# COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

## DEPARTMENT OF COMPUTER SCIENCE

### PROGRAMME STRUCTURE & SYLLABUS
### [2021 ADMISSIONS ONWARDS]

**M.TECH. COMPUTER SCIENCE &ENGINEERING WITH SPECIALIZATION IN DATA SCIENCE & ARTIFICIAL INTELLIGENCE**

# SYLLABUS
# FOR
# OUTCOME BASED EDUCATION

In **Master of Technology (M.Tech.)** Degree Program
in **COMPUTER SCIENCE &ENGINEERING WITH SPECIALIZATION IN DATA SCIENCE & ARTIFICIAL INTELLIGENCE**

*For the student admissions starting from the academic year 2021-2022*

## Program Outcomes (PO) for the M.Tech. Program in COMPUTER SCIENCE &ENGINEERING WITH SPECIALIZATION IN DATA SCIENCE & ARTIFICIAL INTELLIGENCE

*After the completion of M.Tech. programme, the students will be able to:*

PO1: Elicit deeper and current knowledge through research/exploration leading to development work with a motivation to solve practical problems.

PO2: Communicate effectively through well-written technical documentation as well as audio-visual Presentations.

PO3: Recognize the importance of entrepreneurship and innovation to create value and wealth.

PO4: Acquire mastery in the topic of study at an exceedingly higher level.

**DEPARTMENT OF COMPUTER SCIENCE**
**PROGRAMME STRUCTURE AND SYLLABUS (2021 ADMISSIONS)**

**M. TECH. COMPUTER SCIENCE &ENGINEERING WITH SPECIALIZATION IN DATA SCIENCE & ARTIFICIAL INTELLIGENCE**

**Semester - I**

| Sl. No. | Course code | Course Title | Core /Elective | Credits | Lec | Lab/ Tutorial | Marks |
|---|---|---|---|---|---|---|---|
| 1 | 21-479-0101 | Mathematical Concepts for Computer Science | C | 4 | 4 | 2 | 100 |
| 2 | 21-479-0102 | Machine Learning Algorithms | C | 4 | 4 | 3 | 100 |
| 3 | 21-479-0103 | Design and Analysis of Algorithms | C | 4 | 4 | 3 | 100 |
| 4 | - | Elective I | E | 3 | 4 | 1 | 100 |
| 5 | - | Elective II | E | 3 | 4 | 1 | 100 |
| Total for Semester I | | | | 18 | 20 | 10 | 500 |

Electives

- 21-479-0104: Virtualized Systems
- 21-479-0105: Computational Linguistics
- 21-479-0106: Advanced Optimization Techniques
- 21-479-0107: Algorithms for Modern Data Models
- 21-479-0108: Digital Image and Video Processing
- 21-479-0109: Mathematics for Machine Learning
- 21-479-0110: Number Theory and Cryptography

**Semester - II**

| Sl. No. | Course code | Course Title | Core /Elective | Credits | Lec | Lab/ Tutorial | Marks |
|---|---|---|---|---|---|---|---|
| 1 | 21-479-0201 | Algorithms for Massive Datasets | C | 4 | 4 | 2 | 100 |
| 2 | 21-479-0202 | Probabilistic Graphical Models | C | 4 | 4 | 2 | 100 |
| 3 | 21-479-0203 | Seminar | C | 1 | | 3 | 50 |
| 4 | - | Elective III | E | 3 | 4 | 1 | 100 |
| 5 | - | Elective IV | E | 3 | 4 | 1 | 100 |
| 6 | - | Elective V | E | 3 | 4 | 1 | 100 |
| Total for Semester II | | | | 18 | 20 | 10 | 550 |

Electives

- 21-479-0204: Bioinformatics
- 21-479-0205: Programming Massively Parallel Processors
- 21-479-0206: Deep Learning
- 21-479-0207: Modelling Cyber Physical Systems
- 21-479-0208: Algorithmic Game Theory
- 21-479-0209: Deep Learning for Computer Vision
- 21-479-0210: Image and Video Coding
- 21-479-0211: Reinforcement Learning
- 21-479-0212: Natural Language Processing with Deep Learning

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Semester - III** | | | | | | | |
| 1 | 21-479-0301 | Elective - VI | E | 2 | 0 | 10 | 50 |
| 2 | 21-479-0302 | Project & Viva Voce | C | 16 | 0 | 20 | 350 |
| | | | Total for Semester III | 18 | 0 | 30 | 400 |
| **Semester - IV** | | | | | | | |
| 1 | 21-479-0401 | Project & Viva Voce | C | 18 | 0 | 30 | 500 |
| Total credits for Degree: **72** | | | | | | | |

## 21-479-0101: Mathematical Concepts for Computer Science

Core/Elective: **Core** Semester: **1** Credits: **4**

### Course Description

This course introduces the study of mathematical structures that are fundamentally discrete in nature. The course is intended to cover the main aspects which are useful in studying, describing and modeling of objects and problems in the context of computer algorithms and programming languages.

### Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Analyse the different methods for proving the correctness of the theorems and problems.
CO2: Understand the basic concepts of number theory.
CO3: Understand the basic aspects of graph theory.
CO4: Evaluate the performance of various graph-based algorithms.
CO5: Understand the fundamentals of probability theory.
CO6: Apply various probability density functions and its moments to solve problems.

### Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1
CO4: PO1, PO4
CO5: PO1
CO6: PO1, PO4

### Course Content

1. Introduction – proofs – propositions – predicates and quantifiers – truth tables – first order logic – satisfiability – pattern of proof – proofs by cases – proof of an implication – proof by contradiction – proving iff – sets – proving set equations – Russell's paradox – well-ordering principle – induction – invariants – strong induction – structural induction

2. Sums – arithmetic, geometric and power sums – approximating sums – harmonic sums – products – Stirling's approximation for finding factorial-Pigeon hole principle – parity – number theory – divisibility – gcd – Euclid's algorithm – primes.

3. Graph theory – simple graphs – isomorphism – subgraphs – weighted graphs – matching problems – stable marriage problem – graph coloring – paths and walks – shortest paths – connectivity – Eulerian and Hamiltonian tours – travelling salesman problem – trees – spanning trees – planar graphs – Euler's formula – directed graphs – strong connectivity – relations – binary relations – surjective and injective relations symmetry, transitivity, reflexivity, equivalence of relations – posets and dags – topological sort.

4. Probability – events and probability spaces – conditional probability – tree diagrams for computing probability – sum and product rules of probability – A posteriori probabilities – identities of conditional

probability – independence – mutual independence – birthday paradox – random variables – indicator random variables.

5. Probability distribution functions – Bernoulli, Uniform, Binomial, Poisson, Normal distributions – Expectation – linearity of expectations – sums of indicator random variables – expectation of products – variance and standard deviation of random variables – Markov and Chebyshev's theorems – Bounds for the sums of random variables.

## References


1. Eric Lehman, F Thomson Leighton, Albert R Meyer, Mathematics for Computer Science, 1e, MIT, 2010.
2. Susanna S. Epp, Discrete Mathematics with Applications, 4e, Brooks Cole, 2010.
3. Gary Chartrand, Ping Zhang, A First Course in Graph Theory, 1e, Dover Publications, 2012.
4. Michael Sipser, Introduction to Theory of Computation, 3e, Cengage, 2014.
5. Sheldon Ross, A First Course in Probability, 9e, Pearson, 2013.
6. Tom Leighton, and Marten Dijk. 6.042J Mathematics for Computer Science.Fall 2010. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu.
7. John Tsitsiklis. 6.041SC Probabilistic Systems Analysis and Applied Probability. Fall 2013. Massachusetts Institute of Technology: MIT OpenCourseWare. https://ocw.mit.edu
8. Igor Pak. 18.315 Combinatorial Theory: Introduction to Graph Theory, Extremal and Enumerative Combinatorics. Spring 2005. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu
9. Albert Meyer. 6.844 Computability Theory of and with Scheme. Spring 2003. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu.
10. Shai Simonson , Theory of Computation, http://www.aduni.org/courses/theory/

## 21-479-0102: Machine Learning Algorithms

Core/Elective: **Core** Semester: **1** Credits: **4**

### Course Description

Machine learning is programming computers to optimize a performance criterion using example data or past experience. This course is to discuss many methods that have their bases in different fields: statistics, pattern recognition, neural networks, artificial intelligence, signal processing, control, and data mining. Major focus of the course is on the algorithms of machine learning to help students to get a handle on the ideas, and to master the relevant mathematics and statistics as well as the necessary programming and experimentation.

### Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Demonstrate strength and weakness of Machine Learning approaches.

CO2: Appreciate the underlying mathematical relationships within and across algorithms and different paradigms of Machine Learning.

CO3: Utilize dimensionality reduction techniques for feature selection.

CO4: Examine methods for model building and fine tuning.

CO5: Experiment with Machine learning tools.

CO6: Apply Machine Learning algorithms to many real world problems.

### Mapping with Program Outcomes

CO1: PO1

CO2: PO1

CO3: PO1, PO4

CO4: PO1, PO4

CO5: PO1, PO4

CO6: PO1, PO4

### Course Content

1. Machine Learning – Examples of Machine Learning applications – Supervised Learning: Learning a class from examples – Learning multiple classes – Regression – Model selection – Bayesian Decision Theory: Classification – Discriminant functions – Association rules – Parametric methods: MLE – Bayes estimator – Parametric classification – Tuning model complexity

2. Multivariate Methods – Classification – Regression – Dimensionality reduction: LDA – PCA – Factor Analysis – ICA – Locally Linear Embedding – MDS- Probabilistic Learning: Gaussian Mixture Models- EM algorithm- Nearest Neighbor Methods – Distance Measures

3. Support Vector Machines: Optimal separation – Kernels – SVM algorithm – Extensions to SVM – Optimization and Search: Least-squares optimization – conjugate gradients – Search: Search techniques – Exploitation and exploration – Simulated annealing

4. Learning with trees: Decision trees – CART – Ensemble Learning: Boosting – Bagging – Random Forests – Unsupervised Learning: K-Means algorithm – Vector quantization – SOM algorithm – Markov Chain Monte Carlo Methods

5. Graphical Models: Bayesian Networks – Markov Random Fields – HMMS – Tracking Methods – Deep Belief Networks: Hopfield Network – Boltzmann Machine – RBM – Deep Learning

**References**


1. Ethem Alpaydin, Introduction to Machine Learning, 3e, MIT Press, 2014.
2. Tom M. Mitchell, Machine Learning, McGraw Hill Education; 1e, 2017.
3. Stephen Marsland, Machine Learning, An Algorithmic Perspective, 2e, CRC Press, 2015.
4. Giuseppe Bonaccorso, Machine Learning Algorithms, 1e, Packt Publishing Limited, 2017.
5. Ethem Alpaydin, Machine Learning- The New AI, MIT Press, 1e, 2016.
6. Andrew Ng, Machine Learning Yearning, ATG AI (Draft version), 1e, 2018.
7. Rohit Singh, Tomi Jaakkola, and Ali Mohammad. *6.867 Machine Learning.* Fall 2006. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu
8. Andrew Ng, https://www.coursera.org/learn/machine-learning

## 21-479-0103: Design and Analysis of Algorithms

Core/Elective: **Core** Semester: **1** Credits: **4**

**Course Description:**

The course covers the foundational algorithms in depth. The course helps in understanding the working and complexity of the fundamental algorithms and to develop the ability to design algorithms to attack new problems.

**Course Outcomes (CO)**

After the completion of the course, the students will be able to:

CO1: Understand the basic concepts of design and analysis of fundamental algorithms.
CO2: Develop the ability to design algorithms to attack new problems.
CO3: Prove the correctness of algorithms.
CO4: Develop the ability to analyze the complexity of algorithms.
CO5: Understand Complexity classes, concepts of P and NP problems

**Mapping with Program Outcomes**

CO1: PO1
CO2: PO1, PO4
CO3: PO2, PO4
CO4: PO1, PO2
CO5: PO1, PO4

**Course Content**

1. Introduction to design and analysis of algorithms, models of computation, correctness proofs, insertion sort, computational complexity, Master theorem , proof of Master theorem, merge sort, heaps, heap sort, binary search, binary search trees.

2. Graph algorithms, BFS and DFS, Dijkstra’s algorithm, proof of correctness of Dijkstra’s algorithm, Complexity analysis of Dijkstra’s algorithm , Negative weight edges and cycles , Bellman-Ford algorithm, proof of correctness and complexity of Bellman-Ford, All pairs shortest paths, Floyd-Warshall algorithm, proof of correctness and complexity, Minimum Spanning Trees , Prim’s algorithm, Cut property, Kruskal’s algorithm, proof of correctness and complexity analysis of Kruskal's Algorithm, Maximum-Flow networks, Ford-Fulkerson method, proof of correctness and complexity, Edmonds-Karp algorithm

3. Probability review, Experiments, outcomes, events, Random variables, Expectation, Linearity of Expectation, Indicator Random Variables, Hiring Problem, Quicksort , Best case and Worst case complexity, Randomized Quicksort , Average case complexity , Hashing, Chaining, Open Addressing, Universal Hashing, Perfect Hashing , Analysis of hashing operations

4. Dynamic Programming , Rod-cutting problem, Recursive formulation, Bottom-up reformulation of

recursive algorithms, Optimal Substructure Property, Matrix chain multiplication, Complexity of dynamic programming algorithms, Sequence Alignment , Longest common subsequence, Greedy algorithms, Optimal substructure and greedy-choice properties , 0-1 and fractional Knapsack problems, Huffman coding

5.P vs NP, NP Hardness, Reductions, Travelling Salesman Problem, NP-Completeness, SAT, 2-SATand 3-SAT, Vertex Cover

**References**


1.Thomas H. Cormen et al, Introduction to Algorithms, 3e, MIT Press, 2009.

2.Jon Kleinberg, Eva Tardos, Algorithm Design, 2e, Pearson, 2015.

3. Robert Sedgewick, Kevin Wayne, Algorithms, 4e, AW Professional, 2011.

4. Steven S. Skiena, The Algorithm Design Manual, 2e, Springer, 2011.

# 21-479-0104: Virtualized Systems

Core/Elective: **Elective** Semester: **1** Credits: **3**

## Course Description

Virtualization provides the benefit of reducing the total cost of ownership and improving the business agility. This course systematically introduces the concepts and techniques used to implement the major components of virtual servers behind the scene. It discusses the details on hypervisor, CPU scheduling, memory management, virtual I/O devices, mobility, and etc.

## Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Explain virtualization concepts.
CO2: Illustrate the merits of server virtualization.
CO3: Explain CPU virtualization.
CO4: Identify the roles of type 1 and type 2 hypervisors.
CO5: Explain the memory management techniques for virtualized systems.
CO6: Explain the methods of virtualization at I/O levels.
CO7: Outline the virtualization support for different cloud computing models.

## Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1, PO4
CO4:.PO1, PO4
CO5:.PO1, PO4
CO6: PO1
CO7:.PO1, PO2

## Course Content

1. Overview: Why server virtualization –History and re-emergence – General structures – Architectures comparison - Commercial solutions – VMWare, Xen

2. Virtual machines: CPU virtualization -Privileged instructions handling -Hypervisor – Paravirtualization - Hardware-assisted virtualization - Booting up - Time keeping - CPU scheduling- Commercial examples.

3. Memory management in virtualization: partitioning –reclamation –ballooning. Memory sharing.OS-level virtualization –VMWare –Red Hat Enterprise Virtualization

4. I/O virtualization: Virtualizing I/O devices -monolithic model -virtual I/O server - Virtual networking –tunneling –overlay networks - Commercial examples. Virtual storage: Granularity - file system level–blocks level

5. Virtualized computing: Virtual machine based distributed computing - elastic cloud computing clustering - cold and hot migration - Commercial examples - Challenges and future trends

## References


1. Jim Smith, Ravi Nair , Virtual Machines: Versatile Platforms for Systems and Processes, 1e, Morgan Kaufmann, 2005
2. Sean Campbell, Applied Virtualization Technology -Usage models for IT professionals and Software Developers, 1e, Intel Press, 2006
3. Matthew Portnoy , Virtualization Essentials, 1e, JW, 2012
4. George Trujillo, Charles Kim, Steve Jones, Rommel Gracia, Justin Murray , Virtualizing Hadoop, VM Press, 2015

## 21-479-0105: Computational Linguistics

Core/Elective: **Elective** Semester: **1** Credits: **3**

### Course Description:

Computational Linguistics deals with statistical and rule based modelling of natural languages from a computational point of view. This course is intended to give a comprehensive coverage of language processing fundamentals like morphology, Syntax, Semantics and pragmatics. Application of various computational models in application domains like Machine translation, information retrieval etc. is also dealt with.

### Course Outcomes (CO)

After the completion of the course, the students will be able to:

**CO1 :** Understand the fundamentals of written language processing
**CO2:** Applying theses fundamentals in real world problems like POS tagging, Corpus development, WordNet, Dialogue processing, document retrieval, Machine translation etc etc
**CO3:** Creating resources for less resource languages
**CO4:** Case study of various typical Language processing tools.

### Mapping with Program Outcomes

CO1: PO1,PO2,PO4
CO2:PO1,PO2,PO3,PO4
CO3:PO1,PO2,PO3,PO4
CO4:PO1,PO2,PO3,PO4

### Course Content

1. Words- Regular Expressions and Finite Automata-Morphology and Finite State Transducers-Probabilistic Models of Pronunciation and Spelling -N grams

2. Word Classes and Part-of-Speech Tagging-MM Taggers- probabilistic Context Free Grammars for English Syntax-Parsing with Context Free Grammars- probabilistic parsing- Features and Unification-Language and Complexity

3. Semantics-Representing Meaning-canonical forms-FOPC-ambiguity resolution-scoping phenomena-Semantic Analysis-syntax driven semantic analysis-Lexical Semantics-Word Sense Disambiguation and Information Retrieval

4. Discourse-Reference Resolution -Text Coherence -Dialog and Conversational Agents-Dialogue acts-dialogue structure

5. Statistical alignment and machine translation-clustering- text categorization

## References

1. James Pustejovsky, Amber Stubbs, Natural language annotation for machine learning, 1e, O'Reilly, 2013
2. Alexander Clark and Chris Fox, The handbook of Computational linguistics and natural language processing, 1e, Willey-Blackwell, 2012
3. Grant S Ingersoll, Thomas Morton, Andrew L Farris, Taming Text: How to Find, Organize, and Manipulate It, 1e, Manning Publications 2013
4. Daniel Jurafsky and James Martin , Speech and Language Processing, 2e, Pearson, 2013
5. Christopher D. Manning and HinRich Schütze, Foundations of statistical natural language processing, 1e, MIT press, 1999

# 21-479-0106: Advanced Optimization Techniques

Core/Elective: **Elective** Semester: **1** Credits: **3**

## Course Description

This course is about the well-known population-based optimization techniques developed during the last three decades. This course emphasizes on the advanced optimization techniques to solve large-scale problems especially with nonlinear objective functions.

## Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Understand the basic concepts of optimization and its applications.
CO2: Understand the mathematical representation and classical methods for solving optimization problems.
CO3: Explain and demonstrate working principles of various population based optimization techniques.
CO4: Explain and Demonstrate working principle of various Hybrid Algorithms for optimization.

## Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1, PO2, PO4
CO4: PO1, PO2, PO4

## Course Content

1. Introduction to optimization- formulation of optimization problems-Review of classical methods-Linear programming-Nonlinear programming-Constraint optimality criteria-constrained optimization-Population based optimization techniques
2. Genetic Algorithm-Introduction-Working principle-Representation-selection-fitness assignment-reproduction-cross over-mutation-constraint handling-advanced genetic algorithms- Applications-Artificial Immune Algorithm-Introduction-Clonal selection algorithm- Negative selection algorithm-Immune network algorithms-Dendritic cell algorithms
3. Differential Evolution-Introduction-Working principles-parameter selection-advanced algorithms in Differential evolution-Biogeography-Based Optimization-Introduction-Working Principles-Algorithmic variations
4. Particle Swarm Optimization-Introduction- Working principles- Parameter selection- Neighborhoods and Topologies-Convergence-Artificial Bee Colony Algorithm-Introduction- Working principles-Applications-Cuckoo search based algorithm-Introduction- Working principles- Random walks and the step size-Modified cuckoo search
5. Hybrid Algorithms-Concepts- divide and conquer- decrease and conquer-HPABC-HBABC-HDABC-HGABC-Shuffled Frog Leaping Algorithm-- Working principles -Parameters- Grenade Explosion Algorithm-Working principle-Applications

## References


1. Dan Simon, Evolutionary Optimization Algorithms, 1e, Wiley, 2013
2. Xin-She Yang, Engineering Optimization: An Introduction with Meta-heuristic Applications, 1e, Wiley, 2010
3. S.S. Rao, Engineering Optimization: Theory and Practice, 4e,New Age International, 2013
4. R. VenkataRao, Teaching Learning Based Optimization Algorithm: And Its Engineering Applications, 1e, Springer, 2016

# 21-479-0107: Algorithms for Modern Data Models

Core/Elective: **Elective** Semester: **1** Credits: **3**

## Course Description:

There exist both algorithmic and statistical challenges in modern large-scale applications and data analysis. This course describes the randomization and probabilistic techniques for modern computer science, with applications ranging from combinatorial optimization and machine learning to communication networks. The course covers the core material to advanced concepts. Also the emphasis is on methods useful in practice.

## Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Relate the advanced concepts of probability theory and modern applications.
CO2: Explain the uncertainty in prediction due to intervention of random variables..
CO3: Examine random graphs and their properties.
CO4: Analyze evolutionary algorithms.
CO5: Interpret algorithms for evolving data streams
CO6: Build algorithms for new problems with volume of data

## Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1, PO4
CO4:.PO1, PO4
CO5:.PO1, PO4
CO6: PO1, PO4

## Course Content:

1. Probability: Expectations - Tail Bounds - Chernoff Bound – Balls and Bins – Probabilistic Method – Markov chains and Random walks
2. Entropy, Randomness, and Information: Measure of randomness – Monte Carlo Method – Markov Chain Monte Carlo Method
3. Graph models and algorithms– Random graph Models- Algorithms for graph generation - Random graphs as models of networks, Power laws, Small world Phenomena
4. Components of evolutionary algorithms – Example applications – Genetic algorithms – Evolution strategies – Evolutionary programming
5. Sampling, sketching, data stream models, read-write streams, stream-sort, map-reduce - Algorithms in evolving data streams

## References


1. Michael Mitzenmacher, Eli Upfal, Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis, 2e, Cambridge University Press, 2017

2. Rajeev Motwani and PrabhakarRaghavan, Randomized Algorithms, Cambridge University Press; Reprint edition, 2010
3. S. Muthukrishnan, Data Streams: Algorithms and Applications, 1e, Now Publishers, 2005
4. Charu C. Aggarwal, Data Streams: Models and Algorithms, 1e, Springer, 2006
5. Agoston E. Eiben, J.E. Smith , Introduction to evolutionary computing, 1e, Springer, 2010

## 21-479-0108: Digital Image and Video Processing

Core/Elective: **Elective** Semester: **1** Credits: **3**

### Course Description

The aim of this course is to inculcate a comprehensive knowledge about various Digital Image and Video Processing techniques. The objectives are to give an in-depth knowledge about the basic theory and algorithms related to Digital Image and Video Processing, provide awareness about the current technologies and issues, provide hands-on experience in using computers to process digital images and Videos using Python and OpenCV library.

### Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Understand the fundamental concepts of signal and image processing systems.
CO2: Evaluate the different spatial filters for image enhancement and restoration.
CO3: Analyze images in the frequency domain using various transforms.
CO4: Apply the different spatial and frequency domain filters on color images.
CO5: Evaluate the performance of different image segmentation algorithms.
CO6: Understand the fundamental concepts of video processing systems.
CO7: Evaluate the different algorithms for motion estimation from videos.
CO8: Understand the techniques for depth estimation from stereo images or videos.
CO9: Develop any image or video processing application.

### Mapping with Program Outcomes

CO1: PO1
CO2: PO1, PO4
CO3: PO1
CO4: PO1, PO4
CO5: PO1, PO4
CO6: PO1
CO7: PO1, PO4
CO8: PO1
CO9: PO1, PO2, PO4

### Course Content

1. Signals: Impulse Sequence - Exponential Sequence - Periodic Sequence. Linear Systems - Shift-Invariant systems - Linear Shift Invariant (LSI) systems – Convolution - Correlation. Image Transforms: Fourier Transform - Discrete Fourier Transform - Z- transform – KL Transform. Causal Systems - Random Signals - Stationary Process - Markov Process.

2. Intensity Transformation and Spatial Filtering: Intensity Transformation Functions. Histogram Processing: Histogram Equalization - Histogram Matching. Image enhancement: Arithmetic/Logic operations - Image Subtraction - Image Averaging. Spatial Filtering: Smoothening Spatial Filters - Sharpening Spatial Filters - Laplacian Filter - Unsharp masking - High Boost Filter. Gradient operators: Edge detection filters. Frequency Domain Smoothening - Frequency Domain Sharpening Filters - Laplacian in Frequency domain - Homomorphic Filtering.

3. Image degradation/Restoration process model - Noise probability density functions - Spatial Filtering: Mean Filters - Order-statistics filter - Adaptive Filters - Periodic Noise Reduction –Band-reject filters - Band-pass filters - Notch filters. Inverse filtering - Wiener filtering - Performance measures. Color image processing: Color fundamentals - Color models – RGB, CMYK – HIS - Color image smoothening and sharpening – Color image histogram - Color edge detection.

4. Point and line detection - Hough Transform. Image Segmentation: Fundamentals – Thresholding – Otsu's optimum global thresholding - Region-based segmentation: Region growing - Region Splitting and Merging - Segmentation using Morphological Watersheds.

5. Color video processing: Video display - Composite versus component video - Progressive and interlaced scan. Motion estimation: Optical flow - pixel based motion estimation - block matching algorithm - deformable block matching algorithm - Global and region based motion estimation - multiresolution motion estimation - Feature based motion estimation. Stereo and multi-view sequence processing: Depth perception - Stereo imaging principle - Disparity estimation.

## References


1. Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", 4th Ed., Pearson, March 2017.
2. Anil K. Jain, "Fundamentals of Digital Image Processing", Pearson, 1st Ed., 1988.
3. William K. Pratt, "Digital Image Processing: PIKS Scientific Inside", John Wiley & Sons, 4th Ed., 2007.
4. Azriel Rosenfield, Avinash C. Kak, "Digital Picture Processing", Morgan Kaufmann, 2nd Ed., 1982.
5. Bernd Jahne, "Digital Image Processing", Springer, 6th Ed., 2005.
6. Yao Wang, Jorn Ostermann, Ya-Qin Zhang, "Video Processing and Communications", Pearson, 1st Ed., 2001.
7. Alan C. Bovik, "The Essential Guide to Video Processing", Academic PRess, 2nd Ed., 2009
8. A. Murat Tekalp, "Digital Video Processing", Prentice Hall, 2nd Ed., 2015.

## 21-479-0109: Mathematics for Machine Learning

Core/Elective: **Elective** Semester: **1** Credits: **3**

### Course Description

The aim of this course is to inculcate a comprehensive knowledge about mathematical formalisms required to understand machine learning concepts. The course introduces in detail linear algebra, probability concepts, optimization, and some of the applications

### Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Outline the fundamental concepts of linear algebra.
CO2: Illustrate matrix diagonalization.
CO3: Analyze the process of backpropagation.
CO4: Apply Bayes theorem.
CO5: Analyze the gradient descent algorithm
CO6: Examine linear programming problems.
CO7: Build some of the basic machine learning applications .

### Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1
CO4: PO1, PO4
CO5: PO1, PO4
CO6: PO1, PO4
CO7: PO1, PO4

### Course Contents:

1. Linear Algebra – vectors – matrices – systems of linear equations – vector spaces – linear independence –basis and rank – linear mappings – affine spaces – Norms – lengths and distances – angles and orthogonality– orthonormal basis – inner product of functions – orthogonal projections – rotations

2. Determinant and trace – eigenvalues and eigenvectors – cholesky decomposition – eigendecomposition and diagonalization – singular value decomposition – matrix approximation – Partial differentiation – gradients – gradients of vectors and matrices – higher order derivatives – backpropagation and automatic differentiation – multivariate Taylor series

3. Probability review – conditioning and independence – Bayes theorem – counting – discrete and continuous random variables – discrete and continuous probability distributions – Gaussian distribution – Bayesian inference – limit theorems – estimation – conjugacy and exponential family – inverse transform – sampling from distributions

4. Optimization – gradient descent – choosing the right step size – gradient descent with momentum – stochastic gradient descent – constrained optimization and Lagrange multipliers – convex optimization – linear programming – quadratic programming – Empirical risk minimization – probabilistic modeling and inference – directed graphical models

5. Applications: linear regression – parameter estimation – Bayesian Linear Regression – PCA – Maximum Variance Projections – Low-Rank Approximations – Gaussian mixture models – Parameter learning via maximum likelihood – EM Algorithm – Support Vector Machines – Separating Hyperplanes – Primal and Dual forms – The Kernel Trick

**References:**


1. Gilbert Strang, Linear Algebra and Learning from Data, Wellesley-Cambridge Press, 2019
2. Marc Peter Deisenroth et al., Mathematics for Machine Learning, 1e, Cambridge Press, 2020, Ebook: https://mml-book.com
3. Mehryar Mohri et al., Foundations of Machine Learning, 2nd Edition, The MIT Press, 2018
4. Gilbert Strang, Introduction to Linear Algebra, 5th Edition, Wellesley-Cambridge Press, 2016
5. James Stewart, Multivariable Calculus, 7th Edition, Cengage Learning, 2011
6. Dimitri P. Bertsekas, John N. Tsitsiklis, Introduction to Probability, 2nd Edition, Athena Scientific, 2008.
7. Morris H. DeGroot, Mark J. Schervish, Probability and Statistics, 4th Edition, Pearson, 2011

## 21-479-0110: Number Theory and Cryptography

**Core/Elective: Elective** **Semester: 1** **Credits: 3**

### Course Description:

The course provides an introduction to basic number theory, where the focus is on computational aspects with applications in cryptography. Applications to cryptography are explored including symmetric and public-key cryptosystems. Modern cryptographic methods are also discussed.

### Course Outcomes (CO)

After the completion of the course, the students will be able to:

CO1: Demonstrate the number theoretic foundations of modern cryptography
CO2: Analyze cryptographic and number theoretic algorithms
CO3: Build cryptographic and number theoretic algorithms
CO4: Illustrate public key cryptosystems
CO5: Relate modern cryptographic techniques

### Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1, PO4
CO4: PO1, PO2
CO5: PO1, PO2

### Course Content:

1. Divisibility, Division Algorithm, Euclidean Algorithm, Congruences, Complete Residue systems, Reduced Residue systems, Fermat's little theorem, Euler's Generalization, Wilson's Theorem, Chinese Remainder Theorem, Euler Phi-function, multiplicative property, Finite Fields, Primitive Roots, Quadratic Residues, Legendre Symbol, Jacobi Symbol, Gauss's lemma, Quadratic Reciprocity Law

2. Primality Tests, Pseudoprimes, Carmichael Numbers, Fermat's pseudoprimes, Euler pseudoprimes, Factorization by Pollard's Rho method, Simple Continued Fraction, simple infinite continued fractions, Approximation to irrational numbers using continued fractions, Continued Fraction method for factorization.

3. Traditional Cryptosystem, limitations, Public Key Cryptography Diffie-Hellman key exchange, Discrete Logarithm problem, One-way functions, Trapdoor functions, RSA cryptosystem, Digital signature schemes, Digital signature standards, RSA signature schemes, Knapsack problem, ElGamal Public Key Cryptosystem, Attacks on RSA Cryptosystem: Common modulus attack, Homomorphism attack, timing attack, Forging of digital signatures, Strong primes, Safe primes, Gordon's algorithm for

generating strong primes.

4. Cubic Curves, Singular points, Discriminant, Introduction to Elliptic Curves, Geometry of elliptic curves over reals, Weierstrass normal form, point at infinity, Addition of two points, Bezout's theorem, associativity, Group structure, Points of finite order

5. Elliptic Curves over finite fields, Discrete Log problem for Elliptic curves, Elliptic Curve Cryptography, Factorization using Elliptic Curve, Lenstra's algorithm, ElGamal Public Key Cryptosystem for elliptic curves

## References


1. James S. Kraft and Lawrence C. Washington, An Introduction to Number Theory with Cryptography, 1e, CRC Press, 2013
2. Jill Pipher, Jeffrey Hoffstein, Joseph H. Silverman, An Introduction to Mathematical Cryptography, 2e, Springer, 2014
3. Christof Paar and Jan Pelzl, Understanding Cryptography, 1e, Springer, 2010
4. G.H.Hardy and Edward M Wright, An Introduction to theory of numbers, 1e, Oxford, 2008
5. : Song Y.Yan, Computational Number Theory & Modern Cryptography, 1e, Wiley, 2013

## 21-479-0201: Algorithms for Massive Datasets

Core/Elective: **Core** Semester: **2** Credits: **4**

### Course Description

Big Data concerns large-volume, complex, growing data sets with multiple, autonomous sources. With the fast development of networking, data storage, and the data collection capacity, Big Data is now rapidly expanding in all science and engineering domains. The traditional data mining algorithms also need to be adapted for dealing with the ever-expanding datasets of tremendous volume.

### Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Build practical skills on developing MapReduce jobs.
CO2: Explain the search algorithms that perform efficiently on massive datasets.
CO3: Explain the algorithms for data stream processing.
CO4: Explain the link analysis methods in the context of social networks and Page rank algorithms.
CO5: Demonstrate the power of some of the online algorithms for massive data.
CO6: Explain the randomized, approximate and one-pass algorithms for mining from massive datasets.

### Mapping with Program Outcomes

CO1: PO1, PO4
CO2: PO1
CO3: PO1
CO4:.PO1
CO5:.PO1, PO4
CO6: PO1, PO2

### Course Content

1. Introduction to MapReduce – the map and reduce tasks, MapReduce workflow, fault tolerance. - Algorithms for MapReduce – matrix multiplication, relational algebra operations- Complexity theory for MapReduce

2. Locality-Sensitive Hashing - shingling of documents, min-hashing. Distance measures, nearest neighbors, frequent itemsets- LSH families for distance measures, Applications of LSH- Challenges when sampling from massive data

3. Mining data streams – stream model, stream data sampling, filtering streams – bloom filters, counting distinct elements in a stream - Flajolet-Martin algorithm. Moment estimates - Alon-Matias-Szegedy algorithm, counting problems for streams, decaying windows

4. MapReduce and link analysis- PageRank iteration using MapReduce, topic-sensitive

PageRank - On-line algorithms – Greedy algorithms, matching problem, the adwords problem – the balance algorithm

5. Computational model for data mining – storage, cost model, and main memory bottleneck. Hash based algorithm for mining association rule – improvements to a-priori, park-chen-yu algorithm, multistage algorithm, approximate algorithm, limited-pass algorithms – simple randomized algorithm, Savasere, Omiecinski, and Navathe algorithm, Toivonen algorithm

**References**


1. Jure Leskovec, Rajaraman, A., & Ullman, J. D., Mining of Massive Datasets, Cambridge University Press, 2e, 2016
2. Charu C. Aggarwal, Data Streams: Models and Algorithms, 1e, Springer, 2007
3. Michael I Jordan et.al , Frontiers in Massive Data analysis, 1e, National Academies Press, 2013
4. Nathan Marz & James Warren, Big Data: Principles and best practices of scalable realtime data systems, Manning Publications, 2015

# 21-479-0202: Probabilistic Graphical Models

Core/Elective: **Core** Semester: **2** Credits: **4**

## Course Description

Probabilistic Graphical models (PGM) are a foundation for understanding many methods of artificial intelligence, machine learning and estimation. Machine learning provides algorithms for solving problems by using training data. This course will give insight into how to formulate problems so that machine learning can be used effectively. Building good models can help learn with less data by constraining the learning space. Bayesian models are at the heart of most estimation methods. Formulation of these models is the first step in developing an estimation algorithm. The estimation itself is in many cases just inference on the model given some evidence. Approximate inference techniques such as those covered in this course are important in solving many very hard estimation problems in science and engineering. Data scientists, machine learning enthusiasts, engineers, and those who curious about the latest advances in machine learning will find PGM interesting

### Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Demonstrate application of Probability and Graph Theory in reasoning.
CO2: Discuss how different graphs represent both factorization and independent relations.
CO3: Utilize message passing algorithms for inference.
CO4: Examine methods for learning uncertainties in a model's parameters.
CO5: Experiment with graph building tools.
CO6: Apply Bayesian networks and Markov networks to many real world problems.

### Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1, PO4
CO4: PO1, PO4
CO5: PO1, PO4
CO6: PO1, PO4

## Course Content:

1. Probabilistic reasoning: Representing uncertainty with probabilities – Random variables and joint distributions – Independence – Querying a distribution - Graphs

2. Representation: Bayesian Network (BN) representation – Independencies in BN – Factorizing a distribution – D-separation- Algorithm for D-separation – From distributions to Graphs

3. Undirected Graphical Models: Factor products – Gibbs distribution and Markov networks – Markov network independencies – Factor graphs – Learning parameters – Conditional Random Fields

4. Gaussian Network Models: Multivariate Gaussians – Gaussian Bayesian networks – Gaussian Markov Random Fields – Exact Inference: variable elimination- Sum-product and belief updates – The Junction tree algorithm

5. Learning: Learning Graphical Models – Learning as optimization – Learning tasks – Parameter estimation – Structure learning in BN – Learning undirected models – Actions and decisions

## References:

1. Daphne Koller, Nir Friedman, Probabilistic Graphical Models- Principles and Techniques, 1e, MIT Press, 2009
2. Richard E. Neapolitan, Learning Bayesian Networks, 1e, Pearson, 2019
3. Christian Borgelt, Rudolf Kruse and Matthias Steinbrecher, Graphical Models- Methods for data analysis and Mining, 2e, Wiley, 2009
4. David Bellot, Learning Probabilistic Graphical Models in R, Packt Publishing, 1e, 2016
5. Luis Enrique Sucar, Probabilistic Graphical Models, 1e, Springer Nature, 2015
6. Coursera: https://www.coursera.org/specializations/probabilistic-graphical-models

## 21-479-0203: Seminar

Core/Elective: **Core** Semester: **2** Credits: **1**

### Course Description

The student has to prepare and deliver a presentation on a research topic suggested by the department before the peer students and staff. They also have to prepare a comprehensive report of the seminar presented.

### Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Find out relevant information for the topic.
CO2: Define clearly the topic for discussion.
CO3: Deliver the content effectively.
CO4: Organize the content with proper structure and sequencing.
CO5: Demonstrate the academic discussion skills to emphasize, argue with clarity of purpose using evidence for the claims.
CO6: Show ability to evaluate and reflect on critical questions.
CO7: Show attempts to reach across diverse disciplines and bring other schools of thoughts into the discussions.

### Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1, PO2
CO4:.PO1
CO5:.PO1
CO6: PO1, PO4
CO7:.PO1, PO4

# 21-479-0204: Bioinformatics

Core/Elective: **Elective** Semester: **2** Credits: **3**

## Course Description

Present fundamental concepts from molecular biology, computational problems in molecular biology and some efficient algorithms that have been proposed to solve them.

## Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Understand and appreciate basic concepts of molecular Biology and Human genome project.
CO2: Illustrate and explain various sequence alignment algorithms.
CO3: Understand the basic concepts of Fragment Assembly of DNA and demonstrate various algorithms for the same.
CO4: Demonstrate and evaluate different algorithms for identifying optimal phylogenetic trees.
CO5: Understand the concepts of structure prediction in molecular biology.
CO6: Understand and demonstrate an algorithm in the literature for the domain.

## Mapping with Program Outcomes

CO1: PO1
CO2: PO1, PO4
CO3: PO1
CO4: PO1, PO4
CO5: PO1, PO4
CO6: PO1, PO2, PO4

## Course Content

1. Basic concepts of molecular Biology-Proteins-Nucleic acids– genes and genetic synthesis – translation-transcription- protein Synthesis- Chromosomes- Maps and sequences- human genome project- sequence databases
2. Strings-Graphs-Algorithms- Comparing 2 sequences- Global & Local comparison-General Gap Penalty Function-Affix gap penalty function-comparing multiple sequences-Star alignments-Tree alignments-Database Search-PAM matrices BLAST-FAST –Issues
3. Fragment Assembly of DNA-Biological Background –Models-Algorithms-Heuristics-Physical Mapping of DNA-Restriction site Mapping-site models-Internal Graph Models –Hybridization Mapping-Heuristics
4. Phylogenetic Trees –Binary Character States-Parsimony and Compatibility in Phylogenies- Algorithm for Distance Matrices-Additive Trees- Genome rearrangements-Oriented Blocks- unoriented Blocks
5. Molecular Structure Prediction- RNA secondary structure prediction-Protein Folding problems-Protein threading-Computing with DNA-Hamilton Path Problems.

## References


1. Neil James and Pavel A Pevzner, An introduction to Bioinformatics Algorithms, 4e, OUPress, 2014
2. ZhumurGhosh, BibekanandMallick , Bioinformatics : Principles and Applications, OUPress, 2015
3. Concord Bessant, Darren Oakley, Ian Shadforth, Building Bioinformatics Solutions, OUPress, 2014
4. Peter Clote and Rolf Backofen, Computational Molecular Biology-An introduction, 1e, Wiley Series, 2000

# 21-479-0205: Programming Massively Parallel Processors

Core/Elective: **Elective** Semester: **2** Credits: **3**

## Course Description

It used to be the case that parallel computing was confined to giant supercomputers. But nowadays it is literally everywhere - even in the small mobile handsets that most of us carry around. This course introduces parallel computing with a strong emphasis on programming.

## Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Illustrate the parallel programming paradigm
CO2: Identify the benefits of GPU programming model.
CO3:Examine the CUDA programming architecture
CO4: Assess programs written for single-processor systems and convert into efficient parallel programs
CO5: Develop basic parallel programs in CUDA.
CO6: Apply parallel programming to real world applications

## Mapping with Program Outcomes

CO1: PO1
CO2: PO1, PO4
CO3: PO1, PO4
CO4: PO1, PO4
CO5: PO1, PO4
CO6: PO1, PO4

## Course Content:

1. Introduction - parallel computing - more speed or parallelism - languages and models - sequential vs parallel - concurrent, parallel, distributed - parallel hardware architecture - modifications to the von Neumann Model.

2. Evolution of GPU - GPGPU - introduction to data parallelism - CUDA program structure - vector addition kernel - device global memory and data transfer

3. CUDA thread organization - mapping threads to multi-dimensional data - assigning resources to blocks - synchronization and transparent scalability - thread scheduling and latency tolerance

4. Memory access efficiency - CUDA device memory types - performance considerations - global memory bandwidth - instruction mix and thread granularity -floating point considerations

5 Parallel programming patterns - convolution - prefix sum - sparse matrix and vector multiplication - application case studies - strategies for solving problems using parallel programming.

## References:

1. David B. Kirk, Wen-mei W Hwu, Programming Massively Parallel Processors, 2e, Morgan Kaufmann, 2012
2. Peter Pacheco, Introduction to Parallel Programming, 1e, Morgan Kaufmann, 2011
3. Shane Cook, CUDA Programming: A Developer's Guide to Parallel Computing with GPUs, 1e, Morgan Kaufmann, 2012
4. Jason Sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, 1e, AW Professional, 2010

## 21-479-0206: Deep Learning

Core/Elective: **Elective** Semester: **2** Credits: **3**

### Course Description:

Deep learning is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. This course describes deep learning techniques used by practitioners in industry, including deep feedforward networks, regularization, optimization algorithms, convolutional networks, sequence modeling, and practical methodology. This course is useful to students planning careers in either industry or research, and for software engineers who want to begin using deep learning in their products or platforms

### Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1:Understand the need for Deep learning, Feed forward networks, Learning XOR, Gradient based Learning, Hidden units.
CO2: Differentiate between training error and generalization error, Underfitting and Overfitting.
CO3: Identify Regularization strategies, Dataset Augmentation, Adversarial Training.
CO4: Demonstrate the working of Convolution Operation, Sparse interactions, Parameter sharing, Equivariant representations, Pooling.
CO5: Describe Recurrent Neural Networks, Recurrent networks with a single output
CO6: Understand different types of Autoencoders, Undercomplete Autoencoders, Regularized Autoencoders, Dimensionality Reduction.
CO7: Explain Deep generative models like Boltzmann Machines, Restricted Boltzmann Machines.

### Mapping with Program Outcomes

CO1: PO1
CO2: PO1, PO4
CO3: PO1, PO4
CO4:.PO1, PO2,PO4
CO5:.PO1, PO2, PO4
CO6: PO1, PO4
CO7:.PO1, PO4

### Course Content:

1. Deep Networks: Feed forward networks – Learning XOR- Gradient based Learning – Hidden units – Architecture design- Back propagation – Differentiation algorithms

2. Regularization for Deep Learning: Penalties-Constrained optimization-Under constrained problems-Dataset augmentation-Semi Supervised learning- Sparse representation- Adversarial training-Optimization for training deep models: Basic algorithms-Algorithms with adaptive learning rates

3. Convolutional Networks: Convolution-Pooling-Variants of pooling- Efficient convolutional algorithms – Recurrent and Recursive Nets: Recurrent Neural Networks-Deep Recurrent Networks-Recursive Neural Networks- Explicit memory

4. Linear Factor Models: Probabilistic PCA- ICA – Slow feature analysis – Sparse coding – Autoencoders: UndercompleteAutoencoders – Regularized Autoencoders- Learning Manifolds- Applications of Autoencoders – Representation learning

5. Deep generative models: Boltzmann Machines – RBM- Deep Belief Networks-Deep Boltzmann Machines-Convolutional Boltzmann Machines- Directed generative Nets

## References

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, 1e, MIT Press, 2017
2. Nikhil Buduma and Nicholas Locascio, Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms, 1e, Shroff/O'Reilly, 2017
3. Josh Patterson and Adam Gibson, Deep Learning: A Practitioner's Approach, 1e, Shroff/O'Reilly, 2017

# 21-479-0207: Modeling Cyber Physical Systems

**Core/Elective: Core** **Semester: 2** **Credits: 3**

## Course Description:

Cyber-Physical Systems (CPS) is a new frontier for computer systems that is transforming the way people interact with engineered systems. CPS applications include systems such as aircraft, automotive, medical devices, process control, and critical infrastructure. Unlike the traditional computer systems, the interplay between the cyber and the physical systems in CPS brings significant challenges in the modeling, design, analysis and verification of such systems. The complex, interdisciplinary nature of CPS requires a unique approach for the education of CPS. This course introduces Modeling formalism of Cyber-Physical Systems (CPS), Modeling of physical and cyber systems, and software synthesis from these modeling formalisms.

## Course Objectives:

*After the completion of the course, the students will be able to:*

CO1: Categorize the essential modeling formalism of Cyber-Physical Systems (CPS).
CO2: Analyze the functional behavior of CPS based on standard modeling formalism.
CO3: Improve specific software CPS using existing synthesis tools.
CO4: Contrast CPS requirements based on operating system and hardware architecture constraints.
CO5: Analyze and verify the correctness of CPS implementations against system requirements and timing constraints.

## Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1, PO4
CO4: PO1
CO5: PO1, PO4

## Course Content:

1. Introduction to Cyber Physical System: Cyber physical system: Definition Applications, Design Process for Cyber Physical System: Modeling, Design, Analysis: Modelling continuous dynamics, Newtonian Mechanics, Actor models, Properties that actors and the systems: Causal Systems, Memoryless Systems, Linearity and Time Invariance, Stability. Feedback control

2. Modeling Discrete Systems :Discrete Systems ,State, Finite-State Machines: Transitions, The occurrence of reaction, Update functions, Determinacy and Receptiveness, Extended State Machines, Nondeterministic Finite State Machines , Behaviors and Traces

3. Hybrid Systems: Actor Model for State Machines, Continuous Inputs, State Refinements, Classes of Hybrid Systems: Timed Automata, Higher-Order Dynamics, Supervisory control

4. Composition of State Machines: Concurrent Composition: Side-by-Side Synchronous Composition,

Side-by-Side Asynchronous Composition, Shared Variables, Cascade Composition, General Composition, Hierarchical state machines

5. Concurrent Models of Computation : Structure of Models, Synchronous-Reactive Models: Feedback Models, Well-Formed and ill-Formed Models, Constructing a Fixed Point, Dataflow Models of Computation: Dataflow Principles, Synchronous Dataflow ,Dynamic Dataflow, Structured Dataflow, Process Networks, Timed Models of Computation: Time-Triggered Models, Discrete Event Systems, Continuous-Time Systems

**References:**


1. Edward Ashford Lee, Sanjit Arunkumar Seshia, Introduction to Embedded Systems - A Cyber-Physical Systems Approach, 2e, MIT Press, 2017
2. Rajeev Alur , Principles of Cyber-Physical Systems, 1e, MIT Press, 2015
3. Raj Rajkumar, Dionisio de Niz, Mark Klein, Cyber-Physical Systems, 1e, AW Professional, 2017
4. Peter Marwedel, Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things, 3e, Springer, 2017


Online courses: Coursera: https://www.coursera.org/learn/cyber-physical-systems-1

## 21-479-0208: Algorithmic Game Theory

Core/Elective: **Elective** Semester: **2** Credits: **3**

### Course Description

Game theory is a branch of mathematics and economics which models interactions of agents as games. Algorithmic game theory is the intersection of game theory and computer science. This course introduces algorithmic game theory in an application-oriented manner.

### Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Explain the fundamental concepts of non-co-operative and cooperative game theory
CO2: Distinguish between standard game models and solution concepts.
CO3: Illustrate a variety of advanced algorithmic techniques and complexity results for computing game theoretical solution concepts
CO4: Identify rationale of decision making in games.
CO5:Apply solution concepts, algorithms, and complexity results to unseen games that are variants of known examples.
CO6: Compare the state of the art in some areas of algorithmic research, including new developments and open problems

### Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1
CO4: PO1,PO4
CO5: PO1, PO4
CO6: PO1

**Course Content:**

1. Introduction to game theory – strategies, costs, payoffs – solution concepts – finding equilibria – games with sequential moves – games with simultaneous moves – discrete strategies, continuous strategies – mixed strategies – games with incomplete information – expected payoffs – Prisoner's dilemma and repeated games – Nash equilibrium – Computational complexity of Nash equilibrium

2. Games on networks – congestion games – selfish routing – Nash and wardrop equilibria for networks – price of anarchy – pricing network edges – network design with selfish agents – economic aspects of internet routing

3. Epistemic game theory – Modeling knowledge – rationality and belief – common belief in rationality – game strategies and perfect recall – cryptography and game theory – modeling cryptographic algorithms as games – multi-party computations – MPC and games

4. Mechanism design – general principles – social choice – incentives – algorithms mechanism design – distributed aspects – cost-sharing mechanisms – mechanism design without money – house

allocation problem – stable matchings

5. Voting – evaluation of voting systems – strategic manipulation of votes – auctions – types of auctions – winner's curse – bidding strategies – fairness in auctions

**References**


1. Avinash K. Dixit et al., Games of Strategy, 4e, W. W. Norton & Company, 2014
2. Noam Nisan et al., Algorithmic Game Theory, 1e, Cambridge University Press, 2007
3. Steven Tadelis, Game Theory: An Introduction, 1e, Princeton University Press, 2013.
4. Michael Maschler, et al., Game Theory, 1e, Cambridge University Press, 2013.
5. Andres Perea, Epistemic Game Theory: Reasoning and Choice, 1e, Cambridge University Press, 2012

# 21-479-0209: Deep Learning for Computer Vision

Core/Elective: **Elective** Semester: **2** Credits: **3**

## Course Description

Deep Learning has fundamentally changed the landscapes of a number of areas in artificial intelligence, including Computer Vision (CV), Natural Language Processing (NLP) and game playing. Intersection of Deep Learning and CV have emerged with interesting results serving a benchmark for the advances in one of the most important tasks in artificial intelligence. This course introduces the state of the art of deep learning research and its successful applications to major CV tasks, including image classification, object detection, image captioning, video classification, and image generation. The course will help the students to gain practical insights into tools and techniques to implement CV projects effectively and also to outline and analyze various research frontiers of CV in the deep learning era.

## Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Understand the foundations of Computer Vision.
CO2: Evaluate the different feature extraction techniques.
CO3: Develop an object recognition system using handcrafted features and classifiers.
CO4: Understand the concepts of neural networks and logistic regression.
CO5: Develop an image classification system using minimal neural network.
CO6: Understand the fundamentals of Convolutional Neural Network (CNN).
CO7: Develop an image classification system using CNN.
CO8: Develop an image classification system using pre-trained models or auto-encoders.
CO9: Understand the fundamentals of adversarial networks.
CO10: Develop a video processing application using CNN.

## Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1, PO4
CO4: PO1
CO5: PO1, PO4
CO6: PO1
CO7: PO1, PO4
CO8: PO1, PO2, PO4
CO9: PO1
CO10: PO1, PO4

## Course Content

1. Introduction to Computer Vision - Image Filtering - Interest Point Detection - Feature Extraction - Geometric features - SIFT, SURF, HOG, WLD, LBP. Recognition: Geometry-based – Appearance-based. Applications: Object recognition - Face recognition. Implementation: Object Recognition using hand-crafted features and classifiers.

2. Neural Networks – Stochastic Gradient Descent – Backpropagation – Logistic Regression – Softmax. Implementation:Image Classification using minimal neural network.

3. Convolutional Neural Networks: Building Blocks – Hyperparameter Tuning – Learning – Visualizing CNNs – Batch Normalization and Dropout – Deconvnets. Implementation: Simple Image classification using CNN.

4. Transfer Learning – Pre-trained Models – Autoencoders. Implementation: Image Classification using Pre-trained models/Autoencoders.

5. Generative Adversarial Network (GAN) – Attention Mechanism – YOLO. Application: Video Classification-Streaming CNN for action recognition-3D convolution for temporal learning– Segmenting and captioning videos. Implementation: Video Classification / Summarization / Anomaly Detection using CNN.

## References


1. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning, The MIT Press, 2017
2. Charu C Aggarwal. Neural Networks and Deep Learning, Springer, 2018
3. Eugene Charniak. Introduction to Deep Learning, The MIT Press, 2019
4. Linda G. Shapiro, George C. Stockman, "Computer Vision", Prentice Hall, 1st Ed., 2001.
5. Richard Szeliski, "Computer Vision: Algorithms and Applications", Springer, 1st Ed., 2010.
6. David A. Forsyth, Jean Ponce, "Computer Vision: A Modern Approach", 2nd Ed., 2011.
7. Simon J. D. Prince, "Computer Vision: Models, Learning, and Inference", Cambridge University Press, 1st Ed., 2012.
8. Ramesh Jain, Rangachar Kasturi, Brian G. Schunck, "Machine Vision", McGraw-Hill, 1st Ed., 1995.

# 21-479-0210: Image and Video Coding

Core/Elective: **Elective** Semester: **2** Credits: **3**

## Course Description

The aim of this course is to give a rigorous introduction into the fundamental concepts of data compression with strong emphasis on the mathematical techniques and its applications to image and video coding. The main objectives of the course are to understand how digital data can be compressed using either lossless or lossy techniques, to provide a strong mathematical background in the field of coding theory, to expose the students to the standard compression techniques used in various coding standards and to expose the students to the latest image and video coding standards.

## Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Understand the mathematical preliminaries of Lossless compression techniques.
CO2: Analyse the significance of uniquely decodable codes and prefix codes.
CO3: Evaluate the performance of Huffman code and its variants.
CO4: Evaluate the performance of Arithmetic and Integer Arithmetic coding.
CO5: Understand the fundamentals of dictionary and prediction based compression techniques.
CO6: Evaluate the performance of various dictionary-based compression methods.
CO7: Evaluate the performance of various context-based predictive compression methods.
CO8: Understand the mathematical preliminaries of Lossy compression techniques.
CO9: Evaluate the different quantization techniques.
CO10: Understand the basic concepts of Differential and Transform based coding.
CO11: Compare and contrast different content-dependent video coding techniques.
CO12: Compare and contrast different image and video compression standards.
CO13: Develop an image or video compression technique.

## Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1, PO4
CO4: PO1, PO4
CO5: PO1
CO6: PO1, PO4
CO7: PO1, PO4
CO8: PO1
CO9: PO1
CO10: PO1
CO11: PO1
CO12: PO1, PO4
CO13: PO1, PO2, PO4

## Course Content

1. Introduction: Compression Techniques - Modeling and Coding. Mathematical Preliminaries for Lossless compression: Information Theory – Models - Coding: Uniquely decodable codes - Prefix codes - Kraft-McMillan Inequality. Huffman Coding: Minimum Variance Huffman Codes - Length of Huffman Codes - Adaptive Image Compression Standards: JPEG - JPEG 2000 - JPEG XR - JPEG-LS - JPEG XT - JPEG Pleno. Video Compression Standards: MPEG-4 - H.263 - H.264/AVC - H.265/HEVC - AVS China - Dirac.ffman Coding - Golomb codes - Rice codes - Tunstall codes. Arithmetic Coding: Integer Arithmetic Coding.

2. Dictionary Techniques: Static Dictionary - Digram coding - Adaptive Dictionary - LZ77 - LZ78 - LZW. Context-based Compression: Prediction with partial match - Burrows-Wheeler Transform – CALIC - Run-Length Coding – JBIG – JBIG2.

3. Mathematical Preliminaries for Lossy Coding: Distortion Criteria - Rate Distortion Theory. Scalar Quantization: Quantization problem - Uniform Quantizer - Lloyd-Max Quantizer - Adaptive Quantization - Non-uniform Quantization - Entropy-Coded Quantization. Vector Quantization: LBG Algorithm - Tree Structured and Structured Vector Quantizers. Differential Coding: Basic algorithm – DPCM. Transform Coding.

4. Content dependent video coding: Temporal prediction and Transform coding - Two dimensional shape coding - Joint shape and texture coding - Region based and object based video coding - Knowledge based video coding - Semantic video coding - Layered coding system - Scalable video coding.

5. Image Compression Standards: JPEG - JPEG 2000 - JPEG XR - JPEG-LS - JPEG XT - JPEG Pleno. Video Compression Standards: MPEG-4 - H.263 - H.264/AVC - H.265/HEVC - AVS China - Dirac.

## References


1. Khalid Sayood, “Introduction to Data Compression”, Morgan Kaufmann Publishers, 4th Ed., 2012.
2. David Salomon, “Data Compression – The Complete Reference”, Springer, 4th Ed., 2006.
3. Alistair Moffat, Andrew Turpin, “Compression and Coding Algorithms”, Kluwer Academic Publishers, 1st Ed., 2002.
4. Vasudev Bhaskaran, Konstantinos Konstantinides, “Image and Video Compression Standards”, Kluwer Academic Publishers, 2nd Ed., 2003.
5. Mark Nelson, Jean-Loup Gailly, “The Data Compression Book”, John Wiley & Sons, 2nd Ed., 1995.
6. John Miano, “Compressed Image File Formats”, Addison Wesley Professional, 1st Ed., 1999.
7. Peter Wayner, “Compression Algorithms for Real Programmers”, Morgan Kaufmann, 1st Ed., 1999.
8. Yao Wang, Jorn Ostermann, Ya-Qin Zhang, "Video Processing and Communications", Pearson, 1st Ed., 2001.
9. Alan C. Bovik, "The Essential Guide to Video Processing", Academic PRess, 2nd Ed., 2009
10. A. Murat Tekalp, "Digital Video Processing", Prentice Hall, 2nd Ed., 2015.

## 21-479-0211: Reinforcement Learning

Core/Elective: **Elective** Semester: **2** Credits: **3**

### Course Description

The course aims to introduce the concepts reinforcement learning and to impart an understanding of how reinforcement learning -- along with supervised and unsupervised learning -- form a building block of modern artificial intelligence. The course will provide a solid introduction to the field of reinforcement learning and students will learn about the core challenges and approaches, including generalization and exploration.

### Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Define the key features of reinforcement learning that distinguishes it from AI and non-interactive machine learning
CO2: Demonstrate the ability to formulate a given problem as a reinforcement problem with all ingredients.
CO3: Implement in code common RL algorithms
CO4:. Describe the exploration vs exploitation challenge
CO5: Compare and contrast at least two approaches for addressing the above challenge.

### Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1,PO4
CO4: PO1,PO4
CO5: PO1, PO4

### Course Content:

1. The Reinforcement Learning problem: evaluative feedback, non-associative learning, Rewards and returns, Markov Decision Processes, Value functions, optimality and approximation

2. Bandit Problems: Explore-exploit dilemma, Binary Bandits, Learning automata, exploration schemes Dynamic programming: value iteration, policy iteration, asynchronous DP, generalized policy iteration

3. Monte-Carlo methods: policy evaluation, roll outs, on policy and off policy learning, importance sampling Temporal Difference learning: TD prediction, Optimality of TD(0), SARSA, Q-learning, R-learning, Games and after states

4. Eligibility traces: n-step TD prediction, TD(lambda), forward and backward views, Q(lambda), SARSA(lambda), replacing traces and accumulating traces.

5. Function Approximation: Value prediction, gradient descent methods, linear function approximation, Control algorithms, Fitted Iterative Methods Policy Gradient methods: nonassociative learning -

REINFORCE algorithm, exact gradient methods, estimating gradients, approximate policy gradient algorithms, actor-critic methods Hierarchical RL: MAXQ framework, Options framework, HAM framework, Option discovery algorithms

### References


1. R. S. Sutton and A. G. Barto; Reinforcement Learning - An Introduction. 2e, MIT Press (2018) eBook: http://incompleteideas.net/book/the-book-2nd.html
2. Marco Wiering and Martijn van Otterlo (Editors); Reinforcement Learning: State-of-the Art, Springer (2012)
3. Csaba Szepesvari; Algorithms for Reinforcement Learning, Morgan and Claypool Publishers (2010)
4. David Silver: https://www.davidsilver.uk/teaching/

## 21-479-0212: Natural Language Processing with Deep Learning

Core/Elective: **Elective** Semester: **2** Credits: **3**

### Course Description

Deep learning has fundamentally changed the landscapes of a number of areas in artificial intelligence, including speech and natural language, vision, robotics, and game playing. Intersection of deep learning and NLP have emerged with interesting results serving a benchmark for the advances in one of the most important tasks in artificial intelligence. This course introduces the state of the art of deep learning research and its successful applications to major NLP tasks, including speech recognition and understanding, dialogue systems, lexical analysis, parsing, knowledge graphs, machine translation, question answering, sentiment analysis, social computing, and natural language generation

### Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Outline the foundations of speech and written language processing
CO2: Analyse various NLP tasks through the lens of deep learning
CO3: Implement an NLP project effectively
CO4: Analyze various research frontiers of NLP in the deep learning era

### Mapping with Program Outcomes

CO1: PO1
CO2: PO1,PO4
CO3: PO1,PO4
CO4: PO1,PO4

### Course Content:

1. Word embeddings: SVD based methods – Iteration based methods – Word2vec – optimization Language models: Unigram, Bigram – CBOW – Skip-Gram Model - GloVe – Evaluation of Word Vectors: Intrinsic and Extrinsic tasks – Word Window Classification Implementations: Gensim Word2vec, Doc2Vec, FastText -GloVe using gradient descent

2. Neural Networks Architectures: feed-forward computation – representational power – back propagation and computation graphs Implementations: Neural network and Deep neural network in ML framework

3. Linguistic structure: Dependency parsing – N-gram language models – Recurrent Neural Networks and language models – Sequence modeling: Recurrent and Recursive Neural Nets Implementations: Character level RNN, Chatbot using recurrent sequence to sequence models

4. Machine translation – Seq2Seq learning - Attention models– ConvNets for NLP

5. Natural language generation – Coreference resolution – Constituency parsing

## References


1. Dan Jurafsky and James H. Martin. Speech and Language Processing, Prentice Hall, 3rd ed, 2019
2. Jacob Eisenstein. Natural Language Processing, The MIT Press, 2019
3. Yoav Goldberg, Graeme Hirst.. Neural Network Models for Natural Language Processing, Morgan and Claypool Life Sciences, 2017
4. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning, The MIT Press, 2017
5. Charu C Aggarwal. Neural Networks and Deep Learning, Springer, 2018 6. Eugene Charniak. Introduction to Deep Learning, The MIT Press, 2019
6. Natural Language Processing in Deep Learning by Stanford (Coursera)
7. Deep Learning Specialization by deeplearning.ai
8. Applied AI with Deep Learning by IBM (Coursera)
9. Introduction to pyTorch and Machine Learning (Udemy)
10. Practical Deep Learning with pyTorch (Udemy)

## 21-479-0301: Elective VI

Core/Elective: **Elective** Semester: **3** Credits: **2**

### Course Description

A credit-based MOOC course of minimum 12 weeks duration or three non-credit based MOOC courses of 4-weeks duration from SWAYAM/NPTEL/any other platforms approved by the Department.

## 21-479-0302: Dissertation & Viva Voce

Core/Elective: **Core** Semester: **3** Credits: **16**

### Course Description

The dissertation work spans two semesters. Through the dissertation work, the student has to exhibit the knowledge in terms of engineering or technological innovation or research ability to solve the contemporary problem. On completion of the first part of the work, the student shall submit an interim dissertation report. The qualitative and quantitative results of the work will be evaluated through a viva-voce exam.

### Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Demonstrates in depth knowledge and thoughtful application through the detailed analysis of the problem chosen for the study
CO2: Assess the gap by acquiring knowledge about the previous works, and its interpretation and application
CO3: Demonstrates the design of the proposed methodology and its merits.
CO4: Organize the interim dissertation content with proper structure and sequencing
CO5: Demonstrate the academic discussion skills to emphasize, argue with clarity of purpose using evidence for the claims.

### Mapping with Program Outcomes

CO1: PO1, PO4
CO2: PO1, PO4
CO3: PO1, PO4
CO4:.PO2
CO5:.PO2

## 21-479-0401: Dissertation & Viva Voce

Core/Elective: **Core** Semester: **4** Credits: **18**

### Course Description

The dissertation work spans two semesters. Through the dissertation work, the student has to exhibit the knowledge in terms of engineering or technological innovation or research ability to solve the contemporary problem. On completion of the work, the student shall submit a final dissertation report. The qualitative and quantitative results of the work will be evaluated through a viva-voce exam.

### Course Outcomes (CO)

*After the completion of the course, the students will be able to:*

CO1: Demonstrates in depth knowledge and thoughtful application through the detailed analysis of the problem chosen for the study
CO2: Assess the gap by acquiring knowledge about the previous works, and its interpretation and application
CO3: Demonstrates the design of the proposed methodology and its merits.
CO4: Organize the interim dissertation content with proper structure and sequencing
CO5: Demonstrate the academic discussion skills to emphasize, argue with clarity of purpose using evidence for the claims.
CO6: Show ability to evaluate and reflect on critical questions.

### Mapping with Program Outcomes

CO1: PO1, PO4
CO2: PO1, PO4
CO3: PO1, PO4
CO4:.PO2
CO5:.PO2
CO6: PO1, PO4

**Learning Outcomes and Assessment**

Each course's learning outcomes will be assessed based on one or many methods, including the internal written tests, quizzes, presentations, seminars, assignments in the form of lab exercises, and group projects. The above assessment methods will be attentively created to support the intended learning outcomes that have been set out for a particular course. The program outcome attainment is measured using the CO/PO mappings.