File Compression and Analysis System

Mentor: Mr. Piyush Agarwal

Team Members:

Nupur Choudhary (Team Leader)

Dev Verma

Yuvika Gupta

Project Objectives

1 Design a robust system for compressing and decompressing files across text/PDF, images, and videos.

2 Implement lossless algorithms to ensure data integrity for text/PDF and select image formats.

3 Utilize lossy algorithms to optimize file size for images and videos.

4 Create an archiving mechanism to compress and decompress files while preserving their structure and metadata.
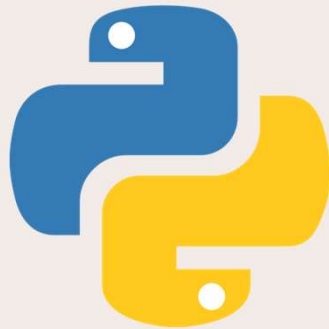
5 To visualize and analyze compression efficiency for better understanding and optimization.

TECHNOLOGY STACK

Backend

Python Libraries:

- Numpy
- OpenCV
- PIL
- JSON
- FFMPEG

Frontend

GUI:

- HTML5
- CSS3
- JavaScript

Visualization:

- Matplotlib

Project Overview

- A versatile system for file compression, tailored to file types:

- Text/PDF: Lossless LZ77 (.lz77) and Huffman Coding (.bin) preserve all data.

- Images: Lossy Color/Grayscale Quantization (.jpg) for size reduction; Lossless Deflate (.zlib) and PNG/WebP (.png/.webp) for quality.

- Videos: Lossy HEVC and AVC (.mp4) for high compression efficiency.

A versatile system for file compression, tailored to file types:

TXT **Text/PDF**
- Lossless LZ77 (.lz77)
- Huffman Coding (.bin)
  preserve all data

Images
- Lossy Color/Grayscale Quantization (.jpg)
  for size reduction

Videos
- Lossy HEVC and AVC (.mp4)
  for high compression efficiency

Workflow/Architecture

- Process Overview:

1. Select files (.txt, .csv, .pdf, .jpg, .png, .mp4).
2. Detect file type (text/PDF, image, video).
3. Choose compression algorithm:
   - Text/PDF: LZ77 or Huffman.
   - Images: ColorLossy, GrayscaleLossy, Deflate, or PNG/WebP.
   - Videos: HEVC or AVC.
4. Compress file and save output.
5. Optional decompression (LZ77, Huffman, Deflate).
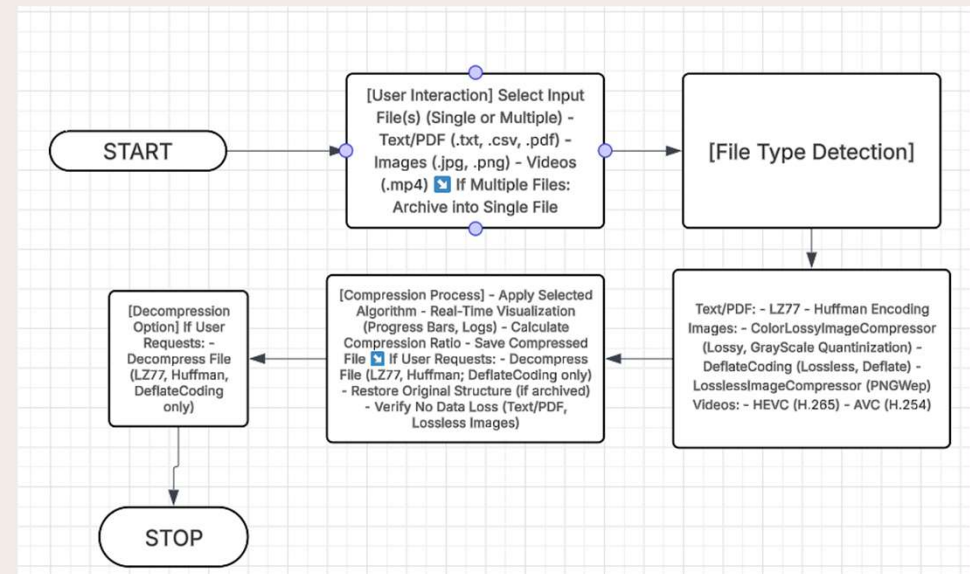
Roles and Responsibilities

Nupur Choudhary

- Designed and implemented LZ77 and Huffman algorithms for text/PDF
- Ensured lossless compression accuracy

Yuvika Gupta

- Developed four image compression methods (ColorLossy, GrayscaleLossy, Deflate, PNG/WebP)
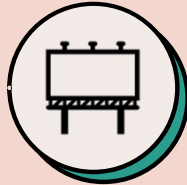- Optimized performance

Dev Verma

- Integrated HEVC and AVC codecs using FFmpeg
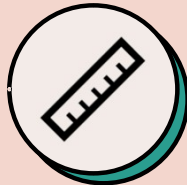- Built Tkinter-based file selection UI.
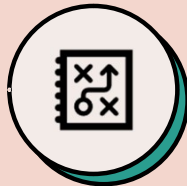
Current Achievements

Developed 8 compression algorithms for diverse file types

Text/PDF: LZ77 and Huffman ensure zero data loss for critical files.

Videos: HEVC and AVC deliver compact, high-quality outputs.

Images: Dual approach with lossy (Quantization) and lossless (Deflate, PNG/WebP) methods.

Implemented user-friendly file selection UI with Tkinter for videos.