Foundations of Data Science - CS F320

A REPORT
ON

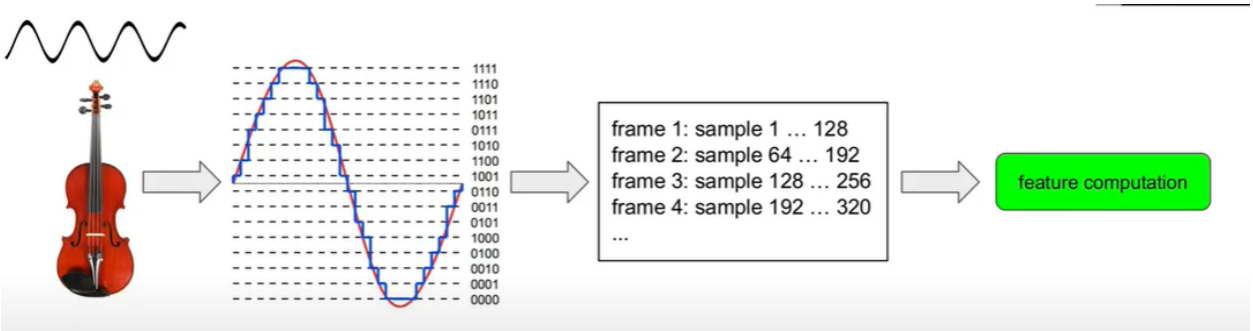
Feature Extraction: Image, Video, Audio and Text Data

By

Uday Mittal (2019B4A70662P)
Anubhav Srivastava (2018A8PS0030P)

Audio features

Time Domain Feature extraction:



The analog signal is passed through an ADC which samples and the signal to stored digitally. The data is converted to frames through a process called framing. A frame represents a duration of signal which is perceivable by human. Then we can extract time domain features from this data.

Amplitude Envelope

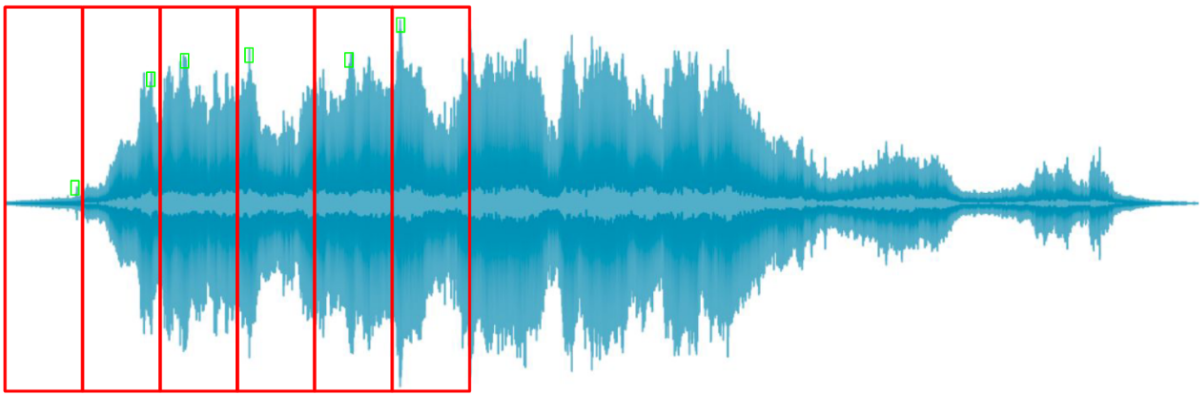
Amplitude envelope is maximum value of all samples in a frame. It gives rough idea of loudness. However, it is sensitive to outliers. It is used for onset detection i.e. when a note starts.

Mathematically:

$$AE_t = \max_{k=t\cdot K}^{(t+1)\cdot K-1} s(k)$$

Where, AEt is amplitude envelope at tth frame,
s(k) is amplitude of kth sample,
K is the frame size.

On a waveform:



Red boxes represents the different frames. Green box represents the amplitude envelope in that frame.

Root Mean Square Energy

Root Mean Square Energy is the RMS value of all samples in that frame. It is an indicator of loudness. Comparing to AE, RMS energy is less sensitive to outliers because it takes more data into account. It is used for audio segmentation i.e. identifying new segments in an audio.

Zero crossing rate

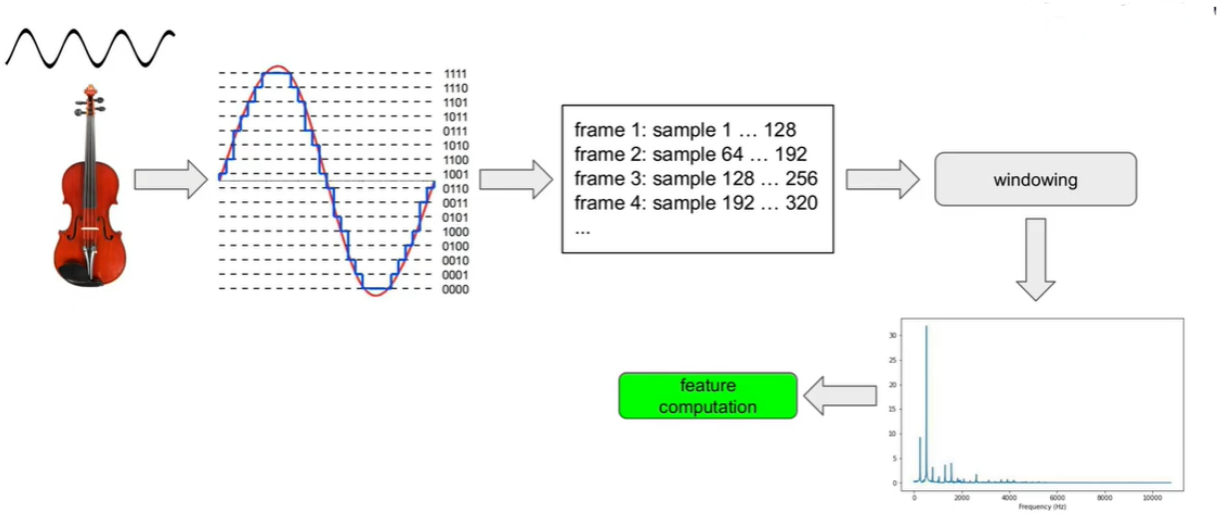
Zero crossing rate is the number of times a signal crosses horizontal axis. It is used for classification of percussion sounds vs pitched sounds. Percussion sounds tend to have a higher number of zero crossings.

Mathematically in t th frame :

$$ZCR_t = \frac{1}{2} \cdot \sum_{k=t \cdot K}^{(t+1) \cdot K - 1} | \, sgn(s(k)) - sgn(s(k+1)) \, |$$

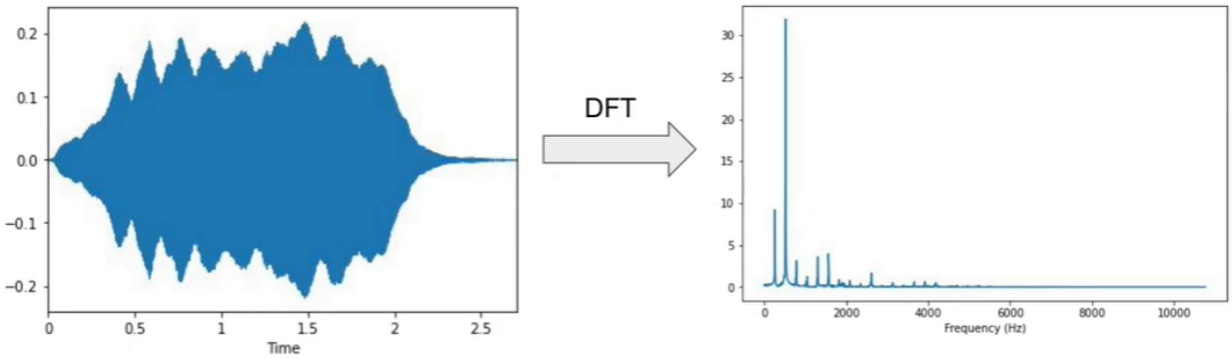
Here, we are comparing amplitudes at consecutive samples. For every zero crossing we get ZCR = 1.

Frequency Domain feature extraction:



The process is same upto framing in time domain as well as in frequency domain. To remove high frequency components which are not part of time domain representation, we apply a windowing function. Then to convert it into frequency domain, we apply fourier transform to get spectrum. Now, we can extract frequency domain features from this data.

Discrete Fourier Transform:



A Discrete Fourier Transform on an audio signal gives the frequency components present in that audio signal for the whole duration.
DFT gives:

- Frequency bins
- Fourier coefficients corresponding to frequencies in the audio signal

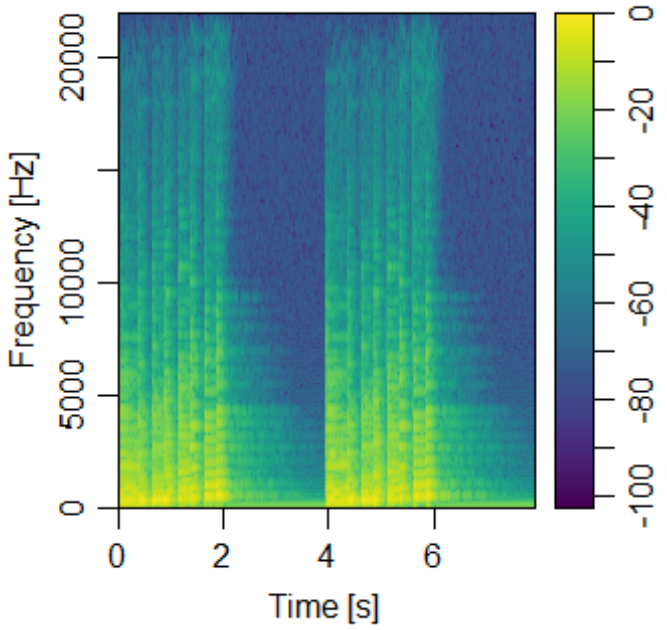
Short Time Fourier Transform:

A key information missing from DFT is about different frequencies present at different points of time in an audio signal. A Short Time Fourier Transform describes the frequency components present in the audio signal at different intervals of time. It is obtained by applying DFT to each frame of the signal.
STFT gives:

- 2D matrix - (frequency bins,frames)
- Fourier coefficients corresponding to each frequency and each frame

Spectrogram

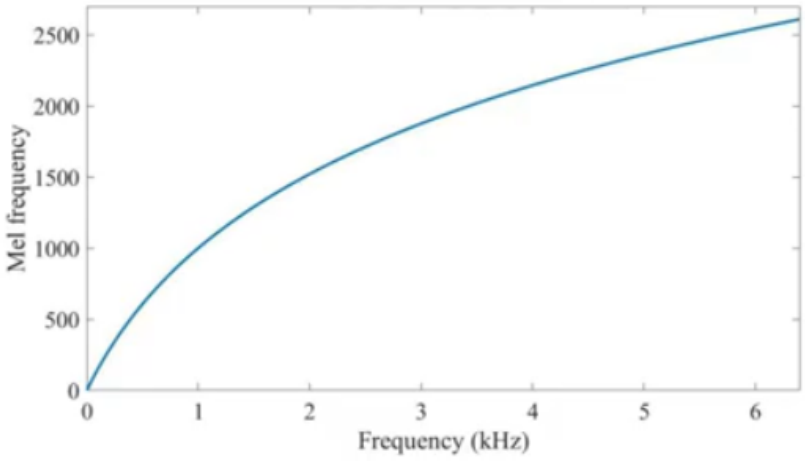
We obtain a spectrogram by applying STFT to audio signal. Below is a spectrogram plotted for audio of notes played on a piano.



We can see different frequency components at different times. This gives a very good visual representation of an audio signal. We can feed this spectrogram as an image to a ML algorithm for classification.

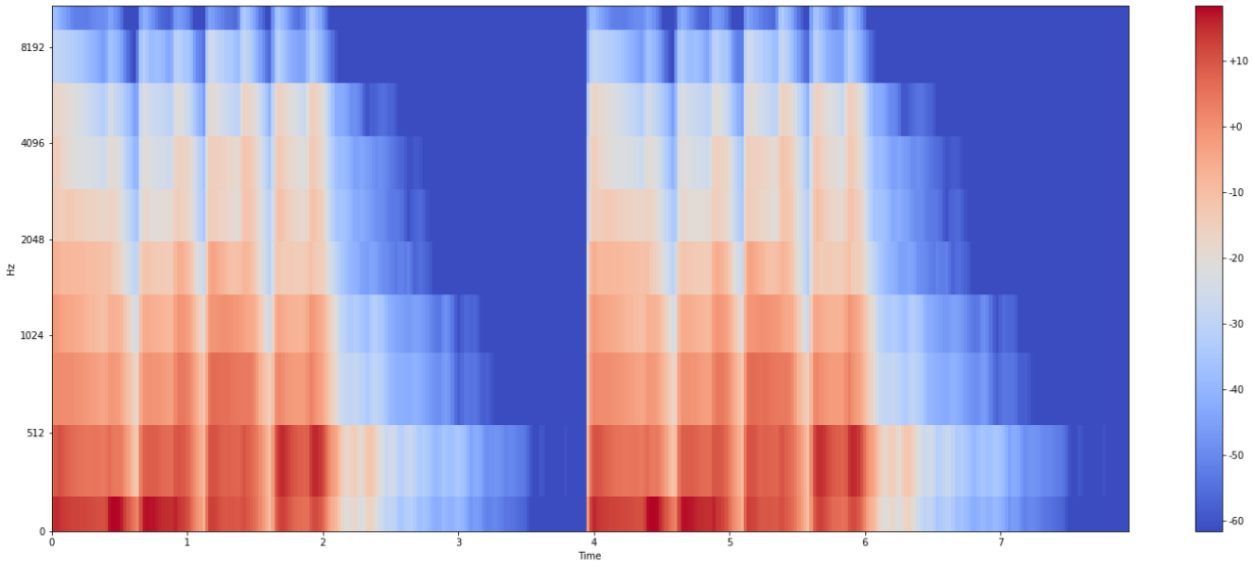
Mel-Spectrogram

We conduct two simple experiments, first we hear frequencies from 100 Hz to 200 Hz, then, we hear frequencies from 1500 Hz to 1700 Hz. We find that frequencies in the second experiment are perceived more closely related to each other than in the first experiment even though difference in frequencies in both is 200 Hz. This shows that humans perceive frequencies in a logarithmic scale and not on a linear scale.
Hence, we arrive at Mel-scale:



$$m = 2595 \cdot log(1 + \frac{f}{500})$$

Mel-scale is a logarithmic scale. Equal distance on the scale have same perceptual distance.
A mel-spectrogram is a spectrogram where frequencies are converted to mel-scale.
Below is the mel-spectrogram for audio of notes played on a piano.

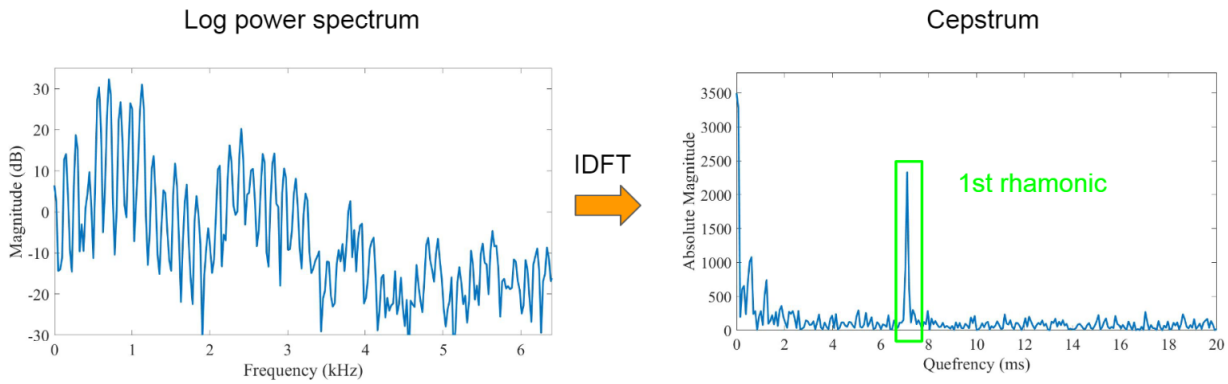


MFCC

MFCC is a feature of choice for seismic signals, audio recognition and music processing. MFCC is obtained by computing the cepstrum after mel-scaling of the log-amplitude spectrum. It represents distinct unit of sound analogous to the shape of vocal tract.
Cepstrum contains the information of rate of change of spectral bands. It can be computed as:

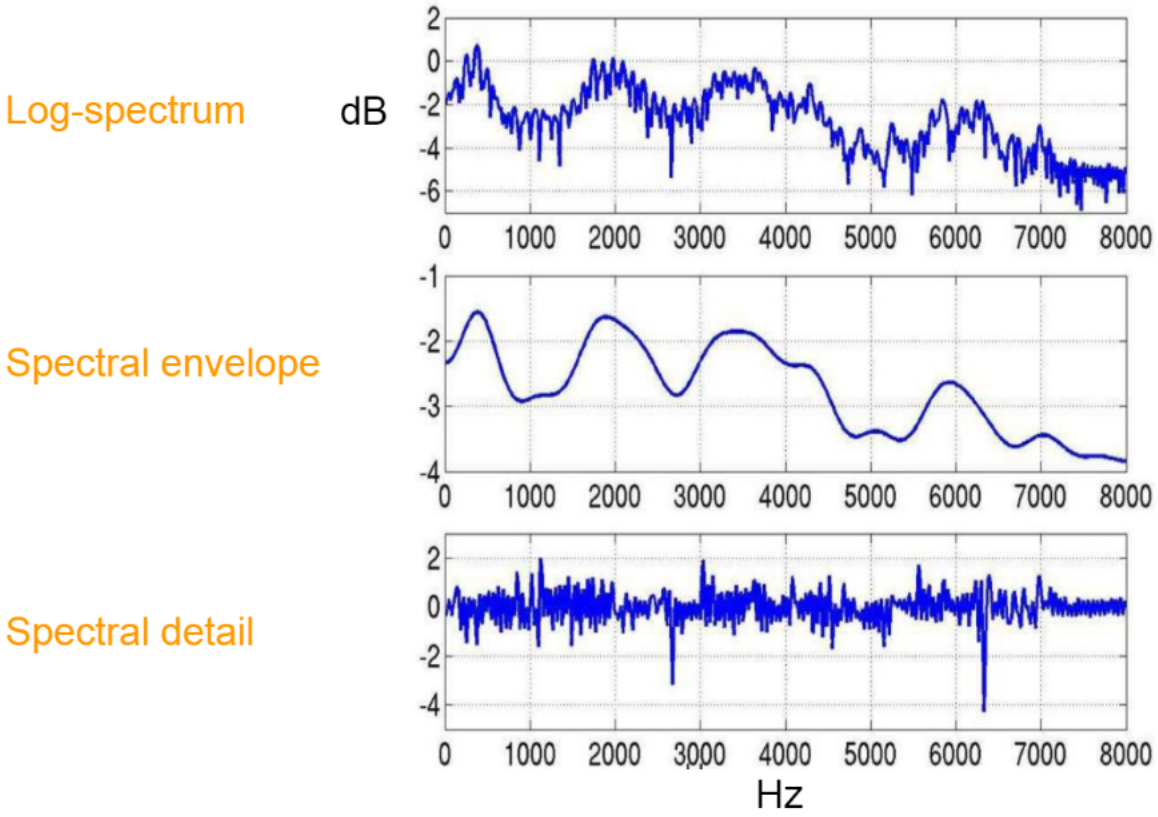
$$C(x(t)) = F^{-1}[log(F[x(t)])]$$

Where x(t) is the time domain signal,
F(x(t)) is the fourier transform of audio signal,
F^-1 is the inverse fourier transform.
Hence, a cepstrum of time domain signal is the inverse fourier transform of log spectrum.



1st rhamonic is the frequency associated with fundamental frequency in the original audio signal.

Log spectrum is composed of spectral envelope and spectral detail.



Spectral envelope peaks are called formants. Formants are a very important feature as they identify a sound. This is analogous to the way our vocal tract compresses/expands to produce a specific sound.

We are not interested in spectral detail because they do not contain much information.

In moving from frequency domain to quefrency domain (by IDFT) we get a physical separation of spectral envelope and spectral detail. Quefrency associated with spectral envelope is very less than quefrency associated with spectral detail. Usually a low pass filter is applied (called liftering) to remove higher frequencies (due to spectral detail).

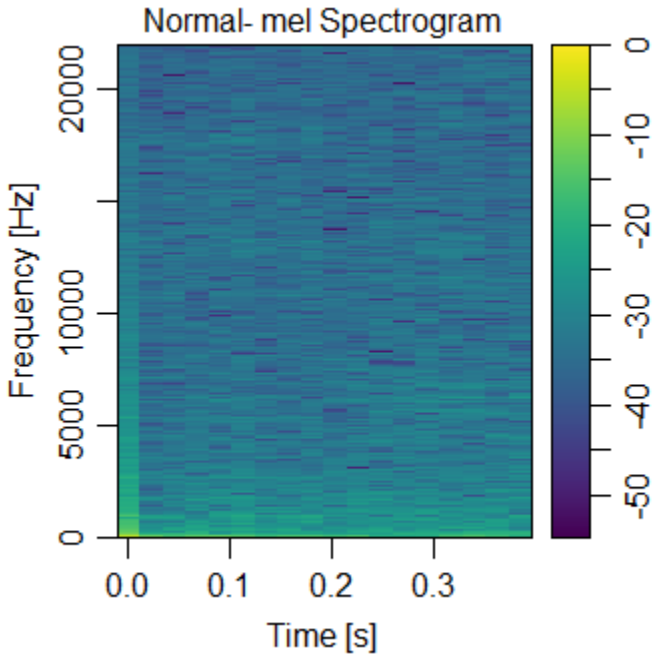
MFCC advantages:

- Describes a large structure of spectrum
- Ignores the fine spectral structures
- Works well in speech and music processing

MFCC disadvantages:

- Not robust to noise
- Biased for human perception
- Not efficient for synthesis of audio

Below is the spectrogram with MFCC function for notes on piano audio signal:



Text

Stemming

Is the process of reducing words to their word stem(root word). It is used in sentiment analysis/classification.

Example: (history,historical)--->histori , (finally,finale,finalized) ----> finc

Lemmatization

Is the process of reducing words to a meaningful word instead of root word.

Lemmatization is more computationally expensive compared to stemming. It is used in chat bots where output has to be perceived by a human.

Example: (history,historical)--->history , (finally,finale,finalized) ----> final

Bag of Words (BOW)

We can understand BOW representation with a simple example. Say, we have 3 sentences:

Sentence 1: He is a good boy.

Sentence 2: She is a good girl.

Sentence 3: Boy and girl are good.

After removing shortwords (words which do not contribute meaning to sentences like isn't,aren't etc.):

Sentence 1: good boy

Sentence 2: good girl

Sentence 3: boy and girl good

This can be represented as a dataset with 3 features:

Sentence No.	good	boy	girl
1	1	1	0
2	1	0	1
3	1	1	1

In another similar example for a paragraph of 31 sentences, we get:

sentences - List (31 elements)

Inde	Type	Size	Value
0	str	31	I have three visions for India.
1	str	143	In 3000 years of our history, people from all over the ...
2	str	182	From Alexander onwards, the Greeks, the Turks, the Moguls, the Portugu ...
3	str	46	Yet we have not done this to any other nation.
4	str	29	We have not conquered anyone.
5	str	122	We have not grabbed their land, their culture, their h ...
6	str	4	Why?
7	str	104	Because we respect the freedom of others.That is why my ...
8	str	115	I believe that India got its first vision of this in 1 ...
9	str	80	It is this freedom that we must protect and nurture and ...
10	str	43	If we are not free, no one will respect us.
11	str	41	My second vision for India's development.

corpus - List (31 elements)

Inde	Type	Size	Value
0	str	18	three vision india
1	str	69	year history people world come invaded u captured land conquered mind
2	str	85	alexander onwards greek turk mogul portuguese british french dutch cam ...
3	str	15	yet done nation
4	str	16	conquered anyone
5	str	51	grabbed land culture history tried enforce way life
6	str	0	
7	str	43	respect freedom others first vision freedom
8	str	55	believe india got first vision started war independence
9	str	34	freedom must protect nurture build
10	str	18	free one respect u
11	str	31	second vision india development

(detailed code in code file)
'Corpus' variable is obtained after lemmatization and removing of stop words from 'sentences' variable. The document matrix obtained from corpus:

X - NumPy object array

	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0

'X' is the document matrix obtained using BOW representation. It has 31 rows (as there are 31 sentences) and 114 columns (as there are 114 unique words).
Disadvantages of BOW:

- We loose word order
- No weightage to words, so don't know which words are more important

Term Frequency and Inverse Document Frequency (TF-IDF)
Term Frequency, TF = (Frequency of word in a sentence)/(Number of words in the sentence)
Inverse Document Frequency, IDF = log[(number of sentences)/(freq of word in all sentences)
We can take the same example we took in BOW after removing shortwords:

Sentence 1: good boy
Sentence 2: good girl
Sentence 3: boy and girl good
Now, we compute TF matrix:

	Sentence 1	Sentence 2	Sentence 3
good	1/2	1/2	1/3
boy	1/2	0	1/3
girl	0	1/2	1/3

Now, we compute IDF vector:

Words	IDF
good	log(3/3)=0
boy	log(3/2)
girl	log(3/2)

Multiplying both: IDF^T * TF =

Sentence No.	good	boy	girl
1	0	log(3/2)/2	0
2	0	0	log(3/2)/2
3	0	log(3/2)/3	log(3/2)/3

In TF-IDF importance is given to uncommon words.
Advantage of TF-IDF over BOW is that in TF-IDF we can attach more weightage to particular words. Here, ‘boy’ is given more weightage than ‘good’ in sentence 1 whereas both were given same weightage in BOW.
However, temporal order of words is not preserved.

Extracting Features from 2D Color Images

Features of an image can be extracted using various content of the image. Some of them are

1) Color
2) Texture
3) Shape
4) Position
5) Dominant Edges and Regions of Images

Extracting Features using its color:-

The technique is optimal and is very efficient.

1) Every image can be represented in the form of pixels. To illustrate the image in the state of pixels.
2) Any color can be formed by mixing Red, Green, and Blue (RGB) colors in a particular proportion.
3) Hence each pixel has a specific value of the RGB colors.
4) Represent each pixel in the ratio of the RGB colors in an array.
5) Take the average of the Red, Green, and Blue decomposition of all the pixels.
6) Represent them as R_{avg}, G_{avg}, and B_{avg}.
7) R_{avg} = , G_{avg} = , and B_{avg} = Where n represents the number of pixels in the image, and R(n), G(n), and B(n) denote the specific value of the RGB color of a pixel.
8) Using the averages of RGB colors, the data sets can be classified as per the various need of the problem.
9) In the code attached with the report, a sample is presented, with which the pixel values can be extracted from the images.
10) The data set is an extensive data set of fruit images.

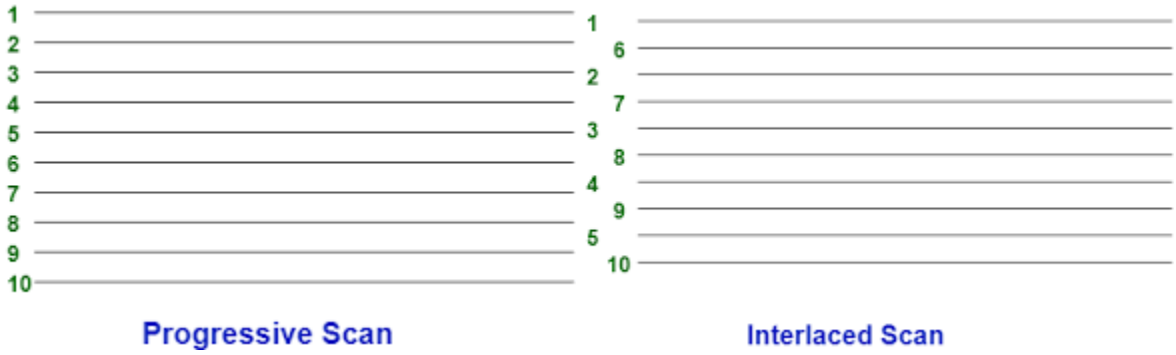
Video Processing

Definition of Video Signal – A video signal is any sequence of time-varying images. An image is a spatial distribution of intensities that remain constant with time, while a time-varying image has a spatial intensity distribution which varies with time. A video signal is a series of images, which are called frames.

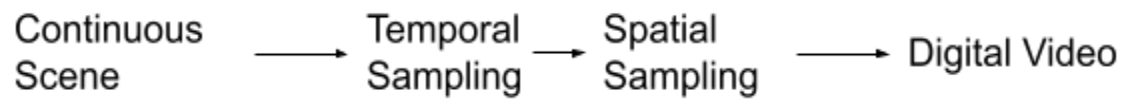
Analogue Video Signal – The most common consumer display mechanism still uses analog display devices such as Cathode-Ray Tube (CRT). Until all the terrestrial and satellite broadcasters become digital, analog video signals will remain significant. The there principal analog video signal formats are National Television Systems Committee (NTSC), Phase Alternate Line (PAL), and Sequential Color with Memory (SECAM). All three are video formats in which the information is captured by CRT and is scanned from left to right to create a sequence of intensity signal. The system takes advantage of human vision and allows sound movement reproduction in the scene at a relatively low field rate of 50 field/sec for PAL and SECAM and 60 field/sec for NTSC.

Progressive Scan patterns are used for high-resolution displays like computer CRT monitors, digital cinema projects, etc. In this technique, each frame of picture information is scanned thoroughly to create a video signal.

In an interlaced scan pattern, each picture's odd and even lines are read out in two separate scans of odd and even fields, respectively. This allows sound reproduction of movement in the scene at a relatively low field rate.



Digital Video – In a digital video, the picture information is digitized spatially and temporally, and the resultant pixel intensities are quantized.



Spatial Sampling – The sensitivity of the Human Visual System (HVS) varies according to the spatial frequency of an image. In the digital representation of the image, the value of each pixel needs to be quantized using some finite precision. In practice, 8 bits are used per luminous sample.

Temporal Sampling – A video consists of a sequence of images displayed in rapid succession to give an illusion of continuous motion. If the time gap between the successive frames is too large, the viewer will observe shaky motion. The sensitivity of HVS also drops off significantly at high frame rates. In most video formats, the temporal sampling rates of 24 frames/sec or above are used.

Video Formats – A digital video consists of video formats that are displayed at various prescribed video formats. For example – NTSC uses a frame rate of 30 frames/sec. The format of the frame specifies the size of individual frames in terms of pixels.

Format	***Luminance Pixel Resolution***	***Application***
Sub – QCIF	128 x 96	Mobile Multimedia
QCIF	176 x 144	Video conferencing
CIF	352 x 288	Video Conferencing
4CIF	704 X 576	Standard Definition Television (SDTV)
16CIF	1408 x 1152	High Definition Television (HDTV)

Frame Type – There are three types of video frames – I-frame, P-frame, and B-frame. 'I' stands for Intra – coded frame, 'P' stands for Predictive structure, and 'B' stands for Bidirectional predictive frame.

Video Processing – the different areas of video processing includes –

1) Video Compression

2) Video Indexing

3) Video Segmentation

4) Video Tracking, etc.

Video Indexing – It is a process of providing watchers with a way to access and navigate the contents easily. A set of representative vital frames is selected to create an efficient index that captures and encapsulates the entire video content.

Subsampling – The basic concept is to reduce the dimension of the input video and thus the number of pixels to be coded before the encoding process. On the receiver side, the decoded images are interpolated for display. This concept is the most elementary technique, which uses specific characteristics of the human eye. This removes subjective redundancy contained in the video data. RGB format is not preferred as R, G, B components are correlated, and transmitting R, G, B components separately is redundant. Instead of this, the image is divided into YUV components (one luminance and two chrominance components).

Video Compression plays a critical role in many digital video applications such as digital libraries, video on demand, and HDTV. Consider a video sequence with a frame size of 176 x 144 pixels at 30 frames per second, and 24 bits per pixel, which would require 18.25 Mbps data. It is impractical to transmit the video sequence over standard telephone lines, where data rates are usually high. Effective video compression could be achieved by minimizing both spatial and temporal redundancy. To compress the video for efficient transmission and storage, temporal redundancy should be used. Spatial redundancy means that neighboring pixels are identical, while Temporal redundancy implies that adjacent frames are similar. Removal of spatial redundancy is generally termed intraframe coding, and reduction of temporal redundancy is termed interframe coding. The spatial redundancy within a frame is minimized by using Discrete Cosine Transform technique. Video compression algorithms can be classified into two types

A) Lossless video compression

B) Lossy video compression

Subsampling – The basic concept is to reduce the dimension of the input video and thus the number of pixels to be coded before the encoding process. At the receiver's side, the decoded images are interpolated for display.

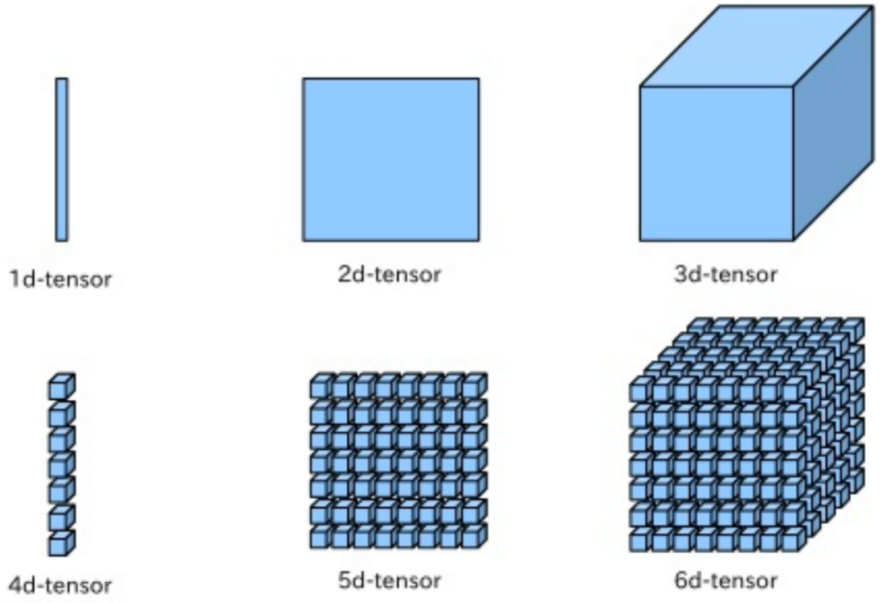
Tensors

Tensors are the primary data structures used by neural networks; the inputs outputs and transformations within neural networks are all represented using tensors.The concept of a tensor is a mathematical generalization of other more specific concepts like scalars, vectors and matrices.

with the number of indices required to access a specific element. There are 0 indices required for a number. When we move to an array or a vector we need one index to refer to a specific element and in a 2d array or a matrix we need two indices to refer to a specific element.

A tensor is characterized by order. A zeroth order tensor is a scalar, first order tensor is a vector, second order tensor is a matrix and nth order tensor is a n dimensional array. Order is equal to the number of indices required to access a specific element. There are 0 indices required for a number. When we move to an array or a vector we need one index to refer to a specific element and in a 2d array or a matrix we need two indices to refer to a specific element.

For different orders, tensors can visualised as:



Since, tensors are a generalisation of n dimensional matrices, it can be used when matrices are not enough. A tensor can represent a set of matrices.

Representation of different types of data as tensors:

Greyscale image ---> 2D tensor (height*width)

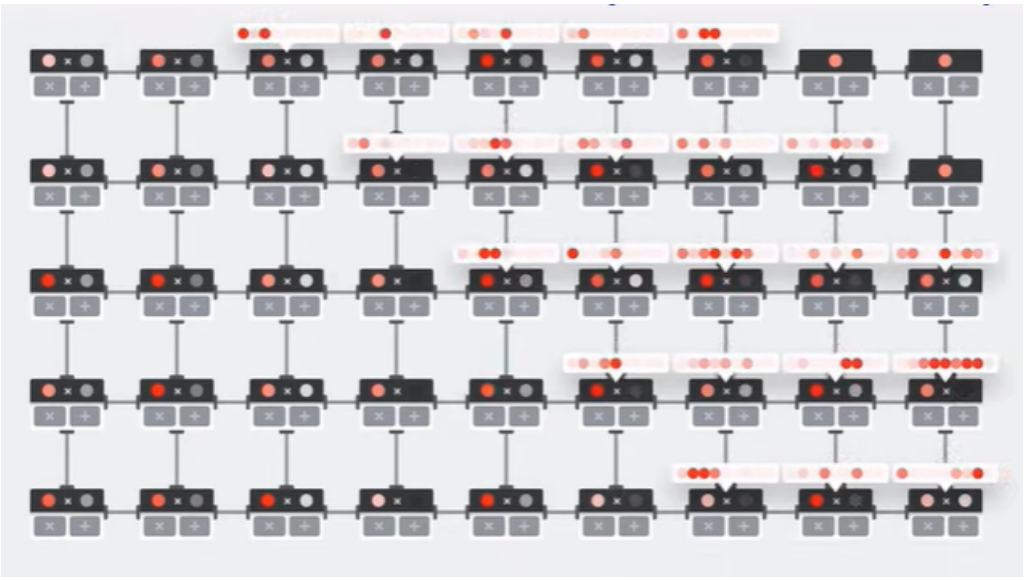
RGB image, Hyperspectral images ----> 3D tensor (height*width*band)

Colour Video ----> 4D tensor (height*width*band*time)

Hence, tensors help us in modelling different types of data.

Google TPU

Tensor Processing Unit or TPU is a custom built integrated circuit developed specifically for machine learning or deep learning applications. It handles matrix multiplication and addition in the form of tensors. As we can see both CPU and GPU read and write the intermediate result
on memory at every calculation so that they can support a wide variety of different algorithms as general purpose processes but this leads back to our fundamental problem of Vonmann bottleneck, ie. for every single calculation in the thousand of ALUs, GPUs/CPUs need to access registers or shared memory to read and store the intermediate calculation results which increases the throughput time.
So Google designed a domain specific architecture which uses a matrix processors that can handle matrix multiplications and additions for neural networks at blazingly fast speeds while consuming much less power and area.
It utilizes reduced precision, a matrix processor, and a minimal design to reduce overhead.
Reduced Precision & Minimal Design: The precision is reduced by using 8-bit integers instead of 32-bit floating point numbers. This reduces the floating point unit overhead size, hence more multipliers can be fit in the same space.
Matrix Processor: It uses a systolic array in order to do a large hard-wired matrix calculation without memory access. This reduce memory access bottleneck problem (as in CPUs/GPUs).



So, even though it has a low clock frequency, TPU is performing tons of operations every cycle.

Tensor Flow

Machine Learning is a complex discipline, and it becomes even more complicated when many complex models have to be implemented. In such cases, machine learning frameworks, such as Google's TensorFlow are used, that ease the process of acquiring the data, training models, serving predictions, and refining future results.

Google Brain Team developed TensorFlow for internal Google use in research and production. TensorFlow is an open-source library for numerical computation and large-scale machine learning. TensorFlow bundles a complex number of machine learning and deep learning models and algorithms and makes them helpful using a common platform. It uses Python to provide an easy and comfortable front-end Application Programming Interface (API) for building applications with the framework. The execution is done on C++ to get everything done in high-performance. Other languages in which tensors could be utilized are JavaScript and Java.

TensorFlow can train models and run deep neural networks for many daily life applications. They include –

1) Handwritten digit recognition

2) Image Recognition

3) Word Embeddings

4) Recurrent Neural Networks

5) Sequence to sequence models for machine translation

6) Natural Language Processing

7) Partial Differential Equations (PDE)

TensorFlow – TensorFlow allows the developers to create dataflow graphs, which are specialized structures that describe how data moves through the graph or a series of nodes. Each node in the graph represents a certain mathematical operation, in which each edge is a multidimensional array or a tensor.

TensorFlow uses Python language, which is easy to learn and understand. Nodes and tensors in TensorFlow are Python objects, and the applications of TensorFlow are applications of Python itself.

The mathematical computations are not done in Python but rather are done in C++. Python directs traffic between the pieces and provides high-level programming abstractions to hook them together.

Benefits of TensorFlow – The most significant benefit of TensorFlow for machine learning is an abstraction. Instead of dedicating resources to implementing algorithms, by using TensorFlow, the developer can focus more on the application's logic.