingerprint id. Face identification is also used in competitive exams nowadays to prevent cheating.

2. Medical imaging

   Analysis of images of internal organs to predict the exact cause of disease or wound and to find the anomalies faster. Even used in medical examinations of Egyptian mummies to analyse the methods used during the mummification process.

Q: How to store an image so that the computer will understand and learn from it?

A: Image stored as 2-d array of pixel

- A pixel is a "picture element", or the basic element of a digital image.

    1. Binary image - pixel can take value 0 or 1, only have two colors either black or white.

    2. Gray scale image - pixel can take values 0,1,2…..,255 in 8-bit format

    3. RGB scale - pixel can take three 8-bit values for Red, Green and Blue color.

       C[C] = R[R] + G[G] + B[B] (each color is a combination of red, green and blue color)

- A **gray scale** image will be stored in a **2d tensor** whereas a **RGB** image will be stored in a **3d tensor** with each layer for red, green and blue.
- **Hyperspectral image processing**

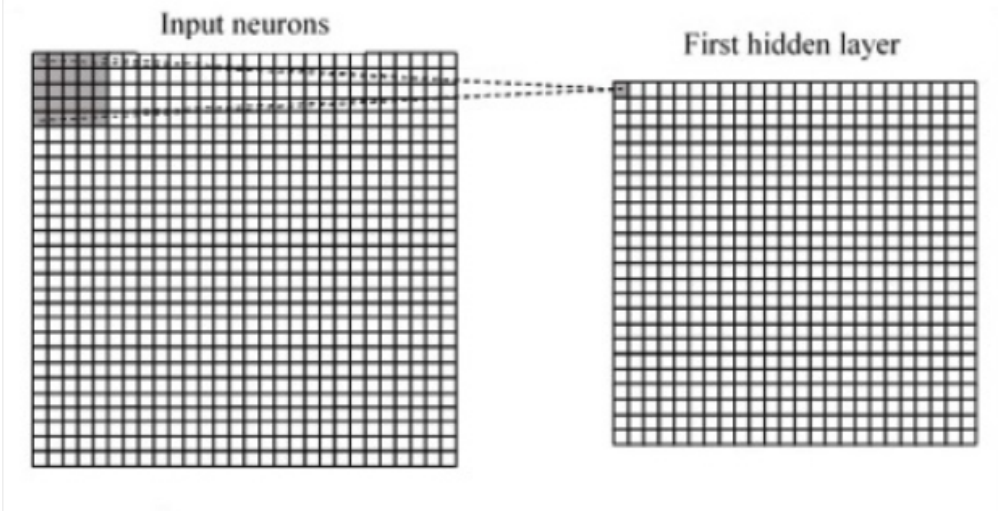
  Hyperspectral image processing includes multiple bands across the electromagnetic spectrum. RGB 3d tensor has three 2-d slices whereas hyperspectral images contain more than three 2-d slices hence it is more informative specially used in satellites for earth observation and collecting images of the cosmos.
- **A question arises which one is better: RGB, Gray scale or hyperspectral image.**

  It does not have a straight answer as we know that RGB or multiple band image processing is more informative but simultaneously computationally extensive. Whereas gray scales are less informative but computationally efficient. So, all have some pros and cons. So basically the choice of representation depends fully upon the application. If we consider earth imaging then multiple band model(3d tensor) would be the best, same for infrared sensing which also needs multiple bands to classify.

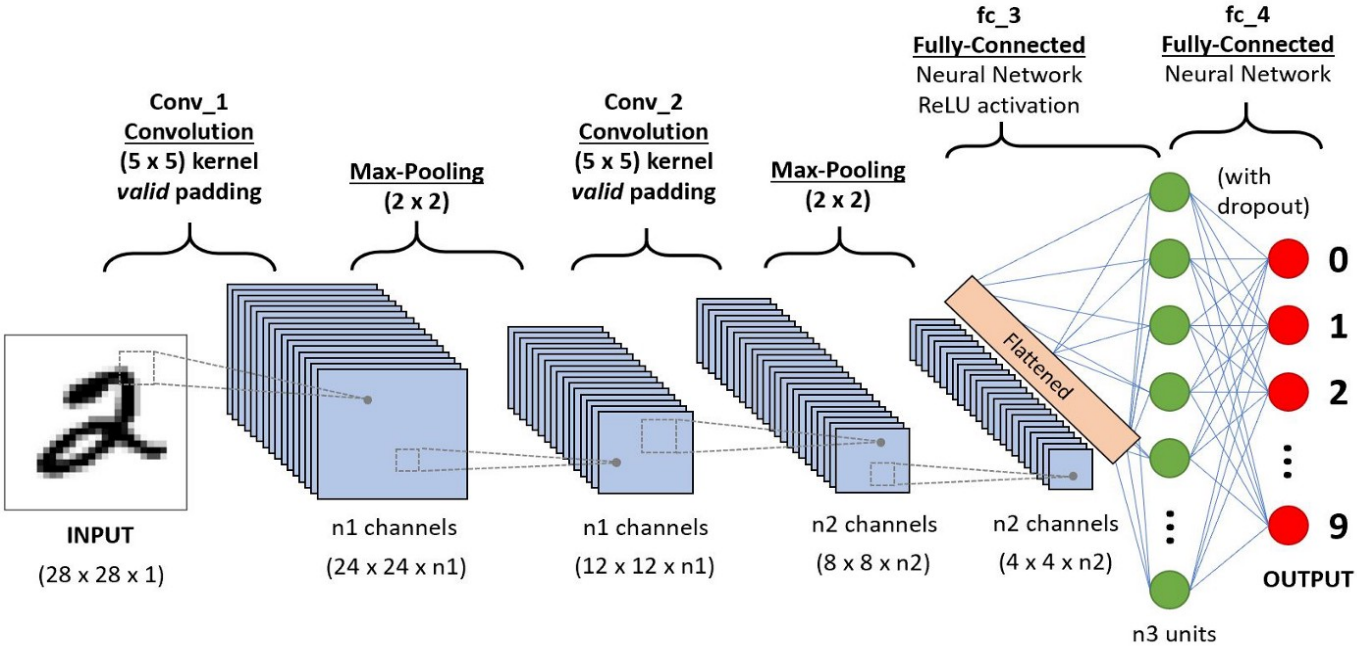
  However in the case of writing recognition software which is used to identify handwritten text which can efficiently be achieved by using Gray scale which is 2d tensor. Normal face id now found in electronic devices can be achieved by gray or RGB scale no need for more bands.
- **How to extract features from an image?**

  CNN(convolutional neural network) is the best one to extract features from an image, applying the weight at each layer and reducing the dimension of the matrix.

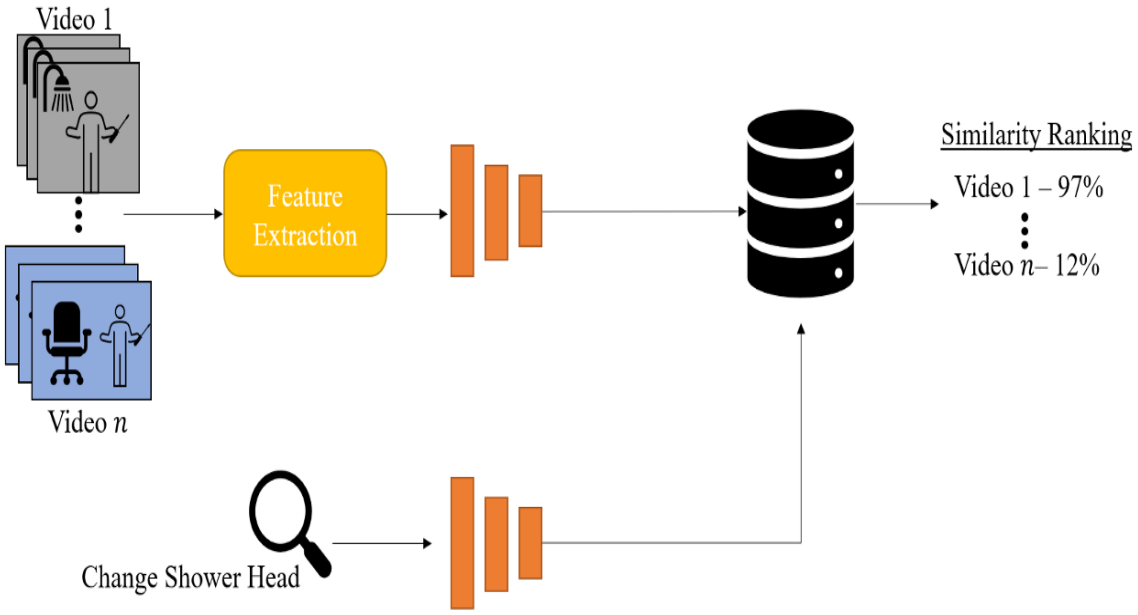
- **How do CNN reduce the dimension?**



Whole process of CNN



VIDEO

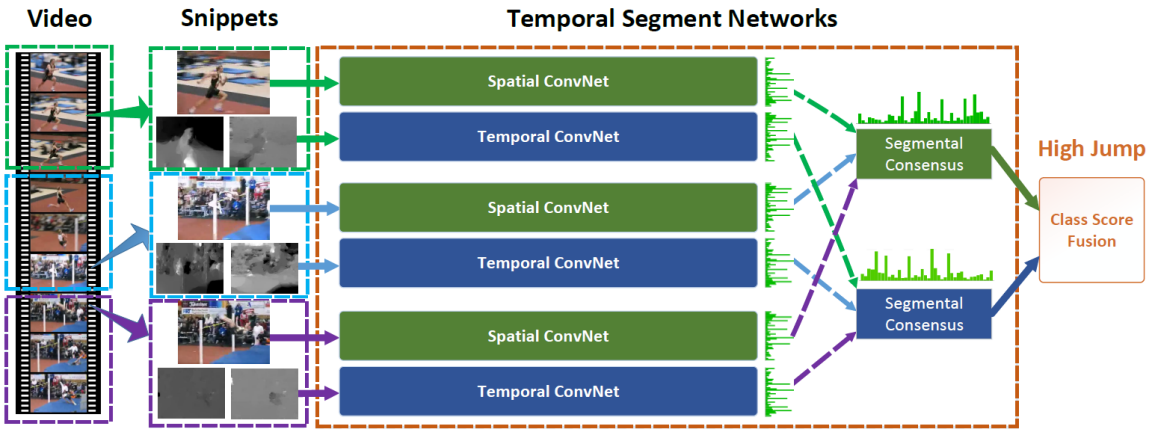


How do we define a video?

Video can be defined as a collection of images which in turn are 2-d arrays of pixels(in any format). Hence video classification is the same as image classification, where extracting feature from video is equivalent to extract feature from each image of the video with the help of CNN.
One way of processing over a video is to process over every frame of the video but this method is not feasible for large video files So, to solve this problem a method is very popular called Temporal segment network which in place of processing every frame firstly divide the video into finite equal interval segments then select random frames from them.

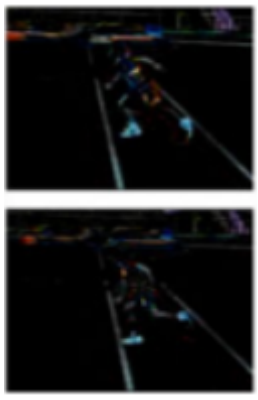
Temporal Segment Network

It is based on segment based sampling for action understanding in videos. Instead of working on a single framer or a short frame stack, temporal segment networks operate on a sequence of short **snippets** sampled from the entire video. These snippets represent the whole video and also have reasonable computational cost. Each snippet has a prediction of its own and there is a **consensus function** which aggregates all predictions from snippets into a video level prediction.

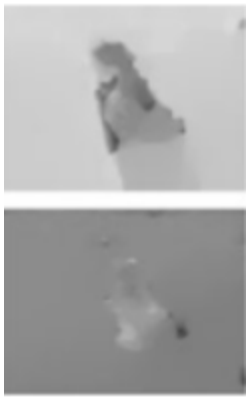


- A video V is divided into equal duration k segments {S1,S2,....,Sk}.
- One snippet Tk from its corresponding segment Sk. Each snippet can have a duration of 1 RGB frame or 5 Optical Flow frames and 5 RGB difference frames.
- Two extra formats(other than RGB image and optical flow field) to store a snippet are:
    - Warped optical flow
    - Stacked RGB Difference

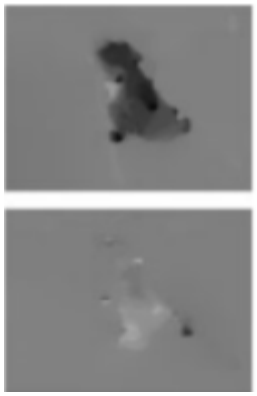
RGB



RGB Difference



Optical flow



Wrapped optical flow

- TSN models a sequence of snippets {T1,T2,……..,Tk}
    - TSN(T1,T2,.....,Tk) = H(G(F(T1;W), F(T2;W),.....,F(Tk;W)))
    - F = form prediction for each snippet
    - G = calculate consensus score for all snippets
    - H = calculate probability for each action class based on consensus score from G.
- The consensus function is an aggregate function. There are five types of aggregate functions:
    - Max-pooling
    - Average pooling
    - top-K pooling
    - Weighted average
    - Attention weighting

We observed that including RGB differences in the dataset increased the accuracy of the model. Similarly, including optical flow data, which is hard to compute, further increased the model performance.

TEXT

Information extraction in case of text is to find relationships between phrases and word and their frequency among different categories of text document. There are many methods in order to do the same.

Text Preprocessing

- Additional spaces, hyperlinks, punctuations, special characters, numbers are removed.
- Text is lowercased and stop words and non-english words are removed. Further, lemmatization with the help of position tags is performed.

N-gram Language model

N-gram is a sequence of N tokens or words. For a sentence "*Birla Institute of Technology and Science was incepted as an Institute with Dr. G.D. Birla as Founder Chairman.*"

- **Unigram** => "*Birla", "Institute", "of", "Technology", "and", "Science", "was", "incepted", "as", "an", "Institute", "with", "Dr.", "G.D",. "Birla", "as", "Founder", "Chairman"*.
- **Bigram** => *"Birla Institute", "Institute of", "of Technology" or "technology and"*.
- **Trigram** => *"Birla Institute of", "Institute of Technology", "of Technology and" or "Technology and Science"*.

How does the N-gram model work?

The N-gram model works on the principle of probability of finding a word based on the previous words in the sentence. In the above example the probability of occurrence of *"incepted"* depends on the probability of occurrence of *"Birla Institute of Technology and Science was"*.

Chain rule of probability

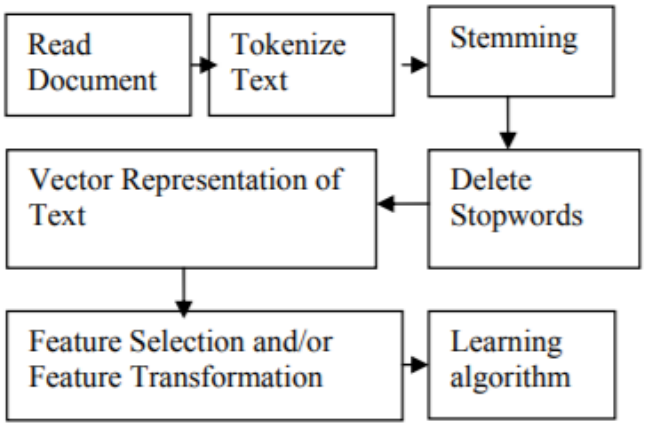
P(w1…...wn) = P(w1) . P(w2|w1) . P(w3|w1 w2) . P(w4|w1 w2 w3) …….P(wn|w1 w2…...wn-1)

One problem that arises is the cost of computation. In the above chain rule, the cost of computation is not feasible for big data. So in order to solve this problem we will use an assumption called Markov assumption which will reduce computation cost to a great extent.

Simplification assumption(Markov assumption)

P(wk | w1….wk-1) = P(wk | wk-1)

Classification



AUDIO

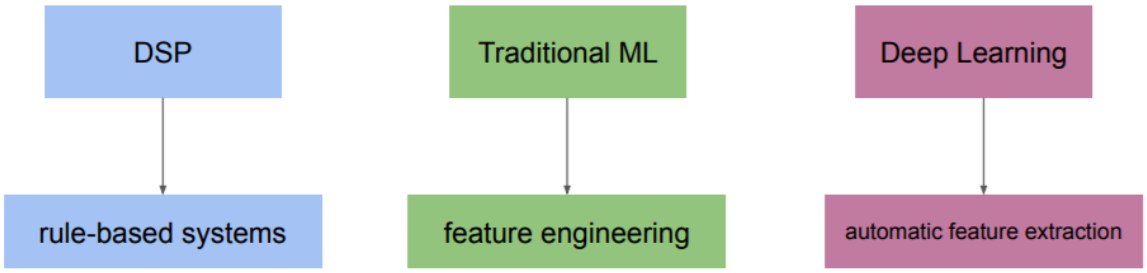
Importance of Audio features

- Description of sound
- Different features capture different aspects of sound
- Build intelligent systems

Audio features categorisation

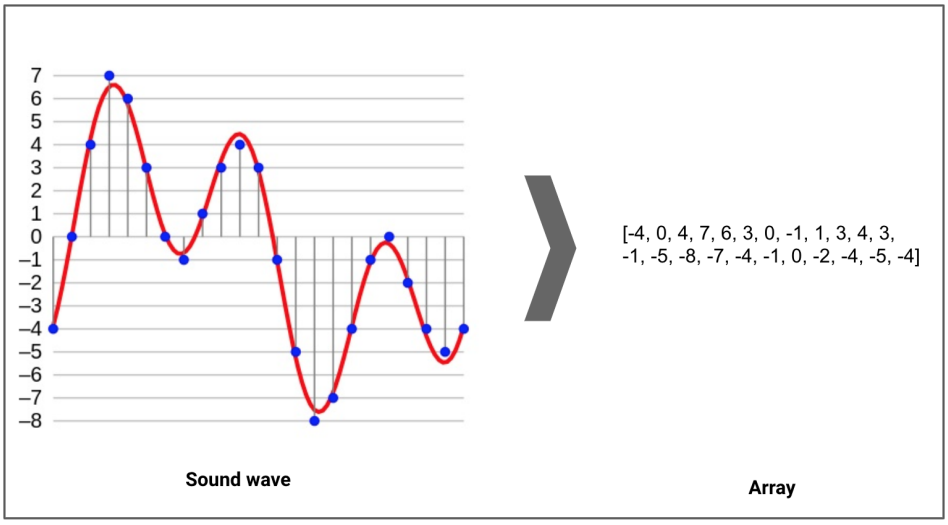
- Level of Abstraction
    - High level (instruments, notes, melody, genre)
    - Mid level (pitch and beat related patterns, fluctuations)
    - Low level (amplitude envelope, energy, zero crossing rate)
- Temporal scope
    - Instantaneous (~50ms)
    - Segment level (seconds)
    - Global
- Music Aspect
    - Beat
    - Timbre
    - Pitch
    - Harmony
- Signal Domain
    - Time domain
    - Frequency domain
    - Time frequency representation
- Machine Learning approach
    - Traditional machine learning (amplitude envelope, zero crossing rate, spectral flux)
    - Deep learning

Types of intelligent audio systems



What is Sampling and Sampling frequency?

In signal processing, sampling is the reduction of a continuous signal into a series of discrete values. The sampling frequency or rate is the number of samples taken over some fixed amount of time. A high sampling frequency results in less information loss but higher computational expense, and low sampling frequencies have higher information loss but are fast and cheap to compute.



Implementation Details

Librosa

It is a Python module to analyze audio signals in general but geared more towards music. It includes the nuts and bolts to build a MIR(Music information retrieval) system.

Spectrogram

A spectrogram is a visual way of representing the signal strength, or “loudness”, of a signal over time at various frequencies present in a particular waveform. Not only can one see whether there is more or less energy at, for example, 2 Hz vs 10 Hz, but one can also see how energy levels vary over time.

A spectrogram is usually depicted as a heat map, i.e., as an image with the intensity shown by varying the color or brightness.

Feature extraction from Audio signal

Every audio signal consists of many features. However, we must extract the characteristics that are relevant to the problem we are trying to solve. The process of extracting features to use them for analysis is called feature extraction. Let us study a few of the features in detail.

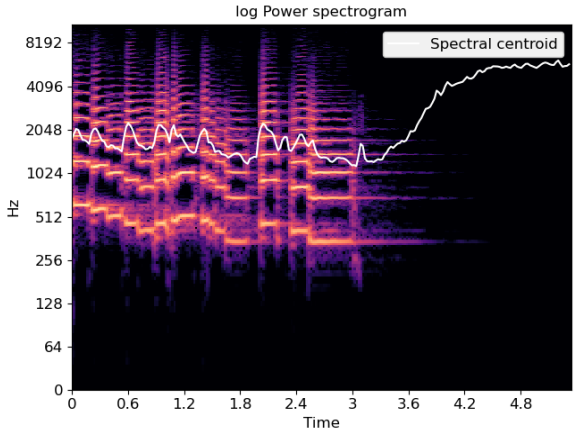
The spectral features (frequency-based features), which are obtained by converting the time-based signal into the frequency domain using the Fourier Transform, like fundamental frequency, frequency components, spectral centroid, spectral flux, spectral density, spectral roll-off, etc.

1. Spectral Centroid

The spectral centroid indicates at which frequency the energy of a spectrum is centered upon or in other words It indicates where the ” center of mass” for a sound is located. This is like a weighted mean:

$$f_c = \frac{\sum_k S(k) f(k)}{\sum_k S(k)}$$

where S(k) is the spectral magnitude at frequency bin k, f(k) is the frequency at bin k.



Spectral Centroid plotted using a Librosa function

2. Spectral Rolloff

It is a measure of the shape of the signal. It represents the frequency at which high frequencies decline to 0. To obtain it, we have to calculate the fraction of bins in the power spectrum where 85% of its power is at lower frequencies.

3. Spectral Bandwidth

The spectral bandwidth is defined as the width of the band of light at one-half the peak maximum (or full width at half maximum [FWHM]) and is represented by the two vertical red lines and λSB on the wavelength axis.

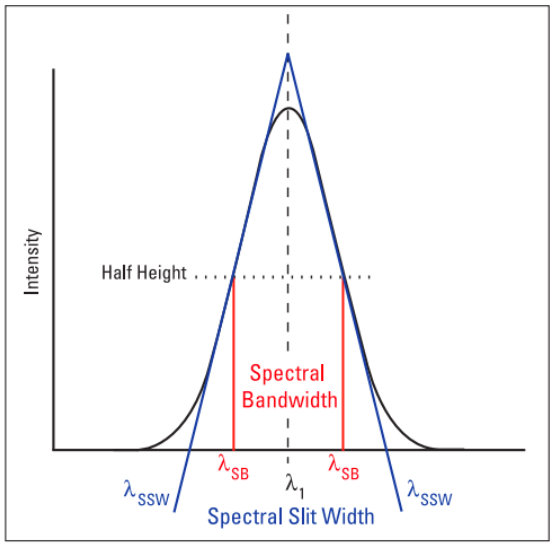


Figure 1: Gaussian intensity distribution of wavelengths emerging from the monochromator. The spectral bandwidth is defined by the red boundaries and λ_{SB}. The spectral slit width is depicted by the blue boundaries and λ_{SSW}.

4. Zero-Crossing Rate

A very simple way for measuring the smoothness of a signal is to calculate the number of zero-crossing within a segment of that signal. A voice signal oscillates slowly — for example, a 100 Hz signal will cross zero 100 per second — whereas an unvoiced fricative can have 3000 zero crossings per second.

$$zcr = \frac{1}{T-1}\sum_{t=1}^{T-1}\mathbb{I}\{s_t s_{t-1} < 0\}$$

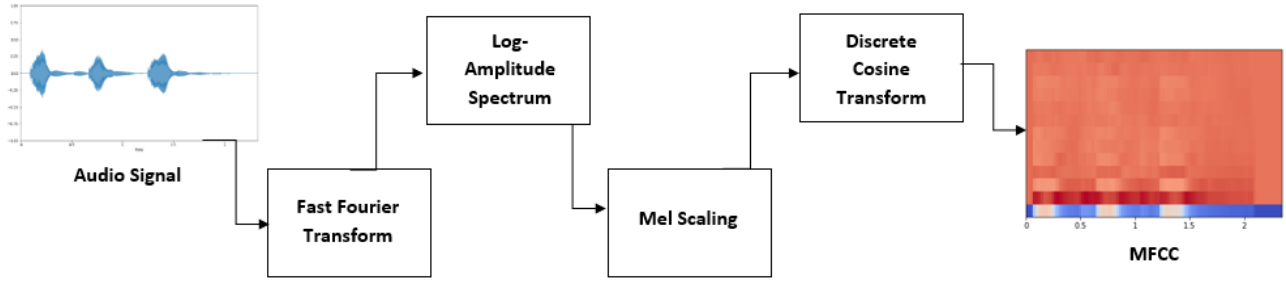
Fig. 4. Formula to calculate the Zero Crossing Rate

$$s_t \text{ is the signal of length } t$$
$$II\{X\} \text{ is the indicator function } (=1 \text{ if } X \text{ true, else } =0)$$

It usually has higher values for highly percussive sounds like those in metal and rock. Now let us visualize it and see how we calculate zero crossing rate.

5. Mel-Frequency Cepstral Coefficients(MFCCs)

The Mel frequency cepstral coefficients (MFCCs) of a signal are a small set of features (usually about 10–20) which concisely describe the overall shape of a spectral envelope. It models the characteristics of the human voice.



Steps to extract MFCCs from an audio signal.

6. Chroma feature

A chroma feature or vector is typically a 12-element feature vector indicating how much energy of each pitch class, {C, C#, D, D#, E, …, B}, is present in the signal. In short, It provides a robust way to describe a similarity measure between music pieces.

Comparison of Traditional Machine Learning and Deep Learning approaches

Traditional Machine Learning approach considers all or most of the features from both time and frequency domain as inputs into the model. Features need to be hand-picked based on its effect on model performance. Some widely used features include Amplitude Envelope, Zero-Crossing Rate (ZCR), Root Mean Square (RMS) Energy, Spectral Centroid, Band Energy Ratio, and Spectral Bandwidth.

Deep Learning approach considers unstructured audio representations such as the spectrogram or MFCCs. It extracts the patterns on its own. By late 2010s, this became the preferred approach since feature extraction is automatic. It's also supported by the abundance of data and computation power.
Commonly used features or representations that are directly fed into neural network architectures are spectrograms, mel-spectrograms, and Mel-Frequency Cepstral Coefficients (MFCCs).

REFERENCES

1. https://neptune.ai/blog/what-image-processing-techniques-are-actually-used-in-the-ml-industry
2. https://www.analyticsvidhya.com/blog/2019/01/build-image-classification-model-10-minutes/
3. https://www.quora.com/What-kind-of-data-structures-could-be-used-in-an-image-processing-project
4. https://www.mathworks.com/matlabcentral/answers/242350-pixel-is-a-2d-or-3d-please-answer-me-thanks
5. Sangwine, Stephen J., and Robin EN Horne, eds. *The colour image processing handbook*. Springer Science & Business Media, 2012.
6. https://www.analyticsvidhya.com/blog/2019/09/step-by-step-deep-learning-tutorial-video-classification-python/
7. https://towardsdatascience.com/image-classification-in-10-minutes-with-mnist-dataset-54c35b77a38d
8. https://www.analyticsvidhya.com/blog/2019/08/3-techniques-extract-features-from-image-data-machine-learning-python/
9. https://towardsdatascience.com/image-feature-extraction-traditional-and-deep-learning-techniques-ccc059195d04
10. https://towardsdatascience.com/popular-downstream-tasks-for-video-representation-learning-8edbd8dc19c1
11. https://sh-tsang.medium.com/review-tsn-temporal-segment-network-video-classification-16a2819462f5
12. Wang, Limin, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. "Temporal segment networks for action recognition in videos." *IEEE transactions on pattern analysis and machine intelligence* 41, no. 11 (2018): 2740-2755.
13. https://towardsdatascience.com/how-to-turn-text-into-features-478b57632e99
14. https://towardsdatascience.com/machine-learning-text-processing-1d5a2d638958
15. Ikonomakis, M., Sotiris Kotsiantis, and V. Tampakas. "Text classification using machine learning techniques." *WSEAS transactions on computers* 4, no. 8 (2005): 966-974.
16. https://www.youtube.com/watch?v=ZZ9u1vUtcIA
17. https://www.youtube.com/watch?v=8A-W1xk7qs8&list=RDCMUCZPFjMe1uRSirmSpznqvJfQ&index=3
18. https://www.kdnuggets.com/2020/02/audio-data-analysis-deep-learning-python-part-1.html