CURRICULUM & SYLLABI

M.Tech. EMBEDDED & MACHINE LEARNING SYSTEMS

Effective from AY: 2024-25



NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL

WARANGAL, TELANGANA

Table of Contents

VISION AND MISSION OF THE INSTITUTE 3
VISION AND MISSION OF THE DEPARTMENT 3
PROGRAM EDUCATIONAL OBJECTIVES 4
PROGRAM OUTCOMES 5
CURRICULUM 6
THE OVERALL CREDIT STRUCTURE 9
SYLLABI 10
PROFESSIONAL ELECTIVES 43

Vision and Mission of the Institute

National Institute of Technology Warangal

VISION

Towards a Global Knowledge Hub, striving continuously in pursuit of excellence in Education, Research, Entrepreneurship and Technological services to the society

MISSION

- Imparting total quality education to develop innovative, entrepreneurial and ethical future professionals fit for globally competitive environment.
- Allowing stake holders to share our reservoir of experience in education and knowledge for mutual enrichment in the field of technical education.
- Fostering product-oriented research for establishing a self-sustaining and wealth creating centre to serve the societal needs.

Vision and Mission of the Department

Electronics and Communication Engineering

VISION

Create an Educational environment to prepare the students to meet the challenges of modern electronics and communication Industry through state of art technical knowledge and innovative approaches.

MISSION

- To create learning, Development and testing environment to meet ever challenging needs of the Electronic Industry.
- To create entrepreneurial environment and industry interaction for mutual benefit.
- To be a global partner in training human resources in the field of chip design, instrumentation and networking.
- To associate with international reputed institution for academic excellence and collaborative research

M.Tech. – Embedded & Machine Learning Systems (EMS)

Program Educational Objectives

PEO-1	Conceptualize and prescribe the design flow of an embedded machine learning system for a given application.
PEO-2	Acquire knowledge and skill on machine learning algorithms to solve the current challenges in the field of signal and image processing
PEO-3	Analyze, model, design and prototype the machine learning systems with hardware accelerators to meet the specifications
PEO-4	Enhance knowledge to design & develop constrained edge hardware platforms to for machine learning applications
PEO-5	Pursue life-long learning as a means of enhancing the knowledge base and skills necessary to serve the engineering and scientific community.
PEO-6	Contribute as an individual or a member of a team in product-oriented research and demonstrate leadership skills

Program Articulation Matrix

PEO / Mission Statements	**PEO-1**	**PEO-2**	**PEO-3**	**PEO-4**	**PEO-5**	**PEO-6**
To create learning, Development and testing environment to meet ever challenging needs of the Electronic Industry.	**3**	**3**	**3**	**3**		
To create entrepreneurial environment and industry interaction for mutual benefit.	**1**	**2**	**2**	**3**		
To be a global partner in training human resources in the field of chip design, instrumentation and networking.		**1**	**1**	**3**	**2**	**2**
To associate with international reputed institution for academic excellence and collaborative research.				**2**	**3**	**1**

1-Slightly; **2-Moderately;** **3-Substantially**

M.Tech. – Embedded & Machine Learning Systems (EMS)

Program Outcomes

PO-1	Analyze design methodologies for machine learning algorithms on embedded hardware platforms
PO-2	Demonstrate technical skills to handle advanced signal/image processing applications using machine learning models
PO-3	Design, test and validate efficient hardware accelerators for deep learning models for variety of applications
PO-4	Develop parallel programming skills for CUDA environment on GPU architecture in accelerating machine learning algorithms.
PO-5	Characterize and integrate sensors with machine intelligence to solve the real-world applications
PO-6	Communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large

CURRICULUM

M.Tech. Embedded & Machine Learning Systems

1st Semester

S. No.	Code	Course Title	L-T-P	Credits
1	EC16001	Embedded Hardware Platforms and programming	3-0-0	3
2	EC16003	Machine Learning Algorithms and Applications	3-0-0	3
3	EC16005	FPGA Based Systems Design	3-0-2	4
4	EC160XX	Professional Elective – I	3-0-0	3
5	EC160XX	Professional Elective – II	3-0-0	3
6	EC16007	Embedded systems Lab	0-1-2	2
7	EC16009	Machine Learning Lab	0-1-2	2
			Total Credits	**20**

2nd Semester

S. No.	Code	Course Title	L-T-P	Credits
1	EC16002	GPU Architectures and Programming	3-0-0	3
2	EC16004	Hardware Accelerators Design for Machine Learning models	3-0-0	3
3	EC160XX	Professional Elective – III	3-0-0	3
4	EC160XX	Professional Elective – IV	3-0-0	3
5	EC160XX	Professional Elective – V	3-0-0	3
6	EC16006	GPU Programming Lab	0-1-2	2
7	EC16008	Hardware accelerators for machine learning Lab	0-1-2	2
8	EC16096	Minor Project	0-0-0	2
			Total Credits	**21**

3rd Semester

S. No.	Code	Course Title	L-T-P	Credits
1	EC16089	Seminar and Technical Writing	0-0-0	2
2	EC16093	Comprehensive Viva-Voce	0-0-0	2
3	EC16097	Dissertation: Part-A	0-0-0	8
4	EC16091	Summer Internship/Research Experience	0-0-0	2
			Total Credits	**14**

4th Semester

S. No.	Code	Course Title	L-T-P	Credits
1	EC16098	Dissertation: Part-B	0-0-0	12
			Total Credits	**12**

Professional Elective Courses:

Professional Elective – I		
S. No.	**Code**	**Course Title**
1	EC16021	Embedded RTOS
2	EC16023	Statistical Modelling and Optimization Techniques
3	EC16025	Machine Learning for Signal Processing
4	EC16027	IoT Architecture & Computing

Professional Elective – II		
S. No.	**Code**	**Course Title**
1	EC16029	VLSI DSP Architectures
2	EC16033	Machine Learning in Robotics
3	EC36025	VLSI System Design

Professional Elective – III		
S .No.	**Code**	**Course Title**
1	EC16022	Hardware and Software Co-design of Embedded Systems
2	EC16024	Machine Vision and Pattern Recognition
3	EC16026	Wireless Technologies
4	EC26026	Advanced Computer Architecture

Professional Elective – IV		
S. No.	**Code**	**Course Title**
1	EC16028	Synthesis and Optimization of Digital Circuits
2	EC16030	Tiny ML & Edge AI
3	EC16032	Applied Machine Learning: Ensemble Learning
4	EC16034	Verification Methodologies: UVM

Professional Elective – V		
S. No.	**Code**	**Course Title**
1	EC16036	Hardware Security for Embedded Systems
2	EC16038	Selected Topics in Machine Learning Systems
3	EC16040	Artificial Intelligence in Biomedical Applications
4	EC16042	Machine Learning for Speech and Audio Processing
5	EC16044	Deep Learning Architectures

The Overall Credit Structure

Course Category	**Credits**
Program Core	24
Professional Elective	15
Minor Project	2
Seminar and Technical Writing	2
Summer Internship / Research Experience	2
Comprehensive Viva-Voce	2
Dissertation	20
Total Graded Credit Requirement	67

SYLLABI

M. Tech. EMBEDDED & MACHINE LEARNING SYSTEMS

1st Semester

EC16001 **3-0-0 (3)**

Embedded Hardware Platforms and Programming

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand the embedded system design challenges
CO-2	Familiarize with the ARM processor architecture and programming
CO-3	Understand the various communication interfaces and protocols
CO-4	Explore the various hardware platforms for AI& ML applications
CO-5	Develop the software drivers for embedded hardware components and communication protocols

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	1	1	1	2	1
CO-2	2	1	1	2	2	1
CO-3	2	1	1	2	2	1
CO-4	3	2	2	3	2	1
CO-5	2	2	1	2	2	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to Embedded Systems: Embedded systems Overview, Characteristics of embedded computing applications. Design Challenges, Common Design Metrics, Embedded systems Design flow. Processor Technology, IC Technology, Trade-offs. Unified Modeling Language (UML), Activity, diagram, class diagram, component diagram, use-case diagram, sequence diagram. UML specification examples.

Introduction to ARM Processors: Introduction to ARM processors, Evolution of ARM processors, pipeline organization, ARM Processor cores and CPU cores. Introduction to ARM Cortex-M Processors, ARM Cortex-M4 processor ‘s architecture, Programmer‘s model, Special registers, Operation Modes, Memory map, Memory access attributes and overview of Interrupts and exceptions. Keil Microcontroller Development Kit for ARM, Typical program compilation flow, Sample arithmetic and logical assembly language programs

Embedded Systems Interfacing: Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C), RS-232, USB, CAN, IrDA, Bluetooth, Zigbee, AMBA bus protocols.

Embedded hardware platforms: Introduction and specifications of hardware SBC platforms for AI and ML applications: Raspberry Pi 4, Jetson Nano, Jetson Xavier NX, Jetson AGX Xavier, Google Coral, Rock Pi, Beagle Bone AI and PYNQ boards.

Programming on Embedded hardware platforms: Board setup, installation of libraries, and application code development in Python, C and C++ on different embedded hardware platforms.

Learning Resources:

Text Books:

1. Joseph Yiu, The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors, Newnes Publications; 2013, 3rd edition,.
2. Ata Elahi-Trever Arjeski, —ARM Assembly language with hardware experimentǁ, Springer Int. Publishing, 2015.
3. Santanu Chattopadhya, —Embedded System Designǁ, PHI, 2023,2nd Edition.
4. Articles on Embedded hardware platforms and programming.

Reference Books:

1. Wrox, — Professional Embedded ARM Developmentǁ
2. William hohl and Christoper Hinds, —ARM assembly language fundamentals and Techniquesǁ CRC,2nd edition,2015.

EC16003 **3-0-0 (3)**

Machine Learning Algorithms and Applications

Pre-Requisites: **None**

Course Outcomes:

CO-1	Design and implement machine learning solutions to regression, classification, and clustering problems.
CO-2	Appreciate the underlying mathematical relationships within and across Machine Learning algorithms and the paradigms of supervised and un-supervised learning
CO-3	Apply the algorithms to a real-world problem, optimize the models learned and report on the expected accuracy that can be achieved by applying the models
CO-4	Build and train deep neural networks and identify key architecture parameters
CO-5	Implement deep learning algorithms and solve real-world problems

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	1	1	1	2	1
CO-2	2	2	1	1	2	1
CO-3	2	2	1	3	3	1
CO-4	2	2	1	2	2	1
CO-5	2	1	1	2	2	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Linear Regression: Basics of linear regression, its assumptions, limitations and industry applications. Least square based and Gradient Descent Based Regression, Multiple linear regression, Polynomial regression.

Logistic Regression: Univariate and Multivariate Logistic Regression for classification ML. Implementation in Python, evaluation metrics and industry applications are covered.

Classification Algorithms : classification algorithms, naïve Bayes: the basics of Bayes Theorem, Naïve Bayes classifier and implementation in a Spam-Ham classifier, K-Nearest Neighbours Computational geometry; Voronoi Diagrams, Delaunay Triangulations, K-Nearest Neighbour algorithm; Wilson editing and triangulations

Support Vector Machine: the SVM algorithm, its working, kernels and implementation. , Tree Models: Basics of Tree models, their structure, splitting techniques, pruning and ensembles to form Random Forests are covered here.

Unsupervised Learning-Clustering: Distance measures, Different clustering methods (Distance, Density, Hierarchical), Iterative distance-based clustering; K-Means Clustering Algorithm and Image Quantization, basics of Principal Component Analysis its working and implementation in Python.

Introduction to Neural Networks: The Intuition of Neural Networks, Activation Functions in Neural Networks, Fully Connected Layer and Back Propagation, Image Classification, Loss Functions, Optimization and Regularization: Overfitting and Capacity, Cross Validation, Feature, Selection, Regularization, Hyperparameters

Convolutional Neural Networks: The CNN, its structure, layers and working. Implementation of CNN using Keras, Image classification and hyper-parameter tuning

Learning Resources:

Text Books:

1. Christopher Bishop, Pattern Recognition and Machine Learning, Springer 2006.
2. M.Gopal, Applied Machine Learning, Mc Graw Hill education,2018

Reference Books:

1. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Introduction to Statistical Learning, Springer, 2013.
2. Richard Duda, Peter Hart, David Stork, Pattern Classification, John Wiley & Sons, 2001, 2nd Ed..
3. Tom M. Mitchell, Machine Learning, Tata McGraw Hill Education (India) 2013

EC16005 **3-0-2 (4)**

FPGA Based Systems Design

Pre-Requisites: **None**

Course Outcomes:

CO-1	Analyze the Digital Circuits for different applications
CO-2	Understand the architectures of programmable logic devices for implementing digital systems
CO-3	Design, develop and implement the digital circuits on FPGAs using VHDL and Verilog coding
CO-4	Develop test strategies for Digital Systems
CO-5	Design robust Digital Systems

Mapping of Cos and Pos:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	2	2	1	1	1
CO-2	1	1	3	1	1	1
CO-3	2	2	3	2	2	1
CO-4	1	1	2	2	2	1
CO-5	1	1	2	2	2	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction: Fpgas/Cplds: Evolution of programmable devices, FPGA Design flow, Commercially available FPGA/CPLD, Building blocks of FPGAs/CPLDs, Configurable Logic block functionality, Routing structures, Input/output Block, Impact of logic block functionality on FPGA performance, Model for measuring delay.

Design Using Hdl's: VHDL and Verilog modeling concepts, Behavioral, Dataflow and Structural architecture descriptions: Concurrent and Sequential statements, Event driven Simulation.

Digital Designs Building Blocks: Tristate buffers, multiplexers, latches, flip-flops, registers, counters, arithmetic and logic circuits (ALU Design), Finite State Machines. Adder/Subtraction, Divisors, Multipliers, Parallel prefix adders using Signed Magnitude, Complement formats & IEEE Floating Point Arithmetic's.

Design Methodology: Synchronous Systems, Top-Down Design, Register Transfer Level Design, Algorithmic State Machines, and Synthesis. Design Pitfalls. Concepts of CISC & RISC MIPS Processors.

Implementation Issues: Design Case Studies using ASM**,** Hardware Testing & Design for testability, test vectors, fault analysis for Combinational and Sequential Circuits. Microprogramming.

Learning Resources:

Text books:

1. Wayne Wolf, FPGA based digital system design, Published by Prentice-Hall.2015.
2. Cem Ünsalan Yeditepe,The Ohio State University, Digital System Design with FPGA Implementation Using Verilog and VHDL, 2017,TMH Publishers

Reference Books:

1. John V. Old Field, Richrad C. Dorf, Field Programmable Gate Arrays, Wiley, 2008.
2. Charles H.Roth,Jr., LIzy Kurian John, Digital Systems Design, The University of Texas at Austin,2007.
3. William J Dally and John W Poulton, Digital System Engineering Published by Cambridge University Press,1998.
4. Joseph Cavanaugh, Digital Design and Verilog fundamentals: Cambridge University Press Published by CRC Press,2008,1st Edition.
5. Franklin P Processer and David E Winkel, The Art of Digital Design: An Introduction to Top down design: Published by Prentice-Hall,1987,2nd Edition.

EC16007 **0-1-2 (2)**

Embedded Systems Lab

Pre-requisites: EC16001

Course Outcomes:

CO-1	write assembly language and C programs for arithmetic operations
CO-2	Interface I/O devices ADC, DAC, Keypad, SSD, with microcontroller-based systems
CO-3	Develop the driver programs to on-chip and external Peripheral devices
CO-4	Develop the communication software to on-chip and external communication interfaces UART, I2C,SPI ,wi-fi and bluetooth

Mapping of course outcomes with program outcomes:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	1	1	2	2	1	1
CO-2	2	1	2	2	1	1
CO-3	2	1	2	2	1	1
CO-4	2	1	2	2	1	1

List of Experiments:

1. Write a simple program for arithmetic operations – addition, subtraction, multiplication and division of 16 or 32 – bit numbers
2. Flashing of LEDS using Shift Register
3. Interfacing ADC
4. Interfacing DAC
5. Interfacing 7-Segment LED.
6. Interfacing of Analog Keypad.
7. Interrupt using on board push button
8. Interfacing real time clock.
9. Interfacing stepper motor.
10. Interfacing temperature sensor.
11. Interfacing Bluetooth module.
12. Interfacing Real Time Clock
13. Interfacing of micro SD Card.
14. Interfacing Wi-Fi Module
15. On-Chip Timer Programming
16. On-Chip I2C and SPI Programming
17. On-Chip UART Programming
18. On-Chip PWM and ADC Programming

EC16009 **0-1-2 (2)**

Machine Learning Lab

Pre-Requisites: None

Course Outcomes:

CO-1	Articulate the strengths, weaknesses and assumptions of a selection of machine learning algorithms in relation to structured and unstructured data
CO-2	Implement and compare the performance metrics of various machine learning algorithms
CO-3	Translate the real-world problems into a well posed learning problem that can be solved by a suitable machine learning algorithm
CO-4	To analyze the key computations underlying deep learning and use them to train and build DNN.

Mapping of course outcomes with program outcomes:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	1	2	-	1	-	1
CO-2	2	2	1	2	-	1
CO-3	2	2	2	2	2	1
CO-4	1	2	2	1	2	1

List of Experiments:

Implement the following concepts using python with supporting applications.
Cycle-1:

1. Familiarizing with Anaconda and Jupyter for importing modules and dependencies for ML Familiarization with NumPy, Panda and Matplotlib by Loading Dataset in Python
2. **Linear regression***:* Predict the profit of a company/House price from a dataset using the concept of linear regression. Implement the speech recognition model (NLP) from a speech/audio dataset using the concept of linear regression
3. **Logistic regression***:* a. Identify whether the patient has diabetes or not from diabetes dataset using Logistic regression

   b. Implement the speech to text model (NLP- Speech recognitions system) from a speech dataset using the concept of linear regression
4. **Polynomial regression** *:* a.Determine the quality of wine using wine dataset with the help of polynomial regression

b.Implement the speech recognition model (NLP) from a speech/audio dataset using the concept of polynomial regression

5. **K-means clustering***:* Apply the concept of K-means clustering for image segmentation problem (Brain tumor and Lung images)/Color quantization
6. Write a program to implement k-Nearest Neighbor algorithm to classify the iris data set to demonstrate the working of the decision tree based ID3 algorithm.
7. Write a program to implement the k-Nearest Neighbor algorithm for image classification and distance metric learning for large margin with image classification applications using k-nearest neighbor.
8. PCA/LDA: Reduce the dimensionality of a dataset for Face recognition system

Cycle-2:

9. Design an Artificial neural network for Digit classification using Back Propagation Algorithm for MNIST Data set. Train MLP using Gradient descent algorithm by applying Linear, Sigmoid, tanh, and ReLu activation functions
10. **Digit recognition using CNN**: Identify the digits 0-9 from MNIST data and CIFR 10 set using CNN
11. **Image Classification using CNN**: Classify cats and dogs using CNN from the given dataset
12. **Sentiment Analysis / Text classification**: Using RNN analysis the sentiment of the text/
    a. classify the text into domains.
13. LSTM (Long Short-Term Memory Networks)/ARIMA--- Implementation biomedical signals (like EEG, ECG, EMG) classifications and disease prediction.
14. Implementation of Speech recognition system for various languages model using LSTM & GRU
15. Implement a data augmentations and transfer learning concept in medical Image data Classification (like Pneumonia, Tumor etc.)
16. Yolo implementation for Object detection
17. Project related to application of machine learning in healthcare, traffic surveillance system, smart agriculture and advanced biometric system.

Learning Resources:

Text Books:

1. S. N.Sivanandam and S. N.Deepa, Introduction to neural networks using Matlab, 2006.
2. Simon 2. Haykin, Neural Networks and Learning Machines, PHI, 2008,3rdEdition.

2nd Semester

EC16002 **3-0-0 (3)**

GPU Architectures and Programming

Pre-requisites:

Course Outcomes:

CO-1	Understand working proficiency with CUDA, algorithmic GPU programming and parallel computing
CO-2	Comprehend with classic scientific computing algorithms and problems
CO-3	Optimize GPU code and debug GPU code
CO-4	Analyze architecture specific details like memory access coalescing, shared memory usage, GPU thread scheduling
CO-5	Apply deep learning algorithms on embedded GPUs

Mapping of course outcomes with program outcomes

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	1	1	3	1	1
CO-2	2	1	1	2	1	1
CO-3	2	2	1	3	1	1
CO-4	3	1	2	3	1	1
CO-5	2	1	2	3	1	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

GPU architectures: Introduction to the ideas of parallelism and the GPU programming model CPU vs GPU Parallelizing algorithms on paper, First CUDA program

CUDA programming: Hardware of Graphics Processing Units and parallel communication patterns, Brief on GPU architecture, Basics of CUDA C, Floating point precision and support on GPUs

Parallel primitives and algorithms on GPU: The CUDA programming language will be mastered while learning how to implement these algorithms., Matrix Operations, Stencil – Image Blurring, Filters, Gauss-Jacobi-Finite difference updates for PDEs, Histogram, binning 1, Reduce – Maximum and Minimum – Summation, Prefix-sum (Scan) Algorithm – Radix Sort, Generating Cummulative Distributions , Complex algorithms – N-body solutions

Optimizing GPU Applications: Coalesced Memory Transactions, Grid Blocks, Thread Blocks, domain decomposition, Asynchronous Kernels and Multistreaming Possible Items: Libraries on GPU, cuBLAS Thrust, cuFFT, cuRAND, Multi-node GPU processing, Multi-GPU per node processing, CUDA in other languages (Python/Fortran), Scaling.

Deep learning on GPUs: Deep learning on GPUs, Combining graphics and compute, Display the results of computations– Interactive systems, Collision detection with voxelized solid (Gargoyle), Ray tracing in CUDA

kernels, or ray tracing cores, Microsoft DXR (DX12 API), Vulkan, NVIDIA OptiX / RTX, NVIDIA Turing: “World‘s First Ray Tracing GPU”- Quadro RTX, Geforce RTX

Learning resources:

Reading:

1. GPUs for Graphics: OpenGL 4.0 Shading Language Cookbook, 2nd Edition.

2. Jason Sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Publisher: Addison-Wesley Professional,2013, 3rd Edition.

EC16004 **3-0-0 (3)**

Hardware Accelerators Design for Machine Learning Models

Pre-Requisites: **None**

Course Outcomes:

CO-1	Explore the various DNN models and development resources
CO-2	Understand the hardware implementation strategies for DNN
CO-3	Develop the memory optimization and computational optimization techniques to DNN
CO-4	perform the near-data processing and benchmark evaluation to DNN models
CO-5	Design and implement the accelerator logic to DNN models

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	2	2	1	2	1
CO-2	2	2	3	1	2	1
CO-3	3	2	3	2	2	1
CO-4	2	2	2	2	1	1
CO-5	1	1	3	1	-	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Overview of DNNs: Convolutional Neural Networks (CNNs), Popular DNN Models, DNN development resources**:** Frameworks, Models, Popular Data Sets for Classification, And Data Sets for Other Tasks.

Evolution of hardware platforms for Deep Learning: CPUs, GPUs, FPGAs, DSPs, accelerators; Hardware considerations in inference and training, Accelerate Kernel Computations on CPU and GPU Platforms, Energy-Efficient Dataflow for Accelerators, DNN data handling characteristics, Weight Stationary (WS), Output Stationary (OS), No Local Reuse (NLR), Row Stationary (RS). Accelerating the convolution operation: Algorithms, Data flow patterns, Memory reuse Case-study on writing a custom GPU

Kernel for accelerating convolution, optimizing networks: Weight quantization, network compression, sparse operations, zero forwarding, learning with hardware in the loop, learning and inference on low-memory devices.

Memory and compute: Optimizations to CNNs such as tiling, loop optimizations, batching, quantization, pruning, Cache Blocking, four convolution strategies (Direct, GEMM, FFT and Winograd), Model-size aware and system-aware pruning of CNNs, MLPerf Benchmark for evaluating DNN accelerators.

Near-data processing: DRAM, SRAM, Non-volatile Resistive Memories**,** Sensors, co-design of DNN models and hardware: Reduce Precision, Reduced Number of Operations and Model Size. Benchmarking metrics DNN evaluation and comparison**,** Metrics for DNN Models, Metrics for DNN Hardware.

Deep Learning on Systolic Array and Tensor Processing Unit (TPU) v1 to v4 , Distinct Characteristics of Training and Inference , Architectures of TPU v1, v2, v3 and v4; comparison between their architectures , Comparison of CPU, TPU and GPU, Deep Learning on FPGA and Microsoft's Brainwave Architecture , Deep Learning techniques on FPGA; efficacy of FPGAs for binarized neural networks (BNNs)

Learning Resources:

Text Books:

1. Shiho Kim, Ganesh Chandra Deka,Hardware Accelerator Systems for Artificial Intelligence and Machine Learning, Volume 122 - March 28, 2021, 1st Edition,

Learning Resources:

1. V. Sze, Y. -H. Chen, T. -J. Yang and J. S. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," in Proceedings of the IEEE, vol. 105, no. 12, pp. 2295-2329, Dec. 2017, doi: 10.1109/JPROC.2017.2761740.
2. https://docs.amd.com/r/2022.2-English/ug896-vivado-ip/Xilinx-Resources

EC16006 **0-1-2 (2)**

GPU Programming Lab

Pre-Requisites: **EC16002**

Course Outcomes:

CO-1	Explore basic CUDA Implementations
CO-2	Understand host, device and global functions
CO-3	Write basic CUDA programs like vector addition, Dot Product and Matrix operations
CO-4	Understand OpenCL concepts and Basic Program Structure of Numba Pro
CO-5	Understand and Implement computer vision applications for optimizing high performance computing and machine learning applications

Mapping of course outcomes with program outcomes:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	1	1	2	1	1
CO-2	1	1	-	2	1	1
CO-3	1	1	-	2	1	1
CO-4	1	1	-	2	-	1
CO-5	-	2	-	2	1	1

List of Experiments:

1. Basic Programming and CUDA implementation
2. Programs -Hello world, a Kernel Call and Passing Parameters.
3. Vector Sum and Dot Product
4. Matrix-Matrix Multiplication
5. Use of shared memory to reduce Global Memory Traffic
6. Program on warp divergence issue
7. Implementation of Min/Max/Sum reduction algorithm
8. General purpose GPU computing with PyCUDA and PyOpenCL
9. An overview of OpenCL, Important OpenCL concepts and Basic Program Structure, NumbaPro
10. Profiling Code and Improving GPU Performance
11. GPU Computing Applications- Case Studies in Machine Learning for computer vision applications

Learning Resources:

Text Books:

1. David B. Kirk, Wen-mei W. Hwu,Programming Massively Parallel Processors: A Hands-on Approach Elsevier, 2010,3rdEdition.

Reference Books:

1. Shane Cook: CUDA Programming: A Developer's Guide to Parallel Computing with GPUs Applications of GPU computing series Morgan Kaufmann,Newnes, 2012.
2. Dr. Brian Tuomanen:Hands-On GPU Programming with Python and CUDA: Explore high-performance parallel computing with CUDA, Packt Publishing Ltd, 2018.

EC16008 **0-1-2 (2)**

Hardware Accelerators for ML Lab

Pre-requisites: **EC16004**

Course Outcomes:

CO-1	Design effective computing units for complex operations in ML
CO-2	Integrate PL and PS and develop HW/SW co-design model
CO-3	Develop an efficient accelerator for ML Models
CO-4	Develop the complete DL model and fit on constrained edge device.
CO-5	Develop the light weight DNN model to run on edge TPU

Mapping of course outcomes with program outcomes:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	1	1	2	2	2	1
CO-2	1	-	3	1	1	1
CO-3	1	1	3	1	2	1
CO-4	1	1	3	1	2	1
CO-5	1	1	1	-	2	1

List of Experiments:

1. Design of processing element (MAC unit) Using VIO and ILA.
2. Perform matrix multiplication Using Vitis IDE/SDK tool
3. Design of array multiplier IP using HLS C code.
4. Integration of HLS IP with zynq processor and control of DUT with the processor (zed board)
5. Integration of HLS IP with zynq processor and control of DUT with the processor (pynq-z2 board)
6. Design accelerator for direct convolution operation using HLS C code
7. Design accelerator for GEMM based convolution operation
8. Design accelerator for winograd (algorithmic based) convolution operation
9. Develop accelerator for parallel pooling operation
10. Develop accelerator for dense layers of CNN
11. Design accelerator for different activation functions (Tanh, Sigmoid).
12. Design a fixed point type accelerator architecture for experiments (6 – 11).
13. Develop accelerator for ANN model for any standard dataset
14. Develop accelerator for CNN model for any standard dataset

15. Implement accelerator architectures by varying the number of processing elements.
16. Design accelerator with different reuse techniques and analze the performance
17. Develop accelerator for RNN/LSTM DL model
18. ML Programs execution with ARM processor (PYNQ-Z2)
19. Train and build a light model for edge TPU
20. Retrain a classification model for Edge TPU using post-training quantization (with TF2)
21. Retrain Efficient Det for the Edge TPU with TensorFlow Lite Model Maker
22. Run the trained model on edge TPU and analyze the FPS

Learning Resources:

1. Wayne Wolf " FPGA based System Design" Pearson India Education services ,11th Edition.
2. Shiho Kim, Ganesh Chandra Deka,Hardware Accelerator Systems for Artificial Intelligence and Machine Learning, Volume 122 - March 28, 2021, 1st Edition.

https://docs.amd.com/r/2022.2-English/ug896-vivado-ip/Xilinx-Resources

EC16096 **0-0-3 (2)**

Minor Project

Pre-Requisites: **None**

Course Outcomes:

CO-1	Apply engineering principles to real-world projects
CO-2	Plan and monitor project tasks individually or as a team
CO-3	Demonstrate practical experience in project execution
CO-4	Communicate project findings clearly through reports and presentations

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	3	3	3	3	3	-
CO-2	2	2	2	2	2	-
CO-3	2	2	2	2	2	-
CO-4	1	1	1	1	1	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Description:

Students are expected to choose real world or relevant problems and apply the engineering principles learned, to solve the problem through building prototypes or simulations or writing codes or establishing processes/synthesis/correlations etc. The department constituted panel can decide the suitability and worthiness of the project

Evaluation Criteria:

The student will be evaluated by the panel based on the below criteria. Weightage for each criterion will be determined by the panel and will be informed to the students.

Criteria	Description	Weightages
I	Identification of Problem Domain	
II	Study of Existing Systems and establishing clear objectives	
III	Planning of project and work distribution within the team	
IV	Proper Documentation and Technical Writing	
V	Presentation and Response to questions	

Evaluation Criteria-CO Mapping

CO / Criteria	CO1	CO2	CO3	CO4
I	X			
II	X			
III		X	X	
IV				X
V				X

3rd Semester

EC16089 **0-0-0 (2)**

Seminar and Technical Writing

Pre-Requisites: **None**

Course Outcomes:

CO-1	Consolidate ideas based on expert talks attended
CO-2	Prepare a well-organized report employing elements of critical thinking and technical writing
CO-3	Demonstrate the ability to describe, interpret and analyze the subject matter and develop competence in presenting

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	-	2	-	-	3	3	-	2	3	-	3	-	-	-	-
CO-2	2	-	2	-	-	3	3	-	2	3	-	3	-	-	-	-
CO-3	2	-	2	-	-	3	3	-	2	3	-	3	-	-	-	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Description:

In Seminar and Technical Writing, every student is expected to prepare a well-organized report based on one / all of the following:

- by attending at least 5 expert lectures/ invited talks/ Seminar/ Popular lectures etc. organized by the institute/any of the departments, ideally in a specific domain or with the same theme.
- prepare a business or marketing plan based on patent search

The student is expected to consolidate the ideas from these lectures/patent searches and may even include material from other sources to strengthen the content of the report. The student should prepare a well-organized report based on the above and present it to the panel constituted by the department, for evaluation.

Evaluation Criteria:

The student will be evaluated by the panel based on the below criteria.

Criteria	Description	Weightages
I	Clarity on the topic	
II	List of lectures attended	
III	Report	
IV	Presentation	
V	Response to questions	

Evaluation Criteria-CO Mapping

CO / Criteria	CO-1	CO-2	CO-3
I	X		
II	X		
III		X	
IV			X
V			X

EC16093 **0-0-0 (2)**

Comprehensive Viva-Voce

Pre-Requisites: **None**

Course Outcomes:

CO-1	Comprehend the knowledge gained in the course work
CO-2	Demonstrate the ability in problem solving
CO-3	Communicate effectively using engineering terminology

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	3	3	-	-	-	-	-	2	-	3	-	-	-	-
CO-2	3	3	3	3	-	-	-	-	-	2	-	3	-	-	-	-
CO-3	2	2	2	2	-	-	-	-	-	3	-	3	-	-	-	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Description:

In Comprehensive viva-voce each student will be evaluated for their overall comprehension of the course work and laboratory training that they have undergone. The students will be expected to answer questions orally, write down simple equations, draw plots, schematics, write simple code etc. as questioned by the panel. Assessment will be done by the panel based on the student response.

EC16097 **0-0-0 (8)**

Dissertation: Part-A

Pre-Requisites: **None**

Course Outcomes:

CO-1	Identify a topic in advanced areas of Embedded and Machine Learning Systems.
CO-2	Review literature to identify gaps and define objectives and scope of the work.
CO-3	Employ the ideas from literature and develop research methodology.
CO-4	Develop a model, experimental set-up and / or computational techniques necessary to meet the objectives.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	3	2	3	1		3
CO-2	3	2	3	1		3
CO-3	3	3	3	3	3	3
CO-4	3	3	3	3	3	3

1 - Slightly; 2 - Moderately; 3 - Substantially

Description:

Students are expected to choose real-world contemporary problem and apply the engineering principles learned, to solve the problem through building prototypes or simulations or writing codes or establishing processes/synthesis/correlations etc. The department constituted panel will decide the suitability and worthiness of the project.

Dissertation Evaluation:

i. The dissertation shall be submitted as per the schedule given in the academic calendar.
ii. The dissertation supervisor will periodically review the progress of the student and finally give his/her assessment of the work done by the student.
iii. The Dissertation Part – A will be evaluated for 100 marks, with the following weightages:

Sub-component	**Weightage**
a) Periodic evaluation by Guide	40 marks
b) Mid-term review	20 marks
c) End Semester viva-voce examination	40 marks

Evaluation Criteria

The student will be evaluated by the panel based on the below criteria. Weightage for each criterion will be determined by the panel and will be informed to the students.

Criteria	Description	Weightages
I	Selection of Topic	
II	Literature Survey	
III	Defining the Objectives and Solution Methodology	

IV	Performance of the Task	

Evaluation Criteria - CO Mapping:

CO / Criteria	CO-1	CO-2	CO-3	CO-4
I	X			
II		X		
III			X	
IV				X

Refer to the M.Tech. – Regulations for any further information regarding mid-term review and end semester evaluation, etc.

EC16091 **0-0-0 (2)**

Summer Internship/Research Experience

Pre-Requisites: None

Course Outcomes:

CO-1	Apply engineering principles to real-world problems, gaining practical experience
CO-2	Plan, manage and execute the work with ethical consideration
CO-3	Review the social and environmental impact of the work
CO-4	Communicate the learnings through report and presentation

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	3	3	3	-	-	-	-	-	-	3	-	-	-	-
CO-2	2	2	2	2	2	-	-	3	3	-	3	3		-	-	-
CO-3	-	-	-	-	-	3	3	-	-	-	-	3	--	-	-	-
CO-4	-	-	-	-	-	-	-	--	-	3	-	3	-	-	--	-

1 - Slightly; 2 - Moderately; 3 - Substantially

Description:

Every student has to undergo either a Summer Internship / Research project. The summer internship may be undergone in an Industry/Research organization or any premier academic Institution, including NIT Warangal for 6-8 weeks. The research project shall be registered under the guidance of any faculty member in the institute. The student is required to submit a report and present the work before an evaluation committee, nominated by the Head of the Department.

Evaluation Criteria:

The student will be evaluated by the panel based on the below criteria. Weightage for each criterion will be determined by the panel and will be informed to the students.

Criteria	Description	Weightages
I	Relevance of the area of work	
II	Performance of the Task	
III	Crucial learnings from the work	
IV	Report Preparation	
V	Presentation and Response to questions	

Evaluation Criteria-CO Mapping

CO / Criteria	CO-1	CO-2	CO-3	CO-4
I	X			
II		X		
III			X	
IV				X
V				X

4th Semester

EC16098 **0-0-0 (12)**

Dissertation: Part-B

Pre-Requisites: **EC16097**

Course Outcomes:

CO-1	Identify the materials and methods for carrying out experiments/develop a code.
CO-2	Execute the research methodology with a concern for society, environment and ethics
CO-3	Analyse, discuss and justify the results/trends and draw valid conclusions.
CO-4	Prepare the report as per recommended format and present the work orally adhering to stipulated time.
CO-5	Explore the possibility to publish/present a paper in peer reviewed journals/conference proceedings without plagiarism.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	-	3	3	2	3
CO-2	2	-	3	3	2	3
CO-3	3	-	3	3	2	3
CO-4	3	3	3	3	3	3
CO-5	3	3	3	-	-	3

1 - Slightly; 2 - Moderately; 3 - Substantially

Description:

Students are expected to choose real-world contemporary problem and apply the engineering principles learned, to solve the problem through building prototypes or simulations or writing codes or establishing processes/synthesis/correlations etc. The department constituted panel will decide the suitability and worthiness of the project.

Dissertation Evaluation:

i. The dissertation shall be submitted as per the schedule given in the academic calendar.
ii. The dissertation supervisor will periodically review the progress of the student and finally give his/her assessment of the work done by the student.
iii. The Dissertation Part – B will be evaluated for 100 marks, with the following weightages:

Sub-component	**Weightage**
a) Periodic evaluation by Guide	40 marks
b) Mid-term review	20 marks
c) End Semester viva-voce examination	40 marks

Evaluation Criteria:

The student will be evaluated by the panel based on the below criteria as a continuation of Dissertation Part A. Weightage for each criterion will be determined by the panel and will be informed to the students.

Task	Description	Weightages
IV	Performance of the Task	
V	Dissertation Preparation	
VI	Review (Presentation & Understanding)	
VII	Viva-Voce	
VIII	Publications /Possibility of publication	

Evaluation Criteria-CO Mapping:

CO / Criteria	CO-1	CO-2	CO-3	CO-4	CO-5
IV	X	X			
V				X	
VI			X		
VII				X	
VIII					X

Refer to the M.Tech. – Regulations for any further information regarding mid-term review and end semester evaluation, etc.

Professional Electives

Professional Elective - I

EC16021 **3-0-0 (3)**

Embedded RTOS

Pre-Requisites:

Course Outcomes:

CO-1	Understand the basic concepts of embedded C/C++
CO-2	Understand real-world software development challenges
CO-3	Develop the programs for real time application using C++
CO-4	Apply the concepts of OOPS programming
CO-5	Develop graphical user interface using python

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	1	1	2	2	1	1
CO-2	1	1	2	2	1	1
CO-3	2	2	2	2	2	1
CO-4	1	1	2	2	1	1
CO-5	1	1	-	-	2	1

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Introduction to Embedded Systems: Embedded systems Overview, Embedded Systems Vs General Computing Systems, History of Embedded Systems, Classification, Major Application Areas, Purpose of Embedded Systems, Characteristics of embedded computing applications, Design Challenges, Common Design Metrics

Real-Time Operating Systems: Architecture of the kernel, Tasks and Task Scheduler, Scheduling algorithms, Interrupt Service Routines, Semaphores, Mutex, Mailboxes, Message queues, Event Registers, Pipes, Signals, Timers, Memory management, Priority Inversion problem. Overview of COTS/FOSS operating systems Applications and examples of user level/ Kernel programming and debugging

Learning Resources:

Text Books:

1. David E-Simon: An Embedded software Primer, Pearson Education, 2007.
2. K.V.K.K. Prasad,Real-Time Systems: Concepts Design and Programming, Dreamtech Press, 2005.

Reference Books:

1. Kernighan, Brian W, Ritchie, Dennis M, C Programming language, Prentice Hall,1988 (2nd Edition)

2. Michael J. Pont, Embedded C, Addison-Wesley, Kenneth Lambert, Fundamentals of Python: Course Technology Publishing, Cengage Learning, 2012 ISBN-13: 978-1-111-82270-5

EC16023 **3-0-0 (3)**

Statistical Modelling and Optimization Techniques

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand the concepts of statistics and probability
CO-2	Design and Analyse statistical models
CO-3	Understand the optimization and linear programming techniques
CO-4	Solve constrained and unconstrained optimization problems
CO-5	Interpret the results of various statistical and optimization algorithms.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	1	-	-	-	1
CO-2	2	1	-	-	-	1
CO-3	1	2	-	-	-	1
CO-4	2	2	-	-	-	1
CO-5	1	2	-	-	-	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to Statistical Models and Analysis: An overview of basic probability, Sampling, distributions, linear statistical models, Least squares estimation, Gauss Markov Theorem, Properties of the estimates, Statistical experiments and significance testing: Z test, T Test, Anova Test, Chi Square Test, F Test.

Regression and Prediction: Regression, Different types models for Regression, Procedures, Model Selection, Prediction using regression; multiple linear regression: inference technique for the general linear model, generalized linear models: inference procedures, Classification, Naïve Bayes, logistic regression and lag linear models; Nonparametric Regression, introduction to non-linear modelling: Least squares and estimation of parameters

Optimization and Linear Programming: Types and size of problems, Iterative algorithms and convergence, Basic properties of Linear programs, Convex sets, simplex method, Duality, Interior point methods, Transportation and Network flow problems

Unconstrained Problems: Basic properties and solutions of Algorithms, Basic Descent methods, Conjugate Direction Methods, Quasi-Newton Methods

Constrained Minimization: Constrained Minimization conditions, Primal methods, penalty and Barrier methods, Dual cutting plane methods

Learning Resources:

Text Books:

1. Peter Bruce, Andrew Bruce, Practical statistics for data scientists, O'Reilly, 2020,2nd Edition.
2. David G. Luenberger, Yinyu Ye,Linear and Nonlinear Programming, Springer, 2008,3rd Edition.

References /Suggested Readings:

1. Dobson, A. J. and Barnett, A. G., An Introduction to Generalised Linear Models, , Chapman and Hall/CRC, 2008, 3rd Edition..
2. Krzanowski, W. J., An Introduction to Statistical Modelling, Wiley, 2010.
3. EDWIN K. P. CHONG STANISLAW H. ZAK, An Introduction to Optimization,WILEY INTERSCIENCE, 2013, 4th Edition.

EC16025 **3-0-0 (3)**

Machine Learning for Signal Processing

Pre-Requisites:

Course Outcomes:

CO-1	Use Machine learning methods for signal analysis, modeling and information extraction
CO-2	Introduce the fundamentals of machine learning (ML) techniques useful for various signal processing applications
CO-3	Learn various mathematical methods involved in ML
CO-4	Design models and optimize them efficiently
CO-5	Develop projects to solve real time applications

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	1	2	2	2	1
CO-2	2	1	2	2	2	1
CO-3	1	1	1	2	2	1
CO-4	1	1	1	1	1	1
CO-5	1	1	1	1	2	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to real world signals - text, speech, image, video.

Feature extraction and front-end signal processing - information rich representations, robustness to noise and artifacts, signal enhancement, bio inspired feature extraction.

Basics of pattern recognition, Generative modeling - Gaussian and mixture Gaussian models, hidden Markov models, factor analysis.

Discriminative modeling - support vector machines, neural networks and back propagation.

Wavelet Transform in Machine Learning : From Fourier Transform to Wavelet Transform, The DWT as a filter-bank, Using the Continuous Wavelet Transform and a Convolutional Neural Network to classify signals, Training the Convolutional Neural Network with the CWT, Deconstructing a signal using the DWT, Removing (high-frequency) noise using the DWT, Using the Discrete Wavelet Transform to classify signals, Comparison of the classification accuracies between DWT, Fourier Transform and Recurrent Neural Networks

ML for Speech Recognition: Hidden Markov Models, Finite State Transducers and Dynamic Programming

ML for Image Processing: Transfer Learning, Attention models, Attribute-based learning

ML for Communication: Deep learning for wireless applications

Learning Resources:

Text books:

1. C.M. Bishop,Pattern Recognition and Machine Learning, Springer, 2011, 2nd Edition.
2. R. C. Gonzalez, R. E. Woods, Digital Image Processing , Prentice Hall, 2008, 3rd Edition.
3. L. Rabiner and H. Juang,Fundamentals of speech recognition, Prentice Hall, 1993,1st Edition.

EC16027 **3-0-0 (3)**

IoT Architecture & Computing

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand the various concepts of the IoT and their technologies.
CO-2	Develop the IoT application using different hardware platforms
CO-3	Understanding and Implement the various IoT Protocols
CO-4	Understand the basic principles of cloud computing.
CO-5	Develop and deploy the IoT application into cloud environment

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	1	2	2	2	1
CO-2	2	1	2	2	2	1
CO-3	1	1	1	2	2	1
CO-4	1	1	1	1	1	1
CO-5	1	1	1	1	2	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to IoT – definition, Characteristics, Complete Architectural Stack, IoT enabling Technologies and Challenges. **Sensors and Hardware for IoT** – Hardware Platforms – Arduino, Raspberry Pi, Node MCU. A Case study with any one of the boards and data acquisition from sensors.

IoT Protocols – Infrastructure protocol (IPV4/V6/RPL), Identification (URIs), Transport (Wifi, Lifi, BLE), Discovery, Data Protocols, Device Management Protocols. – A Case Study with MQTT/CoAP usage-IoT privacy, security and vulnerability solutions.

Cloud Architecture Basics – Fundamentals of cloud, cloud architecture standards and interoperability-Cloud types; IaaS, PaaS, SaaS. Benefits and challenges of cloud computing, public, private clouds community cloud, Development environments for service development; Amazon, Azure, Google App cloud platform in industry

IoT and the Cloud - Role of Cloud Computing in IoT, AWS Components, Connecting a web application to AWS, IoT using MQTT, AWS IoT Examples. Security Concerns, Risk Issues, and Legal Aspects of Cloud Computing- Cloud Data Security

Case studies with architectural analysis: IoT applications, Smart City, Smart Water, Smart Agriculture, Smart Energy, Smart Healthcare, Smart Transportation, Smart waste management.

Learning Resources:

Text books:

1. Pethuru Raj and Anupama C. Raman by The Internet of Things: Enabling Technologies, Platforms, and Use Cases, CRC Press,2007,1st Edition.

2. Adrian McEwen, Designing the Internet of Things, Wiley, 2013,1st Edition.

3. Singh, Rajesh. Gupta, LoviRaj. Gehlot, Anita. Singh, Bhupendra. Swain, Mahendra, “Internet of Things with Raspberry Pi and Arduino” United Kingdom: Taylor & Francis Limited, 2021.

Professional Elective - II

EC16029 **3-0-0 (3)**

VLSI DSP Architectures

Pre-Requisites: **None**

Course Outcomes:

CO-1	Architect programmable DSP devices optimizing the performance.
CO-2	Design efficient architectures, algorithms and circuits improving size, power consumption, and speed and round-off noise.
CO-3	Translate effective algorithm design to integrated circuit implementations.
CO-4	Comprehend various sources of errors in implementation of DSP algorithms and device means to control them while implementing the DSP systems as per the specifications

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	1	-	-	1	1	1
CO-2	-	1	2	2	1	1
CO-3	1	1	1	2	1	
CO-4	-	-	1	-	1	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Overview of VLSI Technology and ASIC Design : Implementation Strategies for Digital ICs, CAD Tool for ASICs, Full- Custom/Custom ASICs, Custom FPGA Platforms, Digital ASIC Design, Issues in Digital Integrated Circuit Design, Quality Metrics of Digital Designs, Digital IC Components.

VLSI Digital Signal Processing Architectures: Introduction to Digital Signal Processing Systems of Typical DSP Algorithms (FFT, DFT, LMS, RLS…), Representation of Signal Processing Algorithms, Signal-flow, Data- flow and Dependence Graphs, Iteration Bound.

VLSI Architecture Techniques: Pipelining,: Parallel Processing , Pipelining and Parallel Processing for Low Power Design , Retiming Techniques , Unfolding , Folding , Register Minimization Techniques, Systolic Architecture Design , FIR Systolic Arrays.

Synchronous and Asynchronous Pipeline : Synchronous Pipeline and Clocking Styles, Wave Pipelining Asynchronous Pipeline, Implementation of Computational Units Bit- Level Arithmetic Architectures, Arithmetic Circuits Number Systems and their Effect on Implementation Redundant, Floating- point Representations/Operations , Shifters/Adders/ Comparators, Parallel/Bit- serial Multipliers, Redundant Arithmetic, Finite Word Length Effect, Floating- point to Fixed- point simulation techniques, Parallel and Pipelined Digital Filter Design

Architectures for Programmable DSP devices: Introduction, Basic architectural features, Bus architecture and memory. Data addressing, address generation unit, programmability program execution, speed issues, features for external interfacing, Architectural Trends in recent DSP processors.

Learning Resources:

Text Books:

1. K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation, Wiley, 1999,1st Edition.
2. Magdy A. Bayoumi, VLSI Design Methodologies for Digital Signal Processing Architectures, Publisher,Springer.

EC16033 **3-0-0 (3)**

Machine Learning in Robotics

Pre-Requisites: **None**

Course Outcomes:

CO-1	understand and work with robot hardware and software.
CO-2	Implement, plan machine learning analysis on suitable forms of computer and robotics data.
CO-3	Find out more about training controllers via reinforcement learning and imitation learning
CO-4	Integrate machine learning applications into a practical understanding of robotic models

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	1	1	1	2	1
CO-2	2	2	1	1	2	1
CO-3	2	2	1	3	3	1
CO-4	2	2	1	2	2	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to Robotics: Types and Classification of robots; Science and Technology of Robots

Rigid Body Transformation: Overview of Rigid Body Kinematics; Homogeneous Transformation; Link Transformation Matrices, Forward and Inverse Kinematics & Dynamics of Robots, Planning and Control of Robots ,Robotic vision sensors and their interfacing

Fundamentals of Computer Vision: Image accusation and representation, image transformation, filtering, restoration, morphing, Camera Models, Calibration, Single view geometry, Multiple view geometry, Epipolar geometry, RANSAC •

Position and Orientation: Feature based alignment; Pose estimation; Time varying pose and trajectories, Structure from motion, dense Motion Estimation, Visual Odometry (Semi-direct VO, direct sparse odometry), Bundle Assignment •

Localization and Mapping: Initialization, Tracking, Mapping, geometric SLAM formulations (indirect vs. direct error formulation, geometry parameterization, sparse vs. dense model, optimization approach), Relocalization and map Optimization, Visual SLAM, Examples: Indirect (Feature based) methods (MonoSLAM, PTAM, ORB-SLAM), Direct methods (DTAM, LSD-SLAM), Sensor combinations (IMU, mono vs. Stereo, RGB-Depth), Analysis and parameter studies.

Recognition and Interpretations: Concepts of machine learning and deep learning, sequence modeling, Learning for robotic vision:

Learning Resources:

Text Books:

1. Fu. K.S., Gonzalez R.C. and Lee C.S.G., Robotics: Control, Sensing, Vision and Intelligence, Tata McGraw Hill, 2008,ISE Edition.
2. Ghosal A. Robotics: Fundamental Concepts and Analysis, Oxford University Press, 2006,1st Edition.

Reference Books:

1. H. R. Everett, Sensors for Mobile Robots: Theory and Application, A K Peters/CRC Press, 1995, 1st Edition.

2. Dahiya, Ravinder S., Valle, Maurizio, Robotic Tactile Sensing, Springer, 2013

EC36025 **3-0-0 (3)**

VLSI System Design

Pre-Requisites: **None**

Course Outcomes:

CO-1	Model the behaviour of a MOS Transistor
CO-2	Understanding CMOS Inverter
CO-3	Design combinational and sequential circuits using CMOS gates
CO-4	Identify the sources of power dissipation in a CMOS circuit.
CO-5	Analyze SRAM cell and memory arrays

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PSO-1	PSO-2
CO-1	-	-	2	2	-	-	1	-
CO-2	-	-	2	-	-	-	1	-
CO-3	-	-	2	2	-	-	3	1
CO-4	-	-	2	-	-	-	1	-
CO-5	-	-	-	-	-	2	1	-

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

MOS Transistors, CMOS Logic, CMOS Fabrication and Layout, Design Partitioning, Fabrication, Packaging, and Testing, MOS transistor Theory, Long Channel I-V Characteristics, C-V Characteristics, Non-Ideal I-V Effects, DC Transfer Characteristics. The CMOS Inverter:

The Static CMOS Inverter -An Intuitive Perspective, Evaluating the Robustness of the CMOS Inverter: The Static Behavior, Performance of CMOS Inverter: The Dynamic Behavior.

CMOS Processing Technology, CMOS Technologies, Layout Design Rules, CMOS Process Enhancements, Technology-Related CAD Issues, Manufacturing Issues, Circuit Simulation- A SPICE Tutorial, Device Models, Device Characterization, Circuit Characterization, Interconnect Simulation. Combinational Circuit Design, Circuit Families, Silicon-On-Insulator Circuit Design, Sub Threshold Circuit Design, Sequential Circuit Design, Circuit Design of Latches and Flip-Flops, Static Sequencing Element Methodology, Sequencing Dynamic Circuits, Synchronizers, Wave Pipelining.

Power, Sources of Power Dissipation, Dynamic Power, Static Power, Energy-Delay Optimization, Low Power Architectures, Robustness, Variability, Reliability, Scaling, Statistical Analysis of Variability, Variation-Tolerant Design. Delay, Transient Response, RC Delay Model, Linear Delay Model, Logical Effort of Paths, Timing Analysis Delay Models, Datapath Subsystems, Addition/Subtraction, One/Zero Detectors, Comparators, Counters, Boolean Logical Operations, Coding, Shifters, Multiplication.

Array Subsystems, SRAM, DRAM, Read-Only Memory, Serial Access Memories, Content-Addressable Memory, Programmable Logic Arrays, Robust Memory Design, Special-Purpose Subsystems.

CMOS Testing-The need for testing, Manufacturing test principles, Design strategies for test, Chip level test techniques, System level test techniques, Layout design for improved testability.

Learning Resources:

Text Books:

1. Neil H.E. Weste, David Harris, Ayan Banerjee, CMOS VLSI Design – A Circuits and Systems Perspective, Pearson Education, 2006, 3rd Edition.
2. Neil H. E. Weste Kamran Eshraghian, Principles of CMOS VLSI DESIGN:A Systems Perspective, Pearson Education, 2006, 2nd Edition.

Reference Books:

1. Jan M RABAEY, Digital Integrated Circuits, Pearson Education, 2003, 2nd Edition.
2. Douglas A. Pucknell, Kamran Eshraghian, Basic VLSI Design, PHI,1994, 3rd Edition.

Professional Elective - III

EC16022 **3-0-0 (3)**

Hardware and Software Co-design of Embedded Systems

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand Hardware/Software Co-design
CO-2	Model the data flow and implement the same through software and hardware.
CO-3	Design the hardware controllers through Control Flow Graph Structures.
CO-4	Understand the design principles in SoC Architecture
CO-5	Design CORDIC and Crypto coprocessor architectures

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6
CO-1	1	1	1	-	1	-
CO-2	1	2	2	-	1	1
CO-3	1	2	2	-	1	1
CO-4	1	1	2	-	-	1
CO-5	-	-	-	-	-	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

The Nature of Hardware and Software: Introducing Hardware/Software Co-design, The Quest for Energy Efficiency, The Driving Factors in Hardware/Software Co-design, The Dualism of Hardware Design and Software Design.

Data Flow Modeling and Transformation: Introducing Data Flow Graphs, Analyzing Synchronous Data Flow Graphs, Control Flow Modeling and the Limitations of Data Flow, Transformations.

Data Flow Implementation in Software and Hardware: Software Implementation of Data Flow, Hardware Implementation of Data Flow, Hardware/Software Implementation of Data Flow.Analysis of Control Flow and Data Flow: Data and Control Edges of a C Program, Implementing Data and Control Edges, Construction of the Control Flow Graph, Construction of the Data Flow Graph.

Architectures for Embedded Systems: a. Single processor – coprocessor architecture, mixed-signal architectures, multiprocessor architectures, reconfigurable architectures, Systems on Chip.

Performance Modeling: a. System-level performance modeling vs. low-level performance modeling. b. Modeling of system latency, energy consumption etc for hardware and software. c. Estimation of memory requirements.

System-Level Synthesis and Trade-off Analysis: a. Design of customized digital blocks. b. Hardware/software partitioning. Task binding. c. IP core integration and communication synthesis: Hardware

and software interface synthesis. d. Hardware IP core synthesis: High-level synthesis: behavioral specification of hardware, module set allocation, resource binding, operation scheduling, controller design

Learning Resources:

Text Books:

1. Patrick Schaumont, A Practical Introduction to Hardware/Software Co-design, Springer, 2010.
2. A. Doboli, E. Currie, “Introduction to Mixed-Signal Embedded Design”, Springer, 2010.
3. G. De Micheli, R. Ernst, W. Wolf, Readings in Hardware/Software Co-Design, Morgan Kaufman, 2002
4. Ralf Niemann, Hardware/Software Co-Design for Data flow Dominated Embedded Systems, Springer,1998.

EC16024 **3-0-0 (3)**

Machine Vision and Pattern Recognition

Pre-Requisites: **None**

Course Outcomes:

CO-1	To create algorithms and systems for processing of visual signals (images, videos etc.) for low-level, mid-level, and high-level perceptual tasks
CO-2	Understand the imaging process for a pin-hole camera, lenses, image-statistics such as gradients and edges
CO-3	Perform texture modeling, face recognition, activity recognition, and scene recognition
CO-4	Categorize the various pattern recognition techniques into supervised and unsupervised
CO-5	Illustrate the artificial neural network-based pattern recognition

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	1	1	1	-	1	-
CO-2	1	2	2	-	1	1
CO-3	1	2	2	-	1	1
CO-4	1	1	2	-	-	1
CO-5	-	-	-	-	-	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Image and Video Processing – filtering, frequency analysis, multiresolution/pyramids **Camera Geometry** – pinhole cameras, camera projection matrix, calibration algorithms Stereo – triangulation, epipolar geometry, fundamental matrix, depth from disparity **Features** – edges, gradients, Harris corners, SIFT

Learning: Nearest neighbors, SVMs, Deep learning, neural networks

Motion – Tracking, detection, optical flow

Structure-from-Motion and 3D reconstruction

Light/Shading: Shape-from-shading, photometric stereo

Special Topics: A taste of reinforcement learning, SLAM

Learning Resources:

Textbook:

1. Richard Szeliski, Computer Vision: Algorithms and Application, available for free at http://szelisk.org/Book/.

Reference Textbooks:

1. David Forsyth and Jean Ponce, Computer Vision: A Modern Approach, 2012, 2nd Edition.
2. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision,2004,2nd Edition.
3. Richard O. Duda, Peter E. Hart and David G. Stork, Pattern Classification, 2001, 2nd Edition.

EC16026 **3-0-0 (3)**

Wireless Technologies

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand the fundamentals of wireless technologies for embedded systems and their application in real-time scenarios.
CO-2	Understand wireless technologies and wireless transmission technology.
CO-3	Understand the fundamentals of MAC, routing, transport control protocols, and their design considerations.
CO-4	Understand various wireless networking standards and their connectivity for wireless embedded systems.

Mapping of course outcomes with program outcomes:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	-	-	-	-	-	1
CO-2	-	-	-	-	-	1
CO-3	2	1	2	2	2	1
CO-4	1	1	1	-	2	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to wireless communication: Electromagnetic spectrum, Radio waves, Associated bands, Network and Service Architecture, Cellular and Non-cellular Wireless Networks, and their application in embedded systems.

Wireless Communications understanding, Criteria, Range, Frequency Bands, Power Consumption, Topology, Constrained Devices, Constrained-Node Networks, Data Rate and Throughput, Latency and Determinism, Overhead and Payload. Wireless systems and standards.

Wireless Networking: Multiple access schemes – TDMA, FDMA, CDMA. Random Multiple Access Schemes – ALOHA, slotted ALOHA, CSMA/CA, OFDMA. Performance analysis issues. MAC layer scheduling and connection admission in wireless networks. Resource allocation and mobility management.

Wireless Networking:

i) Wi-Fi standard (IEEE 802.11), IEEE 802.11n (Wi-Fi 4), IEEE 802.11ac (Wi-Fi 5), IEEE 802.11ax (Wi-Fi 6), Channels, Service sets, Media Access Control, Wireless Security, Wired Equivalent Privacy (WEP), WPA2 & 3.

ii) Bluetooth standard (IEEE 802.15.1), Bluetooth 5, Channels, Service sets, Media Access Control, Wireless Security.

iii) RFID & NFC, Physical Layer, MAC Layer, Topology, Security.

iv) Broadband over power lines (BPL) (IEEE 1901.2a), Physical Layer, MAC Layer, Topology, Security.

v) Licensed LTE-M, EC-GSM & NB-IoT standards, data rate, and connectivity.

vi) Unlicensed MIOTY, LoRa & Sigfox standards, data rate, and connectivity.

vii) Cellular 5G, Visible Light, Infra Red, and Satellite Connectivity.

vii). Network Layer: IPv6, 6LoWPAN, RPL, IPSec, Transport Layer: TLS1.3, DTLS, TCP, TCP/UDP, and Application Layer: HTTP, XXMPP, DPWS, SOAP, CoAP, MQTT.

Learning Resources:

References:

1. Dr. Raul Aquino santos, MSc. Arthur Edwards Block,Embedded systems and wireless Technology : Theory and Practical Applications (University of Colima, Mexico) publication ,2012,1 st Edition Boca Raton

2. Frank Adelstein, sandeep K.S. Gupta, Gold Richard III, and Loren Schwiebert,Fundamentals of Mobile and pervasive computing Publisher: McGraw-Hill Education, 2005, ISBN-10:0071412379, ISBN- 13: 978-0071412377 to be verified

EC26026 **3-0-0 (3)**

Advanced Computer Architecture

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand the micro-architectural design of processors.
CO-2	Evaluate performance of different architectures with respect to various parameters
CO-3	Analyze performance of different ILP techniques
CO-4	Identify cache and memory related issues in multi-processors
CO-5	Understand the concept of NOC

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	-	2	2	1	1
CO-2	2	1	2	2	1	1
CO-3	1	1	2	3	1	1
CO-4	1	-	2	2	1	1
CO-5	-	-	1	-	-	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to high performance computing, RISC philosophy and overview of pipelined architecture. Performance evaluation of pipelined architecture. Limitations of scalar pipelines, Instruction level parallelism.

Superscalar architecture: Instruction flow optimization, Handling branches, Advanced optimization in instruction flow, register flow techniques: Register renaming and out of order execution.

Advanced data flow techniques: Instruction reuse and value prediction, Memory data flow, advanced memory data flow architectures, performance evaluation of superscalar architectures.

Multi-threading, Simultaneous multithreaded (SMT) architectures, SMT architecture: Choices, SMT performance on various designs, SMT architecture: OS impact and adaptive architectures

Multi scalar architecture, Multi-core Architectures, Multi-core Interconnect–NOC, Network on-Chip, Cache Coherence, Cache Consistency model, Dynamic Core architectures, GPGPU Architecture, CPU-GPU Integration.

Learning Resources:

Text Books:

1. J.L. Hennessy, and D.A. Patterson, Computer Architecture: A quantitative approach,Morgan Kaufmann Publication, 2012, Fifth Edition.
2. J.P. Shen and M.H. Lipasti, Modern Processor Design, MCGraw Hill, Crowfordsville, 2004,1st Edition.

Other Suggested Readings:

1. NPTEL Courses (https://nptel.ac.in/courses/106103182/)

Professional Elective – IV

EC16028 **3-0-0 (3)**

Synthesis and Optimization of Digital Circuits

Pre-Requisites: None

Course Outcomes:

CO-1	Understand the need for optimization and dimensions of optimization for digital circuits. including Hardware Modeling and Compilation Techniques.
CO-2	Understand basic optimization techniques used in circuits design
CO-3	Understand advanced tools and techniques in digital systems design
CO-4	Explain details of Logic-Level synthesis and optimization techniques for combinational and sequential circuits.
CO-5	Explain the concept of scheduling and resource binding for optimization

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	-	1	-	-	-	1
CO-2	-	1	1	-	1	1
CO-3	-	-	1	1	1	1
CO-4	-	-	-	-	-	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to Synthesis and optimization: Design of Microelectronics circuits, Computer aided Synthesis and Optimization.

Hardware Modeling: HDLs for Synthesis, Abstract models, Compilation and Behavioral Optimization.

Graph theory for CAD for VLSI: Graphs, Combinatorial Optimization, Graph Optimization problems and Algorithms, Boolean Algebra and Applications.

Architectural Synthesis and Optimization: Fundamental Architectural Synthesis problems, Area and Performance Estimation, Strategies for Architectural Optimization, Datapath Synthesis, Control Path Synthesis

Two level Combinational Logic Optimization: Introduction, Logic Optimizations, Operations on Two level Logic Covers, Algorithms for Logic Minimization, Symbolic Minimization and Encoding Problems. Multiple Level Combinational Logic Optimization: Introduction, Models and Transformations for Combinational Networks, The Algebraic Model, The Boolean Model.

Sequential Logic Optimization: Introduction, Sequential Logic Optimization using State based Models, Sequential Logic Optimization using Network Models, L1, L2, L3 45 Implicit FSM Traversal Methods, Testability concerns for Synchronous Circuits

Scheduling Algorithms: Introduction, A Model for Scheduling problems, Scheduling with Resource Constraints, Scheduling without Resource Constraints, Scheduling Algorithms for Extended Sequencing Models, Scheduling Pipelined Circuits. Resource Sharing and Binding: Sharing and Binding for Resource dominated circuits, Sharing and Binding for General Circuits, Concurrent Binding and Scheduling, Resource sharing and Binding for Non – Scheduled Sequencing Graphs

Learning Resources:

Text Books:

1. Giovanni De Micheli, —Synthesis and Optimization of Digital Circuitsll, Tata McGraw-Hill, 2003

Reference Books:

1. Edwars M.D., Automatic Logic synthesis Techniques for Digital Systems, Macmillan New Electronic Series, 1992.

EC16030 **3-0-0 (3)**

Tiny ML & Edge AI

Pre-Requisites: None

Course Outcomes:

CO-1	Understand the basics of TinyML and the current landscape of its applications and challenges.
CO-2	Utilize TensorFlow for deep learning and TensorFlow Lite (TFLite) for TinyML.
CO-3	Apply TinyML for gesture tracking with the Magic Wand project, including data collection, labeling, and training a CNN model.
CO-4	Learn various sensors such as accelerometers, gyroscopes, barometers, and magnetometers, and how to interface with sensor data.
CO-5	Deploy TinyML models on power and performance-constrained devices to solve realword problems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	1	2	1	1	2	1
CO-2	1	2	1	1	2	1
CO-3	1	2	1	2	2	1
CO-4	-	-	-	-	3	1
CO-5	-	-	-	-	2	1

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Introduction: Introduction to TinyML, TinyML Landscape, Applications, and Challenges, TinyML Lifecycle and Workflow, Model Compression Techniques. Fundamentals of ML and TinyML, Pruning ML models, Quantization Aware Training (QAT) and Post Training Quantization (PTQ), Knowledge Distillation, Tiny Deep Learning, TensorFlow Lite (TFLite) for TinyML.

TinyML for Keyword Spotting, Background on Keyword Spotting and Streaming Audio, Challenges and Constraints in Keyword Spotting, Keyword Spotting Architecture and Data Collection, Model Training, Evaluation Metrics, and Deployment

TinyML for Visual Wake Words, Introduction to Visual Wake Words and Its Challenges, Visual Wake Words Dataset, Mobile Nets, Transfer Learning for Visual Wake Words, Model Training, Evaluation Metrics, and Deployment.

TinyML for Anomaly Detection, Background on Anomaly Detection and Signal Processing, Real and Synthetic Datasets, Unsupervised Learning, Threshold Choice, Model Training, Evaluation Metrics, and Deployment

Magic Wand, Gesture Tracking through Bluetooth, CNN for Magic Wand Sketch, Data Collection and Labeling, Model Training, Evaluation Metrics, and Deployment

TinyML for Predictive Maintenance, Background on Predictive Maintenance Solutions and Industry Applications, Sensors, Sensor Data, and Interface, Accelerometer, Gyroscope, Barometer, and Magnetometer, TinyML Framework for Predictive Maintenance, Model Training, Evaluation Metrics, and Deployment

Smart Lock Audio Recognition using TinyML, Audio classification for deploying sensitive smart lock model, Data processing on audio data ○ Generate, train, and test a TensorFlow model using the SensiML Python SDK, Compile and flash the model to the edge device and display the inferred classes in the SensiML Open Gateway user interface.

Learning Resources:

Textbook

1 Pete Warden and Daniel Situnayake,TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers, O'REILLY Publisher ,2020,1st Edition.

References

1. Gian Marco odice,TinyML Cookbook

Additional Resources:

https://uncg.on.worldcat.org/oclc/1135326041

TinyML Foundation: https://www.tinyml.org

Canvas Platform: https://cainvas.ai-tech.systems/gallery/

EC16032 **3-0-0 (3)**

Applied Machine Learning: Ensemble Learning

Pre-Requisites: None

Course Outcomes:

CO-1	Explainingthemainstepsinvolvedindatapreparationandwhichtechniques can be used to improve a given data set
CO-2	Introduce a general machine learning pipeline, including techniques to avoid overfitting
CO-3	Provide practical experience on applying machine learning techniques to scientific and/or commercial applications
CO-4	Provide practical experience on applying machine learning techniques to scientific and/or commercial application
CO-5	Introduce a general machine learning pipeline, include techniques to a void overfitting

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	1	2	1	1	2	1
CO-2	1	2	1	1	2	1
CO-3	1	2	1	2	2	1
CO-4	-	-	-	-	3	1
CO-5	-	-	-	-	2	1

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Introduction to Ensemble Learning: Ensemble Learning Model Combination Schemes, Voting, Error-Correcting Output Codes, Bagging: Random Forest Trees, Boosting: Adaboost, Stacking.

Ensemble Learning Techniques: Gradient Boosting Machines (GBM), Extreme Gradient Boosting (XGBoost), CatBoost Stacking Random Subspace Method (Random Subspace Ensembles), Random Forest Variants

Voting Based Ensemble learning : CNN-LSTM

Stacking based Ensemble method: XGBoost, CNN,RNN, CNN LSTM

Algorithm based on Bagging and Boosting: Bagging Algorithm

Bootstrap Sampling, Prediction Aggregation

The fundamentals of real-time machine learning: Introduction, Batch machine learning vs. real-time machine learning, From analyzing historical data to real-time ML, online prediction with real-time features and continual (online) learning, Benefits of real-time machine learning

Learning Resources:

Textbook

1. Pete Warden and Daniel Situnayake,TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers , O'REILLY Publisher,2020,1st Edition.
2. Gian Marco Lodice,TinyML Arm computer library Cookbook,2020.

Additional Resources:

https://uncg.on.worldcat.org/oclc/1135326041

TinyML Foundation: https://www.tinyml.org

Canvas Platform: https://cainvas.ai-tech.systems/gallery/

EC16034 **3-0-0 (3)**

Verification Methodologies: UVM

Pre-Requisites: None

Course Outcomes:

CO-1	Understand the features and capabilities of the UVM class library for System Verilog
CO-2	use the building blocks of the UVM class library
CO-3	Understand the configuration databases, factory override, Transaction Level Modeling (TLM)
CO-4	Integrate scoreboards, multichannel sequencers and Register Models
CO-5	Build a framework for UVM Testbench.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	1	-	-	-	-	
CO-2	-	2	1	-	-	--
CO-3	1	-	-	--	-	-
CO-4	-	-	-	3	-	-
CO-5	-	-	-	-	-	3

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Verification Methodology Overview : Introduction to Verification Methodology, Verification Process, Reusable TB, Verification Environment Architecture, Constraint Random Coverage, Driven Verification, Verification Methodologies & Summary

Introduction to UVM, UVM Concepts, UVM SoC TB, VVM AHB UVC, UVM SOC TB Examples, UVM TB Architecture and Base Class Hierarchy**,** UVM Testbench Architecture, UVM Base Class Hierarchy, **UVM Factory,** Importance of using factory , Registration Process, Create Method and Factory Overriding, **UVM-Stimulus Modeling & Testbench Overview**

Predefined Methods and Field Registration Process, Overriding the predefined do- methods, UVM-TB Overview, UVM Phases & Reporting Mechanism, UVM Phases-Necessity of Phases &pre-run Phases, UVM Phases - Run Phase, post-run Phases and Objection Mechanism, UVM Reporting Mechanism, UVM TLM Ports - Blocking put and Configuration

UVM TLM Ports – Blocking put and get ports, UVM TLM Ports - TLM FIFO and Analysis Parts, UVM Configuration - Introduction to Configuration Facility, UVM Configuration-Configuration class and Configuration of Virtual interface, UVM –Creating UVM Testbench Components

Creating UVM TB Components –Sequencers & Drivers, Creating UVM TB Components- Monitor, Agents, Env and Testcases , **UVM Sequences, Module IX,** UVM Sequences - Introduction and Sequence item flow,

UVM Sequences - Starting the sequences and Default Sequence, UVM Virtual Sequences & Virtual Sequencers – Introduction, UVM Virtual Sequences & Virtual Sequencers – implementation

Text Books:

1. Sharon Rosenberg , A Practical Guide to Adopting the Universal Verification Methodology (UVM) Second Edition, 2013 Springer
2. Ray Salemi , The UVM Primer: A Step-by-Step Introduction to the Universal Verification Methodology, 2013.

Professional Elective - IV

EC16036 **3-0-0 (3)**

Hardware Security for Embedded Systems

Pre-Requisites: **None**

Course Outcomes:

CO-1	Comprehend the basics of embedded firmware, hardware and software vulnerabilities their causes
CO-2	Identify the vulnerabilities related to embedded systems using state of the art tools technologies
CO-3	Realize the principle aspects of a comprehensive security strategy in Embedded systems
CO-4	Understand and apply countermeasures against the introduced attacks

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	1		1	1	1	1
CO-2	2	1	1	1	1	1
CO-3	2	-	1	-	1	1
CO-4	-	-	-	-	-	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to Embedded Systems: Embedded hardware units, Embedded system software, Device drivers and interrupt services, Inter-process communication and synchronization of processes

Embedded System Security and Trust: Physical attacks, Side channel analysis, Trusted integrated circuit, Trusted platform module (TPM), Hardware Trojans, Cryptographic hashing, Stack-based attacks against embedded systems (Code injection and return-oriented programming), Physically unclonable functions, Fault injection attacks, Reverse engineering, Supply chain security and trust

Embedded Hardware Security and Hacking: Securing external memory, JTAG/Debug port considerations, Physical attack vectors, Tamper detection and logging, Soldering techniques, Board analysis methodology, Component Identification, Device instrumentation, Bus monitoring and decoding, Access via JTAG

Embedded Software Security and Exploitation: Fundamentals of embedded software security, Common firmware vulnerabilities, Software vulnerabilities in ARM/MIPS/etc, Embedded code vulnerabilities, Assembly code analysis, Exploitation techniques on ARM/MIPS/x86,Defenses against ARM exploits, Security practices for embedded software, Defensive software architectures, Defensive hardware interfaces

Text Books:

1. Tehranipoor, Mohammad, Wang, Cliff (Eds.), Introduction to Hardware Security and Trust, Springer,2011.

2. David Kleidermacher and Mike Kleidermacher, Embedded Systems Security: Practical Methods for Safe and Secure Software and Systems Development, Elsevier Science, Newnes Publication,2012.

Reference Books:

1. Louis Goubin and Mitsuru Matsui, Cryptographic Hardware and Embedded Systems, Springer-Verlag Berlin and Heidelberg GmbH & Co. KG

Other Suggested Readings:

1. https://onlinecourses.nptel.ac.in/noc22_cs48
2. https://onlinecourses.nptel.ac.in/noc22_cs93

EC16038 **3-0-0 (3)**

Selected Topics in ML

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand scope of Generative AI and their applications
CO-2	Analyze popular LLM architectures
CO-3	Able to pre-train and fine-tune processes in GPT
CO-4	Image generation using popular AI tools

Mapping of course outcomes with program outcomes:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	1	1	1	1	-	1
CO-2	-	1	1	1	-	1
CO-3	-	1	1	1	1	1
CO-4	-	1	-	-	-	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to Generative AI: Definition and scope of Generative AI, Overview of generative models and their applications, Importance of Generative AI in various domains, Brief discussion on ethical considerations and challenges.

Language Models and LLM Architectures: Introduction to language models and their role in AI, Traditional approaches to language modeling, Deep learning-based language models and their advantages, Overview of popular LLM architectures: RNNs, LSTMs, and Transformers

Understanding GPT (Generative Pre-trained Transformer): Introduction to GPT and its significance, Pre-training and fine-tuning processes in GPT, Architecture and working of GPT models, Overview of GPT variants and their use cases, Auto encoders and Generative Models: Auto-encoders and unsupervised learning , Variational Auto encoder , Generative Adversarial Networks (GANs)

Prompt Engineering: Enhancing Model Outputs: Understanding the concept and significance of prompt engineering, Strategies for designing effective prompts Techniques for controlling model behavior and output quality, Best practices for prompt engineering **in generative AI**

Use Cases of Generative AI: Overview of various domains and industries benefiting from Generative AI, Use cases in natural language processing, content generation, and creative applications, Case studies highlighting successful implementations, Potential future applications and emerging trends.

Some research papers and discussion

Learning Resources:

Text Books:

1. Alger Fraley, “The Artificial Intelligence and Generative AI Bible: The Most Updated and Complete Guide”2023.
2. Jacob Emerson by “Ripples of Generative AI: How Generative AI Impacts, Informs and Transforms Our Lives, Ingram spark publication,2023.

References :

1. David Foster, “Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play”.
2. I Almeida “Artificial Intelligence Fundamentals for Business Leaders: Up to Date With Generative AI”
3. Harish Bhat, “Demystifying Prompt Engineering: AI Prompts at Your Fingertips (A Step-By-Step Guide)”.

EC16040 **3-0-0 (3)**

Artificial Intelligence in Biomedical Applications

Pre-Requisites: None

Course Outcomes:

CO-1	Understand ML algorithms for biomedical disease classification
CO-2	Different disease classification using DL models
CO-3	Understand the nature of ECG, EMG, EEG signals
CO-4	EEG, ECG, EMG classification using AI techniques

Mapping of course outcomes with program outcomes:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	1	1	1	1	1	1
CO-2	1	2	-	-	2	1
CO-3	-	1	-	-	2	1
CO-4	-	2	-	2	2	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to AI & Machine-learning (ML): Exploratory data analysis, Handling big-data sets, Hypothesis testing, Unsupervised learning and clustering

Biomedical signal origin, dynamics and modelling: The nature of biomedical signals:Electrocardiogram (ECG), Electroencephalogram (EEG), Electromyogram (EMG) etc., Biomedical signal origin & dynamics of ECG, ECG;ECG lead system, ECG signal characteristics; Non-Stationarities in ECG signal, Cardiac Arrhythmias, EEG data acquisition, EEG lead system, EEG signal characteristics

Analysis of ECG: Cardiac Arrhythmia Detection, Pre-processing of ECG, Estimation of heart rate in ECG-Harmonic analysis - Heart rate monitoring, Pitch detection by using AI/ML.

Analysis of EEG & EMG, EEG rhythms, Detection of neurological disorders, Template matching for EEG, spike and wave detection. Estimation of the spectrum in EEG. Detecting artefacts in EEG & their characteristics using AI/ML.

AI in Biomedical Engineering: Predicting cardiovascular risk using regression , Classifying cancer sub-types using clustering , Identifying synergistic drug combinations using RF, Predicting diabetic retinopathy using decision trees,

AI in medical imaging: Diagnosing breast cancer from biopsy images using NN & Transfer Learning, microbiome and data mining.

Learning Resources:

References:

1. Jorge Garza Ulloa, Applied Biomedical Engineering using Artificial Intelligence and cognitive models. Elsevier Science and Technology Publications 2021.

2. Eugene N. Bruce, Biomedical Signal Processing and Signal Modeling, John Wiley & Sons, 2000

3. Adam Gacek , Witold Pedrycz Editors, ECG Signal Processing, Classification and Interpretation _A Comprehensive Framework of Computational Intelligence", Springer

4. Tony J. Cleophas and Aeilko H. Zwinderman. 2015. Machine Learning in Medicine - a Complete Overview. Springer.

EC16042 **3-0-0 (3)**

Machine Learning for Speech and Audio Processing

Prerequisites:

Course Outcomes:

CO-1	Understand the fundamentals of AI & ML and their applications
CO-2	Analyze supervised and unsupervised machine learning algorithms for classification
CO-3	analyze the pre-processing techniques and machine learning applications in the areas of Speech processing
CO-4	analyze the pre-processing techniques and machine learning applications in the areas of Audio processing,
CO-5	analyze the pre-processing techniques and machine learning applications in the areas of Bio-Medical Signal processing,

Mapping of course outcomes with program outcomes:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	1	1	1	1	1
CO-2	1	1	1	1	1	1
CO-3	2	2	-	2	2	1
CO-4	1	2	-	2	2	1
CO-5	1	1	-	1	1	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to AI & ML: Introduction to AI, Definition, Types of AI, Subsets of AI., Future of Artificial Intelligence, Types of Learning, Examples of Machine Learning applications - Learning associations, Training versus Testing, Data Analytics for processing

Fundamentals of Classification and Estimation: Bayesian Probability Theory, classification and Estimation, Feature Extraction Techniques , Modeling of Statistical Distributions , Basic Classification Schemes ,

Probabilistic Models: K-Means Clustering, Gaussian Mixture Models (GMMs), Expectation-Maximization (EM) Algorithm

Modeling Sequential Data: Hidden Markov Models (HMMs), Estimation and Classification with HMMs , Linear Dynamical Systems (LDS)

Modelling Using Non-Negative Matrix Factorization (NMF): Non-negative Matrix Factorization, Generalized Kullback–Leibler Divergence NMF as a dimension reduction technique, Objective Function in NMF, Use Case, Applications , Advantages of NMF

Deep Neural Networks Applications for speech and audio processing: word segmentation (WS), information extraction (IE), relation extraction (RE), named entity recognition (NER), parts of speech

tagging(POS), coreference resolution, parsing, word sense disambiguation (WSD), speech recognition, text to speech (TTS), machine translation (MT) Recurrent Neural Nets for Audio Classification: Multiclass audio segmentation, Classifying Audio Music Genres Using CNN and RNN

Learning Resources:

Reading:

1. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
2. R. O. Duda, P. E. Hart, Pattern Classification, WILEY, (2001).

References:

1. George Kuddrayvtsev, Fundamentals of Computer Vision, 2020.
2. John G. Proakis, Dimitris G. Manolakis, Digital Signal Processing, Pearson Education, 2016, 4Th edition.
3. R.C. Gonzalez and R.E. Woods, “Digital Image processing”, Pearson Education, 2nd ed. New Delhi. 2004.
4. Ranga Raj. M. Rangayyan, Biomedical Signal Analysis: A Case, study approach, John Wiley & sons Inc, 2002, 1st Edition.

EC16044 **3-0-0 (3)**

Deep Learning Architectures

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand the basic concepts of neural networks and deep learning methods
CO-2	Know the basic model types used in deep learning, e.g., Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Generative Adversarial Networks (GANs)
CO-3	Build and train deep neural networks and identify key architecture parameters
CO-4	Modify state-of-the-art deep learning architectures for a new dataset/task. And re-train and tune hyper parameters of several classes of deep learning methods.
CO-5	Know the suitability of specific deep learning methods to various real world data domains such as the ones arising from text, images, and videos.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	1	1	1	2	1
CO-2	2	2	1	1	2	1
CO-3	2	2	1	3	3	1
CO-4	2	2	1	2	2	1
CO-5	2	1	1	2	2	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to Deep Learning: Neural Networks, The basic building blocks of deep learning, Training of Neural Networks.

Convolutional Neural Networks (CNNs): CNN architectural details, CNN architectures – LeNet, AlexNet, VGG, Resenet, Googlenet, Inceptionnet. Training a CNNs: weights initialization, batch normalization, hyperparameter optimization, Understanding and visualizing CNNs

Recurrent Neural Networks (RNNs): RNNs, Need of RNN, The basic building blocks of RNNs and other architectural details, Study of some special RNN architectures: GRU, LSTM, Seq2Seq models, Attention mechanism.

Unsupervised deep learning: Self-organized maps, Autoencoders

Generative models: Architectural and training of Generative Adversarial Networks (GANs) , Restrictive Boltzmann Machines (RBMs), Stacking RBMs, Belief nets, Learning sigmoid belief nets, Deep belief nets Adversarial attacks on Neural Networks.

Learning Resources:

Text Books:

1. Ian Good fellow, Yoshua Bengio and Aaron Courville, Deep learning, In preparation for MIT Press, Available online: http://www.deeplearningbook.org, 2016

Reference Books:

1. Satish Kumar, Neural Networks - A Class Room Approach, Tata McGraw-Hill, 2013, Second Edition.
2. Charu C. Aggarwal Neural Networks and Deep Learning Springer International Publishing, 2018,1st Edition.
3. Andrew W. Trask, Grokking Deep Learning Manning Publications,2019.