COURSE CONTENT

DIGITAL SIGNAL AND IMAGE PROCESSING LABORATORY								
VI Semester: ECE								
Course Code	**Category**	**Hours / Week**			**Credits**	**Maximum Marks**		
AECD45	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		-	-	2	1	40	60	100
Contact Classes: Nil	**Tutorial Classes: Nil**	**Practical Classes: 45**				**Total Classes: 45**		
Prerequisite: Signals and Systems								

I. COURSE OVERVIEW:

This laboratory course is concerned with the implementation of digital signal processing algorithms using different computational platforms such as MATLAB and Python tools that give core knowledge to develop the real time applications in signal and image processing. It focuses on the convolution, discrete Fourier transform, fast Fourier transform algorithms, digital filter design and image processing techniques. Digital signal and image processing applications are used in speech processing, image processing, audio and video data compression, and communication systems.

II. COURSES OBJECTIVES:

The students will try to learn:

I. The Python programming for signal and image processing applications.
II. The implementation of fundamental digital signal processing techniques, such as filtering, transformations, and spectral analysis.
III. The concepts of image processing including enhancement, segmentation, and feature extraction.
IV. The how to design and analyze the algorithms for real-time signal and image processing tasks.
V. The concepts of digital signal and image processing to practical problems.

III. COURSE OUTCOMES:

At the end of the course students should be able to:

CO1 Analyze signals in the time and frequency domains using techniques like FFT, DFT, and filtering
CO2 Develop Python-based projects that integrate signal and image processing techniques to solve engineering problems.
CO3 Implement basic operations on signals and images using Python libraries such as NumPy, SciPy, OpenCV, and Matplotlib.
CO4 Apply spatial and frequency domain techniques for image enhancement and restoration.
CO5 Design and implement image segmentation techniques using edge detection, thresholding, and clustering
CO6 Evaluate and optimize algorithms for real-time signal and image processing applications.

IV. LIST OF EXPERIMENTS:

WEEK-1: DISCRETE-TIME SYSTEMS
Analysis and implementation of discrete-time systems using MATLAB to understand the behavior of the systems through practical computation and visualization.

WEEK-2: FIR AND IIR FILTER DESIGN
Design of Low Pass, High Pass, and Band Pass FIR filters, as well as Butterworth and Chebyshev IIR filters using MATLAB.

WEEK-3: OPTIMUM EQUIRIPPLE FIR DIGITAL FILTERS
Design of optimum equiripple FIR digital filters using MATLAB to analyze their advantages in signal processing applications.

WEEK-4: FIR DIGITAL FILTER USING WINDOW METHOD
Design and implementation of FIR digital filters using the window method in MATLAB.

WEEK-5: IIR DIGITAL FILTER USING BUTTERWORTH AND BILINEAR TRANSFORMATION

Design and analysis of IIR digital filters using the Butterworth design and Bilinear Transformation techniques in MATLAB.

WEEK -6: SAMPLING RATE CONVERSION
Design and implementation of sampling rate conversion using MATLAB.

WEEK -7: SINE WAVE GENERATION
Generation of sine waves using a lookup table and designing of FIR filters using the frequency sampling method in MATLAB.

WEEK -8: INTRODUCTION TO IMAGE PROCESSING
Design and simulation of basic image processing techniques using Python.

WEEK -9: ARITHMETIC OPERATIONS ON IMAGES
Design of arithmetic operations on images are performed using Python.

WEEK -10: HISTOGRAM OF AN IMAGE
Generating and analyzing the histogram of an image using Python.

WEEK -11: IMAGE ENHANCEMENT
Design and simulation of various image enhancement techniques using Python.

WEEK -12: COMPRESSION OF AN IMAGE
Design and simulate image compression algorithms using Python.

WEEK -13: IMAGE SEGMENTATION
Design and implement various image segmentation techniques using Python.

WEEK -14: MORPHOLOGICAL PROCESSING FOR IMAGE SEGMENTATION
Implementation of image segmentation using morphological processing techniques using Python.

V. TEXT BOOKS:

1. Discrete-Time Signal Processing, I second edition, A. V. Oppenheim, R. W. Schafer, J. R. Buck, Prentice Hall, 1999.
2. Real-Time Digital Signal Processing, from MATLAB to C With TMS320C6x DSK,î T. B. Welch, C. H.
3. G. Wright, and M. G. Morrow, Taylor & Francis, 2006.
4. Real-time digital signal processing: Implementations and applications, S. M. Kuo, B. H. Lee, and W. Tian, second edition, John Wiley & Sons, 2006.
5. Anil K. Jain, “Fundamentals of Digital Image Processing”, Pearson, 2002.

VI. REFERENCE BOOKS:

1. Digital Signal Processing Implementation using the TMS320C6000TM Platform, Naim Dahnoun, Prentice Hall, 2000.
2. Digital Signal Processing Using MATLAB , Sanjit Mitra, McGraw-Hill, 1999.
3. Proakis and Manolakis, “Digital Signal Processing”, 3rd edition, Prentice Hall, 1996.
4. Little and Shure, “Signal Processing TOOLBOX User’s Guide for use with MATLAB”, MathWorks, Inc., 1992.

VII. ELECTRONICS RESOURCES:

1. https://ccrma.stanford.edu/~jos/sasp/Example_Overlap_Add_Convolution.html
2. https://rt-dsp.com/3rd_ed/app_a/App_CCS_5_1_dsk6713.pdf
3. Mathworks. The Mathworks Web Site: http://www.mathworks.com/. Technical report, Mathworks, 2001
4. W. Eaton. Octave web site http://www.che.wisc.edu/octave. Technical report, Univ. Wisconsin, 2001.
5. G. Campbell. DataLab Users’ Manual. Technical report, University of Ulster, Interactive Systems Centre, Report isc/94/015/r, available from http://www.cs.qub.ac.uk/~J.Campbell/dl/dlprg.a, 1994.
6. https://sisu.ut.ee/imageprocessing/documents
7. https://www.geeksforgeeks.org/reading-image-opencv-using-python/

VIII. MATERIALS ONLINE

1. Course Content
2. Lab Manual