Department of Computer Science

Details of Lesson Plan

S.No.	Particulars	Details
1.	Course Name	Artificial Intelligence with Computer Vision
2.	Course Code	COM-601
3.	Academic Year	2023-2024
4.	Semester	6th
5.	Number of Lesson plans	40
6.	Faculty Assigned	Shubham Gupta

Faculty Signature

Lesson Plan No. 1	Course Name: Artificial Intelligence with Computer Vision Topic: Introduction to Artificial Intelligence: Basic of AI	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the basic concepts and definitions of Artificial Intelligence (AI). b. Explain the different types of AI: Narrow AI, General AI, and Superintelligence. c. Identify key AI techniques such as machine learning, natural language processing, and computer vision. d. Discuss real-world applications of AI across various industries.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a brainstorming session: Ask students what they know about Artificial Intelligence and where they have seen it applied in their daily lives (e.g., voice assistants like Siri, recommendation systems on Netflix). - Provide a simple definition of AI: Artificial Intelligence is the simulation of human intelligence by machines, especially computer systems, enabling them to perform tasks that typically require human intelligence, such as learning, reasoning, and problem-solving. 2. **Development** (30 minutes) a. Basic Concepts of AI: Explain that AI is a broad field that includes subfields such as machine learning, natural language processing, robotics, and computer vision. Define AI as a branch of computer science that aims to create systems capable of performing tasks that require intelligence, such as perception, decision-making, language understanding, and learning. b. Types of AI: Narrow AI (Weak AI): Explain that narrow AI is designed and trained for a specific task (e.g., facial recognition, language translation). It performs well in specialized domains but cannot generalize to different tasks. Example: Google Translate or image recognition systems. General AI (Strong AI): Describe general AI as the theoretical

	concept of an AI that possesses human-level cognitive abilities and can perform any intellectual task that a human can. Example: A machine that could learn new tasks on its own and reason in a general way like a human (currently hypothetical). Artificial Superintelligence: Introduce the concept of artificial superintelligence as AI that surpasses human intelligence in all areas, including creativity, problem-solving, and social skills. Example: Theoretical future AI systems that could outperform humans in every task (also currently hypothetical). c. Key AI Techniques: Machine Learning: Explain that machine learning is a subset of AI that focuses on building systems that can learn from and make decisions based on data. Introduce the idea of supervised, unsupervised, and reinforcement learning. Example: A machine learning algorithm predicting whether an email is spam or not. Natural Language Processing (NLP): Discuss NLP as the field of AI that enables computers to understand, interpret, and respond to human language. Example: Chatbots, voice assistants like Siri or Alexa. 3. Exercise (5 minutes) – - Assign students to research and present an AI application in a specific industry (e.g., healthcare, finance, or retail). Ask them to describe how AI is used, the benefits it provides, and any potential challenges or ethical considerations. - Afterward, have students share their findings and discuss the role of AI in transforming industries. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the key concepts of AI, including the different types of AI (narrow, general, superintelligence), the core AI techniques, and the various applications across industries. 2. Encourage students to think about how AI might continue to evolve and impact society in the future. 3. Allow time for questions and discussion on the ethical and societal implications of AI. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "How is narrow AI different from general AI?" and "What are the key applications of AI in the healthcare industry?" 3. Conduct an online quiz or group activity at the end of the class to

	assess students' understanding of the basic concepts of AI and their ability to identify AI applications in real-world scenarios. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2	Course Name: Artificial Intelligence with Computer Vision Topic: history of AI	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the origins and evolution of Artificial Intelligence (AI). b. Identify the key milestones and breakthroughs in AI development. c. Discuss the different periods of AI research, including the early beginnings, the AI winters, and the modern AI resurgence.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a discussion: Ask students if they are familiar with any historical figures or events that shaped the field of AI. - Explain that understanding the history of AI helps provide context for its current state and future developments. Introduce the idea that AI has evolved over several decades, with periods of progress and setbacks. 2. **Development** (30 minutes) a. Early Beginnings of AI (1940s–1950s): Discuss the origins of AI in the 1940s and 1950s, when early computing pioneers like Alan Turing began to explore the possibility of machines simulating human intelligence. Key Milestone: Introduce the Turing Test (1950), proposed by Alan Turing, as a method to determine if a machine can exhibit intelligent behavior indistinguishable from that of a human. Example: The Turing Test's influence on modern AI and the development of intelligent systems. b. Birth of AI as a Field (1956): Explain the Dartmouth Conference (1956), organized by John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon, where the term "Artificial Intelligence" was coined. This event marked the official birth of AI as a field of study. Discuss early AI programs such as Logic Theorist (1956) by Allen Newell and Herbert A. Simon, which could prove mathematical theorems. c. The First AI Boom (1956–1970s): Highlight the optimism and early achievements in AI during the 1950s and 1960s, including rule-based systems and early successes in tasks such as game playing (e.g., IBM's Deep Blue for chess).

	Example: Shakey the Robot (1966), the first general-purpose robot to perceive and interact with its environment autonomously. d. AI Winter (1970s–1980s): Explain the concept of AI winter, a period of reduced funding and interest in AI research due to unmet expectations and the limitations of early AI systems. Discuss the challenges that led to the AI winters, including the inability of rule-based systems to handle real-world complexity and the limited computational power of the time. Example: Decline in AI research funding in the 1970s and 1980s after the initial hype failed to deliver on the ambitious promises of early AI pioneers. e. Resurgence of AI (1980s–1990s): Explain how AI saw renewed interest in the 1980s with the rise of expert systems, which used knowledge from human experts to solve specific problems. Key Milestone: Introduce IBM's Watson and Deep Blue, which achieved significant successes in specific domains (e.g., Watson winning Jeopardy! in 2011). Discuss how the development of more powerful computers and better algorithms, including the rise of machine learning and neural networks, helped reignite interest in AI. f. Modern AI Boom (2000s–Present): Discuss the modern resurgence of AI in the 2000s, driven by advancements in machine learning, deep learning, and the availability of vast amounts of data. Key Milestone: Introduce DeepMind's AlphaGo (2016), which defeated the world champion in Go, a complex board game that had long been considered a challenge for AI. Explain how AI has become integrated into many aspects of everyday life, from virtual assistants (e.g., Siri, Alexa) to autonomous vehicles and personalized recommendations (e.g., Netflix, Amazon). 3. Exercise (5 minutes) – - Assign students to research a key historical figure or milestone in AI history. Ask them to present a brief overview of their chosen topic, explaining its significance to the development of AI. - Afterward, have students share their findings and discuss how each milestone contributed to the progress of AI as we know it today. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the major periods in AI history, including the early

	beginnings, the AI winters, and the modern resurgence of AI. 2. Highlight the key milestones and figures that shaped AI development, from the Dartmouth Conference to the rise of machine learning and deep learning. 3. Encourage students to ask questions or share their thoughts on the significance of these milestones and the potential future of AI. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "What were the key factors that led to the AI winters?" and "How did advancements in computing power influence the modern resurgence of AI?" 3. Conduct an online quiz or group discussion at the end of the class to assess students' understanding of the history of AI and their ability to identify key milestones and figures. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 3	Course Name: Artificial Intelligence with Computer Vision Topic: various techniques of Artificial Intelligence: Machine learning	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the role of machine learning (ML) as a technique within artificial intelligence (AI). b. Differentiate between the main types of machine learning: supervised learning, unsupervised learning, and reinforcement learning. c. Identify common machine learning algorithms and their applications.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a discussion: Ask students what they know about machine learning and how it fits within the broader field of artificial intelligence. - Explain that machine learning is a core technique within AI that focuses on creating algorithms and models that enable computers to learn from data and make predictions or decisions without being explicitly programmed. 2. **Development** (30 minutes) a. What is Machine Learning? Define machine learning as a subset of AI where machines are trained to learn patterns from data, adapt to new inputs, and make data-driven decisions or predictions. Example: Image recognition, where a machine learning model learns to classify images based on patterns in the data. b. Types of Machine Learning: Supervised Learning: Explain that supervised learning involves training a model on labeled data, where the input data has corresponding output labels (e.g., classification and regression tasks). Example: Predicting house prices (regression) or classifying emails as spam or not spam (classification). Common Algorithms: Linear regression, logistic regression, decision trees, support vector machines (SVMs). Unsupervised Learning: Define unsupervised learning as a technique where the model is

	trained on unlabeled data and tasked with finding hidden patterns or structures in the data (e.g., clustering or dimensionality reduction). Example: Customer segmentation, where customers are grouped based on purchasing behavior without predefined labels. Common Algorithms: k-means clustering, hierarchical clustering, principal component analysis (PCA). Reinforcement Learning: Introduce reinforcement learning as a type of machine learning where an agent learns to make decisions by interacting with an environment and receiving feedback in the form of rewards or penalties. Example: Training a robot to navigate a maze or teaching an AI to play a game like Go or chess. Common Algorithms: Q-learning, Deep Q Networks (DQN), policy gradient methods. 3. Exercise (5 minutes) – - Assign students to work with a small dataset (e.g., Iris or Titanic dataset) and implement either a supervised or unsupervised machine learning model using Python (with scikit-learn). - Ask students to train and test their models, interpret the results, and identify potential improvements in the model's performance. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the different types of machine learning: supervised, unsupervised, and reinforcement learning, and highlight how each is used to solve different types of problems. 2. Encourage students to reflect on real-world applications of machine learning and how the techniques they've learned could be applied to problems in industries such as healthcare, finance, or transportation. 3. Allow time for questions and discussion on which machine learning approach might be best suited for specific types of data and tasks. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "How does supervised learning differ from unsupervised learning?" and "In what situations would you use reinforcement learning?"

	3. Conduct an online quiz or coding exercise at the end of the class to assess students' understanding of the various machine learning techniques and their ability to apply them to real-world datasets. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 4	Course Name: Artificial Intelligence with Computer Vision Topic: various techniques of Artificial Intelligence: NLP	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the basic concepts and goals of Natural Language Processing (NLP). b. Identify key techniques and algorithms used in NLP, such as tokenization, stemming, lemmatization, and parsing. c. Explain real-world applications of NLP, including sentiment analysis, language translation, and chatbots.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a brainstorming session: Ask students if they have interacted with AI systems that understand or generate language (e.g., Siri, Alexa, Google Translate). - Introduce Natural Language Processing (NLP) as a field of AI focused on enabling computers to understand, interpret, and generate human language. Explain that NLP bridges the gap between human communication and machine understanding. 2. **Development** (30 minutes) a. Overview of NLP: Explain that NLP involves a set of techniques and algorithms that allow computers to process and analyze large amounts of natural language data. This includes tasks such as text classification, machine translation, speech recognition, and sentiment analysis. Emphasize the importance of language in human communication and the challenges of teaching machines to understand nuances like grammar, meaning, and context. b. Key NLP Techniques: Tokenization: Explain tokenization as the process of breaking down text into smaller units, such as words or sentences. Discuss the importance of tokenization as the first step in most NLP tasks. Example: Tokenizing a sentence like "AI is transforming industries" into individual words: ["AI", "is", "transforming", "industries"]. Stemming and Lemmatization: Introduce stemming and

	lemmatization as techniques used to reduce words to their base or root forms. Stemming: Discuss how stemming removes suffixes from words to get their base form (e.g., "running" to "run"). Lemmatization: Explain that lemmatization takes into account the grammatical context and reduces words to their dictionary form (e.g., "running" becomes "run" and "better" becomes "good"). Example: Using stemming to reduce variations of the word "run" (e.g., "running", "runs") to the root "run". 3. Exercise (5 minutes) – - Ask students to share their results and discuss the challenges they faced, particularly in cleaning and preparing the text data for analysis. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the key techniques of NLP, including tokenization, stemming, lemmatization, and named entity recognition. 2. Highlight the wide range of applications for NLP in industries like healthcare, finance, and e-commerce. 3. Encourage students to ask questions or share their thoughts on how NLP could be applied to solve real-world problems in areas they are interested in. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "What are the key differences between stemming and lemmatization?" and "How is NLP used to improve customer service through chatbots?" 3. Conduct an online quiz or coding exercise at the end of the class to assess students' understanding of NLP techniques and their ability to apply them to real-world text data. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 5	Course Name: Artificial Intelligence with Computer Vision Topic: various techniques of Artificial Intelligence: Automation & Robotics	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the role of Artificial Intelligence in automation and robotics. b. Differentiate between automation and robotics and explain their relationship to AI. c. Identify key AI techniques used in robotics, such as machine learning, computer vision, and reinforcement learning. d. Discuss real-world applications of AI in automation and robotics, such as autonomous vehicles, industrial robots, and AI-driven manufacturing systems.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a brainstorming session: Ask students what comes to mind when they think about automation and robotics. Have them share examples of where they've seen AI applied in these fields (e.g., autonomous vehicles, robotic vacuums). - Introduce the concepts of automation and robotics, and explain how Artificial Intelligence enhances these fields by enabling machines to perform tasks autonomously. 2. **Development** (30 minutes) a. Automation and AI: Define automation as the use of technology to perform tasks with minimal human intervention. Discuss the use of AI to make automation more intelligent and capable of handling complex tasks. Example: Automation in business processes, such as AI-powered chatbots for customer service or AI-driven automation in supply chain management. b. Robotics and AI: Define robotics as the branch of technology that involves the design, construction, and operation of robots. Explain how AI plays a crucial role in making robots more adaptable and capable of performing tasks autonomously in dynamic environments. Example: Autonomous robots used in warehouses for picking

	and sorting items. c. Relationship Between Automation, Robotics, and AI: Discuss how AI enhances both automation and robotics by enabling machines to perceive their environment, make decisions, and learn from their experiences. Example: Autonomous cars combining AI for decision-making, automation for performing driving tasks, and robotics for controlling the physical components of the vehicle. d. AI Techniques in Robotics: Machine Learning: Explain how robots use machine learning to improve their performance over time by learning from data. Discuss applications such as predictive maintenance and adaptive control in robotic systems. Example: A robotic arm learning to optimize its movements for faster assembly in a manufacturing line. Computer Vision: Introduce computer vision as a key AI technique that allows robots to interpret visual data from cameras or sensors to understand their environment. Example: Self-driving cars using computer vision to detect obstacles, lane markings, and pedestrians. 3. Exercise (5 minutes) – - Assign students to research a specific application of AI in robotics or automation. Ask them to prepare a short presentation on how AI is used in their chosen example, what tasks the AI performs, and the impact of the technology on the industry. - Afterward, have students share their findings with the class and discuss how AI is transforming different sectors through automation and robotics. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the main concepts of AI in automation and robotics, highlighting the various AI techniques such as machine learning, computer vision, and reinforcement learning. 2. Encourage students to reflect on how AI-driven automation and robotics might continue to evolve and impact industries such as healthcare, manufacturing, and transportation. 3. Allow time for questions and further discussion on the future possibilities and ethical implications of AI in robotics. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "How does reinforcement learning differ

	from traditional programming approaches in robotics?" and "What are the potential benefits and challenges of AI-powered automation in manufacturing?" 3. Conduct an online quiz or group discussion at the end of the class to assess students' understanding of AI techniques in automation and robotics. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 6	Course Name: Artificial Intelligence with Computer Vision Topic: Types of AI	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the different types of AI and their characteristics. b. Differentiate between Narrow AI (Weak AI), General AI (Strong AI), and Superintelligence. c. Identify real-world examples of Narrow AI and discuss the current state of AI research in General AI and Superintelligence. d. Discuss the implications of different types of AI for technology and society.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a discussion: Ask students what they think AI is capable of today and where they see AI systems being used. - Introduce the concept of AI being classified into different types based on its capabilities, ranging from simple systems designed for specific tasks to the possibility of machines that could outperform humans in all cognitive tasks. 2. **Development** (30 minutes) a. Narrow AI (Weak AI): Define Narrow AI as AI that is designed and trained for a specific task. It is highly effective at completing predefined tasks but cannot perform outside its designated function or generalize to other domains. Characteristics: Narrow AI lacks self-awareness and operates within strict boundaries, focusing on single tasks such as image recognition, speech processing, or game playing. Example: Virtual assistants like Siri and Alexa, recommendation systems like Netflix's, and image recognition systems used in social media. Current Applications: Discuss how Narrow AI is widely used across industries today, from healthcare (diagnostic tools) to finance (fraud detection). b. General AI (Strong AI): Define General AI as AI that possesses human-like cognitive abilities and can understand, learn, and apply knowledge across different tasks and domains. General AI would have the ability to perform any intellectual task that a human can do.

	Characteristics: General AI would be capable of reasoning, problem-solving, understanding context, and applying learned knowledge in a broad range of areas without being explicitly programmed for each task. Example (Hypothetical): A machine that could learn to perform complex tasks like driving a car, diagnosing a disease, or writing a book with equal competence, without needing specialized programming. Current Research: Explain that General AI remains a theoretical goal of AI research and has not yet been achieved. Researchers are exploring cognitive architectures, reinforcement learning, and transfer learning as potential pathways toward General AI. c. Artificial Superintelligence (ASI): Define Superintelligence as AI that surpasses human intelligence in every possible field, including creativity, general wisdom, and problem-solving. Characteristics: ASI would be able to outperform the best human minds in every domain, including scientific research, artistic creativity, social intelligence, and strategic decision-making. Example (Hypothetical): A superintelligent AI that could create new technologies, solve global challenges, or redesign entire industries with greater effectiveness than human experts. Ethical Considerations: Discuss the potential risks and ethical implications of superintelligence, such as the possibility of AI systems acting in ways that humans cannot predict or control. Highlight the debates around AI safety and control measures. 3. Exercise (5 minutes) – - Assign students to research and present an example of Narrow AI that they find interesting. Ask them to describe how the AI works, what task it performs, and the impact it has on the industry or society. - Afterward, have students share their findings and discuss how they envision General AI and Superintelligence might develop in the future. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the three types of AI: Narrow AI, General AI, and Superintelligence, emphasizing their differences in capabilities and applications. 2. Encourage students to think critically about the future of AI, the potential for each type, and the societal and ethical implications of advanced AI systems. 3. Allow time for questions and discussion on what students believe

	are the most promising or concerning aspects of AI development. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "What makes General AI fundamentally different from Narrow AI?" and "What ethical considerations should we keep in mind as we move toward more advanced AI systems?" 3. Conduct an online quiz or group activity at the end of the class to assess students' understanding of the types of AI and their implications for technology and society. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 7	Course Name: Artificial Intelligence with Computer Vision Topic: Applications of Artificial Intelligence	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the broad range of applications of AI across various industries. b. Identify how AI is transforming fields such as healthcare, finance, transportation, and entertainment. c. Discuss real-world case studies of AI in action and the benefits and challenges associated with its implementation. d. Explore the ethical considerations and potential societal impacts of AI applications.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a brainstorming session: Ask students where they have seen or experienced AI in their daily lives (e.g., Siri, Netflix recommendations, self-driving cars). - Provide an overview of how AI is being integrated into various industries to improve efficiency, decision-making, and customer experiences. 2. **Development** (30 minutes) a. Healthcare: Explain how AI is revolutionizing healthcare by enhancing diagnosis, treatment planning, and patient care. Example: AI-powered diagnostic tools like IBM's Watson for Oncology, which helps doctors analyze medical data to provide personalized cancer treatments. Additional Applications: Predictive analytics for early disease detection, AI-driven robots assisting in surgeries, virtual health assistants helping patients with routine inquiries. b. Finance: Discuss the role of AI in financial services, from fraud detection to personalized financial advice. Example: AI algorithms used in fraud detection systems that analyze patterns in transaction data to flag suspicious activity. Additional Applications: Robo-advisors providing personalized investment recommendations, AI-driven trading platforms that execute trades at high speeds, credit scoring algorithms that

	assess loan applications. c. Transportation: Explore how AI is being applied to autonomous vehicles, traffic management, and logistics optimization. Example: Self-driving cars like those developed by Tesla and Waymo that use AI to navigate roads, detect obstacles, and make driving decisions. Additional Applications: AI in fleet management to optimize delivery routes, reduce fuel consumption, and manage transportation logistics more efficiently. d. Entertainment: Explain how AI enhances user experiences in entertainment through personalized recommendations, content creation, and interactive gaming. Example: AI recommendation engines used by streaming platforms like Netflix and Spotify to suggest content based on user preferences and behavior. Additional Applications: AI-generated music and art, virtual influencers, and AI-driven characters in video games that adapt to player behavior. 3. Exercise (5 minutes) – - Assign students to research a real-world application of AI in an industry of their choice. Ask them to present a brief overview of how AI is being used, the benefits it provides, and any challenges or ethical considerations associated with its use. - Afterward, have students share their findings and discuss how AI is transforming various industries. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the wide-ranging applications of AI across industries, highlighting the transformative potential of AI in improving efficiency, enhancing decision-making, and personalizing experiences. 2. Emphasize the importance of considering the ethical and societal impacts of AI implementations, such as data privacy, job displacement, and bias in AI algorithms. 3. Allow time for questions and discussion on the future of AI in industries and the potential challenges it may face. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "How is AI changing the way healthcare is delivered?" and "What are the potential risks of AI-driven

	financial services?" 3. Conduct an online quiz or group activity at the end of the class to assess students' understanding of AI applications and their ability to identify real-world use cases across various industries. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 8	Course Name: Artificial Intelligence with Computer Vision Topic: Two Ways Artificial Intelligence Works: Symbolic AI and Data-Based AI	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the two main approaches in AI: Symbolic AI and Data-Based AI. b. Differentiate between Symbolic AI (rule-based systems) and Data-Based AI (machine learning and deep learning). c. Discuss the advantages, limitations, and use cases of each approach. d. Identify real-world examples of both Symbolic AI and Data-Based AI in practice.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a brainstorming session: Ask students if they know different ways in which AI systems learn or perform tasks. - Explain that AI can be categorized into two major approaches: Symbolic AI (based on rules and logic) and Data-Based AI (based on data and patterns). 2. **Development** (30 minutes) a. Symbolic AI (Rule-Based AI): Definition: Explain that Symbolic AI, also known as rule-based AI, is an approach where intelligence is encoded through explicit rules and logical statements, often represented in formal languages. These systems rely on human-defined rules to make decisions and solve problems. Example: Expert systems in the 1980s, such as MYCIN, which used predefined rules to diagnose bacterial infections and recommend treatment. Components: Symbolic AI involves knowledge representation (e.g., ontologies and logical rules), inference engines, and reasoning algorithms. Use Cases: Symbolic AI is effective in domains where the rules are well understood and clearly defined, such as medical diagnosis, legal reasoning, and automated theorem proving. Advantages: Transparent decision-making: Every decision made by the system can be traced back to specific rules.

	Works well in environments with clear and structured rules. Limitations: Rigid and inflexible: Struggles to handle uncertainty, ambiguity, or complex real-world data. Requires human experts to define rules, which can be time-consuming and difficult to scale. b. Data-Based AI (Machine Learning and Deep Learning): Definition: Explain that Data-Based AI relies on data to learn patterns, make predictions, and improve over time. This approach is driven by machine learning and deep learning techniques, where the system learns from examples rather than being explicitly programmed. Example: Image recognition using deep learning, where a neural network is trained on millions of labeled images to recognize objects like cats, dogs, and cars. Components: Data-Based AI involves training algorithms, models (e.g., neural networks, decision trees), and large datasets for learning. Use Cases: Data-Based AI excels in tasks involving large amounts of unstructured data, such as image and speech recognition, natural language processing, and predictive analytics. Advantages: Adaptive and scalable: Can learn from data and improve over time without needing human intervention to define rules. Effective in complex and dynamic environments with large datasets. Limitations: Black-box nature: The decision-making process can be difficult to interpret and explain, especially with deep learning models. Requires large amounts of labeled data for training and significant computational resources. c. Comparison Between Symbolic AI and Data-Based AI: Symbolic AI: Relies on logic, reasoning, and rules defined by humans. Best for structured environments with clear rules but struggles with ambiguity and complex data. d. Data-Based AI: Learns from data to find patterns and make decisions. Best for complex, unstructured environments but requires large datasets and can be difficult to interpret. e. Real-World Examples: Symbolic AI: Expert systems used in legal reasoning, where rules and logical reasoning help lawyers analyze cases. Logic-based chatbots that follow predefined scripts to respond to customer inquiries. Data-Based AI: Google's DeepMind using deep learning to play and master

	video games like AlphaGo. Predictive text models like GPT-3, which use vast amounts of data to generate human-like text. 3. Exercise (5 minutes) – - Assign students to research a specific real-world application of either Symbolic AI or Data-Based AI. Ask them to describe how the AI works, what approach is being used, and why it is suitable for the problem. - Afterward, have students present their findings and discuss the strengths and weaknesses of each AI approach in the given scenario. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the two main approaches to AI: Symbolic AI, which uses rules and logic, and Data-Based AI, which relies on data to learn patterns and make decisions. 2. Emphasize the importance of choosing the right approach based on the problem domain and the nature of the data involved. 3. Encourage students to reflect on how these approaches might evolve and converge in the future, combining the strengths of both Symbolic AI and Data-Based AI. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "In what scenarios would Symbolic AI be more effective than Data-Based AI?" and "What are the key challenges faced by Data-Based AI models?" 3. Conduct an online quiz or group discussion at the end of the class to assess students' understanding of the differences between Symbolic AI and Data-Based AI and their ability to identify real-world applications for each. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1	Course Name: Artificial Intelligence with Computer Vision Topic: Foundation of Machine learning: Terminology used in ML	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the foundational terminology used in machine learning. b. Define key terms such as model, training, testing, features, labels, and overfitting. c. Identify the types of machine learning algorithms: supervised, unsupervised, and reinforcement learning. d. Recognize the difference between common metrics used to evaluate machine learning models.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - what they think machine learning is and whether they have encountered any terms related to it. - Introduce machine learning as a subset of artificial intelligence where machines learn from data, and explain the importance of understanding its terminology before diving into more complex topics. 2. **Development** (30 minutes) a. Model: Explain that a model in machine learning is a mathematical representation of a process or system that learns from data to make predictions or decisions. Example: A linear regression model that predicts house prices based on features like size, location, and number of rooms. b. Training: Define training as the process where the machine learning algorithm learns from a dataset by adjusting parameters to minimize error. Explain that this is done using the training data. Example: Training a decision tree to classify whether an email is spam or not. c. Testing: Explain that after training, the model is tested on a separate dataset (testing data) to evaluate its performance and generalization ability. Example: Testing a classifier on new emails it hasn't seen before to check if it can correctly identify spam. d. Features and Labels: Introduce features as the input variables or attributes used to predict an outcome, and labels as the output variable or target that the model is trying to predict. Example: In a housing price prediction model, the features

	could be the number of bedrooms, size, and location, while the label would be the price. e. Overfitting and Underfitting: Discuss overfitting as a model that learns too much from the training data and performs poorly on new, unseen data, and underfitting as a model that is too simple to capture the underlying patterns in the data. Example: A model that perfectly predicts prices in the training set but fails on new data is overfitting. 3. Exercise (5 minutes) – - Assign students to work with a simple dataset (e.g., the Iris dataset) and implement a basic classification model using scikit-learn. Have them identify the features and labels, train the model, and evaluate it using accuracy as a metric. - Afterward, ask students to share their results and discuss the challenges they faced in understanding the terminology. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the foundational terms covered in the lesson, including model, training, testing, features, labels, and overfitting. 2. Emphasize the importance of understanding these concepts before diving deeper into machine learning algorithms and techniques. 3. Encourage students to ask questions or share their thoughts on how these terms apply to real-world machine learning problems. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "What is the difference between training and testing data?" and "How do you know if your model is overfitting?" 3. Conduct an online quiz or coding exercise at the end of the class to assess students' understanding of the basic terminology used in machine learning. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2	Course Name: Artificial Intelligence with Computer Vision Topic: ML Process (Building the Model)	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the steps involved in building a machine learning model. b. Explain the importance of data preprocessing, model selection, training, evaluation, and tuning. c. Evaluate and improve the performance of a model by adjusting hyperparameters.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a brainstorming session: Ask students if they can recall any steps involved in building a machine learning model from their previous experience or understanding. - Provide an overview of the machine learning process, emphasizing that it involves more than just selecting a model—it's a series of steps that include data preparation, model training, evaluation, and tuning. 2. **Development** (30 minutes) a. Step 1: Problem Definition Define the machine learning task (e.g., classification, regression, clustering). Discuss the importance of clearly defining the problem to choose the right model and evaluation metrics. Example: Predicting whether a customer will churn (classification) or forecasting stock prices (regression). b. Step 2: Data Collection and Preprocessing Explain that data is the foundation of any machine learning model, and its quality directly impacts model performance. Discuss techniques for handling missing data, encoding categorical variables, normalizing or scaling numerical features, and splitting the data into training and testing sets. Example: Using techniques like one-hot encoding for categorical data and scaling numerical features using standardization or normalization. c. Step 3: Model Selection Introduce the concept of model selection. Explain how different algorithms (e.g., decision trees, linear regression, k-nearest neighbors) are suited for different types of problems. Discuss the trade-offs between different models, such as

	complexity, interpretability, and computational efficiency. Example: Choosing between a decision tree and a support vector machine (SVM) for a classification task. d. Step 4: Training the Model Explain the process of training a machine learning model using the training data. Discuss how the model learns by adjusting its internal parameters to minimize error (e.g., using loss functions like MSE for regression or cross-entropy for classification). Example: Training a linear regression model to predict house prices based on features like square footage and location. e. Step 5: Model Evaluation Discuss the importance of evaluating the model on a separate testing set to measure its generalization ability. Introduce common evaluation metrics, such as accuracy, precision, recall, F1 score, and mean squared error. Explain how cross-validation can be used to evaluate the model more robustly by testing it on multiple subsets of the data. Example: Evaluating a classifier's accuracy and F1 score on a test set of customer data. f. Step 6: Hyperparameter Tuning Explain the concept of hyperparameters (e.g., learning rate, depth of decision trees, number of neighbors in kNN) and how they differ from the parameters learned during training. Introduce techniques for tuning hyperparameters, such as grid search and random search, to find the optimal settings for the model. Example: Tuning the number of trees in a random forest classifier using grid search. g. Step 7: Deployment and Monitoring Briefly introduce the deployment phase, where the trained model is put into production to make predictions on new data. Discuss the importance of monitoring the model's performance over time, as data distributions may change, requiring model updates or retraining. 3. Exercise (5 minutes) – - Assign students to work with a simple dataset (e.g., Iris or Titanic dataset) and implement the full machine learning process: data preprocessing, model selection, training, evaluation, and hyperparameter tuning. - Afterward, ask students to share their models and discuss any challenges they faced, particularly with model selection and tuning. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the steps of the machine learning process, emphasizing the

	importance of each step in building an accurate and robust model. 2. Highlight that building a successful machine learning model involves not just selecting an algorithm, but also preparing the data, evaluating performance, and fine-tuning the model to achieve the best results. 3. Encourage students to ask questions or share their thoughts on how they might apply the machine learning process to different types of problems in the real world. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "Why is data preprocessing critical in the machine learning process?" and "How does cross-validation improve model evaluation?" 3. Conduct an online quiz or coding exercise at the end of the class to assess students' understanding of the machine learning process and their ability to implement it in practice. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 3	Course Name: Artificial Intelligence with Computer Vision Topic: Supervised ML	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of supervised learning and its importance in machine learning. b. Identify the key components of supervised learning: features, labels, training data, and test data. c. Differentiate between classification and regression problems.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a brainstorming session: Ask students what they know about machine learning and how they think machines learn from data. - Introduce supervised learning as one of the most common types of machine learning, where the model learns from labeled data (input-output pairs) to make predictions. 2. **Development** (30 minutes) a. Supervised Learning Overview: Explain that supervised learning involves training a model on a labeled dataset, where each data point consists of input features (X) and a corresponding output label (Y). The goal is to learn a mapping from inputs to outputs that can be generalized to new, unseen data. Example: A spam email classifier, where the features are the words in the email, and the label is whether the email is spam or not. b. Classification vs. Regression: Explain the two types of supervised learning tasks: c. Classification: Predicting a discrete label or category (e.g., spam or not spam). d. Regression: Predicting a continuous value (e.g., predicting house prices based on features like size and location). Example: Use the Iris dataset to illustrate classification (predicting the species of a flower) and the Boston housing dataset to illustrate regression (predicting house prices). e. Training and Testing: Discuss the importance of splitting the dataset into training and testing sets. The training data is used to fit the model, while the testing data is used to evaluate the model's performance on unseen data. Example: Splitting the Iris dataset into 80% training data and

	20% testing data for classification. f. Supervised Learning Algorithms: Linear Regression: Explain how linear regression is used for regression tasks to model the relationship between the input features and the output label. Logistic Regression: Introduce logistic regression as a classification algorithm that models the probability of a binary outcome. Decision Trees: Discuss decision trees as a non-linear model used for both classification and regression tasks, which splits the data based on feature values to make predictions. k-Nearest Neighbors (kNN): Introduce kNN as a simple, instance-based learning algorithm that classifies a new data point based on the majority class of its nearest neighbors. 3. Exercise (5 minutes) – - Assign students to work with a dataset (e.g., Titanic or Boston housing dataset) and implement a supervised learning model using scikit-learn. Ask them to select either a classification or regression task, train the model, and evaluate its performance using appropriate metrics. - Afterward, have students present their results and discuss any challenges they faced during the implementation. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the key concepts of supervised learning, including the distinction between classification and regression, the importance of training and testing, and the risks of overfitting and underfitting. 2. Highlight the versatility of supervised learning and its wide range of applications, from image classification to price prediction. 3. Encourage students to ask questions or share their thoughts on how supervised learning could be applied to real-world problems they are interested in. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "What is the difference between classification and regression?" and "How do you prevent overfitting when training a supervised learning model?" 3. Conduct an online quiz or coding exercise at the end of the class to assess students' understanding of supervised learning and their ability to implement basic models. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 4	Course Name: Artificial Intelligence with Computer Vision Topic: Unsupervised ML	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of unsupervised learning and its difference from supervised learning. b. Identify common unsupervised learning tasks, such as clustering and dimensionality reduction. c. Explain the purpose of algorithms like k-means clustering and Principal Component Analysis (PCA).
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a brainstorming session: Ask students what they think unsupervised learning might involve, and how it differs from supervised learning where labels are provided. - Introduce unsupervised learning as a type of machine learning where the model learns patterns and structures from unlabeled data. 2. **Development** (30 minutes) a. Unsupervised Learning Overview: Explain that unsupervised learning involves analyzing and modeling data without any explicit labels. The goal is to uncover hidden structures or patterns in the data. Example: Customer segmentation in marketing, where the task is to group customers into segments based on purchasing behavior without predefined categories. b. Clustering: Discuss clustering as one of the most common unsupervised learning tasks. Explain that clustering algorithms group similar data points together based on their features. Example: Grouping news articles by topic without knowing the categories in advance. c. k-Means Clustering: Introduce the k-means clustering algorithm, which divides the data into k clusters by minimizing the variance within each cluster. Explain the steps of the k-means algorithm: randomly initializing centroids, assigning each point to the nearest centroid, updating centroids, and iterating until convergence. Discuss how the value of k is chosen (e.g., using the elbow method). Example: Applying k-means clustering to the Iris dataset to

	group flowers into clusters based on their features (e.g., petal length, width). d. Hierarchical Clustering: Introduce hierarchical clustering as another clustering technique that builds a hierarchy of clusters, either agglomeratively (bottom-up) or divisively (top-down). Discuss dendrograms as a visual representation of the clustering hierarchy and how to select the number of clusters by cutting the dendrogram at an appropriate level. Example: Hierarchical clustering of animals based on their physical characteristics. 3. Exercise (5 minutes) – - Assign students to work with a dataset (e.g., customer data or Iris dataset) and implement k-means clustering or PCA using scikit-learn. Ask them to explore the clusters or reduced features and interpret the results. - Afterward, have students present their findings and discuss the challenges they faced, particularly with selecting the number of clusters or interpreting principal components. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the key concepts of unsupervised learning, including clustering and dimensionality reduction, and how they differ from supervised learning tasks. 2. Highlight the importance of unsupervised learning in exploratory data analysis and discovering hidden patterns in data. 3. Encourage students to ask questions or share their thoughts on how unsupervised learning could be applied to real-world problems they are interested in. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "How does k-means clustering work, and what are its limitations?" and "When would you use PCA in a machine learning pipeline?" 3. Conduct an online quiz or coding exercise at the end of the class to assess students' understanding of unsupervised learning and their ability to implement basic models. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 5	Course Name: Artificial Intelligence with Computer Vision Topic: Reinforcement ML	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the core principles of reinforcement learning (RL) and how it differs from supervised and unsupervised learning. b. Explain the key concepts in RL: agent, environment, state, action, reward, and policy. c. Identify common reinforcement learning algorithms such as Q-learning and Deep Q Networks (DQN).
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a brainstorming session: Ask students what they think reinforcement learning might involve and how it differs from supervised or unsupervised learning. - Introduce reinforcement learning (RL) as a type of machine learning where an agent learns to make decisions by interacting with an environment to maximize cumulative rewards. 2. **Development** (30 minutes) a. Reinforcement Learning Overview: Explain that reinforcement learning involves an agent interacting with an environment in discrete time steps. The agent takes an action in the environment, receives feedback in the form of a reward, and transitions to a new state. The goal is to learn a policy that maximizes the cumulative reward over time. Example: A robot learning to navigate a maze by trial and error, receiving rewards for reaching the goal and penalties for hitting walls. b. Key Concepts: Agent: The decision-maker or learner in the environment. Environment: The external system with which the agent interacts. State: A representation of the current situation in the environment. Action: The set of possible moves or decisions the agent can take. Reward: Feedback from the environment indicating the success or failure of an action. Policy: A strategy or mapping from states to actions that the agent follows to maximize rewards. Example: In a game environment like CartPole, the agent's

	actions might be moving the cart left or right, and the reward could be a positive score for keeping the pole upright. 3. Exercise (5 minutes) – - Assign students to work with a reinforcement learning environment (e.g., CartPole or MountainCar from OpenAI Gym). Ask them to implement Q-learning or DQN and train an agent to solve the environment. - Afterward, have students present their results and discuss the challenges they faced, particularly in balancing exploration and exploitation. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the key concepts of reinforcement learning, including the agent-environment interaction, policy learning, and reward maximization. 2. Highlight the importance of exploration vs. exploitation and how RL is applied in real-world scenarios like robotics and autonomous systems. 3. Encourage students to ask questions or share their thoughts on how reinforcement learning could be applied to different problems they are interested in. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "What is the difference between Q-learning and Deep Q Networks?" and "How do agents balance exploration and exploitation in reinforcement learning?" 3. Conduct an online quiz or coding exercise at the end of the class to assess students' understanding of reinforcement learning and their ability to implement basic models. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 6	Course Name: Artificial Intelligence with Computer Vision Topic: Types of Problems Solved Using Machine Learning: Regression, Classification, Clustering	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the different types of machine learning problems: regression, classification, and clustering. b. Differentiate between the three types of problems and their respective approaches. c. Identify real-world scenarios where regression, classification, and clustering are used.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a brainstorming session: Ask students if they know the types of tasks machine learning can solve. Explain that machine learning is commonly used to solve problems in three categories: regression, classification, and clustering. - Provide an overview of how each problem type is defined and what distinguishes them from one another. 2. **Development** (30 minutes) a. Regression: Explain that regression is used for predicting continuous outcomes based on input features. Discuss how regression models map input data to continuous values. Example: Predicting house prices based on features like size, location, and number of bedrooms. Algorithm Example: Linear regression, where a straight line is fit to the data points to predict future values. Demonstrate how to implement a linear regression model using a dataset (e.g., Boston housing dataset) in Python with scikit-learn. b. Classification: Explain that classification is used for predicting discrete labels or categories based on input features. Discuss how classification models assign input data to predefined categories. Example: Classifying emails as spam or not spam based on the words they contain. Algorithm Example: Logistic regression or decision trees,

	where the model learns to classify data points into one of several categories. Demonstrate how to implement a classification model using a dataset (e.g., Iris dataset) in Python with scikit-learn. c. Clustering: Explain that clustering is an unsupervised learning task used to group similar data points together based on their features, without predefined labels. Discuss how clustering algorithms divide the data into meaningful groups or clusters. Example: Grouping customers into segments based on their purchasing behavior without predefined categories. Algorithm Example: k-means clustering, which assigns data points to k clusters based on feature similarity. 3. Exercise (5 minutes) – - Assign students to work with small datasets (one for each problem type) and implement basic machine learning models to solve regression, classification, and clustering problems. - Afterward, ask students to share their models and results, discussing how they approached each problem and the challenges they faced. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the differences between regression, classification, and clustering, highlighting the appropriate use cases for each type of problem. 2. Encourage students to reflect on real-world scenarios they are interested in and how these different types of machine learning tasks could be applied to solve problems in those areas. 3. Allow time for questions and further discussion on how to approach more complex tasks that may combine multiple problem types. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "How do you know if a problem is best approached as a classification task?" and "What is the key difference between clustering and classification?" 3. Conduct an online quiz or coding exercise at the end of the class to assess students' understanding of the different types of machine learning problems and their ability to implement basic models. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 7	Course Name: Artificial Intelligence with Computer Vision Topic: Exploratory Data Analysis on Real world Problem	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of Exploratory Data Analysis (EDA) and its role in the data science process. b. Perform EDA to discover patterns, relationships, and anomalies in real-world datasets. c. Use visualization and summary statistics to interpret and communicate findings from data.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a discussion: Ask students why they think understanding the data is crucial before applying machine learning models. - Introduce Exploratory Data Analysis (EDA) as a critical first step in the data science process, where the goal is to summarize the main characteristics of the data, often using visualizations and summary statistics. 2. **Development** (30 minutes) a. Overview of EDA: Explain that EDA involves inspecting the dataset to discover patterns, detect anomalies, test hypotheses, and check assumptions. Emphasize that EDA helps data scientists decide how to preprocess the data, which features to use, and what models might be appropriate for analysis. Example: Analyzing a sales dataset to determine seasonality trends and identify any outliers in sales performance. b. Steps of EDA: Data Loading and Inspection: Demonstrate how to load a real-world dataset using pandas and inspect its structure (e.g., checking for missing values, data types, and basic statistics). Example: Loading the Titanic dataset and using df.info() and df.describe() to understand the dataset structure and key statistics. Summary Statistics: Explain how to compute summary statistics (mean, median, mode, standard deviation) for continuous variables and frequency counts for categorical variables.

	Example: Calculating the average age and fare of Titanic passengers, as well as the distribution of passengers by class. Univariate Analysis: Show how to perform univariate analysis to explore individual features using histograms, boxplots, and bar charts. Example: Plotting the distribution of passenger ages on the Titanic using a histogram, or visualizing the survival rate by class using a bar chart. Bivariate Analysis: Explain how to explore relationships between two variables using scatter plots, correlation matrices, and cross-tabulation. Example: Analyzing the relationship between passenger age and fare using a scatter plot, or calculating the correlation between numerical features in a dataset. Multivariate Analysis: Introduce multivariate analysis techniques to explore relationships among multiple variables simultaneously, such as pair plots and heatmaps. Example: Creating a pair plot using seaborn to visualize relationships among multiple features (e.g., age, fare, and class) and using a heatmap to display the correlation matrix. Handling Missing Data and Outliers: Discuss strategies for handling missing data (e.g., imputation or removal) and detecting outliers using visualizations like boxplots. Example: Identifying missing values in the Titanic dataset and handling them appropriately, such as filling in missing ages with the median age. Data Visualization: Emphasize the importance of data visualization in EDA for uncovering patterns and communicating insights effectively. Show how to create visualizations using matplotlib and seaborn. Example: Visualizing the survival rate by gender and class on the Titanic using a bar plot. c. Real-World EDA Example: Walk through a real-world EDA example using a dataset (e.g., the Titanic dataset or a customer transaction dataset). Guide students through the steps of data inspection, summary statistics, univariate and bivariate analysis, and visualization. d. Demonstrate how insights from EDA inform decisions about data preprocessing and feature engineering before applying machine learning models. 3. Exercise (5 minutes) – - Assign students to perform EDA on a real-world dataset of their choice (e.g., sales data or customer data). Ask them to explore the dataset, visualize key features, and identify any patterns or anomalies.

	- Afterward, have students present their findings and discuss the insights they gained from EDA, including any potential issues they identified in the data (e.g., missing values, outliers). Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the key concepts of EDA, including the use of summary statistics and visualizations to understand and interpret data. 2. Emphasize that EDA is a crucial step in the data science process that guides data preprocessing and feature selection for further analysis. 3. Encourage students to ask questions or share their thoughts on how they would approach EDA for different types of data. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "Why is EDA important before building a machine learning model?" and "What insights can you gain from visualizing your data?" 3. Conduct an online quiz or EDA exercise at the end of the class to assess students' understanding of EDA and their ability to apply it to real-world datasets. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 8	Course Name: Artificial Intelligence with Computer Vision Topic: Exploratory Data Analysis on Real world Problem	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of Exploratory Data Analysis (EDA) and its role in the data science process. b. Perform EDA to discover patterns, relationships, and anomalies in real-world datasets. c. Use visualization and summary statistics to interpret and communicate findings from data.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a discussion: Ask students why they think understanding the data is crucial before applying machine learning models. - Introduce Exploratory Data Analysis (EDA) as a critical first step in the data science process, where the goal is to summarize the main characteristics of the data, often using visualizations and summary statistics. 2. **Development** (30 minutes) a. Overview of EDA: Explain that EDA involves inspecting the dataset to discover patterns, detect anomalies, test hypotheses, and check assumptions. Emphasize that EDA helps data scientists decide how to preprocess the data, which features to use, and what models might be appropriate for analysis. Example: Analyzing a sales dataset to determine seasonality trends and identify any outliers in sales performance. b. Steps of EDA: Data Loading and Inspection: Demonstrate how to load a real-world dataset using pandas and inspect its structure (e.g., checking for missing values, data types, and basic statistics). Example: Loading the Titanic dataset and using df.info() and df.describe() to understand the dataset structure and key statistics. Summary Statistics: Explain how to compute summary statistics (mean, median, mode, standard deviation) for continuous variables and frequency counts for categorical variables.

	Example: Calculating the average age and fare of Titanic passengers, as well as the distribution of passengers by class. Univariate Analysis: Show how to perform univariate analysis to explore individual features using histograms, boxplots, and bar charts. Example: Plotting the distribution of passenger ages on the Titanic using a histogram, or visualizing the survival rate by class using a bar chart. Bivariate Analysis: Explain how to explore relationships between two variables using scatter plots, correlation matrices, and cross-tabulation. Example: Analyzing the relationship between passenger age and fare using a scatter plot, or calculating the correlation between numerical features in a dataset. Multivariate Analysis: Introduce multivariate analysis techniques to explore relationships among multiple variables simultaneously, such as pair plots and heatmaps. Example: Creating a pair plot using seaborn to visualize relationships among multiple features (e.g., age, fare, and class) and using a heatmap to display the correlation matrix. Handling Missing Data and Outliers: Discuss strategies for handling missing data (e.g., imputation or removal) and detecting outliers using visualizations like boxplots. Example: Identifying missing values in the Titanic dataset and handling them appropriately, such as filling in missing ages with the median age. Data Visualization: Emphasize the importance of data visualization in EDA for uncovering patterns and communicating insights effectively. Show how to create visualizations using matplotlib and seaborn. Example: Visualizing the survival rate by gender and class on the Titanic using a bar plot. c. Real-World EDA Example: Walk through a real-world EDA example using a dataset (e.g., the Titanic dataset or a customer transaction dataset). Guide students through the steps of data inspection, summary statistics, univariate and bivariate analysis, and visualization. d. Demonstrate how insights from EDA inform decisions about data preprocessing and feature engineering before applying machine learning models. 3. Exercise (5 minutes) – - Assign students to perform EDA on a real-world dataset of their choice (e.g., sales data or customer data). Ask them to explore the dataset, visualize key features, and identify any patterns or anomalies.

	- Afterward, have students present their findings and discuss the insights they gained from EDA, including any potential issues they identified in the data (e.g., missing values, outliers). Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the key concepts of EDA, including the use of summary statistics and visualizations to understand and interpret data. 2. Emphasize that EDA is a crucial step in the data science process that guides data preprocessing and feature selection for further analysis. 3. Encourage students to ask questions or share their thoughts on how they would approach EDA for different types of data. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "Why is EDA important before building a machine learning model?" and "What insights can you gain from visualizing your data?" 3. Conduct an online quiz or EDA exercise at the end of the class to assess students' understanding of EDA and their ability to apply it to real-world datasets. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1	Course Name: Artificial Intelligence with Computer Vision Topic: Introduction to Image Processing, Computer Vision, and Computer Graphics	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the fundamental concepts of image processing, computer vision, and computer graphics. b. Distinguish between these three fields and their real-world applications. c. Identify common algorithms and techniques used in image processing and computer vision. d. Explain the role of computer graphics in simulation, rendering, and visualization.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Ask questions. what they know about the terms 'image processing,' 'computer vision,' and 'computer graphics.' - Present some real-world applications, such as face recognition, 3D modeling, and computer-generated imagery (CGI). - Briefly outline the objectives of the lesson. 2. **Development** (30 minutes) a. Image Processing: Introduce the basic concepts, such as image acquisition, filtering, enhancement, and segmentation. Show examples of digital image transformations. b. Computer Vision: Discuss how computer vision enables computers to interpret and make decisions based on visual input. Explain key techniques such as edge detection, object recognition, and motion analysis. c. Computer Graphics: Explain the fundamentals of rendering, 3D modeling, and animation. Discuss the role of computer graphics in video games, virtual reality, and simulations. d. The Optimality Principle - Define the principle: a route is optimal if all of its sub-paths are optimal. - Provide mathematical backing and theoretical examples. - Real-world example: Discuss how delivery services optimize routes for multiple deliveries. 3. Exercise (5 minutes) –

	- Engage students in a brief group exercise: Assign them the task of identifying real-world applications of image processing, computer vision, and computer graphics. - Ask students to share examples and briefly explain how each technology is applied in their examples. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the main points covered: definitions and distinctions between image processing, computer vision, and computer graphics. 2. Highlight key technologies and their real-world applications. 3. Allow students to ask questions and discuss any doubts or thoughts they may have. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Use reflective questions (What? Why? How?) to encourage deeper understanding and critical thinking. 3. Conduct an online quiz at the end of the class to assess knowledge retention. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2	Course Name: Artificial Intelligence with Computer Vision Topic: Computer Vision - Low-level, Mid-level, High-level	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the hierarchical structure of computer vision tasks: low-level, mid-level, and high-level vision. b. Differentiate between low-level, mid-level, and high-level vision tasks. c. Identify common algorithms and techniques used at each level of computer vision. d. Explain the importance of integrating low, mid, and high-level vision for complete scene understanding.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a brainstorming session: Ask students what they know about the different levels of tasks in computer vision. - Present real-world applications of computer vision that use low, mid, and high-level vision, such as autonomous vehicles or facial recognition systems. - Briefly outline the objectives of the lesson. 2. **Development** (30 minutes) a. Low-level Vision: Explain the fundamental tasks that involve direct interpretation of image data, such as edge detection, noise removal, and basic image filtering. Discuss how these processes are crucial for extracting raw features from images. Example Algorithms: Edge detection (Sobel, Canny), filtering (Gaussian blur). b. Mid-level Vision: Introduce the concept of grouping features and patterns to detect objects or regions in the image. Discuss tasks such as segmentation, feature extraction, and motion tracking. Example Algorithms: Region-based segmentation, Hough Transform, Optical Flow. c. High-level Vision: Discuss the process of interpreting objects and scenes, recognizing complex structures and making decisions based on the visual data. This includes tasks like object recognition, scene understanding, and activity recognition. Example Algorithms: Convolutional Neural Networks (CNNs)

	for object detection, R-CNN, YOLO. 3. Exercise (5 minutes) – - Divide students into small groups. Assign each group one level of computer vision (low, mid, or high). Have them identify real-world applications or systems that utilize their assigned vision level. - Afterward, ask them to present their findings briefly. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the main points covered: the definitions and differences between low, mid, and high-level vision tasks in computer vision. 2. Discuss how these levels work together to form complete visual processing systems in applications like self-driving cars or smart cameras. 3. Encourage students to ask questions or discuss their thoughts on the integration of these levels in complex systems. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions (What tasks fall under low-level vision? Why is high-level vision critical for decision-making in computer vision systems?). 3. Conduct an online quiz at the end of the class to assess knowledge retention. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 3	Course Name: Artificial Intelligence with Computer Vision Topic: Overview of Diverse Computer Vision Applications	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the wide range of applications of computer vision across different industries. b. Identify key applications of computer vision in fields such as healthcare, automotive, security, and entertainment. c. Explain the core computer vision techniques used in these applications. d. Appreciate the impact of computer vision on society and technology.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Begin with a brainstorming session: Ask students where they have encountered computer vision applications in their daily lives (e.g., facial recognition on smartphones, self-driving cars). - Provide a brief overview of the lesson's objectives and explain how computer vision impacts various industries. 2. **Development** (30 minutes) a. Healthcare Applications: Explain how computer vision is used in medical imaging for tasks like detecting tumors, analyzing X-rays, and assisting surgeries. Discuss techniques such as segmentation, pattern recognition, and deep learning. Example: Automated tumor detection using MRI scans. b. Automotive Industry: Discuss the role of computer vision in autonomous vehicles, such as lane detection, object recognition, and driver monitoring systems. Example: Tesla's Autopilot system using camera-based vision for autonomous driving. c. Security and Surveillance: Explore how computer vision is used in facial recognition systems, anomaly detection, and monitoring in security cameras. Example: Facial recognition in airports and public spaces for identifying individuals. d. Entertainment and Augmented Reality (AR): Discuss applications of computer vision in video games, virtual reality, and AR. Highlight how computer vision enhances user experiences by creating interactive environments. Example: Pokémon Go using computer vision for real-world

	object detection and interaction. e. Retail and E-commerce: Explain how computer vision enables personalized shopping experiences, product recognition, and automated checkout systems. Example: Amazon Go stores using computer vision to eliminate traditional checkout processes. f. Provide short video clips or software demonstrations illustrating these applications in action. 3. Exercise (5 minutes) – - Have students work in pairs to identify one specific industry application of computer vision that was not covered in the lesson. They should research and present their findings briefly, explaining the core technology behind it. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the various applications discussed, emphasizing the diversity and significance of computer vision in different fields. 2. Encourage students to reflect on how computer vision might evolve and impact other areas in the future. 3. Allow students to ask questions or share their thoughts on the most interesting applications. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Ask reflective questions like: "Which industry do you think benefits the most from computer vision technology?" 3. Conduct an online quiz or discussion at the end of the class to evaluate knowledge retention and encourage active participation. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 4	Course Name: Artificial Intelligence with Computer Vision Topic: Light-Models	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the basic principles of light models in computer graphics. b. Differentiate between various light models such as ambient, diffuse, and specular lighting. c. Explain how light models are used to create realistic rendering in 3D environments. d. Identify key algorithms and techniques for simulating lighting in computer graphics.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Begin with a brainstorming session: Ask students how light affects objects in the real world and what factors determine how objects appear to the human eye. - Briefly introduce the topic of light models and explain that they are crucial in computer graphics for simulating realistic lighting and shading. 2. **Development** (30 minutes) a. Ambient Lighting: Introduce the concept of ambient light as the simplest form of lighting. Explain how it simulates the uniform light that comes from no particular direction and affects the overall brightness of a scene. Example: Basic ambient lighting in a 3D game engine. b. Diffuse Lighting: Explain how diffuse lighting simulates light that hits an object and scatters in all directions. Discuss its importance in giving objects a matte appearance, where the intensity of light depends on the angle between the light source and the surface. Example: Diffuse lighting on rough surfaces like walls or fabric. c. Specular Lighting: Introduce the concept of specular lighting, which models the reflection of light from shiny surfaces. Explain how this type of lighting creates highlights and simulates reflections that change depending on the viewer's position. Example: Specular highlights on a polished metal surface or water. d. Phong Reflection Model: Discuss the Phong reflection model

	as a combination of ambient, diffuse, and specular lighting. Explain its role in creating realistic shading in computer-generated imagery (CGI). Example: The Phong model applied in rendering a 3D object with light reflecting from different angles. e. Demonstrate these lighting techniques using a 3D rendering software like Blender or Unity, showing how they affect objects in a scene. 3. Exercise (5 minutes) – - Assign students to work in pairs to explore the effects of changing light parameters in a simple 3D scene. They can use a tool like Blender or Unity to adjust ambient, diffuse, and specular lighting settings and observe the changes. - Ask them to briefly present the effects of these changes on the appearance of objects in their scene. Use Nearpod to collect responses and discuss the answers.
Closure	1. Assign students to work in pairs to explore the effects of changing light parameters in a simple 3D scene. They can use a tool like Blender or Unity to adjust ambient, diffuse, and specular lighting settings and observe the changes. 2. Ask them to briefly present the effects of these changes on the appearance of objects in their scene. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assign students to work in pairs to explore the effects of changing light parameters in a simple 3D scene. They can use a tool like Blender or Unity to adjust ambient, diffuse, and specular lighting settings and observe the changes. 2. Ask them to briefly present the effects of these changes on the appearance of objects in their scene. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 5	Course Name: Artificial Intelligence with Computer Vision Topic: Image Acquisition and Backgrounds: Adaptive Background Subtraction	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of image acquisition and its importance in computer vision. b. Explain the concept of background subtraction and its use in motion detection. c. Describe the adaptive background subtraction technique and how it improves performance in dynamic environments. d. Implement basic adaptive background subtraction algorithms in a computer vision framework.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a brief brainstorming session: Ask students how cameras capture images and why background detection is crucial in applications like surveillance or video conferencing. - Provide an overview of the lesson's objectives, introducing the key concepts of image acquisition and background subtraction. 2. **Development** (30 minutes) a. Image Acquisition: Introduce the process of capturing images using cameras or sensors. Discuss different types of image acquisition devices such as digital cameras, webcams, and thermal cameras, and explain how images are digitized. Example: Explain how a digital camera captures light and converts it into a pixelated image that can be processed by computer vision algorithms. b. Background Subtraction: Introduce the concept of background subtraction in the context of detecting moving objects in a scene. Explain how background subtraction helps in identifying foreground objects by comparing each new frame to a background model. Example: Simple background subtraction in a video stream to detect moving objects such as people or cars. c. Adaptive Background Subtraction: Explain the limitations of static background subtraction methods, especially in dynamic environments where lighting conditions change or the background is not entirely static. Introduce adaptive background subtraction as a solution that continuously updates

	the background model based on changes in the scene. Example Algorithms: Gaussian Mixture Model (GMM), Running Average Method. 3. Exercise (5 minutes) – - Assign students a small coding task where they implement a simple background subtraction algorithm using OpenCV. Ask them to compare the results of static and adaptive methods on a video feed with changing backgrounds. - Afterward, ask them to share their observations and challenges encountered during implementation. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the concepts of image acquisition and background subtraction, emphasizing the need for adaptive techniques in dynamic environments. 2. Discuss real-world applications of adaptive background subtraction, such as in video surveillance and autonomous vehicle systems. 3. Encourage students to ask questions or share their thoughts on how background subtraction can be improved further. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "Why is adaptive background subtraction necessary in dynamic environments?" 3. Conduct an online quiz or demonstration at the end of the class to evaluate understanding and practical application of the concept. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 6	Course Name: Artificial Intelligence with Computer Vision Topic: Camera Geometry and Calibration	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the basic principles of camera geometry in computer vision. b. Explain the concepts of intrinsic and extrinsic camera parameters. c. Describe the process of camera calibration and its importance in accurate 3D reconstruction. d. Implement basic camera calibration techniques using common tools.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a brainstorming session: Ask students how cameras capture 3D scenes in 2D images and why understanding camera geometry is crucial for computer vision applications such as 3D reconstruction and robotics. - Provide a brief overview of the lesson's objectives, introducing the key concepts of camera geometry and calibration. 2. **Development** (30 minutes) a. Camera Geometry: Introduce the concept of camera geometry, focusing on how cameras project a 3D scene onto a 2D image plane. Explain the pinhole camera model, which is a common model for understanding this projection. Example: Show how the camera's focal length and field of view determine how the 3D world is captured as a 2D image. b. Intrinsic Parameters: Explain intrinsic parameters, which define the internal characteristics of the camera such as focal length, optical center, and distortion coefficients. These parameters are crucial for understanding how the camera interprets the world. Example: Discuss how camera lens distortion affects the final image and how it can be corrected using intrinsic parameters. c. Extrinsic Parameters: Introduce extrinsic parameters, which describe the position and orientation of the camera in the world. These parameters help map the relationship between the camera's coordinate system and the world's coordinate system. Example: Show how extrinsic parameters affect the camera's viewpoint and orientation in a 3D scene. d. Camera Calibration: Discuss the process of camera calibration, which is used to estimate both intrinsic and extrinsic

	parameters. Explain how calibration corrects for distortion and ensures that the camera accurately captures the geometry of the scene. Example Techniques: Demonstrate camera calibration using a calibration pattern (e.g., a chessboard) and tools like OpenCV or MATLAB's calibration functions. e. Demonstration: Show a practical implementation of camera calibration using OpenCV. Guide students through the steps of capturing calibration images, detecting key points, and estimating camera parameters. 3. Exercise (5 minutes) – - Assign students to work with a set of calibration images and implement the camera calibration process using OpenCV or MATLAB. Ask them to compute the intrinsic and extrinsic parameters for their camera and compare the corrected image with the original. - Afterward, ask them to share their observations and challenges encountered during the calibration process. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the main points of camera geometry and calibration, emphasizing the importance of accurate calibration for computer vision tasks such as 3D reconstruction and augmented reality. 2. Discuss how calibration is used in various real-world applications like autonomous vehicles, robotic navigation, and AR/VR. 3. Encourage students to ask questions or share their thoughts on challenges in camera calibration and potential solutions. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "Why is camera calibration crucial for accurate 3D reconstruction?" 3. Conduct an online quiz or demonstration at the end of the class to evaluate understanding and practical application of camera geometry and calibration. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 7	Course Name: Artificial Intelligence with Computer Vision Topic: Stereo Interest Operators	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of stereo vision and how stereo interest operators are used in computer vision. b. Explain the role of interest operators in detecting and matching key points across stereo image pairs. c. Identify common stereo interest operators and their use in depth estimation and 3D reconstruction. d. Implement basic stereo matching techniques using stereo interest operators in a computer vision framework.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Begin with a brainstorming session: Ask students how our eyes work together to perceive depth and how this concept is translated into computer vision systems. - Introduce stereo vision and the importance of detecting corresponding points in two images to estimate depth. Briefly outline the objectives of the lesson. 2. **Development** (30 minutes) a. Stereo Vision: Introduce the concept of stereo vision, explaining how two cameras positioned at different angles capture slightly different images, which can be used to estimate depth and create 3D reconstructions. Example: Explain how stereo vision is used in robotics, self-driving cars, and 3D modeling to perceive the world in three dimensions. b. Interest Operators: Explain what interest operators are—algorithms that detect keypoints or corners in images that are distinctive and can be reliably matched across multiple images. Discuss the importance of these operators in stereo vision for finding corresponding points between the left and right images. Example: Corners, edges, and blobs in images that serve as keypoints. c. Common Stereo Interest Operators: Introduce common stereo interest operators such as Harris Corner Detector, Scale-Invariant Feature Transform (SIFT), and Speeded-Up Robust Features (SURF). Explain how these operators detect keypoints that are stable under different lighting conditions, scale, and rotation.

	d. Harris Corner Detector: Discuss how this operator detects corners, which are strong interest points in images. e. Stereo Matching: Discuss how stereo interest operators are used in stereo matching algorithms to find corresponding points between two images. Explain the process of finding disparities between matched points to calculate depth. Example: Implement stereo matching using OpenCV's stereo matching functions, showing how depth maps are generated from stereo image pairs. 3. Exercise (5 minutes) – - Ask students to compare the effectiveness of different interest operators in terms of accuracy and speed. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the main points covered: stereo vision, the role of interest operators, and their application in stereo matching and depth estimation. 2. Discuss real-world applications of stereo interest operators, such as in augmented reality, autonomous vehicles, and medical imaging. 3. Allow students to ask questions or share their thoughts on the challenges and potential improvements in stereo matching techniques. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "How do stereo interest operators contribute to accurate depth estimation in computer vision?" 3. Conduct an online quiz or coding exercise at the end of the class to evaluate students' understanding and practical application of stereo interest operators. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 8	Course Name: Artificial Intelligence with Computer Vision Topic: Feature Extraction.	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of feature extraction in computer vision and machine learning. b. Explain the importance of feature extraction in reducing the dimensionality of data while preserving essential information. c. Identify common feature extraction techniques for images and signals. d. Implement basic feature extraction methods using a computer vision framework.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Begin with a brainstorming session: Ask students what they understand by the term "feature" and how features can be used to describe data in a machine-readable form. - Introduce the concept of feature extraction and explain how it is a critical step in many computer vision and machine learning tasks, such as object recognition, classification, and clustering. 2. **Development** (30 minutes) a. Definition of Features: Explain what features are—distinctive attributes, measurements, or patterns extracted from raw data that help in identifying or classifying objects. Discuss features in different contexts, such as edges and corners in images, or frequency components in signals. b. Example: Discuss common image features such as edges, textures, shapes, and colors. c. Importance of Feature Extraction: Explain why feature extraction is crucial for reducing the dimensionality of raw data while retaining relevant information. Discuss how it simplifies the learning process and improves model performance by focusing on the most important characteristics of the data. Example: Reducing a high-resolution image to a set of edges for object detection in a machine learning model. d. Feature Extraction Techniques: Introduce common techniques used in feature extraction, both for images and other types of data 3. Exercise (5 minutes) –

	- Assign students to work with a sample image and implement two feature extraction techniques, such as edge detection and corner detection, using OpenCV. Ask them to experiment with different algorithms and compare the results. - Afterward, have students present their observations on the effectiveness of the chosen techniques. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the main points covered: the importance of feature extraction, different techniques, and their applications in computer vision and machine learning. 2. Discuss real-world applications of feature extraction in fields such as object detection, face recognition, and pattern recognition. 3. Encourage students to ask questions or share their thoughts on how feature extraction can be optimized or adapted to different data types. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "Why is feature extraction important for reducing data dimensionality?" and "How does feature extraction improve machine learning models?" 3. Conduct an online quiz or coding exercise at the end of the class to assess understanding and practical application of feature extraction techniques. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1	Course Name: Artificial Intelligence with Computer Vision Topic: Introduction to Image Processing, Computer Vision, and Computer Graphics	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the fundamental concepts of image processing, computer vision, and computer graphics. b. Distinguish between these three fields and their real-world applications. c. Identify common algorithms and techniques used in image processing and computer vision. d. Explain the role of computer graphics in simulation, rendering, and visualization.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Ask questions. what they know about the terms 'image processing,' 'computer vision,' and 'computer graphics.' - Present some real-world applications, such as face recognition, 3D modeling, and computer-generated imagery (CGI). - Briefly outline the objectives of the lesson. 2. **Development** (30 minutes) a. Image Processing: Introduce the basic concepts, such as image acquisition, filtering, enhancement, and segmentation. Show examples of digital image transformations. b. Computer Vision: Discuss how computer vision enables computers to interpret and make decisions based on visual input. Explain key techniques such as edge detection, object recognition, and motion analysis. c. Computer Graphics: Explain the fundamentals of rendering, 3D modeling, and animation. Discuss the role of computer graphics in video games, virtual reality, and simulations. d. The Optimality Principle - Define the principle: a route is optimal if all of its sub-paths are optimal. - Provide mathematical backing and theoretical examples. - Real-world example: Discuss how delivery services optimize routes for multiple deliveries. 3. Exercise (5 minutes) –

	- Engage students in a brief group exercise: Assign them the task of identifying real-world applications of image processing, computer vision, and computer graphics. - Ask students to share examples and briefly explain how each technology is applied in their examples. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the main points covered: definitions and distinctions between image processing, computer vision, and computer graphics. 2. Highlight key technologies and their real-world applications. 3. Allow students to ask questions and discuss any doubts or thoughts they may have. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Use reflective questions (What? Why? How?) to encourage deeper understanding and critical thinking. 3. Conduct an online quiz at the end of the class to assess knowledge retention. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2	Course Name: Artificial Intelligence with Computer Vision Topic: Computer Vision - Low-level, Mid-level, High-level	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the hierarchical structure of computer vision tasks: low-level, mid-level, and high-level vision. b. Differentiate between low-level, mid-level, and high-level vision tasks. c. Identify common algorithms and techniques used at each level of computer vision. d. Explain the importance of integrating low, mid, and high-level vision for complete scene understanding.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a brainstorming session: Ask students what they know about the different levels of tasks in computer vision. - Present real-world applications of computer vision that use low, mid, and high-level vision, such as autonomous vehicles or facial recognition systems. - Briefly outline the objectives of the lesson. 2. **Development** (30 minutes) a. Low-level Vision: Explain the fundamental tasks that involve direct interpretation of image data, such as edge detection, noise removal, and basic image filtering. Discuss how these processes are crucial for extracting raw features from images. Example Algorithms: Edge detection (Sobel, Canny), filtering (Gaussian blur). b. Mid-level Vision: Introduce the concept of grouping features and patterns to detect objects or regions in the image. Discuss tasks such as segmentation, feature extraction, and motion tracking. Example Algorithms: Region-based segmentation, Hough Transform, Optical Flow. c. High-level Vision: Discuss the process of interpreting objects and scenes, recognizing complex structures and making decisions based on the visual data. This includes tasks like object recognition, scene understanding, and activity recognition. Example Algorithms: Convolutional Neural Networks (CNNs)

	for object detection, R-CNN, YOLO. 3. Exercise (5 minutes) – - Divide students into small groups. Assign each group one level of computer vision (low, mid, or high). Have them identify real-world applications or systems that utilize their assigned vision level. - Afterward, ask them to present their findings briefly. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the main points covered: the definitions and differences between low, mid, and high-level vision tasks in computer vision. 2. Discuss how these levels work together to form complete visual processing systems in applications like self-driving cars or smart cameras. 3. Encourage students to ask questions or discuss their thoughts on the integration of these levels in complex systems. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions (What tasks fall under low-level vision? Why is high-level vision critical for decision-making in computer vision systems?). 3. Conduct an online quiz at the end of the class to assess knowledge retention. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 3	Course Name: Artificial Intelligence with Computer Vision Topic: Overview of Diverse Computer Vision Applications	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the wide range of applications of computer vision across different industries. b. Identify key applications of computer vision in fields such as healthcare, automotive, security, and entertainment. c. Explain the core computer vision techniques used in these applications. d. Appreciate the impact of computer vision on society and technology.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Begin with a brainstorming session: Ask students where they have encountered computer vision applications in their daily lives (e.g., facial recognition on smartphones, self-driving cars). - Provide a brief overview of the lesson's objectives and explain how computer vision impacts various industries. 2. **Development** (30 minutes) a. Healthcare Applications: Explain how computer vision is used in medical imaging for tasks like detecting tumors, analyzing X-rays, and assisting surgeries. Discuss techniques such as segmentation, pattern recognition, and deep learning. Example: Automated tumor detection using MRI scans. b. Automotive Industry: Discuss the role of computer vision in autonomous vehicles, such as lane detection, object recognition, and driver monitoring systems. Example: Tesla's Autopilot system using camera-based vision for autonomous driving. c. Security and Surveillance: Explore how computer vision is used in facial recognition systems, anomaly detection, and monitoring in security cameras. Example: Facial recognition in airports and public spaces for identifying individuals. d. Entertainment and Augmented Reality (AR): Discuss applications of computer vision in video games, virtual reality, and AR. Highlight how computer vision enhances user experiences by creating interactive environments. Example: Pokémon Go using computer vision for real-world

	object detection and interaction. e. Retail and E-commerce: Explain how computer vision enables personalized shopping experiences, product recognition, and automated checkout systems. Example: Amazon Go stores using computer vision to eliminate traditional checkout processes. f. Provide short video clips or software demonstrations illustrating these applications in action. 3. Exercise (5 minutes) – - Have students work in pairs to identify one specific industry application of computer vision that was not covered in the lesson. They should research and present their findings briefly, explaining the core technology behind it. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the various applications discussed, emphasizing the diversity and significance of computer vision in different fields. 2. Encourage students to reflect on how computer vision might evolve and impact other areas in the future. 3. Allow students to ask questions or share their thoughts on the most interesting applications. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Ask reflective questions like: "Which industry do you think benefits the most from computer vision technology?" 3. Conduct an online quiz or discussion at the end of the class to evaluate knowledge retention and encourage active participation. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 4	Course Name: Artificial Intelligence with Computer Vision Topic: Light-Models	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the basic principles of light models in computer graphics. b. Differentiate between various light models such as ambient, diffuse, and specular lighting. c. Explain how light models are used to create realistic rendering in 3D environments. d. Identify key algorithms and techniques for simulating lighting in computer graphics.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Begin with a brainstorming session: Ask students how light affects objects in the real world and what factors determine how objects appear to the human eye. - Briefly introduce the topic of light models and explain that they are crucial in computer graphics for simulating realistic lighting and shading. 2. **Development** (30 minutes) a. Ambient Lighting: Introduce the concept of ambient light as the simplest form of lighting. Explain how it simulates the uniform light that comes from no particular direction and affects the overall brightness of a scene. Example: Basic ambient lighting in a 3D game engine. b. Diffuse Lighting: Explain how diffuse lighting simulates light that hits an object and scatters in all directions. Discuss its importance in giving objects a matte appearance, where the intensity of light depends on the angle between the light source and the surface. Example: Diffuse lighting on rough surfaces like walls or fabric. c. Specular Lighting: Introduce the concept of specular lighting, which models the reflection of light from shiny surfaces. Explain how this type of lighting creates highlights and simulates reflections that change depending on the viewer's position. Example: Specular highlights on a polished metal surface or water. d. Phong Reflection Model: Discuss the Phong reflection model

	as a combination of ambient, diffuse, and specular lighting. Explain its role in creating realistic shading in computer-generated imagery (CGI). Example: The Phong model applied in rendering a 3D object with light reflecting from different angles. e. Demonstrate these lighting techniques using a 3D rendering software like Blender or Unity, showing how they affect objects in a scene. 3. Exercise (5 minutes) – - Assign students to work in pairs to explore the effects of changing light parameters in a simple 3D scene. They can use a tool like Blender or Unity to adjust ambient, diffuse, and specular lighting settings and observe the changes. - Ask them to briefly present the effects of these changes on the appearance of objects in their scene. Use Nearpod to collect responses and discuss the answers.
Closure	1. Assign students to work in pairs to explore the effects of changing light parameters in a simple 3D scene. They can use a tool like Blender or Unity to adjust ambient, diffuse, and specular lighting settings and observe the changes. 2. Ask them to briefly present the effects of these changes on the appearance of objects in their scene. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assign students to work in pairs to explore the effects of changing light parameters in a simple 3D scene. They can use a tool like Blender or Unity to adjust ambient, diffuse, and specular lighting settings and observe the changes. 2. Ask them to briefly present the effects of these changes on the appearance of objects in their scene. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 5	Course Name: Artificial Intelligence with Computer Vision Topic: Image Acquisition and Backgrounds: Adaptive Background Subtraction	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of image acquisition and its importance in computer vision. b. Explain the concept of background subtraction and its use in motion detection. c. Describe the adaptive background subtraction technique and how it improves performance in dynamic environments. d. Implement basic adaptive background subtraction algorithms in a computer vision framework.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a brief brainstorming session: Ask students how cameras capture images and why background detection is crucial in applications like surveillance or video conferencing. - Provide an overview of the lesson's objectives, introducing the key concepts of image acquisition and background subtraction. 2. **Development** (30 minutes) a. Image Acquisition: Introduce the process of capturing images using cameras or sensors. Discuss different types of image acquisition devices such as digital cameras, webcams, and thermal cameras, and explain how images are digitized. Example: Explain how a digital camera captures light and converts it into a pixelated image that can be processed by computer vision algorithms. b. Background Subtraction: Introduce the concept of background subtraction in the context of detecting moving objects in a scene. Explain how background subtraction helps in identifying foreground objects by comparing each new frame to a background model. Example: Simple background subtraction in a video stream to detect moving objects such as people or cars. c. Adaptive Background Subtraction: Explain the limitations of static background subtraction methods, especially in dynamic environments where lighting conditions change or the background is not entirely static. Introduce adaptive background subtraction as a solution that continuously updates

	the background model based on changes in the scene. Example Algorithms: Gaussian Mixture Model (GMM), Running Average Method. 3. Exercise (5 minutes) – - Assign students a small coding task where they implement a simple background subtraction algorithm using OpenCV. Ask them to compare the results of static and adaptive methods on a video feed with changing backgrounds. - Afterward, ask them to share their observations and challenges encountered during implementation. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the concepts of image acquisition and background subtraction, emphasizing the need for adaptive techniques in dynamic environments. 2. Discuss real-world applications of adaptive background subtraction, such as in video surveillance and autonomous vehicle systems. 3. Encourage students to ask questions or share their thoughts on how background subtraction can be improved further. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "Why is adaptive background subtraction necessary in dynamic environments?" 3. Conduct an online quiz or demonstration at the end of the class to evaluate understanding and practical application of the concept. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 6	Course Name: Artificial Intelligence with Computer Vision Topic: Camera Geometry and Calibration	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the basic principles of camera geometry in computer vision. b. Explain the concepts of intrinsic and extrinsic camera parameters. c. Describe the process of camera calibration and its importance in accurate 3D reconstruction. d. Implement basic camera calibration techniques using common tools.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Start with a brainstorming session: Ask students how cameras capture 3D scenes in 2D images and why understanding camera geometry is crucial for computer vision applications such as 3D reconstruction and robotics. - Provide a brief overview of the lesson's objectives, introducing the key concepts of camera geometry and calibration. 2. **Development** (30 minutes) a. Camera Geometry: Introduce the concept of camera geometry, focusing on how cameras project a 3D scene onto a 2D image plane. Explain the pinhole camera model, which is a common model for understanding this projection. Example: Show how the camera's focal length and field of view determine how the 3D world is captured as a 2D image. b. Intrinsic Parameters: Explain intrinsic parameters, which define the internal characteristics of the camera such as focal length, optical center, and distortion coefficients. These parameters are crucial for understanding how the camera interprets the world. Example: Discuss how camera lens distortion affects the final image and how it can be corrected using intrinsic parameters. c. Extrinsic Parameters: Introduce extrinsic parameters, which describe the position and orientation of the camera in the world. These parameters help map the relationship between the camera's coordinate system and the world's coordinate system. Example: Show how extrinsic parameters affect the camera's viewpoint and orientation in a 3D scene. d. Camera Calibration: Discuss the process of camera calibration, which is used to estimate both intrinsic and extrinsic

	parameters. Explain how calibration corrects for distortion and ensures that the camera accurately captures the geometry of the scene. Example Techniques: Demonstrate camera calibration using a calibration pattern (e.g., a chessboard) and tools like OpenCV or MATLAB's calibration functions. e. Demonstration: Show a practical implementation of camera calibration using OpenCV. Guide students through the steps of capturing calibration images, detecting key points, and estimating camera parameters. 3. Exercise (5 minutes) – - Assign students to work with a set of calibration images and implement the camera calibration process using OpenCV or MATLAB. Ask them to compute the intrinsic and extrinsic parameters for their camera and compare the corrected image with the original. - Afterward, ask them to share their observations and challenges encountered during the calibration process. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the main points of camera geometry and calibration, emphasizing the importance of accurate calibration for computer vision tasks such as 3D reconstruction and augmented reality. 2. Discuss how calibration is used in various real-world applications like autonomous vehicles, robotic navigation, and AR/VR. 3. Encourage students to ask questions or share their thoughts on challenges in camera calibration and potential solutions. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "Why is camera calibration crucial for accurate 3D reconstruction?" 3. Conduct an online quiz or demonstration at the end of the class to evaluate understanding and practical application of camera geometry and calibration. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 7	Course Name: Artificial Intelligence with Computer Vision Topic: Stereo Interest Operators	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of stereo vision and how stereo interest operators are used in computer vision. b. Explain the role of interest operators in detecting and matching key points across stereo image pairs. c. Identify common stereo interest operators and their use in depth estimation and 3D reconstruction. d. Implement basic stereo matching techniques using stereo interest operators in a computer vision framework.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Begin with a brainstorming session: Ask students how our eyes work together to perceive depth and how this concept is translated into computer vision systems. - Introduce stereo vision and the importance of detecting corresponding points in two images to estimate depth. Briefly outline the objectives of the lesson. 2. **Development** (30 minutes) a. Stereo Vision: Introduce the concept of stereo vision, explaining how two cameras positioned at different angles capture slightly different images, which can be used to estimate depth and create 3D reconstructions. Example: Explain how stereo vision is used in robotics, self-driving cars, and 3D modeling to perceive the world in three dimensions. b. Interest Operators: Explain what interest operators are—algorithms that detect keypoints or corners in images that are distinctive and can be reliably matched across multiple images. Discuss the importance of these operators in stereo vision for finding corresponding points between the left and right images. Example: Corners, edges, and blobs in images that serve as keypoints. c. Common Stereo Interest Operators: Introduce common stereo interest operators such as Harris Corner Detector, Scale-Invariant Feature Transform (SIFT), and Speeded-Up Robust Features (SURF). Explain how these operators detect keypoints that are stable under different lighting conditions, scale, and rotation.

	d. Harris Corner Detector: Discuss how this operator detects corners, which are strong interest points in images. e. Stereo Matching: Discuss how stereo interest operators are used in stereo matching algorithms to find corresponding points between two images. Explain the process of finding disparities between matched points to calculate depth. Example: Implement stereo matching using OpenCV's stereo matching functions, showing how depth maps are generated from stereo image pairs. 3. Exercise (5 minutes) – - Ask students to compare the effectiveness of different interest operators in terms of accuracy and speed. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the main points covered: stereo vision, the role of interest operators, and their application in stereo matching and depth estimation. 2. Discuss real-world applications of stereo interest operators, such as in augmented reality, autonomous vehicles, and medical imaging. 3. Allow students to ask questions or share their thoughts on the challenges and potential improvements in stereo matching techniques. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "How do stereo interest operators contribute to accurate depth estimation in computer vision?" 3. Conduct an online quiz or coding exercise at the end of the class to evaluate students' understanding and practical application of stereo interest operators. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 8	Course Name: Artificial Intelligence with Computer Vision Topic: Feature Extraction.	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of feature extraction in computer vision and machine learning. b. Explain the importance of feature extraction in reducing the dimensionality of data while preserving essential information. c. Identify common feature extraction techniques for images and signals. d. Implement basic feature extraction methods using a computer vision framework.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Begin with a brainstorming session: Ask students what they understand by the term "feature" and how features can be used to describe data in a machine-readable form. - Introduce the concept of feature extraction and explain how it is a critical step in many computer vision and machine learning tasks, such as object recognition, classification, and clustering. 2. **Development** (30 minutes) a. Definition of Features: Explain what features are—distinctive attributes, measurements, or patterns extracted from raw data that help in identifying or classifying objects. Discuss features in different contexts, such as edges and corners in images, or frequency components in signals. b. Example: Discuss common image features such as edges, textures, shapes, and colors. c. Importance of Feature Extraction: Explain why feature extraction is crucial for reducing the dimensionality of raw data while retaining relevant information. Discuss how it simplifies the learning process and improves model performance by focusing on the most important characteristics of the data. Example: Reducing a high-resolution image to a set of edges for object detection in a machine learning model. d. Feature Extraction Techniques: Introduce common techniques used in feature extraction, both for images and other types of data 3. Exercise (5 minutes) –

	- Assign students to work with a sample image and implement two feature extraction techniques, such as edge detection and corner detection, using OpenCV. Ask them to experiment with different algorithms and compare the results. - Afterward, have students present their observations on the effectiveness of the chosen techniques. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the main points covered: the importance of feature extraction, different techniques, and their applications in computer vision and machine learning. 2. Discuss real-world applications of feature extraction in fields such as object detection, face recognition, and pattern recognition. 3. Encourage students to ask questions or share their thoughts on how feature extraction can be optimized or adapted to different data types. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "Why is feature extraction important for reducing data dimensionality?" and "How does feature extraction improve machine learning models?" 3. Conduct an online quiz or coding exercise at the end of the class to assess understanding and practical application of feature extraction techniques. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1	Course Name: Artificial Intelligence with Computer Vision Topic: Recognition: Space-Time Methods	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of space-time methods in recognition tasks for computer vision. b. Explain how space-time features are extracted from videos or dynamic scenes. c. Identify common space-time recognition methods and their use in tasks like action recognition and motion analysis. d. Implement a basic space-time feature extraction and recognition technique using a computer vision framework.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Ask questions. how they think computers recognize actions or events in videos. Discuss how recognition in dynamic scenes requires understanding changes over time, in addition to spatial features. - Introduce the concept of space-time methods and how they are applied in recognition tasks such as action recognition, gesture detection, and motion analysis. 2. **Development** (30 minutes) a. Space-Time Features: Explain the concept of space-time features, which combine spatial and temporal information from a video or dynamic scene. Discuss how these features capture motion and object behavior over time. Example: Explain how space-time features are used to detect actions like walking, jumping, or hand gestures in a video. b. Space-Time Interest Points (STIP): Introduce the idea of space-time interest points, which are key points in a video that have significant changes in both space and time. Explain how STIP detectors identify these points for further analysis. Example: Detecting changes in pixel intensities over time to identify motion in a video sequence. c. Recognition Methods: Discuss common space-time recognition methods. 3. Exercise (5 minutes) – - Assign students to work with a sample video dataset and implement a space-time feature extraction technique such as HOF or STIP. Have them analyze the extracted features and use

	them to classify simple actions in the video. - Ask students to present their findings and discuss the effectiveness of the chosen method for recognizing actions. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the key concepts of space-time features, interest points, and recognition methods. 2. Discuss real-world applications of space-time methods in fields such as video surveillance, sports analysis, and human-computer interaction. 3. Encourage students to ask questions or share their thoughts on the challenges and future developments of space-time recognition methods. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "How do space-time methods help in recognizing actions in dynamic scenes?" and "What are the advantages of using 3D CNNs for action recognition?" 3. Conduct an online quiz or practical coding exercise at the end of the class to evaluate students' understanding of space-time methods and their application in video recognition. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2	Course Name: Artificial Intelligence with Computer Vision Topic: Discriminant Analysis	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of space-time methods in recognition tasks Understand the concept of Discriminant Analysis in machine learning and statistics. b. Differentiate between Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA). c. Explain how Discriminant Analysis is used for classification tasks.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Ask questions. what they know about classification methods in machine learning and when it is important to classify data into distinct categories. - Provide an overview of Discriminant Analysis as a classification technique and explain that it is used for determining the decision boundary that best separates different classes. 2. **Development** (30 minutes) a. Overview of Discriminant Analysis: Explain the purpose of Discriminant Analysis, which is to find a linear combination of features that best separates two or more classes of objects or events. Example: Classifying species of flowers based on features such as petal length and width using the Iris dataset. b. Linear Discriminant Analysis (LDA): Introduce LDA as a method that assumes each class has the same covariance matrix and finds a linear boundary that separates the classes. Explain how LDA works by maximizing the ratio of between-class variance to within-class variance. Example: Using LDA to classify images into categories like "cat" or "dog" based on pixel values. c. Quadratic Discriminant Analysis (QDA): Introduce QDA as a variant of LDA that does not assume identical covariance matrices for each class. Explain that QDA fits quadratic boundaries for classification tasks. Example: Classifying objects when the classes have different shapes in feature space, such as spherical versus elliptical clusters.

	d. Applications of Discriminant Analysis: Discuss real-world applications of LDA and QDA, including face recognition, disease diagnosis, and fraud detection. Example: LDA used in credit scoring to classify individuals as low or high credit risk. 3. Exercise (5 minutes) – - Assign students to implement LDA and QDA on a dataset of their choice (e.g., Iris dataset). Ask them to fit the models, classify the data, and compare the performance of LDA and QDA on the task. - Afterward, have students present their results and discuss which method performed better and why. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the key differences between LDA and QDA and their use cases in classification tasks. 2. Discuss when to choose LDA versus QDA depending on the dataset and problem at hand. 3. Encourage students to ask questions or share their thoughts on the importance of discriminant analysis in machine learning. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "What assumptions does LDA make about the data?" and "When would QDA be more suitable than LDA for classification?" 3. Conduct an online quiz or coding exercise at the end of the class to assess students' understanding of LDA and QDA. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 3	Course Name: Artificial Intelligence with Computer Vision Topic: Discriminant Analysis	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of space-time methods in recognition tasks Understand the concept of Discriminant Analysis in machine learning and statistics. b. Differentiate between Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA). c. Explain how Discriminant Analysis is used for classification tasks.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Ask questions. what they know about classification methods in machine learning and when it is important to classify data into distinct categories. - Provide an overview of Discriminant Analysis as a classification technique and explain that it is used for determining the decision boundary that best separates different classes. 2. **Development** (30 minutes) a. Overview of Discriminant Analysis: Explain the purpose of Discriminant Analysis, which is to find a linear combination of features that best separates two or more classes of objects or events. Example: Classifying species of flowers based on features such as petal length and width using the Iris dataset. b. Linear Discriminant Analysis (LDA): Introduce LDA as a method that assumes each class has the same covariance matrix and finds a linear boundary that separates the classes. Explain how LDA works by maximizing the ratio of between-class variance to within-class variance. Example: Using LDA to classify images into categories like "cat" or "dog" based on pixel values. c. Quadratic Discriminant Analysis (QDA): Introduce QDA as a variant of LDA that does not assume identical covariance matrices for each class. Explain that QDA fits quadratic boundaries for classification tasks. Example: Classifying objects when the classes have different shapes in feature space, such as spherical versus elliptical clusters.

	d. Applications of Discriminant Analysis: Discuss real-world applications of LDA and QDA, including face recognition, disease diagnosis, and fraud detection. Example: LDA used in credit scoring to classify individuals as low or high credit risk. 3. Exercise (5 minutes) – - Assign students to implement LDA and QDA on a dataset of their choice (e.g., Iris dataset). Ask them to fit the models, classify the data, and compare the performance of LDA and QDA on the task. - Afterward, have students present their results and discuss which method performed better and why. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the key differences between LDA and QDA and their use cases in classification tasks. 2. Discuss when to choose LDA versus QDA depending on the dataset and problem at hand. 3. Encourage students to ask questions or share their thoughts on the importance of discriminant analysis in machine learning. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "What assumptions does LDA make about the data?" and "When would QDA be more suitable than LDA for classification?" 3. Conduct an online quiz or coding exercise at the end of the class to assess students' understanding of LDA and QDA. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 4	Course Name: Artificial Intelligence with Computer Vision Topic: Bayesian Network Classifiers	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of Bayesian Networks in the context of classification tasks. b. Explain the role of conditional probabilities and dependencies in Bayesian Network Classifiers. c. Differentiate between Naive Bayes and more complex Bayesian Network Classifiers.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Ask questions. what they know about probabilistic models and how probabilities can be used to predict outcomes. - Introduce the concept of Bayesian Networks and explain that they model the probabilistic relationships among variables in a network structure for classification tasks. 2. **Development** (30 minutes) a. Overview of Bayesian Networks: Explain Bayesian Networks as graphical models that represent a set of variables and their conditional dependencies using a directed acyclic graph (DAG). Discuss the key components: nodes (variables) and edges (dependencies). Example: A medical diagnosis system where nodes represent symptoms and diseases, and edges represent the probabilistic relationships between them. b. Conditional Probability and Inference: Explain how conditional probability tables (CPTs) are used in Bayesian Networks to quantify the relationships between connected nodes. Discuss how inference in Bayesian Networks is performed by computing the posterior probabilities of the variables given evidence. Example: Given certain symptoms (evidence), infer the probability of a specific disease (hidden variable). c. Naive Bayes Classifier: Introduce the Naive Bayes Classifier as a simple form of a Bayesian Network that assumes all features are conditionally independent given the class label. Explain the pros and cons of this simplifying assumption. Example: Naive Bayes for email spam detection, where the presence of specific words determines whether an email is

	spam or not. d. Bayesian Network Classifiers: Discuss more complex Bayesian Network Classifiers that model dependencies between features. Explain how these networks can capture richer probabilistic relationships compared to Naive Bayes, leading to potentially better classification performance in certain scenarios. Example: A Bayesian Network Classifier for predicting weather conditions based on factors like temperature, humidity, and wind speed, where dependencies between features are modeled explicitly. e. Applications of Bayesian Network Classifiers: Discuss real-world applications, such as medical diagnosis, fault detection in systems, and financial risk assessment. Example: A Bayesian Network Classifier used in medical diagnosis to predict the probability of diseases based on observed symptoms and test results. 3. Exercise (5 minutes) – - Recap the main points covered: the structure of Bayesian Networks, the role of conditional probabilities, and the difference between Naive Bayes and more complex Bayesian Network Classifiers. - Discuss when it is beneficial to use Bayesian Network Classifiers over simpler models like Naive Bayes. - Encourage students to ask questions or share their thoughts on the challenges and future developments in Bayesian network-based classification. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the main points covered: the structure of Bayesian Networks, the role of conditional probabilities, and the difference between Naive Bayes and more complex Bayesian Network Classifiers. 2. Discuss when it is beneficial to use Bayesian Network Classifiers over simpler models like Naive Bayes. 3. Encourage students to ask questions or share their thoughts on the challenges and future developments in Bayesian network-based classification. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "What are the limitations of Naive Bayes, and how do Bayesian Network Classifiers overcome them?" and "In what scenarios would you choose a Bayesian Network

	Classifier over other models?" 3. Conduct an online quiz or coding exercise at the end of the class to assess students' understanding of Bayesian Network Classifiers and their practical implementation. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 5	Course Name: Artificial Intelligence with Computer Vision Topic: Temporal Methods	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of temporal methods and their importance in analyzing sequential data. b. Explain the role of temporal methods in tasks such as time series analysis, action recognition, and predictive modeling. c. Identify common temporal methods such as Hidden Markov Models (HMM), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM).
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Ask questions. how they think computers analyze data that changes over time, such as stock prices, human actions in a video, or weather conditions. - Introduce the concept of temporal methods and explain that they are crucial for understanding and predicting sequential or time-dependent data. 2. **Development** (30 minutes) a. Overview of Temporal Methods: Explain that temporal methods focus on modeling sequential data, where the order of data points matters. Discuss how these methods differ from traditional machine learning models, which assume that data points are independent. Example: Time series forecasting for stock prices or sensor data analysis in smart homes. b. Hidden Markov Models (HMM): Introduce HMMs as a statistical model used for analyzing sequences where the system being modeled is assumed to be a Markov process with hidden states. Explain how HMMs are used in tasks such as speech recognition, bioinformatics, and action recognition. Example: Recognizing human activities from a sequence of actions in video footage using HMMs. 3. Exercise (5 minutes) – - Assign students to work with a time series dataset (e.g., stock prices or temperature data) and implement a basic LSTM model to predict future values. Have them experiment with different model architectures and evaluate their performance.

	- Afterward, ask students to present their results and discuss the challenges of working with sequential data. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the key concepts of temporal methods, including HMMs, RNNs, and LSTM models, and their importance in analyzing time-dependent data. 2. Discuss when it is appropriate to use temporal methods over traditional machine learning models and how to choose the right model for the task at hand. 3. Encourage students to ask questions or share their thoughts on the impact of temporal methods in modern AI applications. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "What are the limitations of RNNs, and how does LSTM address them?" and "What types of problems are best suited for temporal methods?" 3. Conduct an online quiz or coding exercise at the end of the class to assess students' understanding of temporal methods and their practical implementation. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 6	Course Name: Artificial Intelligence with Computer Vision Topic: Model-Based Methods (Markov-Dynamic Networks)	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of Markov-Dynamic Networks (MDNs) and their role in model-based methods. b. Explain how MDNs combine Markov models with dynamic probabilistic models to represent time-dependent processes. c. Identify applications of MDNs in areas such as robotics, control systems, and time-series prediction.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Ask questions. how they think dynamic systems (such as robots navigating an environment) can be modeled over time. - Introduce the concept of Model-Based Methods with a focus on Markov-Dynamic Networks (MDNs) and explain that these methods are used to model systems that evolve over time based on probabilistic transitions. 2. **Development** (30 minutes) a. Overview of Model-Based Methods: Explain that model-based methods rely on probabilistic models that describe how a system behaves over time. Discuss the general idea behind modeling systems that are stochastic and evolve dynamically. Example: Modeling a robot's movement through a grid where the robot's position depends on its previous position and a probabilistic transition. b. Markov Models: Introduce the Markov property, which assumes that the future state of a system depends only on its current state and not on the sequence of states that preceded it. Discuss how Markov models represent this property using states and transitions. Example: A weather prediction model where tomorrow's weather only depends on today's weather conditions. c. Dynamic Bayesian Networks (DBNs): Explain how DBNs extend Bayesian networks to handle temporal data by representing time-evolving systems as a sequence of Bayesian networks, where each network corresponds to a different time step. Example: Using DBNs to track the health of a machine over

	time, based on sensor readings. d. Markov-Dynamic Networks (MDNs): Introduce Markov-Dynamic Networks as a combination of Markov models and dynamic probabilistic networks (such as DBNs). Explain how MDNs capture the probabilistic transitions between states over time while accounting for complex dependencies between variables. Example: Modeling a robot's navigation through an environment where the robot's next position depends on both its current position and observations of the environment, as well as a probabilistic transition model. e. Applications of MDNs: Discuss real-world applications of MDNs, such as in robotics (for planning and navigation), control systems (for system state prediction), and time-series prediction (for financial forecasting or health monitoring). Example: Using MDNs to model a robot's movements in uncertain environments, where the robot's next action depends on both its current state and its observations. 3. Exercise (5 minutes) – - Assign students to work with a dataset (e.g., sensor data from a robot or financial data) and implement a basic MDN model to predict future states based on current observations and probabilistic transitions. - Afterward, ask students to present their models and discuss the challenges of using MDNs for modeling dynamic systems. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the key concepts of Markov models, Dynamic Bayesian Networks, and Markov-Dynamic Networks. 2. Discuss the advantages of using MDNs for modeling systems that evolve over time, and when to choose this method over other modeling approaches. 3. Encourage students to ask questions or share their thoughts on the potential applications and limitations of MDNs in real-world scenarios. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "How do Markov-Dynamic Networks improve upon traditional Markov models?" and "What are the key challenges in implementing MDNs for complex systems?" 3. Conduct an online quiz or coding exercise at the end of the class to assess students' understanding of MDNs and their ability to apply

	the method to time-dependent problems. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 7	Course Name: Artificial Intelligence with Computer Vision Topic: Model-Based Methods (Markov-Dynamic Networks)	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of Markov-Dynamic Networks (MDNs) and their role in model-based methods. b. Explain how MDNs combine Markov models with dynamic probabilistic models to represent time-dependent processes. c. Identify applications of MDNs in areas such as robotics, control systems, and time-series prediction.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Ask questions. how they think dynamic systems (such as robots navigating an environment) can be modeled over time. - Introduce the concept of Model-Based Methods with a focus on Markov-Dynamic Networks (MDNs) and explain that these methods are used to model systems that evolve over time based on probabilistic transitions. 2. **Development** (30 minutes) a. Overview of Model-Based Methods: Explain that model-based methods rely on probabilistic models that describe how a system behaves over time. Discuss the general idea behind modeling systems that are stochastic and evolve dynamically. Example: Modeling a robot's movement through a grid where the robot's position depends on its previous position and a probabilistic transition. b. Markov Models: Introduce the Markov property, which assumes that the future state of a system depends only on its current state and not on the sequence of states that preceded it. Discuss how Markov models represent this property using states and transitions. Example: A weather prediction model where tomorrow's weather only depends on today's weather conditions. c. Dynamic Bayesian Networks (DBNs): Explain how DBNs extend Bayesian networks to handle temporal data by representing time-evolving systems as a sequence of Bayesian networks, where each network corresponds to a different time step. Example: Using DBNs to track the health of a machine over

	time, based on sensor readings. d. Markov-Dynamic Networks (MDNs): Introduce Markov-Dynamic Networks as a combination of Markov models and dynamic probabilistic networks (such as DBNs). Explain how MDNs capture the probabilistic transitions between states over time while accounting for complex dependencies between variables. Example: Modeling a robot's navigation through an environment where the robot's next position depends on both its current position and observations of the environment, as well as a probabilistic transition model. e. Applications of MDNs: Discuss real-world applications of MDNs, such as in robotics (for planning and navigation), control systems (for system state prediction), and time-series prediction (for financial forecasting or health monitoring). Example: Using MDNs to model a robot's movements in uncertain environments, where the robot's next action depends on both its current state and its observations. 3. Exercise (5 minutes) – - Assign students to work with a dataset (e.g., sensor data from a robot or financial data) and implement a basic MDN model to predict future states based on current observations and probabilistic transitions. - Afterward, ask students to present their models and discuss the challenges of using MDNs for modeling dynamic systems. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the key concepts of Markov models, Dynamic Bayesian Networks, and Markov-Dynamic Networks. 2. Discuss the advantages of using MDNs for modeling systems that evolve over time, and when to choose this method over other modeling approaches. 3. Encourage students to ask questions or share their thoughts on the potential applications and limitations of MDNs in real-world scenarios. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "How do Markov-Dynamic Networks improve upon traditional Markov models?" and "What are the key challenges in implementing MDNs for complex systems?" 3. Conduct an online quiz or coding exercise at the end of the class to assess students' understanding of MDNs and their ability to apply

	the method to time-dependent problems. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 8	Course Name: Artificial Intelligence with Computer Vision Topic: Hidden Markov Models	Course No.: COM-601

Objectives	At the end of the lesson the student shall be able to: a. Understand the fundamental principles of Hidden Markov Models (HMMs) and their applications. b. Explain the difference between observable and hidden states in an HMM. c. Identify the key components of HMMs: states, transition probabilities, emission probabilities, and initial probabilities.
Teaching Aids (if any)	a. PowerPoint slides b. Use of Nearpod tool for online quiz c. Videos demonstrating routing in networks
Teaching Development	1. **Introduction** (5 minutes) - Ask questions. if they are familiar with systems where you can observe certain outcomes, but the underlying processes are hidden (e.g., stock market trends or speech recognition). - Introduce Hidden Markov Models (HMMs) as a statistical model used to represent systems where the states are hidden but can be inferred from observable outcomes. 2. **Development** (30 minutes) a. Overview of HMMs: Explain that an HMM is a statistical model in which the system being modeled is assumed to follow a Markov process with hidden states. Discuss how the hidden states evolve over time based on transition probabilities, and how the observed outcomes are generated based on emission probabilities. Example: A simple weather prediction model where the actual weather (hidden states) influences the observed sensor readings (observations). b. Key Components of HMMs: Break down the key components of an HMM: States: The hidden variables that cannot be directly observed (e.g., the true weather condition: sunny, rainy, etc.). Observations: The observable outcomes influenced by the hidden states (e.g., temperature readings or umbrella use). Transition Probabilities: The probabilities of transitioning from one hidden state to another (e.g., the probability of moving from a sunny day to a rainy day). Emission Probabilities: The probabilities of observing a particular outcome given a hidden state (e.g., the likelihood of observing "umbrella" given the state is "rainy").

	Initial Probabilities: The starting probabilities for each hidden state. 3. Exercise (5 minutes) – - Assign students to work with a sample dataset (e.g., a sequence of stock prices or biological sequences) and implement a basic HMM to model the hidden states and predict future observations. Ask them to experiment with different configurations of transition and emission probabilities. - Afterward, have students present their findings and discuss the effectiveness of HMMs in modeling sequential data. Use Nearpod to collect responses and discuss the answers.
Closure	1. Recap the key concepts of Hidden Markov Models, including the distinction between hidden states and observations, and the role of transition and emission probabilities. 2. Discuss the advantages and limitations of HMMs for sequence modeling, and how they compare to other temporal models like Recurrent Neural Networks (RNNs). 3. Encourage students to ask questions or share their thoughts on how HMMs can be applied to different sequence modeling tasks. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Assess student comprehension through informal checks during the lesson, such as questioning and observation. 2. Reflective questions like: "How does the Viterbi algorithm help in identifying the most likely sequence of hidden states?" and "In what scenarios would you prefer using an HMM over other temporal models?" 3. Conduct an online quiz or coding exercise at the end of the class to assess students' understanding of HMMs and their practical application. Spend 5 minutes to evaluate student assimilation of the lesson contents