Multimedia System and Design (Mids) - Jaish Khan

Multimedia is computer information which can be represented through audio, video, animations, text, graphics or images. A *Multimedia Application* is then a collection of multiple media sources.

A **Multimedia System** is then a system capable of processing, storing, generating, manipulating and rendering multimedia. They have 4 characteristics (*computer controlled*, *integrated*, *digital*, *interactive*).

- Tooling required for Multimedia Systems
    - **Hardware** → Capture Devices, Storage Devices, Networks, Display Devices and Computer Systems.
    - **Software** → Software Tools and Applications.

It might also need **Synchronization** as different sources like audio, video and text can get out of sync.

Hypertext → text which contains links to other texts.
Hypermedia → not just text but other types of media as well.

- Examples: World Wide Web, Powerpoint, Adobe Acrobat, Adobe Flash etc.
- Challenges: Distributed Networks, Sequencing, Synchronization.
- Issues: Temporal information storage and representation, Large data requirements, Digital representation.

Data Formats

Different types of data requires differen types of inputs and ways to store it.

Data Format	Input	Storage
Text	Keyboard, OCR, Voice	1B per ASCII >1B per Unicode
Images	Digital Cameras and Scanners	1b/pixel (Black/White) 8b/pixel (Grayscale) 24b/pixel (Truecolor) 32b/pixel (TC with Alpha)
Graphics	Generated by programs	Not much as its based on code
Audio	Microphones	1min, Mono → 5MB 1min, Stereo → 10MB

Data Format	Input	Storage
Video	Video Cameras	1s, HD → 150MB 1min, HD → 9GB

- **Graphics Standards:** OpenGL, PHIGS, GKS.

Data can be **Static** (Discrete) → Text, Images, Graphics or **Dynamic** (Continuous) → Audio, Video, Animations. Data can also be digital or analog and can be converted using **ADC/DAC**

OCR (*Optical Character Recognition*) → A method used to scan text data from physical objects.

Multimedia Data Compression

Because the multimedia can reach enormous sizes (especially video) so compression becomes necessary. There are two ways of doing it:

- **Lossless** → Preserves all data.
  - Examples: Zip (Text), PNG (Images), FLAC (Audio) etc...
- **Lossy** → Discards perceptually less relevant data to achieve higher compression ratios.
  - Examples: JPEG (Images), MP3 (Audio), AVC (Video) etc...

A **codec** (Coder-Decoder) is used to compress and decompress data.

Text

A **Typeface** is a family of characters that include many sizes and styles like Times, Arial, Helvetica etc.

- The size of a font is measured in **Points** and is measured from the top of the ascender to the bottom of the descender. $1p = \frac{1}{72}\,inch = 0.0138\,inch$.
- **Font** → Collection of characters of a single point-size and style belonging to a typeface family like Times 12-point italic. A *Font Family* is a collection of different styles of the same font.
- **Styles** → Bold, Italic etc...
- **Serif** → Small lines attached to the ends of letters.
    - Fonts which have them are called *Serif* Fonts (more readable on printed media) and those without them are called *Sans-Serif* Fonts (more readable on screens).
- **Leading** → Vertical space between lines of text (measured from baseline to baseline).
- **Tracking** → Horizontal space between characters in a block of text (applies as a whole).
- **Kerning** → Horizontal space between individual characters to make them more readable (applies to pairs of characters).
- **X-height** → Height of the lowercase 'x' in a typeface.
- **Other Character Metrics**
    - Baseline → The line on which most characters sit.
    - Ascender → The part of lowercase letters that goes above x-height such as h, b and f.
    - Descender → The part of lowercase letters that goes below baseline suchas p, q and y.
    - Counter → The whitespace inside letters such as O, A and P.
    - Set width → The horizontal space a character takes.
    - Cap height → The height of capital letters.
    - Height → The total height of a character including ascenders/descenders.
    - Cap line → The line which marks the top of uppercase characters.
    - Mean line → The line which marks the top of lowercase characters (ignoring ascenders).

Fonts are bundled in different file types. *Bitmap* fonts can not be altered while *TrueType* and *PostScript* fonts can be. A letter on the screen using dots/pixels.

Legibility

It refers to how easily characters can be identified/distinguished from each other in a text. It is influenced by *Font Size*, *Background and Foreground Color*, *Font Style* and *Spacing*.

Case can also affect legibility in that it is easier to read words with mixture of upper and lower case letters VS all upper case letters. Capitalization Schemes → UPPER CASE, lower case, Title Case (**Intercap**), Sentence case etc.

Design Tips

A designer should use the most legible font available and not use as multiple different types faces. It is called **ransom-note** typography.

- Normal line length is `10` words or `70` characters. On mobiles its even less.
- Use **bold** and *italics* to convey meaning. Also experiment with other styles like underlines, outlines and shadows.
- All caps or All italics makes reading difficult so avoid that. All caps denotes Shouting.
- Adjust spacing between lines (leading) and the spacing between letters in headings to remove gaps (tracking). Also adjust the spacing between individual letters (kerning).
- Use background and foreground colors to increase legibility. **Reverse Type** is light text on dark background.
- Use anti-aliased text as it blends the colors along the edges of letter.
- Use **whitespace** {space: ASCII 32, tab: ASCII 9 and newlines}. A nonbreaking space entity ` ` is used to force spaces into lines of text.
- *Portrait* (Taller-than-Wide) vs *Landscape* (Wider-than-Tall) screen orientation.
- Keep font sizing in your mind. Fonts smaller than 12-points are not very legible on a monitor.

> People blink 3-5 times/minute, using a computer and 20-25 times/minute reading a book. This causes fatigue, dryness and hence makes reading much slower comparatively to reading a book.

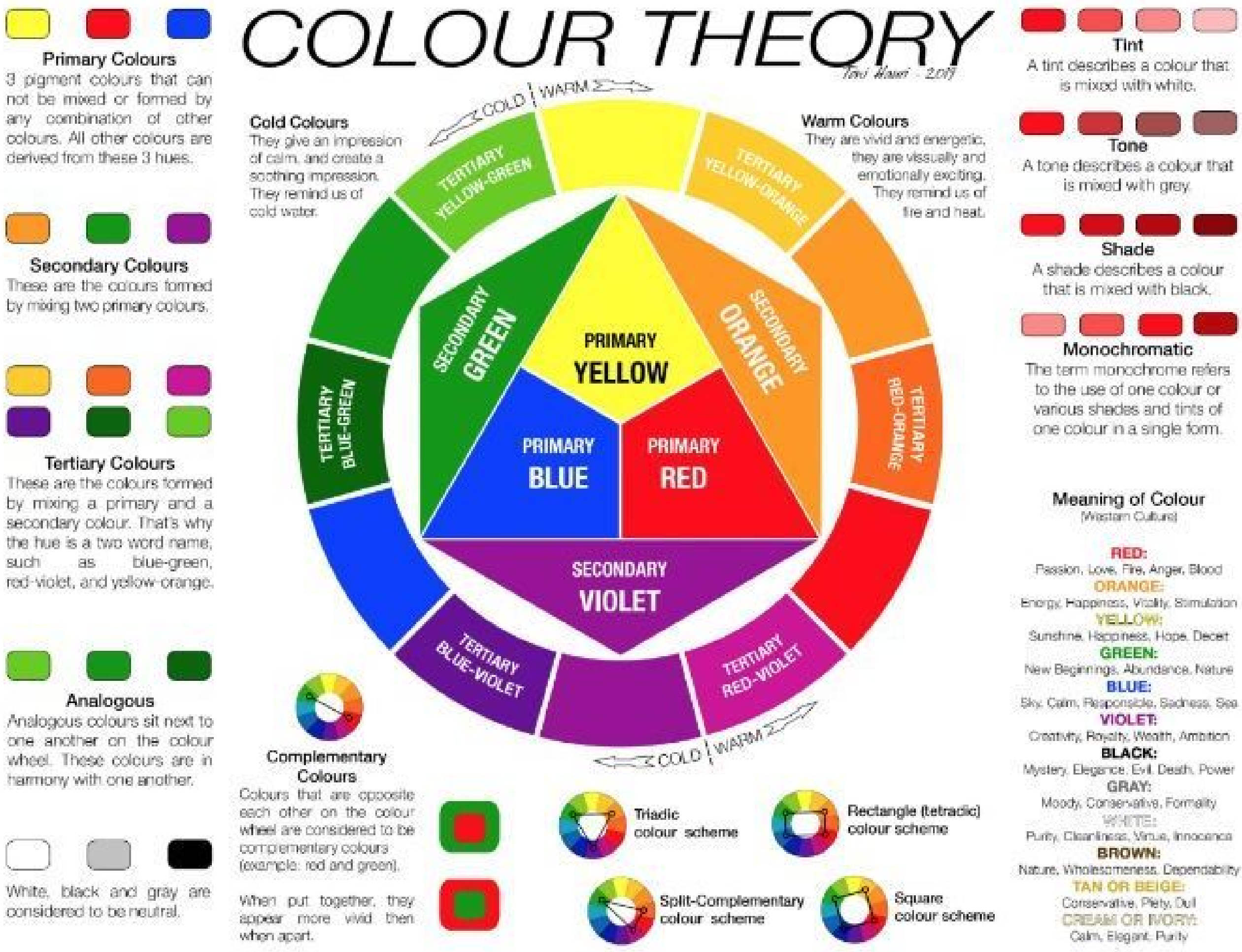
- **ASCII** → 128 characters, only supports English-like languages. Every character takes 1 byte.
- **Unicode** → 65536 characters, supports every character from every single language. Every characters takes 1-4 bytes (depending on the language).

Color

Color conveys meaning which is different depending on the culture. It is affected by light, context and environment.

Color Wheel

When it comes to Art and Design, the RBY (Red, Blue, Yellow) has been used.



Primary Colors → [Red, Blue, Yellow].
Mixing them gives us 3 more:
Secondary Colors → [Orange, Green, Violet].
Mixing them gives us 6 more colors:
Tertiary Colors → [Red-Orange, Red-Violet, Blue-Violet, Blue-Green, Yellow-Green, Yellow-Orange].

White, Black and Gray are considered neutral colors.

Color Strategies

Also called Color Harmony Schemes.

It is the process of choosing colors to be used such that they follow one of these schemes and are visually appealing.

1. *Monochromatic* → Uses variations in lightness and saturation of a single color.
2. *Grayscale* → Uses only black, white and grays.
3. *Analogous* → Uses colors that are adjacent/next to each other.
    1. Warm vs Cool Colors
4. *Complimentary* → Uses colors that are opposite to each other.
5. *Triadic* → Uses 3 colors that are 120° apart.
6. *Split-Complimentary* → Uses a base color and then two colors adjacent to its complimentary.
7. *Tetradic* → Uses 4 colors that are in pairs of two-complementaries.
8. *Discord* → Uses mismatched colors for an eye-catching effect.

Some Terms:

- **Color Contrast** → The difference between the darkest darks and the brightest brights.
- **Color Gamut** → It is the set of all colors that can be produced or recorded.
- **Dithering** → Process of simulating colors that are not in the palette by mixing pixels of different colors.
- **Color Palettes** → The colors that are at our disposal (we can use).

- **Palette Flashing:** Occurs when a series of images, each with its own color palette, are displayed, causing a flashing effect. Solutions include using a single palette for all images or fading images to black or white before displaying the next.

Hex Codes → Used for defining color on the computer. Starts with `#` then 3 pairs (RGB) of 6 hexadecimal numbers.

Color	Hex Code	RGB Values
Red	`#FF0000`	$(255, 0, 0)$
Green	`#00FF00`	$(0, 255, 0)$
Blue	`#0000FF`	$(0, 0, 255)$
Yellow	`#FFFF00`	$(255, 255, 0)$
Magenta	`#FF00FF`	$(255, 0, 255)$
Cyan	`#00FFFF`	$(0, 0, 255)$
White	`#FFFFFF`	$(255, 255, 255)$
Black	`#000000`	$(0, 0, 0)$

Physics of Color

Our human eye has a wall at the back of it called the "retina". This retina has millions of photosensitive cells called *rods* and *cones*. Rods are sensitive to Light while cones come in three variants: Red cone, Blue cone, Green cone. Cones are sensitive to their respective Color.

Our eyes have around 20-30 times more Rods compared to Cones which makes our eyes more sensitive to changes in lightness compared to changes in color.

Color Representation

The most common way to represent color is the **Additive** RGB (Red, Green, Blue) color model, used for displays, and the **Subtractive** CMYK (Cyan, Magenta, Yellow, Key/Black) color model, used for printing.

Color can also be defined using the HSL/HSV model where

- **Hue** → what color it is.
- **Saturation** → how much color there is.
- **Value/Lightness** → how much lightness/darkness there is.
- **Tint** → Color mixed with *white*.
- **Tone** → Color mixed with *gray*.
- **Shade** → Color mixed with *black*.

Then there're also these color models (not important)

1. YCrCb → Luma, Red-difference Chroma, Blue-difference Chroma
2. YIQ → Luma, Orange-Blue Chroma, Purple-Green Chroma
3. YUV → Luma, Blue-luminance difference, Red-luminance difference

and CIE Lab → Used for color management and is device independent.

Images

Image Resolution is the number of pixels in an image. It is written like `horizontal pixels x vertical pixels`.

Frame Buffer → Used to store bitmaps.
Color Histograms → A graph that show color distribution where whites/light are on the left, blacks/dark on the right and grays in the middle.

Storage

1. Bitmap → 1 bit per pixel.
2. 8-bit Grayscale → 8 bits per pixel.
3. 8-bit Color → 8 bits per pixel. Done using a CLUT (Color Lookup Table).
4. True Color → 8 bit per color (8 for Red, 8 for Green, 8 for Blue)
5. True Color with Alpha → Extra 8 bits for Transparency.

Image File Formats

1. JPEG - Joint Photographic Experts Group
2. PNG - Portable Network Graphics
3. GIF - Graphics Interchange Format → LZW Encoding
4. DCT - Discrete Cosine Transform
5. TIFF - Tagged Image File Format
6. EXIF - Exchange Image File Format
7. Animation Formats (FLC, GL, Apple QuickTime etc)
8. Postscript and PDF
9. BMP, WMP, PSD etc

- **Spatial Resolution:** The density of pixels per inch. Measured in ppi (pixels per inch) for monitors and dpi (dots per inch) for printers.
- **Color Resolution:** The number of colors each pixel can display, determined by bit depth.
- **Device Dependence:** Image dimensions depend on the resolution of the output device. Bitmapped images are device-dependent.
- **Vector Graphics:** Created from mathematically defined shapes, allowing for smooth scaling without distortion.

- **Image Size:** A 512×512 grayscale image takes up 1/4 MB, a 512×512 24-bit image takes 3/4 MB with no compression. Overhead increases with image size. Modern high digital cameras (10+ Megapixels) produce approximately 29MB uncompressed images. Compression is commonly applied.

Image Artifacts

These are unwanted abnormalities/distortions that appear due to issues.

1. Compression Artifacts → Occurs due to lossy compression.
2. Motion Blur → Bluriness that shows when the subject/camera moves.
3. Chromatic Aberration → Issues with the lens of the camera causes color fringing.
4. Color Banding → Less bit-depth causes bands of color to appear in gradients.
5. Aliasing → Low-resolution causes jagged/stair-stepped edges.

Camera Attributes

Camera Resolution is measured in Megapixels.

1. **ISO** → sensitivity of the camera's sensor to light.
2. **Aperture** → opening inside the lens that controls how much light enters the camera.
3. **Shutter Speed** → how long the camera's shutter stays open, controlling the amount of time light hits the sensor.

Compression

Text Compression

Text is that one medium where Lossy compression doesn't work. It **must** always be Lossless otherwise we might lose important information.

Data Compression

Zip, **Rar** and **7zip** are three of the most well known file compression formats. *Zip* is universal in nature and is supported on every platform natively.

Linux has the *gzip*, *bzip2*, *xz* and *tar* formats.

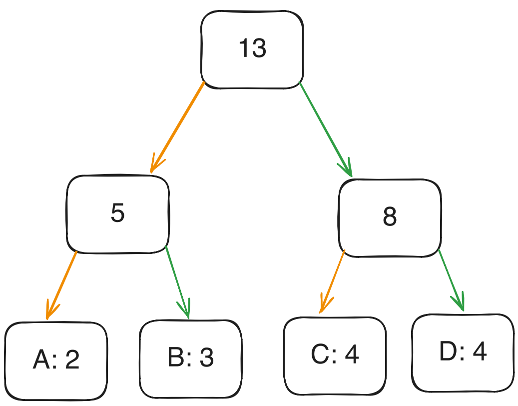
Huffman Encoding

A lossless compression, entropy coding algorithm which is used in JPEG, MP3 and ZIP.

It uses "variable-length" codes to represent input symbols based on their frequency which means that symbols with higher frequency get shorter codes, while rare symbols get longer codes.

Example → We have this string: "AABBBCCCCDDDD"

1. Calculate frequency of each character → A: 2, B: 3, C: 4, D: 4.
2. Create a frequency queue → (A,2), (B,3), (C,4), (D,4).
3. Build the Huffman Tree
    1. Take the two nodes with lowest frequency → (A,2), (B,3) and Create a new node with their sum (5), with A and B as its children.
    2. Now we have → (5), (C,4), (D,4)
    3. Take the two lowest again → (C,4), (D,4) and Create a new node with their sum (8), with C and D as its children.
    4. Finally, combine (5) and (8) → (13).



5. Now, Assign codes by traversing the tree, assigning 0 for left branches and 1 for right branches → A: `00`, B: `01`, C: `10`, D: `11`.
6. Encode the string "AABBBCCCCDDDD" becomes `00 00 01 01 01 10 10 10 10 11 11 11 11`

The encoded bitstring is 26 bits long, compared to the original 96 bits.

LZW Encoding

LZW(Lempel-Ziv-Welch) is a dictionary-based encoding algorithm that builds a dictionary of sequences as it processes the input.

Example → We have the string: "AABABBABCABABBA"

1. Initialize the dictionary, Start with single characters → 1: A, 2: B, 3: C
2. Encode the string

Step	Current	Next	Output	Add to Dictionary
1	A	A	1	4: AA
2	A	B	1	5: AB
3	B	A	2	6: BA
4	AB	B	5	7: ABB
5	B	A	2	(BA already in)
6	BA	B	6	8: BAB
7	C	A	3	9: CA
8	AB	A	5	10: ABA
9	AB	B	5	(ABB already in)
10	BA	-	6	-

Final encoded output → `1, 1, 2, 5, 2, 6, 3, 5, 5, 6`

This sequence of numbers represents the compressed version of the original string. To decompress, we would use these numbers to rebuild the dictionary and reconstruct the original string.

Image Compression

JPEG Compression

The JPEG file format uses a lossy compression technique which removes data and details from an image, that is unnoticeable to the human eyes.

JPEG compression reduces file size by:

1. Converting colors to **YCbCr** format.
2. Using **chroma subsampling** to save less color data.
3. Transforming pixel values into **frequencies** (DCT).
4. **Quantizing** the data by rounding off unimportant values.
5. Compressing with **entropy encoding**.

1. Color Space Conversion (RGB to YCbCr)

- **Why?**: Human eyes are more sensitive to brightness (**luminance, Y**) than color differences (**chrominance, Cb and Cr**).
- **How?**: The image's RGB colors are transformed into **YCbCr** format:
  - **Y** = Luminance (brightness).
  - **Cb** = Blue-difference chrominance.
  - **Cr** = Red-difference chrominance.

2. Chroma Subsampling (Reducing Color Data)

- **Why?**: Our eyes don't notice small changes in color as much as they notice changes in brightness. JPEG takes advantage of this by **storing fewer color details** to save space.
- **How?**: Chroma subsampling reduces the resolution of Cb and Cr channels (by creating 2×2 blocks and taking the average):
  - **4:4:4**: No subsampling (full quality for Y, Cb, and Cr).
  - **4:2:2**: Cb and Cr are halved horizontally.
  - **4:2:0**: Cb and Cr are halved both **horizontally and vertically**.

This means **fewer color values** are saved, while the brightness stays sharp.

3. Discrete Cosine Transform (DCT)

The image is split into **8×8 pixel blocks** for easier processing with values from 0-255. They are then shifted by subtracting 128 from each value making the range from -127 to 128. Each block will undergo compression separately.

- **How?**: converts the pixel values in each 8×8 block into **frequency values** (using the DCT formula):
  - **Low frequencies** = General shapes and smooth gradients.
  - **High frequencies** = Fine details (like edges or noise).

4. Quantization (Simplifying Data)

Less important frequencies (especially high ones) are rounded off to reduce precision.

- **How?**: A Quantization Table is used and every value in it divides a corresponding value in the block.
    - **Lower numbers** (more rounding) = **Smaller file, more quality loss**.
    - **Higher numbers** (less rounding) = **Better quality, larger file**.

5. Entropy Encoding (Further Compression)

The quantized data is compressed using:

- **Zig-Zag Scan** → The 8×8 block is scanned in a zig-zag pattern as this makes it more likely for larger strings of 0s to occur.
- **Huffman Encoding** to assign shorter codes to frequently used values.
[Multimedia System and Design (Mids) - Jaish Khan > Huffman Encoding]

Run-Length Encoding (RLE)

Used for AC Coefficients. To reduce repeating patterns, Instead of listing every 0 we just list how many there are in a sequence.

Example → We got coefficients (after zig-zag scan) in this order

```
[12, 5, 0, 0, 0, -2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Applying RLE converts it into:

```
(0, 12), (0, 5), (3, -2), (1, 1), (15, 3), EOB
```

- (0, 12): No zeros before 12.
- (0, 5): No zeros before 5.
- (3, -2): Three zeros before -2.
- (1, 1): One zero before 1.
- (15, 3): Fifteen zeros before the last non-zero value 3.
- EOB: End of Block. Denotes that the remaining values are zeroes.

Discrete Pulse Code Modulation (DPCM)

Used for DC Coefficients. DPCM is a signal encoding technique for data compression.

Instead of encoding absolute values of each sample (as in PCM), DPCM encodes the difference between the current sample and a predicted value.

Example → We have a simple audio signal: [100, 105, 111, 114, 116]

1. First number (100) is encoded as is.
2. For next numbers, we encode the difference:
    - 105 - 100 = 5
    - 111 - 105 = 6
    - 114 - 111 = 3
    - 116 - 114 = 2

So our DPCM encoding would be: [100, 5, 6, 3, 2]

Image Reconstruction

When you open the JPEG file, the reverse process occurs:

- Decompressed frequency values are used to reconstruct the image.
- Some details (like colors and fine edges) might be lost, especially if the file was highly compressed.

PNG Compression (Not Important)

PNG is a lossless image compression format that uses a combination of filtering and DEFLATE compression.

It supports true color, grayscale, and palette-based images as well as alpha channel support for transparency and gamma correction for cross-platform color consistency.

1. Pixels to Index Stream

Pixels are mapped to an index stream. Each row is called a **scanline**.

2. Filtering

PNG applies a filter to each scanline of the image. This process is called "filtering" and it helps to make the image data more compressible.

There are five filter types:
0. None: No filtering

1. Sub: Subtract the value of the pixel to the left
2. Up: Subtract the value of the pixel above
3. Average: Use the average of the left and upper pixel
4. Paeth: A special adaptive filter

The filter type that produces the smallest output is chosen for each scanline.

Example Original: [20, 30, 40, 30, 20] → Filtered: [20, 10, 10, -10, -10]

The filtered data often has smaller values and more repetition, making it more compressible.

2. DEFLATE Compression

After filtering, PNG uses DEFLATE compression, which is a combination of LZ77 and Huffman coding.

1. LZ77:
    - Looks for repeated sequences in the data
    - Replaces repeated sequences with references to previous occurrences
2. Huffman coding:
    - Assigns shorter codes to more frequent symbols
    - Further compresses the output from LZ77

Audio

Humans have a listening range of 20Hz to 20kHz (20,000Hz) and speech is around 2kHz to 4kHz. The threshold for pain is 120 decibels where **Decibels** is a logarithmic unit of sound pressure/intensity. Extremely high volumes can cause permanent damage.

Digital Noise increase by 6 decibels per bit.

1. Sine Wave
2. Frequency → Number of cycles per second (measured in Hertz).
3. Wavelength → Distance between two peaks. $1/frequency$
4. Amplitude → Highest point/peak of the wave.
5. Phase → Starting point of the wave.

CD Quality Audio: Requires 16-bit sampling at 44.1 kHz. Higher rates exist (e.g., 24-bit, 96 kHz) for audiophiles. Audio can be Mono (Single Channel) vs Stereo (Double Channel) and even Surround (5+1 or 7+1).

Audio Compression:

1. Lossy: MP3, AAC, Ogg Vorbis.
2. Lossless: FLAC, ALAC.

Lossy compression of Audio can introduce artifacts like Noise, Distortion and Attenuation.

Deepfake → Generation of video through AI using samples.
Audio Cloning → Copying someone else's voice.