ARTIFICIAL INTELLIGENCE | ADVANCED | 5 DAYS

Production-Grade Computer Vision & Deep Learning

Build a strong foundation in production-grade Computer Vision systems, from model development to scalable deployment and lifecycle operations. Learn how to deliver high-performance CV pipelines with monitoring, observability, MLOps workflows, and reliability practices used in real-world enterprise environments.

Why Production Computer Vision?

Reduce operational costs

Automate inspection, monitoring, and compliance workflows with reliable CV pipelines.

Production readiness

Deploy optimized, observable CV systems with real-time performance at scale.

Improve quality and safety

Detect defects, anomalies, and security breaches faster than manual methods.

Future-proof capability

Build internal expertise across modern CV stacks, Transformers, and deployment acceleration.

Faster AI adoption

Use transfer learning and foundation models (SAM, CLIP) to deliver results with less data.

What you'll be able to do

- Engineer datasets using data-centric AI practices (annotation strategy, active learning, synthetic data) to improve model outcomes.
- Build robust training baselines using transfer learning with modern architectures and reproducible pipelines.
- Train production-ready object detection models using YOLOv8 and handle hard cases like small-object detection.
- Implement real-time video analytics with object tracking and streaming pipeline reliability patterns.
- Use foundation models (SAM, CLIP) for segmentation and zero-shot workflows with minimal training data.
- Build OCR and document intelligence solutions using transformer-based OCR approaches.
- Optimize models using quantization and acceleration techniques (ONNX, TensorRT) and measure latency/FPS improvements.
- Deploy scalable inference systems using Triton Inference Server with gRPC and dynamic batching.
- Apply CV MLOps practices including dataset versioning, experiment tracking, and production monitoring for drift/outliers.
- Deliver an end-to-end production-style capstone pipeline from data ingestion to deployment and observability.

Who this is for

- AI/ML Engineers building production CV systems
- Data Scientists moving into real-time vision and deployment

- MLOps Engineers supporting vision workloads
- Computer Vision Engineers working on detection, tracking, and segmentation
- Platform and DevOps Engineers supporting GPU inference and serving

Curriculum

01 Module 1: The Production Vision Landscape

- Architecture evolution: ResNet to EfficientNet to Transformers (ViT)
- Data-centric shift: why data quality drives model performance
- Lab: Rapid Transfer Learning with TIMM to build a baseline in under 15 lines of code

02 Module 2: Advanced Data Engineering

- Annotation strategy using CVAT or Label Studio
- Active learning and uncertainty sampling for labeling efficiency
- Synthetic data generation using Stable Diffusion or Copy-Paste augmentation
- Lab: Semi-automated labeling workflow with pre-trained model label suggestions in CVAT

03 Module 3: Production Augmentation

- Albumentations deep dive for robust augmentation
- Domain-specific simulation: rain/fog, ISO noise, motion blur
- Lab: Build an augmentation pipeline that mimics real-world camera degradation

04 Module 4: CNN Deep Dive and Debugging

- Visual debugging with Grad-CAM and activation interpretation
- Failure analysis: background bias and dataset leakage patterns
- Lab: Train ResNet50 for retail classification and debug model errors using Grad-CAM

05 Module 5: Modern Object Detection (YOLOv8)

- YOLOv8 architecture: anchor-free detection, C2f modules, loss functions
- YOLOv8 vs Faster-RCNN vs RetinaNet tradeoffs
- Lab: Train YOLOv8 on a custom manufacturing defect dataset

06 Module 6: The Small Object Problem

- Detecting tiny defects in high-resolution (4K) imagery
- SAHI (Slicing Aided Hyper Inference) and tiling strategies
- Lab: Implement inference tiling to detect defects missed by standard YOLO inference

07 Module 7: Video Analytics and Object Tracking

- Detection vs tracking, occlusions, re-identification concepts
- Tracking algorithms: SORT, DeepSORT, ByteTrack
- Lab: Build a people counting system using YOLOv8 + ByteTrack

08 Module 8: Handling Video Streams

- RTSP streams, buffering, lag, dropped frame handling
- Geofencing and counting lines using shapely and OpenCV
- Lab: Build a real-time intrusion detection pipeline on a mock video feed

09 Module 9: Segmentation and SAM

- Semantic vs instance segmentation (U-Net vs Mask R-CNN)
- Segment Anything Model (SAM) for zero-shot segmentation
- Lab: Use SAM to auto-generate segmentation masks for a dataset

10 Module 10: Zero-Shot Learning with CLIP

- Concept of contrastive language-image pre-training
- Natural language search use cases (catalog search without training)
- Lab: Build a natural language image search engine using CLIP

11 Module 11: Vision Transformers (ViT and DETR)

- ViT architecture: patch embeddings and attention
- DETR: transformer-based end-to-end object detection
- Lab: Fine-tune a ViT using HuggingFace transformers for classification

12 Module 12: OCR and Document Intelligence

- Transformer OCR intro: TrOCR, PaddleOCR
- Layout parsing for receipts/invoices using Donut concepts
- Lab: Extract structured data from shipping labels using PaddleOCR

13 Module 13: Model Optimization

- Quantization: FP32 to FP16 to INT8 tradeoffs
- Pruning fundamentals
- Lab: Post-training quantization of YOLO and FPS benchmarking

14 Module 14: Hardware Acceleration (TensorRT and ONNX)

- Understanding ONNX computational graphs
- TensorRT optimizations: fusion and kernel tuning
- Lab: Convert PyTorch model to ONNX and build a TensorRT engine

15 Module 15: Enterprise Serving (Triton)

- Limitations of simple API serving for production concurrency
- Triton server setup, dynamic batching, multi-model execution
- Lab: Deploy TensorRT model on Triton and query via gRPC

16 Module 16: Edge Deployment (Jetson and DeepStream)

- Edge constraints: thermal, memory, resource limits
- DeepStream SDK intro and GStreamer pipeline concepts
- Lab: Simulate an edge deployment workflow using remote container deployment

17 Module 17: CV MLOps

- Data versioning with DVC for large image datasets
- Experiment tracking with MLflow
- Lab: Create a reproducible DVC + MLflow training workflow

18 Module 18: Monitoring and Observability

- Embedding drift vs pixel drift concepts
- Outlier detection using Deepchecks or Alibi Detect
- Lab: Implement an anomaly alert pipeline for out-of-distribution images

19 Module 19: Capstone Project (Choose One)

- End-to-end pipeline: Data pipeline to training to optimization to deployment
- Option A: Manufacturing visual inspection (YOLOv8 + SAHI + TensorRT + drift detection)
- Option B: Retail smart shelf (detection + CLIP + Triton serving)
- Option C: Security perimeter breach (YOLOv8 + ByteTrack + geofencing + video streams)