Mel-Frequency Cepstral Coefficients

March 14, 2025

1 Abstract

Automatic Speech Recognition (ASR) is a technology that enables machines to interpret and transcribe spoken language into text. It has widespread applications in voice assistants, transcription services, and human-computer interaction. A crucial step in ASR is feature extraction, where meaningful representations of the speech signal are derived to improve classification accuracy. Since raw audio signals contain vast amounts of redundant and irrelevant information, extracting discriminatory features helps to reduce dimensionality and improve recognition performance.

Various feature extraction techniques have been proposed for ASR and audio classification, with different success rates. Features can be derived from either the time-domain signal or a transformed frequency-domain representation, depending on the chosen analysis approach. Some of the most widely used audio features include Mel frequency cepstral coefficients (MFCC), linear predictive coding (LPC), and local discriminant bases (LDB). Among these, MFCC has emerged as one of the most popular and effective feature extraction methods due to its ability to mimic human auditory perception. By representing the short-term power spectrum of a speech signal on the Mel scale, MFCC enhances the robustness of ASR systems, improving accuracy in noisy environments and diverse speech conditions.

2 Why MFCC is Most Eqffective?

2.1 Mimics the Human Auditory System

The human ear perceives frequencies non-linearly, it means we are more sensitive to lower frequencies than higher ones. MFCC uses the Mel scale values, which space frequency bands logarithmically to align with human hearing. This makes MFCC more efficient at capturing speech-relevant information while ignoring irrelevant high-frequency noise.

2.2 Compact and Discriminative Representation of a Word.

Instead of dealing with raw waveforms, MFCC extracts only the essential information related to speech. The first 12-13 MFCC capture the speech spectral envelope, which contains phoneme-specific features. By ignoring fine details that do not contribute to speech recognition, MFCC provides a compact yet highly informative representation.

3 Steps to Compute MFCC

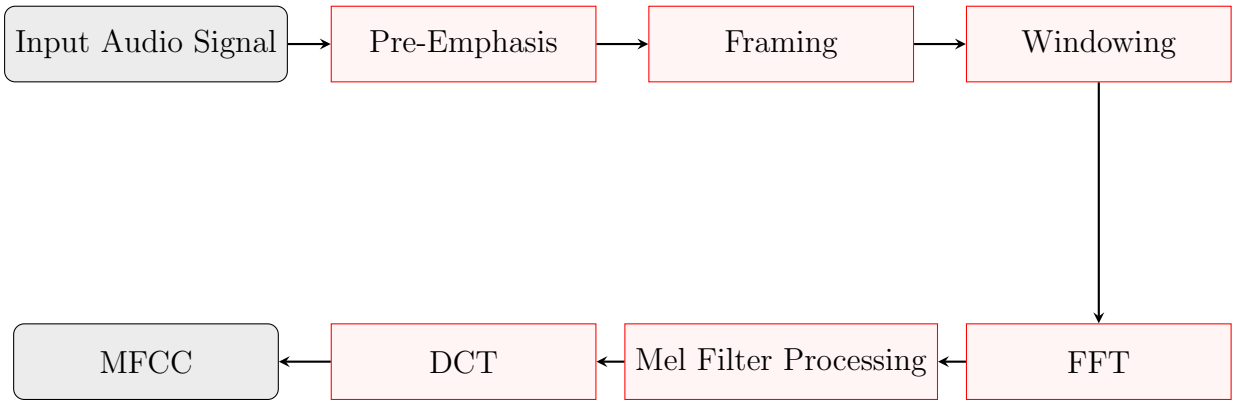

Figure 1: Control Flow of MFCC Computation

In addition to static MFCC coefficients, dynamic features known as delta (first derivative) and delta-delta (second derivative) coefficients are computed. These features capture the temporal variations of speech, enhancing the robustness of ASR systems. Delta features represent the rate of change of MFCCs over time, while delta-delta features capture acceleration trends, making speech feature extraction more effective in real-world conditions.

3.1 A Visual Representation of Each Step

4 Step-by-Step Explanation

We will now go a little slowly through all the steps and explain why each of the steps is necessary and what is done.

An audio is a continuous-time (analog), constantly changing signal of sound, typically consisting of air pressure variations that propagate as waves. However, computers cannot process continuous analog signals directly, so we must digitize them. The continuous signal is captured by a microphone at regular time intervals and converted into an electrical signal for processing, called **sampling points** (which is denoted as $x[n]$, nth sample or $x[t]$, sample measured at time t, $n = t * sampling\ rate$). The number of samples taken per second is called the **sampling rate**.

After sampling, the analog audio signal is represented as a sequence of discrete values (digital samples), which can be stored and processed using digital techniques.

This is the main audio time-domain diagram, and we start working with this. Here Y-axis represents the amplitude of the samples and X-axis represents the sample index.

4.1 Pre-emphasis:

Pre-emphasis is the first step to MFCC computation. The speech signal naturally loses energy at higher frequencies due to the physiological characteristics of the human vocal tract and the properties of sound transmission. This phenomenon, known as spectral tilt,

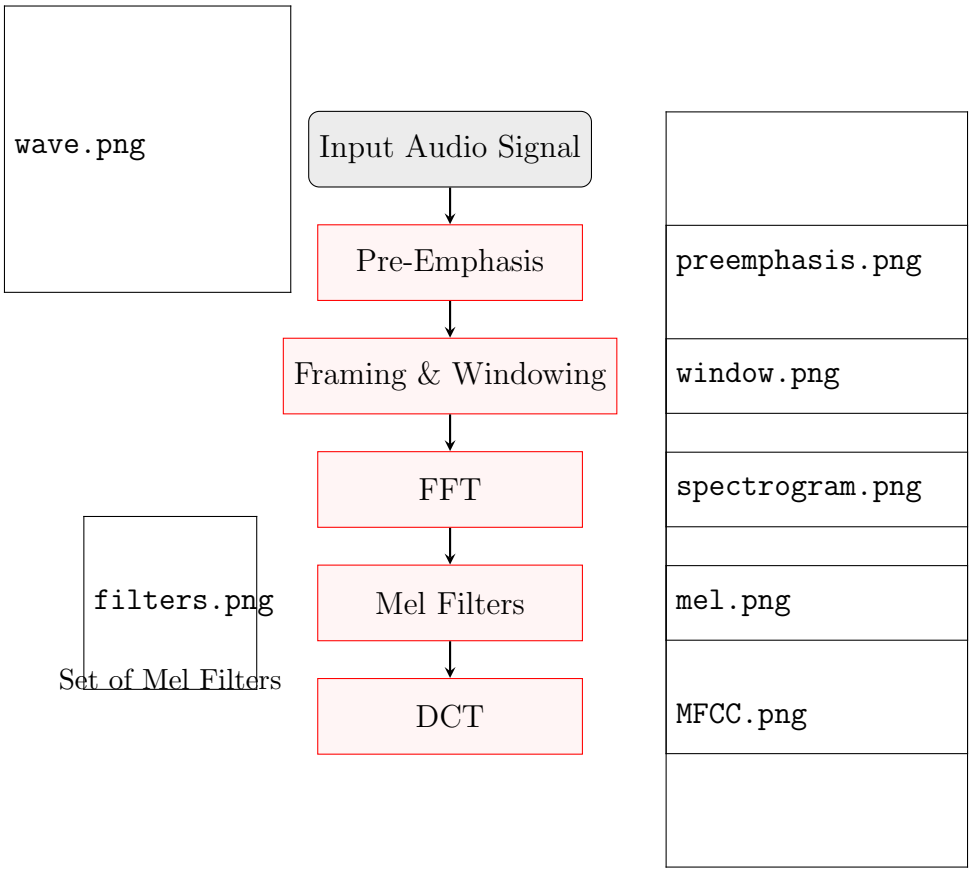

Figure 2: MFCC Computation Flowchart with Corresponding Images

occurs because voiced segments such as vowels contain more energy at lower frequencies than at higher frequencies.

By applying pre-emphasis, high-frequency energy is boosted, making crucial speech details, such as formants and consonants, more discernible. This enhances speech clarity and improves phone detection accuracy in automatic speech recognition (ASR) systems. Additionally, pre-emphasis increases the signal-to-noise ratio, making important speech features stand out against background noise.

Technically, pre-emphasis is implemented using a high-pass filter that amplifies high frequencies while maintaining a balance in the overall. This filter subtracts a fraction (α) of the previous sample ($x[n-1]$) from the current sample ($x[n]$).

The mathematical formulation of this process:

$$y[n] = x[n] - \alpha x[n-1] \tag{1}$$

where:

- $x[n]$ is the input signal (original speech signal).
- $x[n-1]$ is the previous sample of the input signal.
- α is the pre-emphasis coefficient (typically between 0.95 and 0.98).
- $y[n]$ is the output signal (pre-emphasized signal).

Figure 3:

Pre-emphasized Signal: This represents the signal after applying pre-emphasis.The higher frequencies are enhanced, making the waveform sharper and highlighting rapid transitions. This step is crucial for improving feature extraction in speech processing

4.2 Framing:

An audio signal is constantly changing over time, making direct analysis difficult. However, to simplify things, we assume that over short time scales, the signal does not change significantly in a statistical sense. This means that while the individual samples are always fluctuating, the overall characteristics of the signal remain relatively stable within a short time frame. This assumption of statistical stationarity is why we divide the signal into short frames of 20 to 40 milliseconds.

If the frame length is too short, there will not be enough samples to get a reliable spectral estimate. However, if the frame is too long, the assumption of signal's statistical stationarity will be violated, and the signal properties will change too much for consecutive frames, making the analysis inaccurate. To prevent data loss at the edges and maintain continuity of each frame, a overlapping of about 10 milliseconds is used.

Figure 4:

Without overlap, important details at the boundaries might be lost, affecting the quality of signal processing. After framing if N is the length of each frames then the frame samples in $k-th$ frame can be represented by:

$$y_k[n] = y[n + kr] \tag{2}$$

where:

- n: the number of samples in a frame $(0 \leq n \leq N-1)$
- r : hop length.

The given figure represents the signal after applying pre-emphasis and framing. The signal is now divided into short frames for further processing, preserving temporal characteristics while preparing it for feature extraction. The X-axis represents the frames and the Y-axis represents the samples of frames.Here the frames are taken of 20ms with 10ms overlap.

Figure 5:

4.3 Windowing:

In MFCC computation, after framing and scaling, each frame is multiplied by a window function, a process known as windowing. This step smoothens the abrupt transitions at the edges of frames, reducing discontinuities when performing the Discrete Fourier Transform (DFT). Without windowing, the sharp cuts at frame boundaries introduce spectral leakage, causing unwanted frequency components to appear in the spectrum. Spectral leakage distorts the true frequency representation of the signal, making feature extraction less accurate. To mitigate this, a window is commonly used, which tapers the signal gradually at the edges while preserving the main frequency components. This ensures a more accurate spectral representation, leading to better speech recognition and feature

extraction in MFCC processing.

Hanning Window:

$$w(n) = 0.5 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) \tag{3}$$

Blackman Window:

$$w(n) = 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right) \tag{4}$$

This figure represents the signal after applying windowing to the first and second frames. Windowing smooths the edges of each frame, reducing spectral leakage and improving frequency resolution. The X-axis represents the sample points, while the Y-axis represents amplitude variations after applying the window function.

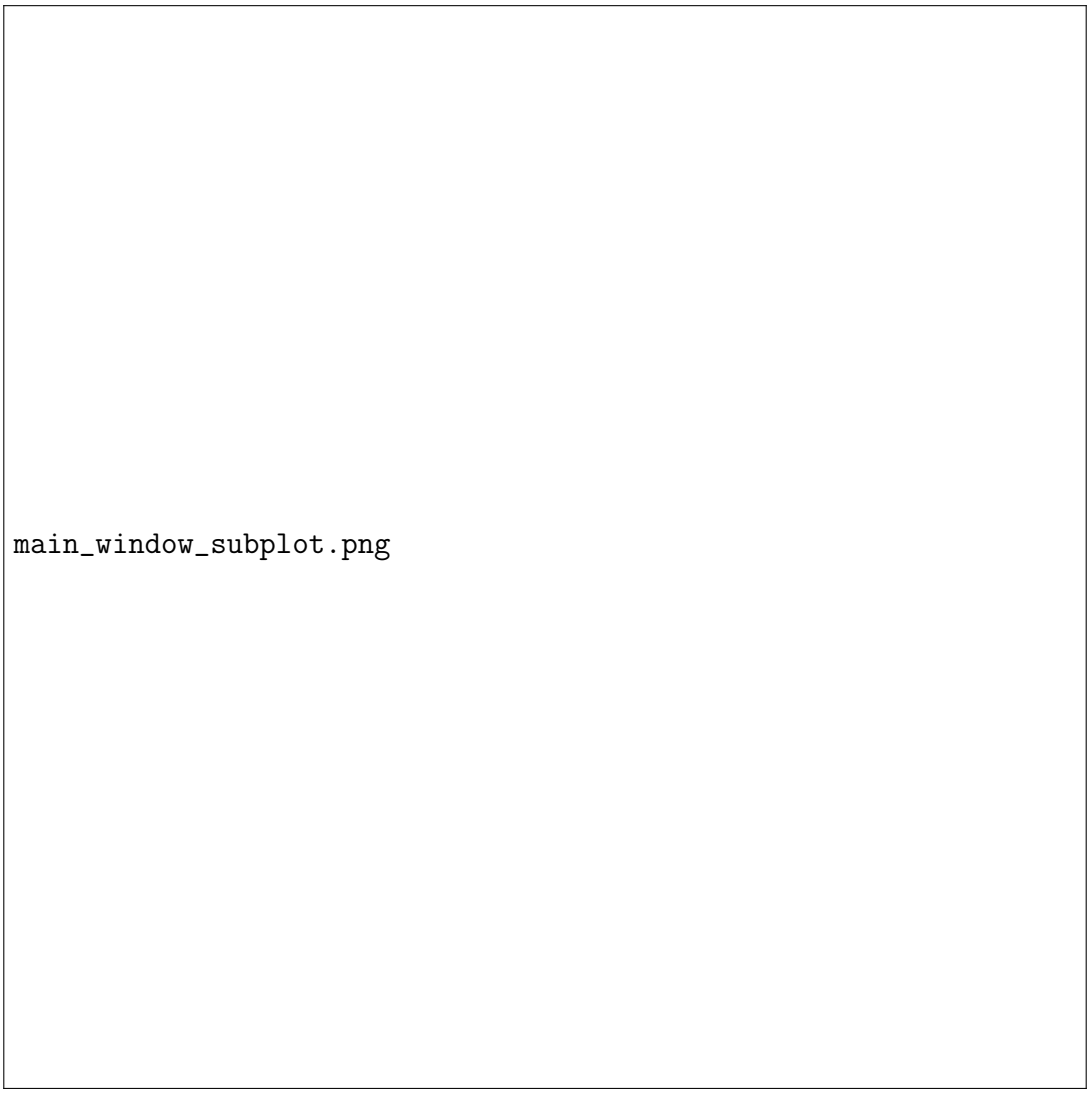

Figure 6:

4.4 Fast Fourier Transformation:

The Fast Fourier Transform (FFT) is an efficient algorithm to compute the Discrete Fourier Transform (DFT). The DFT converts a discrete-time signal from the time domain to the frequency domain, which is crucial for including speech recognition and spectral analysis. However, directly computing the DFT is computationally expensive, requiring $O(N^2)$ operations for an N-point sequence. The FFT significantly reduces this complexity to O(N log N), making it much faster and practical for real-time applications.

In MFCC computation, the FFT plays a crucial role in transforming each framed speech signal from the time domain to the frequency domain. This step is essential because human auditory perception is based on frequency content rather than raw waveforms. The FFT allows us to compute the power spectrum, which mimics how the human cochlea responds to different frequencies.

The cochlea vibrates at different locations depending on the incoming sound's frequency, activating corresponding nerve signals to the brain. Similarly, the periodogram estimate obtained via FFT identifies which frequencies are present in a given frame of speech. This transformation prepares the signal for Mel filter bank processing, which further models human auditory perception.

Discrete Fourier Transform (DFT) Definition:

$$X_k[m] = \sum_{n=0}^{N-1} y_k[n] e^{-j\frac{2\pi}{N}mn}, \quad 0 \leq m \leq M \tag{5}$$

where:

- $X_k(m)$ represents the frequency-domain coefficients,
- $y_k(n)$ is the input time-domain sample sequence,
- $e^{-j\frac{2\pi}{N}kn}$ represents the complex exponential basis function.
- M is the number of frequency bin. Generally, m is chosen in such a way that $M = 2^b, 2^b \geq N$(Frame length);$b \in \mathbb{Z}^+$. If m is properly chosen, then it gives better frequency resolution, otherwise it may increase computational complexity.

Computing this directly requires N multiplications for each frequency component, leading to $O(N^2)$ complexity.

FFT Algorithm: The FFT algorithm is a divide-and-conquer method that recursively divides the DFT into smaller DFTs, reducing the complexity to $O(N \log N)$.

- **Step-by-Step Computation:**
  - **Divide the sequence:** Split the N-point sequence into two $N/2$-point sequences:
    - Even-indexed samples: $y_{k_e}[n] = y_k[2n]$

* Odd-indexed samples: $y_{k_o}[n] = y_k[2n+1]$

– **Compute two smaller DFTs:**

* Compute the DFT of the even-indexed sequence:

$$S_{k_e}(m) = \sum_{n=0}^{N/2-1} y_{k_e}[2n] e^{-j\frac{2\pi}{N/2}2nm} \tag{6}$$

* Compute the DFT of the odd-indexed sequence:

$$S_{k_o}(m) = \sum_{n=0}^{N/2-1} y_{k_o}[2n+1] e^{-j\frac{2\pi}{N/2}2nm} \tag{7}$$

– **Combine the results using the FFT structure:**

* Using the factor:

$$W^m = e^{-j\frac{2\pi}{N/2}m} \tag{8}$$

* The final FFT computation is:

$$S_k(m) = S_{k_e}[m] + W^m S_{k_o}[m], \quad 0 \leq m \leq M \tag{9}$$

After computing FFT we get $S_k(m)$ for each frame(k is the frame index). Then the Periodogram estimate of the power spectrum is calculated. Here, the absolute values of the complex Fourier transformation are taken, and the result is squared.

$$P_k[m] = \frac{1}{N} |S_k[m]|^2 \tag{10}$$

For FFT with 2^b bins,S_k represents an array of 2^b values of kth frame, then the first $2^{b-1}+1$ coefficients are kept based on the symmetry properties of the Discrete Fourier Transform (DFT) for real-valued signals. Where $S_k[0]$ is the DC component, next 2^{b-1} components are unique frequency bins and the rest are mirror values of $S_k[1]$ to $S_k[2^{b-1}]$.

After performing the Fast Fourier Transform (FFT), we obtain the power spectrum, the above two representations are of the first two frames. The X-axis represents the $frequencybins$, while the Y-axis represents the *magnitude*, which is negative due to logarithmic scaling.

Computing the Magnitude Spectrum The FFT output, $X[k]$, is complex, so we compute its magnitude as:

$$|P_k[m]| = \sqrt{\text{Re}(S_k[m])^2 + \text{Im}(S_k[m])^2}$$

which provides the frequency content without phase information.

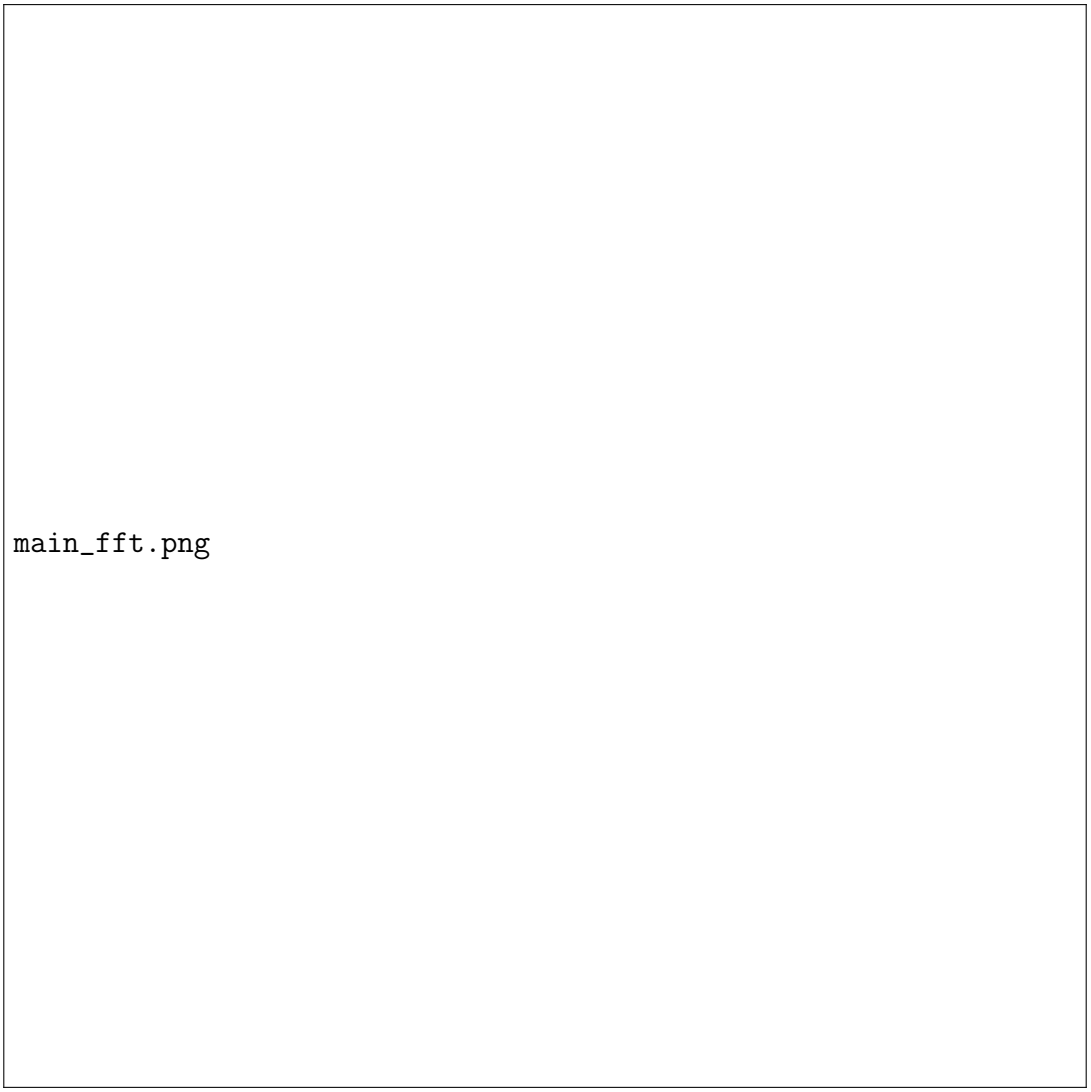

Figure 7:

Converting to Decibels To convert the magnitude spectrum into the dB scale, we use:

$$P_k^{\text{dB}}[m] = 20 \log_{10} \left(\frac{|S_k[m]|}{\max(|S_k[m]|)} \right)$$

And the spectrogram of the audio is

4.5 Mel Scale Filter bank:

After applying the Fast Fourier Transform (FFT) and computing the power spectrum, the next step in MFCC computation is to apply a Mel-spaced filterbank. This step is crucial because it mimics the human auditory perception, which is more sensitive to lower frequencies and less sensitive to higher frequencies. Instead of analyzing all frequencies linearly, we use the Mel scale, which spaces frequencies in a way that aligns with human hearing. The Mel scale is a perceptual frequency scale that reflects how humans perceive sound. Unlike the linear frequency scale (measured in Hertz, Hz), the

Figure 8:

Mel scale is nonlinear, placing more emphasis on lower frequencies where human hearing is more sensitive

The Mel filterbank consists of 20-40 triangular filters (with 26 filters being the standard) that are applied to the power spectrum. Each filter focuses on a particular frequency range, emphasizing certain spectral components while suppressing others.

Convert Frequency (Hz) to Mel Scale

The standard formula for converting a frequency f (in Hz) to the Mel scale is:

$$M(f) = 1125 \cdot \ln\left(1 + \frac{f}{700}\right) \tag{11}$$

where:

- f = frequency in Hz
- $M(f)$ = frequency in Mel units

The number 700 Hz is used as a reference point in the Mel scale formula because: It aligns with human hearing sensitivity – Below 700 Hz, humans perceive frequency changes linearly, but above 700 Hz, perception becomes logarithmic (non-linear).It was chosen based on experimental data.
The coefficient 1125 in the Mel formula comes from the fact that: It ensures a smooth and accurate conversion between Hertz and the Mel scale by approximating the relationship found in studies.

Convert Mel Scale to Frequency (Hz)

To convert a Mel-scale value M back to a linear frequency f (in Hz), use the inverse formula:

$$f(M) = 700 \cdot \left(e^{\frac{M}{1125}} - 1\right) \tag{12}$$

where:

- M = frequency in Mel units
- $f(M)$ = frequency in Hz

To compute the filterbank energies, each triangular filter is multiplied with the power spectrum, and the resulting values are summed. This results in 26 values, each representing the amount of spectral energy captured by a corresponding Mel filter. This step effectively reduces the dimensionality of the spectral representation while retaining perceptually important frequency information, making it a critical step in speech and audio feature extraction.

A detailed explanation of filterbanks calculation

we'll assume sampled at 16kHz.
For convenience in calculation, we will see an example with 10 filterbanks; in reality, 26–40 filterbanks are used. Let the samples be framed into 20ms long frames, which means that each frame contains 320 samples with an overlap of 10ms i.e. of 160 samples. To palce the filters along the frequency bins perfectly, we first have to choose a lower and upper frequency. Let 300 Hz for the lower and 8000 Hz for the upper frequency. Of course, if the speech is sampled at 16000 Hz. Then follow these steps:

1. Using equation 11, convert the upper and lower frequencies to Mels. In our case, 300Hz is 401.25 Mels and 8000Hz is 2834.99 Mels.

2. For this example, we will do 10 filterbanks, for which we need 12 points. This means we need 10 additional points spaced linearly between 401.25 and 2834.99 Mels. This comes out to:

   m(i) = 401.25, 622.50, 843.75, 1065.00, 1286.25, 1507.50, 1728.74, 1949.99, 2171.24, 2392.49, 2613.74, 2834.99

3. Now use equation 12 to convert these back to Hertz:

   h(i) = 300, 517.33, 781.90, 1103.97, 1496.04, 1973.32, 2554.33, 3261.62, 4122.63, 5170.76, 6446.70, 8000

   Notice that our start- and end-points are at the frequencies we wanted.

4. We don't have the frequency resolution required to put filters at the exact points calculated above, so we need to round those frequencies to the nearest FFT bin. This process does not affect the accuracy of the features. To convert the frequencies to FFT bin numbers, we need to know the FFT size and the sample rate:

$$f(i) = \lfloor \frac{(\text{fft bins} + 1) \cdot h(i)}{\text{samplerate}} \rfloor \tag{13}$$

This results in the following sequence:

$$f(i) = 9, 16, 25, 35, 47, 63, 81, 104, 132, 165, 206, 256$$

We can see that the final filterbank finishes at bin 256, which corresponds to 8kHz with a 512-point FFT size.

5. Now we create our filterbanks. The first filterbank will start at the first point, reach its peak at the second point, then return to zero at the third point. The second filterbank will start at the 2nd point, reach its max at the 3rd, then be zero at the 4th, etc. A formula for calculating these is as follows:

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \frac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \leq k \leq f(m) \\ \frac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \leq k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases} \tag{14}$$

where M is the number of filters we want, and $f()$ is the list of $M + 2$ Mel-spaced frequencies.
After this the Mel filterbank energy E_m is computed by applying the Mel-scaled filter $H_m(f)$ to the power spectrum of the signal. This step is crucial in extracting frequency components that mimic human auditory perception.

Expression for E_m Calculation

$$E_{k_m} = \sum_{l=1}^{\frac{N+2}{2}} P_k[l] H_m(l) \tag{15}$$

where:

- E_{k_m} = Energy for the m-th Mel filter, of kth frame.
- $P_k(l)$ = Power spectrum of the ,
- $H_m(l)$ = Weight of the m-th Mel filter at frequency lth bin.,
- N = Number of frequency bins in the FFT.

In reality, 26 Mel fiters are used, and we get a vector of 26 Mel energies for each frame.

Figure 9:

After applying the Mel Filterbank, we obtain a set of 26 Mel energies for each frame. The Mel scale is a perceptually motivated frequency scale that mimics the human ear's response to sound. Instead of linearly spaced frequencies, it applies triangular filters that are spaced logarithmically in the higher frequencies and linearly in the lower frequencies.

4.6 Discrete Cosine Transformation:

After computing the Mel filterbank energies, we obtain a vector of 26 Mel energies for each frame. However, these energy coefficients are highly correlated due to the overlapping nature of the Mel filters. To make the features more compact and independent, we apply the Discrete Cosine Transform (DCT) to the Mel energies.

The DCT serves as a decorrelation step, transforming the 26 correlated energy values into a new set of coefficients, known as Mel-Frequency Cepstral Coefficients (MFCCs). This transformation helps in reducing redundancy and capturing only the most important spectral features. The first few MFCCs contain the most use-

ful information, while the higher-order coefficients capture fine spectral variations, which are often discarded.
Mathematically, the DCT of the Mel energies is computed as:

$$C_{k_i} = \sum_{m=1}^{M} F_{k_m} \cos\left[\frac{\pi i}{M}\left(m - \frac{1}{2}\right)\right], \quad 1 \leq i \leq 26 \tag{16}$$

where:

- C_{k_i} is the i-th MFCC of kth frame,
- $F_{k_m} = \log E_{k_m}$ is the energy of the m-th Mel filter in kthframe,
- M is the total number of Mel filters (typically 26),
- i is the index of the MFCC coefficient,

Here, the cosine functions act as basis functions, and this expression calculates the projection of the 26 Mel energies onto each distinct basis function. Usually, 26 such projections are taken and for ASR first 13 coefficients are kept.

These coefficients serve as a compact and perceptually relevant representation of the speech signal, making them highly useful in speech and speaker recognition tasks. By using only the first 13 coefficients, the system retains the most important phonetic information while reducing noise and redundancy.

After computing the **Discrete Cosine Transform (DCT)** on the **Mel energies**, we obtain the **Mel-Frequency Cepstral Coefficients (MFCCs)**. These coefficients capture the important spectral characteristics of the signal while reducing redundancy.

5 Delta and Delta-Delta Coefficients

MFCC computation assumes that the speech signal is static within each short frame. However, speech is a dynamic process, and important information is carried by how the spectral properties change over time. To capture these temporal variations, we compute Delta (Δ) and Delta-Delta (Δ^2) coefficients, also known as first-order and second-order derivatives of MFCCs.

5.1 1. Delta (Δ) Coefficients – First Derivative

Delta coefficients represent the rate of change of MFCCs over time. They are calculated using a regression formula:

$$\Delta c_t = \frac{\sum_{n=1}^{N} n \cdot (c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2} \tag{17}$$

where:

- c_t is the MFCC coefficient at time t.

Figure 10: MFCC Representation

- N is the window size (usually 2 or 3 frames).
- c_{t+n} and c_{t-n} are the MFCCs at future and past frames.

This formula takes neighboring frames into account to estimate the change in spectral characteristics over time.

5.2 2. Delta-Delta (Δ^2)Coefficients – Second Derivative

Delta-Delta coefficients capture the acceleration of spectral changes by computing the second-order derivative of MFCCs:

$$\Delta^2 c_t = \frac{\sum_{n=1}^{N} n \cdot (\Delta c_{t+n} - \Delta c_{t-n})}{2 \sum_{n=1}^{N} n^2} \tag{18}$$

where:

- $\Delta^2 c_t$ is the second derivative of the MFCC coefficient at time t.

- N is the window size (typically 2 or 3 frames).
- Δc_{t+n} and Δc_{t-n} are the delta coefficients at future and past frames.

5.3 Why Are Delta and Delta-Delta Coefficients Important?

- Delta (Δ) coefficients capture the dynamics of speech, such as whether a phoneme is transitioning from one sound to another.
- Delta-Delta (Δ^2) coefficients help understand how fast or slow the change occurs, capturing speech rhythm and tone variations.
- Including Δ and Δ^2 coefficients along with MFCCs improves the accuracy of speech and speaker recognition systems by making them more robust to variations in speech speed and articulation.

Thus, the final MFCC feature set typically consists of 39 features :

- 13 MFCCs
- 13 Delta MFCCs
- 13 Delta-Delta MFCCs

These enriched features help in speech recognition, speaker identification, and emotion detection by modeling both static and dynamic properties of speech signals.

References

I have taken help from the following books and other sources for information:

1. Lawrence Rabiner, Biing-Hwang Juang, *Fundamentals of Speech Recognition.*
2. John G. Proakis, Dimitris G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, Third Edition.
3. Uday Kamath, John Liu, James Whitaker, *Deep Learning for NLP and Speech Recognition.*
4. Sanjit K. Mitra, *Digital Signal Processing: A Computer-Based Approach*, Second Edition.
5. Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development.*