CONCEPT

Deep Learning and Computer Vision: Neural Networks, CNNs, Object Detection and Deployment

Deep Learning and Computer Vision: Neural Networks, CNNs, Object Detection and Deployment

Introduction:

Deep learning and computer vision projects often stall between a promising notebook and a model that works on real plant images: training diverges, accuracy collapses on new lighting, detectors miss small defects and nobody can run the model on edge hardware. This Core Concept course builds code-first capability in PyTorch, from neural network mechanics and training discipline to convolutional networks, transfer learning, object detection, segmentation, video analytics and deployment. Every session is a lab, and participants finish with an Industrial Vision Model Evaluation Pack built on an industrial image dataset.

Course Objectives:

- Build and train neural networks in PyTorch, selecting activation functions, loss functions and optimisers suited to the task
- Diagnose underfitting and overfitting and control them with regularisation, dropout, normalisation and structured hyperparameter search
- Adapt pretrained convolutional and transformer backbones to small domain image datasets through transfer learning and fine-tuning
- Train and evaluate object detection and segmentation models, reporting IoU, mean average precision and error cases
- Export, compress and benchmark vision models for edge inference and track experiments, drift and model versions
- Deliver an Industrial Vision Model Evaluation Pack with trained model, evaluation report, deployment package and model card

Target Audience:

- Data science staff who build predictive models and now need to work with image and video data
- Machine learning engineering staff responsible for training, packaging and serving neural network models
- Technical analysts who prototype automation and analytics solutions in Python
- Automation and digital engineering staff who integrate camera systems with plant or field software
- Research and development staff who evaluate deep learning methods for new products and processes

Course Outline:

Day 1: Neural Network Foundations and the PyTorch Workflow

- Perceptron, Multilayer Perceptron and Decision Boundary Intuition
- PyTorch Tensors, GPU Devices and Autograd Computational Graphs
- Forward Pass and Backpropagation Through the Chain Rule
- Activation Functions: Sigmoid, Tanh, ReLU and GELU Behaviour
- Loss Functions: Mean Squared Error, Cross-Entropy and Focal Loss

Day 2: Optimisers and the Training Loop

- Stochastic Gradient Descent, Momentum, RMSprop and Adam Optimisers
- Learning-Rate Schedules: Step Decay, Cosine Annealing and Warm-Up
- Weight Initialisation Schemes and Vanishing or Exploding Gradients
- PyTorch Dataset, DataLoader and Mini-Batch Training Loop Build
- Training and Validation Curves with TensorBoard Logging

Day 3: Regularisation, Generalisation and Hyperparameter Search

- L1 and L2 Weight Decay with Early Stopping Callbacks
- Dropout Layers and Their Behaviour at Training Versus Inference Time
- Batch Normalisation and Layer Normalisation Placement
- Hyperparameter Search with Random Search and Optuna Trials
- Reproducibility Controls: Random Seeds, Deterministic Kernels and Config Files

Day 4: Convolutional Neural Networks and Transfer Learning

- Convolution, Stride, Padding, Pooling and Receptive Field Arithmetic
- LeNet-5, AlexNet, VGG and ResNet Residual Block Design
- torchvision Pretrained Backbones for Feature Extraction
- Fine-Tuning Strategies: Frozen Layers, Discriminative Learning Rates and Gradual Unfreezing
- Grad-CAM Saliency Maps for Inspecting CNN Decisions

Day 5: Sequence Models, Transformers and Week-One Integration Lab

- Recurrent Networks, LSTM and GRU Cells for Sensor Time Series
- Self-Attention, Multi-Head Attention and Positional Encoding
- Vision Transformer Patch Embedding Compared with CNN Backbones
- Keras Sequential and Functional API Equivalents of the PyTorch Models
- Guided Lab: Image Classifier Built, Tuned and Explained End to End

Day 6: Image Pipelines, Augmentation and Annotation

- Image Data Pipeline Design: Decoding, Resizing, Normalisation and Caching
- Albumentations Geometric, Photometric and Cutout Transforms
- Mixup and CutMix for Small Industrial Image Sets
- Annotation Formats: COCO JSON, Pascal VOC XML and YOLO Text Labels
- Class Imbalance Remedies: Weighted Sampling and Synthetic Minority Images

Day 7: Object Detection with Single-Stage and Two-Stage Detectors

- Anchor Boxes, Anchor-Free Heads and Non-Maximum Suppression
- YOLO-Family Single-Stage Detector Training with Ultralytics
- Faster R-CNN Two-Stage Detection with torchvision
- Intersection over Union and Mean Average Precision Calculation
- Small-Object and Occlusion Error Analysis on Detector Outputs

Day 8: Segmentation, Tracking and Video Analytics

- U-Net Encoder-Decoder Semantic Segmentation
- Mask R-CNN Instance Segmentation and Polygon Masks
- Dice Coefficient and Mean IoU Segmentation Metrics
- Multi-Object Tracking with SORT and ByteTrack Association
- Video Stream Ingestion with OpenCV and Frame-Sampling Strategies

Day 9: Deployment, Edge Inference, MLOps and Responsible AI

- ONNX Export and ONNX Runtime Inference Benchmarking
- Post-Training Quantisation, Pruning and TensorRT Optimisation for Edge Devices
- Experiment Tracking and Model Registry with MLflow
- Data Drift Monitoring and Retraining Triggers for Deployed Vision Models
- Responsible AI Checks: Dataset Bias Audit, Privacy Masking and Model Cards

Day 10: Capstone: Industrial Vision Model Build and Evaluation

- Industrial Dataset Brief: Surface Defect, Component Counting or Corrosion Detection
- Baseline Training Run and Augmentation Plan Execution
- Evaluation Report with Confusion Matrix, mAP and Latency Measurements
- Deployment Package: ONNX Model, Inference Script and Model Card
- Capstone Presentation and Technical Peer Code Review

Skills You Will Gain:

- Neural Network Implementation
- Gradient-Based Optimisation
- Convolutional Architecture Design
- Transfer Learning
- Object Detection Engineering
- Image Segmentation
- Edge Model Optimisation
- Vision MLOps

Why Attend This Course:

- Return with an Industrial Vision Model Evaluation Pack containing a trained model, evaluation evidence and a deployable package
- Stop guessing when training fails by reading loss curves, gradients and saliency maps to find the cause
- Move from image classification to detection, segmentation and tracking with code that can be reused on the next project
- Compare modelling choices with data scientists and engineers from manufacturing, energy, utilities and logistics during shared labs

Conclusion:

Deep learning delivers value in computer vision only when sound training practice, suitable architectures and honest evaluation meet a model that can run where the images are captured. Week one builds the mechanics: networks, optimisers, regularisation, convolutional backbones, transfer learning and sequence models. Week two adds what a short course leaves out: image pipelines, detection, segmentation, tracking, edge deployment, MLOps and responsible AI checks. The capstone applies all of it to an industrial dataset and produces an Industrial Vision Model Evaluation Pack.