Indian Institute of Space Science and Technology

Thiruvananthapuram

M.Tech. Machine Learning and Computing

Curriculum & Syllabus (Effective from 2026 Admission)

Department of Mathematics

Program Educational Objectives (PEOs)

The M.Tech. in Machine Learning and Computing programme is designed to prepare graduates to:

1. Develop strong theoretical and practical foundations in machine learning, computing, and intelligent systems for solving complex real-world problems.
2. Apply advanced machine learning, data-driven, and computational techniques in interdisciplinary domains, industry, research, and academia.
3. Engage in research, innovation, and lifelong learning in emerging areas of artificial intelligence, machine learning, data science, and computing technologies.
4. Design scalable, efficient, and ethical intelligent systems using modern computational tools, distributed computing platforms, and AI methodologies.
5. Demonstrate professional competence, communication skills, teamwork, and leadership in multidisciplinary and research-oriented environments.

Program Outcomes (POs)

On successful completion of the programme, graduates will be able to:

1. Apply advanced knowledge of machine learning and computing to analyze and solve complex problems.
2. Design, implement, and evaluate machine learning models and intelligent systems for real-world applications.
3. Analyze large-scale and complex datasets using modern computational, statistical, and AI techniques.
4. Use contemporary software tools, programming frameworks, cloud platforms, GPU computing, and distributed systems for machine learning applications.
5. Conduct independent research, literature review, experimentation, and technical documentation in machine learning and related domains.
6. Develop efficient and scalable AI solutions while considering ethical, societal, privacy, and security aspects.
7. Work effectively as an individual and as a member or leader in multidisciplinary teams.
8. Communicate technical concepts, research findings, and project outcomes effectively through reports, presentations, and publications.

9. Adapt to emerging technologies and engage in lifelong learning in machine learning, artificial intelligence, and computing.

10. Contribute to innovation, entrepreneurship, and advanced technological development in academia, industry, and society.

SEMESTER I

CODE	TITLE	L	T	P	C
MA626	Numerical Linear Algebra	3	0	0	3
MA627	Optimization Theory and Applications	3	0	0	3
MA628	Probability and Statistics for Machine Learning	3	0	0	3
MA629	Machine Learning	3	0	0	3
	Elective I	3	0	0	3
	Elective II	3	0	0	3
MA645	Intelligent Systems Lab	0	0	3	1
	Total	18	0	3	19

SEMESTER II

CODE	TITLE	L	T	P	C
MA630	Advanced Machine Learning	3	0	0	3
MA631	Deep Learning	3	0	0	3
	Elective III	3	0	0	3
	Elective IV	3	0	0	3
	Elective V	3	0	0	3
MA646	Distributed Machine Learning Lab	0	0	3	2
MA647	MLOps Lab	0	0	3	2
	Total	15	0	6	19

SEMESTER III

CODE	TITLE	L	T	P	C
MA852	Project Work – Phase I	0	0	0	15
	Total	0	0	0	15

SEMESTER IV

CODE	TITLE	L	T	P	C
MA853	Project Work – Phase II	0	0	0	17

LIST OF ELECTIVES

CODE	TITLE
MA871	Advanced Kernel Methods
MA872	Advanced Optimization
MA876	Bayesian Machine Learning
MA877	Computer Vision
MA869	Discrete Mathematics and Graph Theory
MA880	Explainable AI
MA881	Graphical Models
MA882	Graph Neural Networks
MA883	Large Language Models
MA888	Machine Learning for Remote Sensing
MA884	MLOps
MA885	Riemannian Geometry for Machine Learning
MA867	Reinforcement Learning
MA886	Time Series
MA875	Topological Data Analysis
MA887	Transformers and Vision-Language Models

SEMESTER–WISE CREDITS

Semester	I	II	III	IV	Total
Credits	19	19	15	17	70

SEMESTER I

MA626 Numerical Linear Algebra — 3 credits

Linear Algebra Basics: Vector spaces and R^n; linear subspaces; bases and dimension; vectors; norms and dot products; common vector norms; orthogonality; orthogonal subspaces; linear transformations and matrices; change of basis; orthogonal matrices and projectors; eigenvalues and eigendecomposition; diagonalizable matrices; singular values.

Foundations of Computational Linear Algebra: Standard problems in computational linear algebra; floating-point arithmetic and roundoff errors; stability of numerical algorithms; perturbation theory and condition numbers; vector and matrix norms (induced and Frobenius); compatibility and submultiplicativity; perturbation of the right-hand side and coefficient matrix; properties of condition numbers; computational efficiency, memory usage, parallelization and vectorization; BLAS levels; GPU and out-of-core computations; relevance of numerical stability in large-scale machine learning pipelines.

Direct Methods for Linear Systems and Least Squares Problems: Gaussian elimination and pivoting; error analysis and numerical stability; estimation of condition numbers and practical error bounds; Basic Linear Algebra Subprograms (BLAS); optimization of matrix multiplication using Level-3 BLAS; symmetric positive definite and indefinite systems; band matrices; linear least squares problems; QR decomposition (Householder transformations and Givens rotations); orthogonal matrices; rank-deficient least squares using QR and singular value decomposition; LSQR and other iterative least-squares methods; connections to stochastic gradient descent; performance comparison of algorithms and practical recommendations.

Eigenvalue Problems and Iterative Methods: Eigenvalue and eigenvector computation (power method, inverse iteration, Rayleigh quotient iteration, QR iteration with implicit shifts); Krylov subspace methods (Lanczos and Arnoldi); algorithms for singular value decomposition (bidiagonal QR, Jacobi methods); randomized and approximate methods for large-scale problems (randomized SVD, sketching); iterative methods for linear systems (Jacobi, Gauss–Seidel, successive over-relaxation); conjugate gradient method; preconditioning techniques (Jacobi, incomplete LU, spectral preconditioning); convergence diagnostics; applications to spectral methods and large-scale machine learning.

Course Outcomes:

CO1: Explain the advantages and limitations of iterative solution methods compared to direct methods, including when each is preferable.

CO2: Describe subspace projection methods (in particular Krylov subspaces) and their importance for very large-scale linear systems.

CO3: Given a matrix and problem context, recommend and justify a suitable iterative solution method.

CO4: Analyze the effect of matrix condition number and spectrum on the convergence behaviour of iterative methods.

CO5: Explain the role of preconditioners, describe basic preconditioning techniques, and evaluate trade-offs between computational cost and iteration reduction.

CO6: Distinguish spectral characteristics of different matrix classes and relate them to solver performance.

CO7: Select and justify appropriate methods for computing eigenvalues for different types of matrices.

References:

1. Biswa Nath Datta – *Numerical Linear Algebra and Applications*, SIAM, 2010.
2. Folkmar Bornemann – *Numerical Linear Algebra: A Concise Introduction with MATLAB*, Springer International AG, 2018.
3. Lloyd Trefethen and David Bau – *Numerical Linear Algebra*, SIAM, 1997.
4. William Ford – *Numerical Linear Algebra with Applications using MATLAB*, Academic Press, 2015.

MA627 Optimization Theory and Applications — 3 credits

Optimization: need for unconstrained methods in solving constrained problems; necessary conditions of unconstrained optimization; structure methods; quadratic models; methods of line search; steepest descent method.

Quasi-Newton methods: DFP and BFGS methods; conjugate-direction methods; methods for sums of squares and nonlinear equations.

Linear programming: simplex method; duality in linear programming; transportation problem.

Nonlinear programming: Lagrange multipliers; Karush–Kuhn–Tucker (KKT) conditions; convex programming.

Algorithm for least squares and iterative least squares method.

Course Outcomes:

CO1: Model optimization problems mathematically.

CO2: Understand foundational theory of optimization.

CO3: Become familiar with algorithms for solving optimization problems.

CO4: Develop programming implementations for real-time optimization problems.

References:

1. Chong, E. K. P., and Zak, S. H., *An Introduction to Optimization*, 2nd Edition, Wiley India, 2001.
2. Luenberger, D. G., and Ye, Y., *Linear and Nonlinear Programming*, 3rd Edition, Springer, 2008.
3. Kambo, N. S., *Mathematical Programming Techniques*, East-West Press, 1997.

MA628 Probability and Statistics for Machine Learning — 3 credits

Probability Basics: axioms, conditional probability, law of total probability, Bayes' theorem; random variables and distributions; moments; moment inequalities (Markov, Chebyshev, Hoeffding); common distributions (Bernoulli, Binomial, Poisson, Uniform, Gaussian); joint, marginal, conditional distributions; mean, variance, covariance; independence; multivariate Gaussian; law of large numbers; central limit theorem.

Sampling Theory: important sample statistics and their distributions; parameter estimation using MLE and MAP; bias–variance trade-off; regularization as a probabilistic prior.

Bayesian Inference: Bayes' theorem in learning; conjugate priors; posterior prediction; Bayesian regression and classification.

Core Probabilistic Models: Naïve Bayes; Gaussian mixture models (GMMs); latent variable models; Expectation–Maximization (EM).

Course Outcomes:

CO1: Apply axioms of probability, conditional probability, Bayes theorem, and probability distributions to machine learning problems.

CO2: Analyze multivariate random variables, joint, marginal, and conditional distributions, and multivariate Gaussian models.

CO3: Perform sampling-based statistical inference and parameter estimation using MLE and MAP, and analyze bias–variance trade-offs.

CO4: Implement Bayesian inference techniques for regression and classification with uncertainty quantification.

CO5: Design and analyze core probabilistic machine learning models including Naïve Bayes, GMM, and EM-based latent variable learning.

References:

1. Larry Wasserman – *All of Statistics; A Concise Course in Statistical Inference*, Springer, 2010
2. Kevin P Murphy – *Machine Learning: A Probabilistic Perspective*, MIT Press, 2012
3. Christopher M Bishop – *Pattern Recognition and Machine Learning*, Springer, 2009
4. Sheldon Ross – *A First Course in Probability*, 10th ed., Pearson, 2023

Introduction to Machine Learning: capacity; overfitting and underfitting; regularization techniques; hyperparameter tuning; bias–variance tradeoff; model evaluation and selection; performance metrics for classification and regression; feature engineering and data preprocessing; handling class imbalance; ML frameworks (Python: scikit-learn, PyTorch, TensorFlow).

Supervised Learning: classification and regression concepts; linear regression; logistic regression; Gaussian discriminant analysis (LDA & QDA); Naïve Bayes; ridge regression; linear models for regression and classification; decision trees; k-nearest neighbors; basics of support vector machines (SVMs); multi-class classification techniques; basics of ensemble methods.

Unsupervised Learning: k-means; hierarchical clustering; self-organizing maps; cluster evaluation and validation.

Dimensionality Reduction: feature selection (supervised methods) and principal component analysis (PCA); brief coverage of t-SNE/UMAP for visualization.

Learning Techniques: introduction to semi-supervised learning, reinforcement learning, transfer learning, and active learning.

Course Outcomes:

CO1: Explain fundamental concepts and methodologies of machine learning.

CO2: Apply supervised and unsupervised learning techniques to solve data-driven problems.

CO3: Preprocess and analyze datasets using appropriate feature-selection and dimensionality-reduction methods.

CO4: Implement, evaluate, and compare machine learning models using modern software frameworks and tools.

CO5: Select and justify suitable machine learning approaches for different real-world applications and problem settings.

References:

1. Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer, 2006.
2. Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning*, Springer, 2009 (2nd ed.).
3. Murphy, K. P., *Machine Learning: A Probabilistic Perspective*, MIT Press, 2012.
4. Shalev-Shwartz, S., and Ben-David, S., *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, 2014.
5. Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, MIT Press, 2016.

ELECTIVE I	3 credits
ELECTIVE II	3 credits

MA645 Intelligent Systems Lab 1 credit

Introduction to cloud platforms and cloud notebooks; running AI tasks on cloud with dataset handling; basics of GPU computing including CPU vs GPU concepts; execution of programs using GPU support; performance comparison between CPU and GPU; use of pre-built ML models on cloud/GPU; mini project involving an AI application with demonstration and presentation.

Course Outcomes:

CO1: Use modern computational environments for developing and testing intelligent applications.

CO2: Apply parallel and accelerated computing techniques for efficient execution of AI tasks.

CO3: Evaluate computational resources and execution strategies for machine learning applications.

CO4: Integrate existing AI tools and pre-trained models for practical problem solving.

CO5: Design, implement, and demonstrate a simple intelligent system application.

SEMESTER II

MA630 Advanced Machine Learning 3 credits

Learning Theory: PAC model; Rademacher complexity; growth function and VC-dimension; generalization bounds.

Clustering and unsupervised learning: model-based clustering; spectral clustering; independent component analysis (ICA).

Advanced kernel methods: reproducing kernel Hilbert space (RKHS) concepts; support vector machines for classification and regression; kernel ridge regression; kernel PCA (KPCA); kernel online learning; multiple kernels; graph kernels.

Ensemble learning: bootstrap; bagging; boosting (AdaBoost); random forests; gradient boosting (XGBoost).

Graphical models and geometric machine learning: probabilistic graphical models (Bayesian networks, Markov networks, hidden Markov models); d-separation; conditional independence; inference techniques including variable elimination, belief propagation, max-product algorithm, and junction trees; geometric and topological machine learning concepts; manifold and Riemannian learning; PageRank as an application.

MLOps, explainable AI (XAI), and federated learning: deployment, reproducibility, model interpretability, and privacy-aware distributed training.

Course Outcomes:

CO1: Analyse the theoretical foundations underlying learning, generalization, and optimization behavior of machine learning models.

CO2: Develop advanced machine learning models and solutions for complex, high-dimensional, structured, and unlabelled data.

CO3: Evaluate and compare different learning strategies and algorithms to improve prediction accuracy, robustness, and scalability.

CO4: Model uncertainty, dependencies, and relationships in data using probabilistic and graph-based approaches.

CO5: Design and implement reliable, interpretable, and deployable machine learning systems for real-world applications.

References:

1. Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer, 2006.
2. Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning*, Springer, 2009.
3. Cristianini, N. and Shawe-Taylor, J., *An Introduction to Support Vector Machines and Other Kernel-based Methods*, Cambridge University Press, 2000.
4. Schölkopf, B. and Smola, A. J., *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, 2002.
5. Murphy, K. P., *Machine Learning: A Probabilistic Perspective*, MIT Press, 2012.

MA631 Deep Learning — 3 credits

Introductory concepts: perceptron; multilayer perceptron; deep learning as composition of functions; loss functions; activation functions; vanishing and exploding gradients; backpropagation; deep learning frameworks (TensorFlow, PyTorch, Keras).

Optimization and regularization techniques for deep networks: stochastic gradient descent (SGD) and its variants (Momentum, Nesterov accelerated gradient, AdaGrad, RMSProp, Adam); dropout; L2 regularization (weight decay); max-norm constraints; learning-rate schedules; weight initialization techniques; batch normalization.

Convolutional Neural Networks (CNNs): architectures; convolution, pooling, and padding operations; pretrained models and transfer learning; backpropagation in CNNs.

Sequence modeling: recurrent neural networks (RNNs); vanishing and exploding gradients; gradient clipping; gated architectures (LSTM, GRU); backpropagation through time (BPTT) for RNNs and LSTMs; attention mechanisms; encoder–decoder architectures; Transformer architectures.

Unsupervised learning and representation learning: autoencoders (basic, denoising, and sparse variants); representation learning.

Generative models: generative adversarial networks (GANs); variational autoencoders (VAEs); overview of other deep generative models including normalizing flows and autoregressive models; training challenges and evaluation metrics.

Graph neural networks (GNNs) and overview of emerging deep learning architectures.

Course Outcomes:

CO1: Explain the fundamental principles and learning mechanisms underlying deep learning methods.

CO2: Develop deep learning models for representation learning and prediction tasks.

CO3: Analyze training challenges in deep neural networks and select appropriate strategies to improve optimization and generalization.

CO4: Evaluate deep learning approaches across diverse data modalities and application domains.

CO5: Design and implement intelligent systems using modern deep learning frameworks and computational techniques.

References:

1. Chollet, F., *Deep Learning with Python*, Manning Publications, 2017.
2. Nielsen, M., *Neural Networks and Deep Learning*, Determination Press, 2015.
3. Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, MIT Press, 2016.
4. Zhang, A., Lipton, Z. C., Li, M., and Smola, A. J., *Dive into Deep Learning*, Cambridge University Press, 2023.

ELECTIVE III	3 credits

ELECTIVE IV	3 credits

ELECTIVE V	3 credits

MA646 Distributed Machine Learning Lab — 2 credits

Introduction to distributed and parallel computing concepts for machine learning applications; execution of machine learning tasks in distributed computing environments; distributed dataset storage and processing; parallel execution of machine learning algorithms using multi-core CPU and GPU platforms; implementation of distributed machine learning workflows; performance analysis and scalability evaluation of distributed ML models; experimentation with large-scale datasets and cloud-based AI platforms; mini project involving the development, demonstration, and presentation of a distributed machine learning application.

Course Outcomes:

CO1: Use distributed and parallel computing environments for machine learning applications.

CO2: Execute machine learning workflows using multi-core and GPU-based computing platforms.

CO3: Analyze scalability and computational performance of distributed machine learning systems.

CO4: Implement distributed machine learning solutions using cloud and large-scale data processing platforms.

CO5: Design, develop, and demonstrate a distributed machine learning application for real-world datasets.

MA647 MLOps Lab — 2 credits

Environment setup and structured ML project organization; data handling and versioning; building modular ML pipelines (data preprocessing, training, evaluation); experiment tracking (logging parameters, metrics, and comparison of runs); model versioning and registry concepts; reproducibility practices (configuration management, seeds, environments); workflow automation and pipeline orchestration (basic); model deployment using APIs (Flask/FastAPI) and testing endpoints; containerization using Docker; introduction to CI/CD pipelines for ML (implementation/demo); model monitoring (performance tracking, data drift, basic alerting concepts); mini project involving complete ML lifecycle from data ingestion to deployment and monitoring.

Course Outcomes:

CO1: Develop structured and reproducible machine learning workflows for end-to-end AI applications.

CO2: Implement automated pipelines for data processing, model training, evaluation, and deployment.

CO3: Apply tools and techniques for experiment tracking, model versioning, and workflow management.

CO4: Deploy and manage machine learning models using modern software and containerization technologies.

CO5: Design and demonstrate complete machine learning lifecycle solutions including deployment and monitoring.

SEMESTER III

MA852 PROJECT WORK — PHASE I	15 credits

SEMESTER IV

MA853 PROJECT WORK — PHASE II	17 credits

MA871 Advanced Kernel Methods — 3 credits

Theory of reproducing kernel Hilbert spaces (RKHS); kernel functions and kernel feature maps; representer theorem and regularization in kernel spaces. Kernel-based learning algorithms including support vector machines for classification and regression, kernel ridge regression, kernel principal component analysis, and kernel online learning.

Advanced kernel methods including multiple kernel learning and Bayesian kernel methods. Design of kernels for structured data including graph kernels, string kernels, and kernels for text and sequence data.

Course Outcomes:

CO1: Explain the theoretical foundations of kernel methods including reproducing kernel Hilbert spaces, kernel functions, and the representer theorem.

CO2: Analyze kernel-based learning algorithms such as support vector machines, kernel ridge regression, and kernel principal component analysis.

CO3: Apply kernel methods for classification, regression, and dimensionality reduction tasks.

CO4: Evaluate advanced kernel techniques including multiple kernel learning and Bayesian kernel methods.

CO5: Design and analyze kernel functions for structured data such as graphs, text, and sequences.

References:

1. Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer (2006).
2. Cristianini, N. and Shawe-Taylor, J., *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press (2000).
3. Schölkopf, B. and Smola, A. J., *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press (2001).
4. Shawe-Taylor, J. and Cristianini, N., *Kernel Methods for Pattern Analysis*, Cambridge University Press (2004).

MA872 Advanced Optimization — 3 credits

Unconstrained Optimization: line search methods including Wolfe condition, Goldstein condition, sufficient decrease, and backtracking; Newton's method and quasi-Newton methods; trust region methods including the Cauchy point, algorithms based on the Cauchy point, improvements on the Cauchy point, the Dog-leg method, and two-dimensional subspace reduction; nonlinear conjugate gradient methods including the Fletcher–Reeves method.

Constrained Optimization: penalty methods including quadratic penalty, convergence, nonsmooth penalty functions, and L1 penalty; augmented Lagrangian method; quadratic programming methods including Schur complement, null space method, and active set method for convex QP; sequential quadratic programming; convex programming.

Course Outcomes:

CO1: Explain the theoretical foundations and convergence properties of advanced unconstrained optimization methods.

CO2: Analyze and apply line search, Newton, quasi-Newton, trust region, and nonlinear conjugate gradient methods for solving unconstrained optimization problems.

CO3: Analyze and apply penalty, augmented Lagrangian, quadratic programming, and sequential quadratic programming methods for constrained optimization.

CO4: Select and compare appropriate optimization algorithms based on problem structure, computational requirements, and convergence properties.

CO5: Implement advanced optimization algorithms and evaluate their performance on practical optimization problems.

References:

1. Boyd, S. and Vandenberghe, L., *Convex Optimization*, Cambridge University Press (2004).
2. Nocedal, J. and Wright, S., *Numerical Optimization*, Springer (2006).

MA876 Bayesian Machine Learning — 3 credits

Foundations of Bayesian inference: Bayes theorem; prior, likelihood, posterior; conjugate priors; posterior predictive distribution; Bayesian decision theory.

Bayesian parameter estimation and regression: Bayesian linear regression; prior distributions; posterior inference; predictive uncertainty; regularization from a Bayesian perspective.

Approximate inference methods: Laplace approximation; variational inference (basic ideas); Monte Carlo methods; Markov Chain Monte Carlo (MCMC) including Metropolis–Hastings and Gibbs sampling (conceptual understanding).

Bayesian models for machine learning: Gaussian processes for regression (introductory treatment); Bayesian neural networks (basic concepts and motivation); latent variable models (basic concepts).

Model comparison and selection: marginal likelihood (evidence); Bayes factors; basic information criteria; hyperparameter tuning.

Course Outcomes:

CO1: Explain the fundamental principles of Bayesian inference, including prior distributions, likelihood, posterior distributions, posterior predictive distributions, and Bayesian decision theory.

CO2: Apply Bayesian methods for parameter estimation, regression, and predictive modeling, while quantifying and interpreting predictive uncertainty.

CO3: Analyze and apply approximate inference methods, including Laplace approximation, variational inference, and Monte Carlo methods for Bayesian machine learning.

CO4: Develop and evaluate Bayesian machine learning models, including Gaussian processes, Bayesian neural networks, and latent variable models.

CO5: Compare and select Bayesian models using marginal likelihood, Bayes factors, information criteria, and hyperparameter tuning.

References:

1. Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
2. Kevin P. Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press, 2012.
3. Kevin P. Murphy, *Probabilistic Machine Learning: An Introduction*, MIT Press, 2022.
4. David J.C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003.

MA877 Computer Vision — 3 credits

Introduction to Computer Vision: Image formation, image representation, color spaces, image sampling and quantization, spatial and frequency domains.

Image Processing Fundamentals: Image enhancement, histogram processing, filtering in the spatial and frequency domains, morphological operations, and geometric transformations.

Feature Detection and Representation: Edge detection, corner detection, keypoint detection and description (SIFT, SURF, ORB), texture analysis, sparse representations, and sparse coding.

Feature Extraction and Machine Learning: Feature engineering, dimensionality reduction, feature selection, classification methods (k-NN, SVM, decision trees), and clustering.

Image Segmentation: Thresholding, region-based methods, graph-based segmentation, watershed, semantic segmentation, and instance segmentation.

Object Detection and Recognition: Template matching, sliding window methods, region proposal techniques, and modern object detectors.

Deep Learning for Computer Vision: Fundamentals of neural networks, Convolutional Neural Networks (CNNs), transfer learning, data augmentation, and popular architectures (AlexNet, VGG, ResNet, EfficientNet).

Vision Transformers: Self-attention mechanism, Vision Transformer (ViT), Swin Transformer, and hybrid CNN-Transformer architectures.

Visual Language Models: Multimodal learning, image-text representation, and contrastive learning (e.g., CLIP).

Advanced Applications and Emerging Trends.

Course Outcomes:

CO1: Explain the fundamental principles of image formation, representation, sampling, quantization, and image processing in spatial and frequency domains.

CO2: Apply classical computer vision techniques for feature detection, feature representation, image segmentation, classification, and object recognition.

CO3: Design and implement deep learning models for computer vision tasks, including image recognition, segmentation, and object detection.

CO4: Analyze and compare CNNs, Vision Transformers, and visual language models for computer vision and multimodal learning tasks.

CO5: Develop and evaluate computer vision solutions for real-world applications using classical and modern AI techniques.

References:

1. Prince, S. J. D., *Computer Vision: Models, Learning, and Inference*, Cambridge University Press, 2012.
2. Szeliski, R., *Computer Vision: Algorithms and Applications*, Springer, 2010.
3. Forsyth, D. A. and Ponce, J., *Computer Vision: A Modern Approach*, Prentice Hall, 2002.
4. Palmer, S. E., *Vision Science: Photons to Phenomenology*, MIT Press, 1999.

5. Duda, R. O., Hart, P. E., and Stork, D. G., *Pattern Classification*, 2nd ed., John Wiley & Sons, 2000.

MA869 Discrete Mathematics and Graph Theory — 3 credits

Basic counting principles: pigeonhole principle, inclusion–exclusion principle, recurrence relations, and generating functions; fundamentals of logic, set theory, formal languages, and finite state machines.

Graph theory: undirected and directed graphs, modelling with graphs, trails and cycles, connectivity and trees; graph algorithms including breadth-first search (BFS), depth-first search (DFS), shortest path algorithms, minimum spanning trees, matching, job assignment problem, and network flow problems.

Course Outcomes:

CO1: Apply fundamental principles of logic, set theory, and discrete structures to solve problems in computer science.

CO2: Apply combinatorial techniques, including counting principles, recurrence relations, and generating functions, to analyze discrete problems.

CO3: Model and analyze problems using graph-theoretic concepts, including trees, connectivity, trails, and cycles.

CO4: Apply fundamental graph algorithms, including BFS, DFS, shortest path algorithms, and minimum spanning trees, to solve computational problems.

CO5: Analyze combinatorial optimization problems using matching, assignment, and network flow models.

References:

1. C. L. Liu, *Elements of Discrete Mathematics*, 2nd Ed., Tata McGraw-Hill, 2000.
2. R. P. Grimaldi and B. V. Ramana, *Discrete and Combinatorial Mathematics*, Pearson, 2008.
3. R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*, 2nd Ed., Addison-Wesley, 1994.
4. K. H. Rosen, *Discrete Mathematics and Its Applications*, 6th Ed., Tata McGraw-Hill, 2007.

MA880 Explainable AI — 3 credits

Introduction to Explainable Artificial Intelligence: Motivation and need for XAI; interpretability vs. explainability; transparency, trust, and accountability; black-box nature of machine learning and deep learning models; examples of black-box decision making using neural networks.

Foundations of Explainability: Types and levels of explanations; global vs. local explanations; model-specific vs. model-agnostic methods; intrinsic vs. post-hoc explainability; feature importance, attribution, and surrogate models.

Post-hoc Explainability: Major families of post-hoc methods, including feature attribution, surrogate-based, example-based, counterfactual, and rule-based explanations; LIME, SHAP, permutation importance, Integrated Gradients, and counterfactual explanations.

Explainability for Deep Learning: Saliency and activation-based methods; feature visualization; CAM and Grad-CAM; layer-wise relevance propagation; explaining CNNs, transformers, and other deep neural networks.

Inherently Interpretable (Ante-hoc) AI: Interpretable linear models, decision trees, rule-based models, generalized additive models, sparse and prototype-based models, and interpretable neural architectures.

Evaluation and Limitations of XAI: Fidelity, faithfulness, stability, robustness, comprehensibility, and usefulness of explanations; human-centered evaluation; explanation uncertainty; misleading explanations; bias and fairness; privacy and security.

Advanced and Emerging Topics: Causal explainability; concept-based explanations; XAI for large language and foundation models; multimodal explainability; generative AI for explanations.

Course Outcomes:

CO1: Explain the fundamental concepts, motivation, and importance of Explainable Artificial Intelligence, including interpretability, explainability, transparency, trust, and accountability.

CO2: Distinguish between intrinsic (ante-hoc) and post-hoc explainability and analyze different types of explanations and explainability methods.

CO3: Apply major post-hoc explainability techniques to interpret machine learning and deep learning models.

CO4: Design and evaluate inherently interpretable models and assess the fidelity, faithfulness, stability, robustness, comprehensibility, and usefulness of explanations.

CO5: Analyze the limitations, risks, and emerging directions in Explainable Artificial Intelligence, including causal, concept-based, multimodal, and foundation-model explainability.

References:

1. Molnar, C., *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, 2nd ed., 2022.
2. Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K., and Müller, K.-R. (Eds.), *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, Springer, 2019.
3. Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., et al., "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI," *Information Fusion*, vol. 58, pp. 82–115, 2020.
4. Ribeiro, M. T., Singh, S., and Guestrin, C., "Why Should I Trust You?: Explaining the Predictions of Any Classifier," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.
5. Rudin, C., "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead," *Nature Machine Intelligence*, vol. 1, pp. 206–215, 2019.

MA881 Graphical Models — 3 credits

Introduction to probabilistic graphical models: directed and undirected graphical models, conditional independence, factorization, Bayesian networks, Markov random fields, Gibbs distributions, and conditional random fields.

Static models: linear Gaussian models, mixture models, factor analysis, multivariate Gaussian graphical models, exponential family distributions, and generalized linear models.

Dynamic models: Hidden Markov Models, Kalman filtering, and linear-Gaussian state space models.

Exact inference: variable elimination and related elimination algorithms; connections to dynamic programming; belief propagation, junction tree algorithm, optimal triangulation, and computational complexity results.

Approximate inference: loopy belief propagation, expectation propagation, sampling methods including Markov Chain Monte Carlo, Metropolis–Hastings, Gibbs sampling, and particle filtering.

Learning in graphical models: parameter learning and structure learning including the Chow–Liu algorithm; latent variable models including Latent Dirichlet Allocation, exchangeability and the de Finetti theorem, and inference using collapsed Gibbs sampling.

Course Outcomes:

CO1: Explain the fundamental principles of probabilistic graphical models, including Bayesian networks, Markov random fields, conditional independence, and factorization.

CO2: Analyze static and dynamic graphical models, including mixture models, factor analysis, Hidden Markov Models, Kalman filters, and linear-Gaussian state space models.

CO3: Apply exact inference techniques, including variable elimination, belief propagation, and junction tree algorithms, to probabilistic graphical models.

CO4: Analyze and evaluate approximate inference methods, including loopy belief propagation, expectation propagation, and sampling-based methods such as Markov Chain Monte Carlo.

CO5: Analyze parameter and structure learning methods in graphical models, including the Chow–Liu algorithm and latent variable models such as Latent Dirichlet Allocation.

References:

1. D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*, Cambridge University Press.
2. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
3. D. Barber, *Bayesian Reasoning and Machine Learning*, Cambridge University Press.

MA882 Graph Neural Networks — 3 credits

Foundations of graphs: graph representations; properties of graphs including degree, connectivity, and centrality measures.

Spectral graph theory and its role in learning on graphs.

Different types of graphs: heterogeneous graphs, bipartite graphs, multi-dimensional graphs, hypergraphs, and dynamic graphs.

Graph embeddings and representation learning for graph-structured data.

Graph Neural Networks (GNN): general framework for graph neural networks; graph filters including spectral-based graph filters and spatial-based graph filters; graph pooling; parameter learning in graph neural networks.

Sampling methods in graph neural networks: node-wise sampling methods, layer-wise sampling methods, and subgraph-wise sampling methods.

Variants of graph neural networks: heterogeneous graph neural networks, bipartite graph neural networks, multi-dimensional graph neural networks, hypergraph neural networks, and dynamic graph neural networks.

Course Outcomes:

CO1: Explain the fundamental concepts and properties of graphs relevant to machine learning.

CO2: Apply spectral graph theory and graph embedding techniques for learning on graph-structured data.

CO3: Analyze and implement Graph Neural Network models using spectral-based and spatial-based graph filters, graph pooling, and parameter learning.

CO4: Analyze and evaluate node-wise, layer-wise, and subgraph-wise sampling methods in Graph Neural Networks for efficient and scalable learning.

CO5: Design and implement advanced Graph Neural Network models for heterogeneous, bipartite, multi-dimensional, hypergraph, and dynamic graph-structured data.

References:

1. Ma, Y., and Tang, J., *Deep Learning on Graphs*, Cambridge University Press.
2. Hamilton, W., *Graph Representation Learning*, Morgan & Claypool Publishers.
3. PyTorch Geometric Documentation.

MA883 Large Language Models — 3 credits

Introduction to large language models: evolution of language models, tokenization methods, word and subword embeddings, attention mechanism, and transformer architecture; concepts of pretraining and fine-tuning; evaluation metrics for language models including perplexity, accuracy, BLEU, and ROUGE.

Training and optimization of large language models: training objectives such as masked language modeling and next-token prediction; gradient-based optimization methods including Adam; regularization techniques such as dropout and layer normalization; handling large-scale datasets and distributed training; GPU and TPU acceleration for large-scale model training.

Advanced concepts in large language models: prompt engineering, in-context learning, zero-shot, one-shot and few-shot learning; transfer learning for natural language processing; fine-tuning strategies; efficiency considerations and scaling laws for large models.

Applications and societal aspects of large language models including text generation, summarization, machine translation, question answering, and dialogue systems; ethical considerations such as bias, hallucinations, safety and alignment; practical aspects of deployment and inference optimization.

Course Outcomes:

CO1: Explain the fundamental principles of large language models, including tokenization, embeddings, attention mechanisms, and Transformer architectures.

CO2: Explain the training objectives, optimization techniques, and large-scale training strategies used in modern language models.

CO3: Apply prompt engineering, in-context learning, and fine-tuning approaches for solving natural language processing tasks using large language models.

CO4: Analyze the applications of large language models in tasks such as text generation, summarization, machine translation, and question answering.

CO5: Evaluate ethical, safety, alignment, deployment, and inference considerations associated with large language models.

References:

1. Vaswani, A. et al., *Attention is All You Need*, NeurIPS, 2017.
2. Devlin, J. et al., *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, NAACL, 2019.
3. Brown, T. et al., *Language Models are Few-Shot Learners*, NeurIPS, 2020.
4. Goldberg, Y., *Neural Network Methods for Natural Language Processing*, Morgan & Claypool, 2017.

MA888 Machine Learning for Remote Sensing — 3 credits

Remote Sensing Data and Machine Learning: Characteristics of multispectral, hyperspectral, and SAR data; spectral, spatial, and temporal representations; formulation of remote sensing problems as classification, regression, clustering, and prediction tasks; training and evaluation strategies for spatial data.

Machine Learning for Spectral and Spatial Analysis: Feature representation and dimensionality reduction; PCA and feature selection; supervised learning; unsupervised learning and clustering; spectral and spatial classification; regression of biophysical and geophysical parameters; spatial sampling and spatial autocorrelation.

Deep and Representation Learning for Remote Sensing: Deep learning for multispectral, hyperspectral, and SAR data; spectral-spatial representation learning; transfer learning and domain adaptation; self-supervised and weakly supervised learning; learning with limited and noisy labels; multimodal and multi-temporal remote sensing data.

Advanced Applications and Emerging Methods: Spectral unmixing and sub-pixel analysis; change detection and time-series analysis; land-cover and crop mapping; mineral and geological mapping; environmental and disaster monitoring; uncertainty, interpretability, and generalization of machine learning models for Earth observation.

Course Outcomes:

CO1: Explain the characteristics of multispectral, hyperspectral, and SAR data and the challenges of applying machine learning to remote sensing data.

CO2: Apply machine learning methods to spectral, spatial, and temporal remote sensing problems.

CO3: Develop and evaluate learning models for multispectral, hyperspectral, and SAR data.

CO4: Analyze learning problems involving limited labels, spatial dependence, domain differences, and multimodal remote sensing data.

CO5: Apply modern machine learning approaches to real-world Earth observation problems and evaluate their generalization, uncertainty, and interpretability.

References:

1. Richards, J. A., *Remote Sensing Digital Image Analysis: An Introduction*, Springer, 2013.
2. Mather, P. M. and Koch, M., *Computer Processing of Remotely-Sensed Images: An Introduction*, 4th ed., Wiley, 2011.
3. Bishop, C. M. and Bishop, H., *Deep Learning: Foundations and Concepts*, Springer, 2024.
4. Camps-Valls, G., Tuia, D., Zhu, X. X., and Reichstein, M. (Eds.), *Deep Learning for the Earth Sciences: A Comprehensive Approach to Remote Sensing, Climate Science and Geosciences*, Wiley, 2021.

MA884 MLOps — 3 credits

Introduction to MLOps: motivation for MLOps; challenges in deploying machine learning systems; differences between traditional software engineering and machine learning systems; overview of the machine learning lifecycle; collaboration between data scientists, ML engineers, and DevOps teams.

Data and experiment management: dataset versioning, data validation, experiment tracking, reproducibility of experiments, metadata management.

Model development workflows: training pipelines, feature engineering pipelines, model versioning, model registry, reproducibility and collaboration in machine learning development.

Continuous integration and continuous deployment for machine learning: CI/CD pipelines, automated testing for machine learning models, model packaging and containerization, deployment strategies for machine learning models.

Monitoring and maintenance of machine learning systems: model performance monitoring, data drift and concept drift detection, logging and alerting mechanisms, model retraining strategies.

Scalable machine learning systems: distributed training, serving machine learning models at scale, real-time and batch inference systems.

Ethics, governance and security in machine learning systems: responsible AI, fairness and transparency, privacy issues, model governance and lifecycle management.

Course Outcomes:

CO1: Explain the MLOps lifecycle and analyze the challenges involved in developing, deploying, and managing machine learning systems.

CO2: Apply data and experiment management techniques, including dataset versioning, data validation, experiment tracking, and reproducibility, in machine learning workflows.

CO3: Develop and manage machine learning workflows and pipelines, including feature engineering, model versioning, model registries, and reproducible model development.

CO4: Implement CI/CD pipelines for machine learning systems, including automated testing, containerization, deployment, monitoring, and maintenance strategies.

CO5: Analyze scalable, ethical, governance, privacy, and security considerations in the deployment and lifecycle management of machine learning systems.

References:

1. Treveil, M., *Introducing MLOps*, O'Reilly Media.
2. Burkov, A., *Machine Learning Engineering*, True Positive Inc.
3. Kreuzberger, D., Kühl, N., and Hirschl, S., *Machine Learning Operations (MLOps): Overview, Definition, and Architecture*, IEEE.
4. Documentation of tools such as *MLflow*, *Kubeflow*, and *TensorFlow Extended (TFX)*.

MA885 Riemannian Geometry for Machine Learning — 3 credits

Motivation from machine learning: geometric structure in data and model spaces; the manifold hypothesis; examples of manifolds arising in ML such as spheres, rotation groups, low-rank matrix manifolds, and symmetric positive definite (SPD) matrices. Mathematical preliminaries: smooth manifolds, smooth maps, tangent spaces, submanifolds, and product manifolds.

Riemannian metrics and geodesics in ML models: Riemannian metrics as measures of similarity and distance on curved spaces; geodesics and shortest paths; exponential and logarithm maps; normal coordinates and completeness. Examples on manifolds frequently used in ML such as spheres, the special orthogonal group, and Stiefel and Grassmann manifolds.

Geometry and curvature in representation learning: intuition for sectional, Ricci, and scalar curvature; effects of curvature on data representation and optimization; manifolds of constant curvature; hyperbolic geometry and its applications in hierarchical representation learning and graph embeddings.

Optimization on manifolds for machine learning: Riemannian gradients and Hessians; retractions and parallel transport; Riemannian gradient descent and trust-region methods; constrained optimization interpreted as optimization on embedded manifolds. Applications to orthogonality constraints, low-rank matrix learning, eigenvalue problems, and structured deep learning models.

Course Outcomes:

CO1: Explain geometric concepts such as manifolds, tangent spaces, and Riemannian metrics in the context of machine learning models and data representations.

CO2: Interpret geodesics, exponential and logarithm maps, and curvature, and explain their roles in representation learning and optimization.

CO3: Formulate machine learning problems with structural constraints, such as orthogonality, low-rank, and symmetric positive definite (SPD) constraints, as optimization problems on Riemannian manifolds.

CO4: Apply Riemannian optimization methods, including Riemannian gradient descent and trust-region methods, to machine learning problems on manifolds.

CO5: Apply geometric tools to analyze and process manifold-valued data, including covariance matrices, subspaces, and rotations, and discuss applications of hyperbolic geometry and geometric representation learning.

References:

1. Lee, J. M., *Introduction to Riemannian Manifolds*, Springer.
2. Lee, J. M., *Riemannian Manifolds: An Introduction to Curvature*, Springer.
3. do Carmo, M. P., *Riemannian Geometry*, Birkhäuser.
4. Guigui, N., Miolane, N., and Pennec, X., *Introduction to Riemannian Geometry and Geometric Statistics: From Basic Theory to Implementation with Geomstats*, Springer.
5. Boumal, N., *An Introduction to Optimization on Smooth Manifolds*, Cambridge University Press.

6. Amari, S.-I., *Information Geometry and Its Applications*, Springer.

MA867 Reinforcement Learning — 3 credits

The reinforcement learning problem: agents, environments, rewards, policies, value functions, and exploration versus exploitation; Markov decision processes.

Tabular solution methods including dynamic programming, Monte Carlo methods, and temporal difference learning; SARSA, Q-learning, and eligibility traces.

Approximate solution methods including value function approximation, policy gradient methods, and function approximation techniques.

Planning and learning in reinforcement learning; model-based methods and applications of reinforcement learning in sequential decision making.

Course Outcomes:

CO1: Explain the fundamental concepts of reinforcement learning, including agents, environments, policies, rewards, and Markov decision processes.

CO2: Analyze tabular reinforcement learning algorithms, including dynamic programming, Monte Carlo methods, and temporal difference learning.

CO3: Apply reinforcement learning algorithms, including SARSA and Q-learning, to sequential decision-making problems.

CO4: Evaluate approximate reinforcement learning methods using function approximation and policy gradient techniques.

CO5: Analyze planning and learning strategies in reinforcement learning and their applications in real-world problems.

References:

1. Sutton, R. S. and Barto, A. G., *Reinforcement Learning: An Introduction*, MIT Press, 2017.
2. Tesauro, G., *Temporal Difference Learning and TD-Gammon*, Communications of the ACM, March 1995.

MA886 Time Series — 3 credits

Difference Equations and Solutions; modeling of time series using regression analysis.

ARMA/ARIMA Models: introduction; moving average models MA(q); condition of invertibility; autoregressive models AR(p); Yule–Walker equations; stationarity conditions; autoregressive-moving average models ARMA(p,q); MLE estimation theory.

Forecasting: introduction to forecasting; modeling volatility; ARCH and GARCH processes.

Non-stationary Processes: time series co-integration; co-integration regression; testing of co-integration; vector autoregression and co-integration; co-integration and error correction model.

Structural Breaks; introduction to Vector Auto-Regression (VAR); structural VARs; applications of VARs.

Course Outcomes:

CO1: Explain the fundamental concepts of time series, difference equations, and regression-based modeling of time series data.

CO2: Analyze and apply ARMA and ARIMA models, including moving average and autoregressive models, stationarity conditions, Yule–Walker equations, and parameter estimation.

CO3: Apply time series forecasting methods and analyze volatility using ARCH and GARCH models.

CO4: Analyze non-stationary time series using co-integration, co-integration regression, and error correction models.

CO5: Analyze structural breaks and apply Vector Auto-Regression (VAR) and structural VAR models to multivariate time series and forecasting problems.

References:

1. Shumway, R. H. and Stoffer, D. S., *Time Series Analysis and Its Applications: With R Examples*, Springer.
2. Koller, D. and Friedman, N., *Probabilistic Graphical Models: Principles and Techniques*, MIT Press, 2009.
3. Mills, T. C., *The Econometric Modelling of Financial Time Series*, Cambridge University Press, 1999.
4. Harvey, A. C., *Time Series Models*, Harvester Wheatsheaf, 1993.

MA875 Topological Data Analysis — 3 credits

Introduction to topology for data analysis: motivation from machine learning and data science; shape of data and the role of topology in extracting global structure; simplicial complexes, simplicial homology groups, simplicial approximation, homotopy invariance, and Reeb graphs.

Complexes from point cloud data: construction of simplicial complexes from sampled data; nerves of a cover and the Nerve theorem; Čech complexes and Vietoris–Rips complexes; manifold reconstruction concepts including Hausdorff distance, Gromov–Hausdorff distance, and basic reconstruction results.

Persistent homology: filtered simplicial complexes, filtrations and persistence modules; computation of persistent homology; algorithms for extracting multi-scale topological features from data.

Representations of persistent homology: persistence diagrams, persistence barcodes, persistence landscapes, persistence images, and the mapper algorithm; interpretation of topological summaries in data analysis.

Stability and metrics in TDA: stability theorem for persistence diagrams; bottleneck distance and p-Wasserstein distance; tame functions; stability of persistence representations and robustness of topological summaries.

Applications in machine learning and data analysis: topological features for clustering and classification; topological summaries as features for learning algorithms; connections to graph learning, manifold learning, and geometric data analysis.

Course Outcomes:

CO1: Explain fundamental concepts of topology used in data analysis, including simplicial complexes, homology, homotopy invariance, and Reeb graphs.

CO2: Construct and analyze simplicial complexes from point cloud data using methods such as Čech and Vietoris–Rips complexes, and explain basic manifold reconstruction concepts.

CO3: Analyze multi-scale topological features of data using filtrations and persistent homology, and interpret persistence diagrams, barcodes, landscapes, persistence images, and mapper.

CO4: Evaluate the stability and robustness of topological summaries using metrics such as bottleneck and Wasserstein distances.

CO5: Apply topological features and summaries as representations in machine learning and data analysis tasks, including clustering and classification.

References:

1. Hatcher, A., *Algebraic Topology*, Cambridge University Press (2001).
2. Ghrist, R., *Elementary Applied Topology*, Createspace (2014).
3. Edelsbrunner, H., and Harer, J., *Computational Topology: An Introduction*, American Mathematical Society (2010).

4. Zomorodian, A., and Carlsson, G., "Computing Persistent Homology," *Discrete and Computational Geometry*, 33(2), 249–274 (2005).
5. Dey, T. K., and Wang, Y., *Computational Topology for Data Analysis*, Cambridge University Press (2022).

MA887 Transformers and Vision-Language Models — 3 credits

CNNs and the Need for Attention: Convolutional neural networks, receptive fields, and hierarchical visual representations; limitations of local feature extraction and the need for modeling global relationships in visual data.

Attention and Transformers: Embeddings and vector representations; similarity and dot products; Query, Key, and Value representations; scaled dot-product attention; self-attention and multi-head attention; Transformer architecture and positional encoding; conceptual and mathematical understanding of attention.

Vision Transformers and Large Language Models: Image patches and visual tokens; Vision Transformer (ViT); comparison of CNNs and Vision Transformers; Transformer-based language modeling; tokens, causal attention, and next-token prediction; conceptual understanding of Large Language Models (LLMs) as Transformer-based models.

Vision-Language Models and Applications: Joint representation of images and text; image-text alignment and contrastive learning; CLIP; integration of vision encoders with language models; Vision-Language Models (VLMs); visual question answering, image captioning, image-text retrieval, and visual grounding; applications of VLMs in computer vision and related domains.

Course Outcomes:

CO1: Explain the limitations of CNNs and the motivation for attention-based visual representation.

CO2: Explain the mathematical formulation and operation of self-attention and Transformer architectures.

CO3: Analyze the transition from image patches to visual tokens and explain the principles of Vision Transformers.

CO4: Explain how Transformers form the foundation of Large Language Models and Vision-Language Models.

CO5: Analyze and apply Transformer-based models to contemporary computer vision and multimodal vision tasks.

References:

1. Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, MIT Press, 2016.
2. Vaswani, A., et al., “Attention Is All You Need,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
3. Dosovitskiy, A., et al., “An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale,” *International Conference on Learning Representations (ICLR)*, 2021.
4. Radford, A., et al., “Learning Transferable Visual Models From Natural Language Supervision,” *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.