Lab 03

School of Computer Science Engineering and Technology

Course	B. Tech.	Type	Core
Course Code	CSET301	Course Name	Artificial Intelligence and Machine Learning
Year	2025	Semester	Odd
Date	01/08/2025	Batch	2023–2027

CO-Mapping

	CO1	CO2	CO3	CO4	CO5	CO6
Q1		√		√		

Lab Topic: Preprocessing of Unstructured Data (Text, Video, and Audio)

Objective: **Total Marks: 0.5**

This lab is designed to introduce students to basic data preprocessing techniques for unstructured data specifically text, video, and audio. The objective is to clean, transform, and prepare each type of data to be suitable for downstream machine learning or deep learning tasks.

Problem Statement:

1. Text Data Preprocessing

Text URL: https://raw.githubusercontent.com/dscape/spell/master/test/resources/big.txt

Tasks:

- Load the text file from the provided URL using Python libraries like requests or directly using file reading methods. Remove unwanted characters such as special symbols, digits, and extra spaces to clean the raw text.
- Split the cleaned text into individual words or tokens using a tokenizer (e.g., from the nltk or spaCy library).
- Remove commonly used words (called stop words) like "the", "is", "and", which usually carry less meaningful information.
- Apply stemming to reduce words to their root form (e.g., "running" becomes "run") using tools like PorterStemmer.
- Count and display the most frequent words remaining after preprocessing to gain insights into the content.

2. Audio Data Preprocessing

Audio URL: https://github.com/Jakobovski/free-spoken-digit-dataset/raw/master/recordings/0_george_0.wav

Tasks:

- Load the audio file from the provided URL using an audio processing library such as librosa or scipy.
- Normalize the audio signal so that its amplitude lies within a consistent range (typically between -1 and 1) to improve processing stability.
- Visualize the audio waveform to understand the time-domain representation of the signal. You can use libraries like matplotlib for plotting.
- Extract MFCC (Mel Frequency Cepstral Coefficients) features from the audio. MFCCs are commonly used in audio and speech recognition tasks as they represent the short-term power spectrum of sound.

3. Video Data Preprocessing

Video URL: https://avtshare01.rz.tu-ilmenau.de/avt-vqdb-uhd-1/test_1/segments/bigbuck_bunny_8bit_15000kbps_1080p_60.0fps_h264.mp4

Tasks:

- Load the video from the given URL into your Python environment using a video processing library like OpenCV.
- Read the video frame by frame using a loop and store each frame for further processing.
- For every extracted frame, convert it from RGB (color) to grayscale using OpenCV functions.
- Resize each grayscale frame to a fixed size (e.g., 128×128 pixels) to maintain uniformity for analysis or model input