SUBJECT: COMPUTER VISION (CMVN)

CLASS XII
SEMESTER III

THEORY

FULL MARKS – 20

(MCQ Type Question)

UNIT	Topic	No. of periods assigned	Marks
Unit 1	**Introduction to Computer Vision**	**9**	**3**
Unit 2	**Image Basics and Image processing Techniques**	**18**	**9**
Unit 3	**Image Classification and Segmentation**	**18**	**8**

DETAIL SYLLABUS

UNIT	Topic / Sub Topic	No. of periods assigned
Unit 1	1.1 Definition and applications of computer vision. 1.2 Historical background and development. 1.3 Importance and relevance in modern technology.	**9**
Unit 2	2.1 Understanding digital images: pixels, resolution, color spaces (RGB, grayscale). 2.2 Image representation in computers: matrices and arrays, Image formats: JPEG, PNG, etc. 2.3 Image enhancement: histogram equalization, contrast stretching. 2.4 Image operations (blurring, sharpening, edge detection).Filtering techniques (convolution, Gaussian filter). 2.5 Filtering: smoothing, sharpening, edge detection using techniques like Sobel, Prewitt, and Canny. 2.6 Morphological operations: erosion, dilation, opening, and closing.	**18**
Unit 3	3.1 Feature Extraction: Introduction to feature extraction techniques such as Harris corner detection, SIFT, SURF, and ORB. 3.2 Feature descriptors: Histogram of Oriented Gradients (HOG), Scale- Invariant Feature 3.3 Image Classification: SVM, Decision Trees, Gradient Boosting Machines, Naïve Bayes 3.4 Image Segmentation: Thresholding, Region-based segmentation, Edge- based segmentation, Semantic segmentation, Instance segmentation 3.5 Working with Orange, Data Mining & Visualization tool for Classification tasks	**18**

CLASS XII
SEMESTER IV

THEORY

FULL MARKS – 30

(SAQ AND LAQ Type Question)

UNIT	Topic	No. of periods assigned	Marks
Unit 4	**Deep Learning for Computer Vision**	**24**	**12**
Unit 5	**Convolutional Neural Networks (CNNs)**	**24**	**12**
Unit 6	**Transfer Learning and Pre-trained Models**	**15**	**6**

DETAIL SYLLABUS

UNIT	Topic / Sub Topic	No. of periods assigned
Unit 4	4.1 Neural network architecture overview. 4.2 Basics of convolutional neural networks (CNNs). 4.3 Training CNNs for image classification tasks. 4.4 Transfer learning and fine-tuning pre-trained models like VGG, ResNet, and Google's inception.	**24**
Unit 5	5.1 Basic architecture of CNNs-Convolutional layers, pooling layers, Activation functions (ReLU). Tools and Libraries – Introduction to deep learning frameworks (TensorFlow, PyTorch), Hands-on coding exercises using high-level APIs (Keras). 5.2 Training CNNs-Dataset preparation and preprocessing, Loss functions (cross- entropy), Optimizers (e.g., SGD, Adam), Back-propagation in CNNs. 5.3 Applications of CNNs in Computer Vision-Image classification, Object detection,	**24**
Unit 6	6.1 Concept of transfer learning, Using pre-trained models for computer vision tasks- VGG, ResNet and Inception. 6.2 Fine-tuning pre-trained models for specific tasks. 6.3 Advanced Topics in Computer Vision-Generative Adversarial Networks (GANs), Object tracking, Face recognition.	**15**

PRACTICAL

CLASS XII

FULL MARKS – 40

NO OF PERIODS ASSIGNED – 84

DETAIL SYLLABUS

Sl. No.	Practical	Classes
1	**Image Processing Basics**	12
	Practical: Introduction to image processing techniques using Python **Tasks:** Loading and displaying images using Python. Implementing basic image operations like resizing, cropping, and rotating images. Applying image enhancement techniques such as histogram equalization and contrast stretching. Implementing filtering techniques like blurring, sharpening, and edge detection.	
2	**Feature Extraction**	10
	Practical: Implementing feature extraction techniques. **Tasks:** Implementing Harris corner detection algorithm. Using OpenCV functions to extract SIFT, SURF, and ORB features. Extracting Histogram of Oriented Gradients (HOG) features for object detection. Visualizing extracted features and descriptors.	
3	**Image Classification and Segmentation**	10
	Practical: Building image classification and segmentation models. **Tasks**: Preprocessing image data for classification and segmentation tasks. Training a simple CNN model for image classification using TensorFlow or Keras. Implementing image segmentation using techniques like thresholding and region-based segmentation. Evaluating model performance using accuracy metrics and visualizing segmentation results.	
4	**Training CNNs**	12
	Practical: Hands-on training of convolutional neural networks. **Tasks:** Preparing datasets for training CNNs. Implementing a CNN architecture using TensorFlow or Keras. Training the CNN model on a dataset for image classification or object detection task. Fine-tuning pre-trained CNN models for specific tasks.	
5	**Applications of CNNs in Computer Vision**	10

Sl. No.	Practical	Classes
	Practical: Implementing CNN-based applications. **Tasks:** Building an image classifier using a pre-trained CNN model like VGG or ResNet. Implementing object detection using frameworks like TensorFlow Object Detection API or YOLO (You Only Look Once). Performing semantic segmentation using deep learning models like FCN (Fully Convolutional Network).	
6	**Transfer Learning and Pre-trained Models**	**10**
	Practical: Transfer learning with pre-trained models. **Tasks:** Loading pre-trained CNN models (e.g., VGG, ResNet, Inception) using TensorFlow or Keras. Fine-tuning pre-trained models on a custom dataset for a specific computer vision task. Evaluating the performance of fine-tuned models and comparing it with training from scratch.	
7	**Advanced Topics in Computer Vision**	**10**
	Practical: Exploring advanced computer vision techniques. **Tasks:** Implementing a simple Generative Adversarial Network (GAN) for image generation. Building an object tracking system using techniques like correlation filters or deep learning-based methods. Developing an image captioning system using CNNs and recurrent neural networks (RNNs).	
8	**Tools and Libraries**	**10**
	Practical: Hands-on experience with deep learning frameworks. **Tasks:** Setting up TensorFlow or PyTorch environment. Writing code using high-level APIs (e.g., Keras) for building and training neural networks. Implementing computer vision tasks using pre-built functions and modules from libraries like OpenCV.	

PROJECT

CLASS XII

FULL MARKS – 10

NO OF PERIODS ASSIGNED – 60

DETAIL SYLLABUS

Name of the Project: Advanced Flower Classification System Development		
Objective – The project aims to develop an advanced flower classification system leveraging machine learning and computer vision techniques. Flower classification finds applications in agriculture, botany, ecology, and horticulture. Throughout the project, participants will gain hands-on experience in building a robust flower classification system capable of accurately identifying various types of flowers from images.		
Sl. No.	**Practical**	**Classes**
1	**Understanding Flower Classification**	**15**
	- Introduction to flower classification: Importance, applications, and challenges. - Overview of popular flower classification datasets and benchmarks. - Exploring different approaches to flower classification, including traditional and deep learning-based methods. - Understanding feature extraction techniques for image classification. - Introduction to convolutional neural networks (CNNs) and their application in image classification tasks. - Familiarization with libraries such as TensorFlow or PyTorch for deep learning.	
2	**Data Collection and Preprocessing**	**15**
	- Collecting flower image datasets from publicly available sources or through webscraping. - Data preprocessing techniques: resizing, normalization, and data augmentation. - Introduction to annotation tools for labeling flower images. - Data augmentation methods to increase dataset diversity. - Annotation and labeling of flower images to create a standardized dataset. - Splitting the dataset into training, validation, and test sets.	
3	**Model Training**	**15**

	- Setting up the development environment for model training. - Implementing CNN architectures for flower classification tasks. - Transfer learning: Fine-tuning pre-trained models for flower classification. - Hyperparameter tuning for optimizing model performance. - Training the model using the prepared dataset. - Evaluating model performance using validation datasets.	
4	**Implementation and Optimization**	**15**
	- Integrating the trained flower classification model into an application using Python. - Optimizing the classification algorithm for efficiency and accuracy. - Handling edge cases and improving robustness. - Implementing additional features like species recognition or bloom stage estimation. - Enhancing the user interface and user experience aspects of the application. - Conducting performance profiling and optimization.	