****

**Department of Computer Science**

**Lab Manual**

**of**

**SPEECH PROCESSING AND RECOGNITION**

**MAI574**

**Class Name: V MScAIML**

**Master of Science Artifitial Intelligence and Machine Learning**

**2025-26**

**Prepared by: Alex Kh Verified by: Faculty name:**

**Department Overview**

Department of Computer Science of CHRIST (Deemed to be University) strives to shape outstanding computer professionals with ethical and human values to reshape nation’s destiny. The training imparted aims to prepare young minds for the challenging opportunities in the IT industry with a global awareness rooted in the Indian soil, nourished and supported by experts in the field.

**Vision**

The Department of Computer Science endeavours to imbibe the vision of the University “**Excellence and Service”**. The department is committed to this philosophy which pervades every aspect and functioning of the department.

**Mission**

“To develop IT professionals with ethical and human values”. To accomplish our mission, the department encourages students to apply their acquired knowledge and skills towards professional achievements in their career. The department also moulds the students to be socially responsible and ethically sound.

**Introduction to the Programme**

Machines are gaining more intelligence to perform human like tasks. Artificial Intelligence has spanned across the world irrespective of domains. MSc (Artificial Intelligence and Machine Learning) will enable to capitalize this wide spectrum of opportunities to the candidates who aspire to master the skill sets with a research bent. The curriculum supports the students to obtain adequate knowledge in the theory of artificial intelligence with hands-on experience in relevant domains with tools and techniques to address the latest demands from the industry. Also, c andidates gain exposure to research models and industry standard application development in specialized domains through guest lectures, seminars, industry offered electives, projects, internships, etc.

**Programme Objective**

● To acquire in-depth understanding of the theoretical concepts in Artificial Intelligence and Machine Learning

● To gain practical experience in programming tools for Data Engineering, Knowledge Representation, Artificial intelligence, Machine learning, Natural Language Processing and Computer Vision.

● To strengthen the research and development of intelligent applications skills through specialization based real time projects.

● To imbibe quality research and develop solutions to the social issues.

Programme Outcomes:

PO1 : Conduct investigation and develop innovative solutions for real world problems in industry and research establishments related to Artificial Intelligence and Machine Learning PO2 : Apply programming principles and practices for developing automation solutions to meet future business and society needs.

PO3 : Ability to use or develop the right tools to develop high end intelligent systems

PO4 : Adopt professional and ethical practices in Artificial Intelligence application development

PO5 : Understand the importance and the judicious use of technology for the sustainability of the environment.

**MAI574– SPEECH PROCESSING AND RECOGNITION**

**Total Teaching Hours for Semester: 75 (3+4)**

**Max Marks:150                         Credits: 5**

**Course Objectives**

This course enables the learners to understand fundamentals of speech recognition, speech production and representation. It also enables the learners to impart knowledge on automatic speech recognition and pattern comparison techniques. This course helps the learners to develop automatic speech recognition model for different applications.

**Course Outcomes**

**After successful completion of this course students will be able to**

CO1: Understand the speech signals and represent the signal in time and frequency domain.

CO2: Analyze different signal processing and speech recognition methods.

CO3: Implement pattern comparison techniques and Hidden Markov Models (HMM)

CO4: Develop speech recognition system for real time problems.

**Unit-1         Teaching Hours: 15**

**FUNDAMENTALS OF SPEECH RECOGNITION**

Introduction- The Paradigm for Speech Recognition- Brief History of speech recognition research- The Speech Signal: The process of speech production and perception in human beings- the speech production system- representing speech in time and frequency domain- speech sounds and features.

**Lab Programs:**

1. Implement the task that takes in the audio as input and converts it to text.
2. Apply Fourier transform and calculate a frequency spectrum for a signal in the time domain.

**Unit-2                 Teaching Hours: 15**

**2.1 APPROACHES TO AUTOMATIC SPEECH RECOGNITION BY MACHINE:**

The acoustic phonetic approach-The pattern recognition approach-The artificial intelligence approach.

**2.2 SIGNAL PROCESSING AND ANALYSIS METHODS FOR SPEECH RECOGNITION:**

Introduction- spectral analysis models- the bank of filters front end processor- linear predictive coding model for speech recognition- vector quantization.

**Lab Programs:**

3. Implement sampling and quantization techniques for the given speech signals.

4. Explore linear predictive coding model for speech recognition

**Unit-3         Teaching Hours: 15**

**PATTERN COMPARISON TECHNIQUES**

Speech detection- distortion measure- Mathematical consideration- Distortion measure – Perceptual consideration- Spectral Distortion Measure- Incorporation of spectral dynamic feature into distortion measure- Time alignment and normalization.

**Lab Programs:**

1. Demonstrate different pattern in the given speech signal
2. Implement time alignment and normalization techniques

**Unit-4         Teaching hours: 15**

**THEORY AND IMPLEMENTATION OF HIDDEN MARKOV MODELS:**

Introduction- Discrete time Markov processes- Extension to hidden Markov Models- Coin - toss models- The urn and ball model- Elements of a Hidden Markov Model- HMM generator of observation- The three basic problems for HMM’s- The Viterbi algorithm- Implementation issues for HMM’s.

**Lab Programs:**

7. Implement simple hidden Markov Model for a particular application.

8.Apply Viterbi dynamic programming algorithm to find the most likely sequence of hidden states.

**Unit-5         Teaching Hours: 15**

**TASK ORIENTED APPLICATION OF AUTOMATIC SPEECH RECOGNITION:**

Task specific voice control and dialog- Characteristics of speech recognition applications- Methods of handling recognition error- Broad classes of speech recognition applications- Command and control applications- Voice repertory dialer- Automated call-type recognition- Call distribution by voice commands- Directory listing retrieval- Credit card sales validation.

**Lab Programs:**

9. Demonstrate automatic speech recognition for Call distribution by voice commands.

10. Apply speech recognition system to access telephone directory information from spoken spelled names (Directory listing retrieval).

**Text Books and Reference Books**

[1] Fundamentals of Speech Recognition, Lawrence R Rabiner and Biing- Hwang Juang. Prentice-Hall Publications, 20209.

[2] Introduction to Digital Speech Processing, Lawrence R. Rabiner, Ronald W. Schafer, Now Publishers, 2015.

**Essential Reading / Recommended Reading**

1. Intelligent Speech Signal Processing, Nilanjan Dey, Academic Press, 2019.
2. Speech Recognition-The Ultimate Step-By-Step Guide, Gerardus Blokdyk, 5STARCooks,2021.
3. Automatic Speech Recognition- A Deep Learning Approach, Dong Yu, Li Deng, Springer-Verlag London, 2015.

**Web Resources:**

1. [www.w3cschools.com](http://www.w3cschools.com)
2. <https://www.simplilearn.com/tutorials/python-tutorial/speech-recognition-in-python>
3. <https://realpython.com/python-speech-recognition/>
4. <https://cloud.google.com/speech-to-text/docs/tutorials>
5. <https://www.coursera.org/courses?query=speech%20recognition>
6. <https://pylessons.com/speech-recognition>

CO – PO Mapping

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | PO1 | PO2 | PO3 | PO4 | PO5 |
| CO1 | 3 | 1 |  |  | 1 |
| CO2 | 2 | 2 |  |  | 1 |
| CO3 | 1 | 3 | 1 |  | 2 |
| CO4 | 1 | 1 | 3 | 1 | 3 |

**LIST OF PROGRAMS**

MCA 2023-2024

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl.no | Title of lab Experiment | Page number | RBT | CO |
| 1 | **Sampling and Reconstruction of Speech Signals** |  | L3 | CO1,3 |
| 2 | **Fourier Transform and Frequency Spectrum Analysis of Signals** |  | L3 | CO2,3 |
| 3 | **Speech-to-Text Application for Accessibility** |  | L3 | CO3 |
| 4 |  |  | L3 | CO3 |
| 5 |  |  | L3 | CO3 |
| 6 |  |  | L3 | CO3,4 |
| 7 |  |  | L4 | CO3 |
| 8 |  |  | L3 | CO3 |
| 9 |  |  | L3 | CO4 |
| 10 |  |  | L5 | CO4 |

**Evaluation Rubrics:**  
(1) Implementation: 5 marks.  
(2) Complexity and Validation: 3 marks.  
(3) Documentation & Writing the inference: 2 marks.

**Submission Guidelines:**

* Make a copy of the lab manual template with your <name\_reg:no\_subject name >,
* Copy the given question and the answer (lab code) with results, followed by the conclusion of that lab. Title the lab as lab number.
* Keep updating your lab manual and show the lab manual of that particular lab for evaluation.
* Create a Git Repository in your profile <SPR lab-reg no> . Follow a different branch for each lab <Lab 1, Lab 2…>, and push the code to Git. The link should be provided in Google Classroom along with the PDF of the lab manual.
* Upload the PDF to Google Classroom before the deadline.

**Lab 1**

**Lab Exercise I: Sampling and Reconstruction of Speech Signals**

### **Aim**

To study sampling and reconstruction of speech signals at different sampling rates, evaluate reconstruction using zero-order hold and linear interpolation, and implement the source-filter model to analyze the effect of filtering, sampling, and reconstruction on speech quality.

**(1) Implement sampling and quantization techniques for the given speech signals.**(a) Plot the time domain representation of the original speech signal.  
(b) Sample the speech signal at different sampling rates (e.g., 8kHz, 16kHz, and 44.1kHz).  
(c) Plot sampled speech signal for each of these sampling rates.  
(d) Using the sampled signals from above, reconstruct the signal using:  
(i) Zero-order hold (nearest-neighbor interpolation)  
(ii) Linear interpolation.  
(e) Calculate the Mean Squared Error (MSE) between the original and the reconstructed signals for both methods.  
Write an inference on how sampling rates affect the quality and accuracy of the reconstructed speech signal.

**(2) Implement the source-filter model for a given speech signal and analyze the impact of sampling and reconstruction on the quality of the speech signal.**  
(a) Generate a synthetic speech signal using the source-filter model.  
(i) Create a source signal (e.g., a glottal pulse train for voiced sounds or white noise for unvoiced sounds).  
(ii) Apply a filter that models the vocal tract, represented by an all-pole filter or an FIR filter with formants (resonances of the vocal tract).  
(b) Plot the generated speech signal and analyze the effect of the filter on the original source.  
(c) Sample the speech signal generated above at different sampling rates (e.g., 8 kHz, 16 kHz, 44.1 kHz).  
(d) Reconstruct the signal using a suitable interpolation method (e.g., zero-order hold, linear interpolation).  
(e) Compute the Mean Squared Error (MSE) between the original and reconstructed speech signals.  
Write an inference on tasks such as creating the source-filter model, different sampling rates, and reconstruction of the sampled signals.

**Code with Results**

**A screen shot of a computer

AI-generated content may be incorrect.**

**A computer screen shot of a computer code

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer screen

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.A screen shot of a computer program

AI-generated content may be incorrect.A screen shot of a computer

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.**

**Conclusion /inference**

In this lab, we studied the effects of sampling and reconstruction on speech signals and analyzed the speech production process using the source–filter model. The experiments showed that higher sampling rates such as 44.1 kHz preserve more spectral details of speech, including harmonics and formant structures, resulting in clearer and more natural reconstructions. In contrast, lower sampling rates like 8 kHz discard high-frequency components due to Nyquist limitations, producing muffled or degraded speech quality. Reconstruction using zero-order hold introduced stair-step distortions and higher error, whereas linear interpolation gave smoother results and lower mean squared error, though neither method could recover information lost during undersampling. Through the source–filter model, we confirmed that speech can be represented as an excitation source shaped by vocal tract resonances, and observed that sampling and reconstruction directly affect the preservation of these formant features. Overall, the study concludes that both the sampling rate and reconstruction method significantly influence speech quality, with higher rates and linear interpolation yielding more accurate and intelligible results.

Lab Exercise 2:

**Fourier Transform and Frequency Spectrum Analysis of Signals**

Aim

To study the Fourier Transform and analyze the frequency spectrum of different signals (sinusoidal,

composite, exponential, and rectangular). To compare their time-domain representation with their

frequency-domain characteristics using both the Discrete-Time Fourier Transform (DTFT) and the

Discrete Fourier Transform (DFT).

Questions

Question 1

(a) Generate a basic sinusoidal signal in the time domain. (For example, generate a sine wave with a

frequency of 5 Hz, sampled at 1000 Hz.)

(b) Plot the time-domain waveform of the signal.

(c) Compute the Discrete-Time Fourier Transform (DTFT) and plot the continuous frequency spectrum.

(d) Compute the Discrete Fourier Transform (DFT) and plot the discrete frequency spectrum.

Question 2

(a) Generate a composite signal by adding two or more sinusoidal signals of different frequencies and

amplitudes.

(b) Plot the time-domain waveform of the composite signal.

(c) Compute the Discrete-Time Fourier Transform (DTFT) and plot the continuous frequency spectrum.

(d) Compute the Discrete Fourier Transform (DFT) and plot the discrete frequency spectrum.

Question 3

(a) Generate an exponentially decaying signal.

(b) Plot the time-domain waveform.

(c) Compute the Discrete-Time Fourier Transform (DTFT) and plot the continuous frequency spectrum.

(d) Compute the Discrete Fourier Transform (DFT) and plot the discrete frequency spectrum.

(e) Analyze the relationship between the time-domain waveform and the frequency-domain

representation.

Question 4

(a) Generate a rectangular pulse signal of finite duration in the time domain.

(b) Plot the time-domain waveform.

(c) Compute the Discrete-Time Fourier Transform (DTFT) and plot the continuous frequency spectrum.

(d) Compute the Discrete Fourier Transform (DFT) and plot the discrete frequency spectrum.

(e) Analyze the relationship between the time-domain waveform and the frequency-domain

representation.

**Code with results**

**A screen shot of a graph

AI-generated content may be incorrect.**

**A screen shot of a computer screen

AI-generated content may be incorrect.A screen shot of a graph

AI-generated content may be incorrect.A screenshot of a computer screen

AI-generated content may be incorrect.**

**Inference:**

In this lab, we studied the Fourier Transform and analyzed the frequency spectrum of different signals—sinusoidal, composite, exponential, and rectangular. The experiments showed that simple sinusoids have sharp spectral peaks at their frequencies, while composite signals exhibit multiple peaks corresponding to their components. The exponentially decaying signal demonstrated that faster decay in the time domain leads to a broader frequency spectrum, while slower decay results in a narrower spectrum. The rectangular pulse revealed the time–frequency trade-off, where short-duration pulses spread widely in frequency and longer pulses concentrate energy in narrower bands. Overall, the lab highlighted the fundamental relationship between time-domain behavior and frequency-domain characteristics.

Lab Exercise 3:

**Question: Speech-to-Text Application for Accessibility**

Aim:

To develop a Python-based speech-to-text system that converts spoken commands into text in real time,

provides meaningful user feedback, handles errors gracefully, and allows comparison of different

recognition methods.

This is an open-ended question. You can explore more.

Scenario:

You have been hired as an AI engineer by a tech startup that focuses on enhancing accessibility for people

with disabilities. One of your key responsibilities is to develop a system that allows users to control

devices and input text via voice commands.

The first version of this system requires you to implement a speech-to-text application that converts

spoken commands into text in real time. This will serve as the foundation for future projects, such as

integrating the system with smart devices or accessibility software.

Tasks:

1. Audio Capture(mandatory task )

○ Record spoken input using a microphone, OR use any speech audio file (e.g., .wav,

.flac).

https://drive.google.com/file/d/1BmlRHKnHWVtlM743vcjLGtaK\_aRzd\_Qa/view?usp=d

rive\_link

○ Provide feedback to the user: "Speak something...".

2. Convert Speech to Text(mandatory task )

○ Implement a speech-to-text system using at least two methods (for comparison):

■ Offline: Whisper, Vosk, or similar

■ Online: Google Speech API

○ Display the message: "Recognizing..." while processing.

3. Display Recognized Text(mandatory task )

○ Show the converted text on the screen.

Example: "Speech recognized: 'Turn on the lights in the living room.'"

○ Display "Speech successfully converted to text!" on successful recognition.

4. Handle Errors and Exceptions(mandatory task )

○ Unclear speech (mumbling, low volume): Display a user-friendly message.

Example: "Speech Recognition could not understand audio. Please try speaking more

clearly."

○ Service unavailability (internet/API down): Display an appropriate error message.

5. Provide Feedback at Each Stage

○ Before recording: "Speak something..."

○ During recognition: "Recognizing..."

○ On success: "Speech successfully converted to text!"

○ On failure: Provide meaningful error messages.

**Code with Results:**

A computer screen shot of text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Comparative Analysis Table**

| **Audio Type / Condition** | **Whisper Output** | **Vosk Output** | **Google Cloud Output** | **Coqui STT Output** | **CMU Sphinx Output** | **Notes on Accuracy** |
| --- | --- | --- | --- | --- | --- | --- |
| **Clear male voice** | High accuracy, few errors | High accuracy | Very high accuracy | Moderate to high accuracy | Moderate accuracy | Whisper and Google perform best; CMU Sphinx slightly less accurate |
| **Clear female voice** | High accuracy | High accuracy | Very high accuracy | Moderate accuracy | Moderate accuracy | Google and Whisper handle female voices well; Coqui slightly less precise |
| **Fast speech** | High, slight omissions | Moderate | High | Low to moderate | Low | Whisper and Google handle fast speech better; CMU Sphinx struggles |
| **Noisy background** | Moderate | Low to moderate | High | Low | Low | Google Cloud is most robust to noise; Whisper tolerates moderate noise; others degrade |
| **Soft voice** | High | Moderate | High | Moderate | Low | Whisper and Google handle soft input better; Sphinx often misses words |

**Inference**

The Whisper (base) model is an offline solution that is robust under most conditions, handling fast or soft speech effectively, though it may struggle slightly in very noisy environments. Vosk, also offline, performs well with clear speech but its accuracy decreases with fast or noisy audio. Google Cloud, being an online API, delivers very high accuracy across nearly all scenarios and is particularly robust to noise and soft voices, although it requires an internet connection. Coqui STT is an offline, lightweight solution suitable for embedded or low-power systems, with performance largely dependent on the quality of the microphone and recording environment. CMU Sphinx, also offline and very lightweight, works well for simple, clear audio but tends to struggle with soft voices, background noise, or rapid speech.