CENTRAL BANK OF EGYPT
Egyptian Banking Institute

Center of **Excellence** and **Knowledge Hub**

# Computer Vision Specialization

**100 Hours**

**Blended (Vrt+EBI)**

Center of **Excellence** and **Knowledge Hub**
www.ebi.gov.eg

## Course Overview

This course is a comprehensive, end-to-end specialization in Computer Vision and Deep Learning. It begins by building a solid foundation in machine learning and neural networks, then progressively advances into convolutional architectures, modern detection and segmentation frameworks, generative models, and Vision-Language Models (VLMs) — covering the field up to its current state of the art.

Participants will develop both the theoretical understanding and the practical engineering skills needed to design, train, fine-tune, and deploy computer vision systems in real-world production environments. Every module is paired with hands-on labs using the latest tools and frameworks, and the course culminates in a set of capstone projects that mirror industry use cases across manufacturing, healthcare, retail, and media.

By the time participants complete this track, they will have moved from understanding pixels to building intelligent visual systems that see, understand, and generate the world around them.

## Target Audience:

- Data Scientists and ML Engineers seeking a complete, end-to-end Computer Vision specialization — from ML foundations through state-of-the-art architectures.
- Professionals in banking, telecom, manufacturing, healthcare, and technology sectors aiming to build and deploy production-grade computer vision systems.
- Practitioners who want to master the full modern CV stack — from classical image processing to Vision Transformers, generative models, and Vision-Language Models (VLMs).

## Course Objectives

**By the end of this course, participants will be able to:**

- Build and train deep neural networks and CNNs from scratch and apply transfer learning using modern architectures such as EfficientNet, ResNet, and ConvNeXt.
- Understand and implement Vision Transformers (ViT, Swin, DeiT, MAE) and leverage self-supervised models such as DINO and DINOv2 for visual feature learning.
- Design and deploy object detection pipelines using the full YOLO family (v5 through v10), Faster R-CNN, DETR, and open-vocabulary detectors like Grounding DINO.
- Implement semantic, instance, and panoptic segmentation systems using SegFormer, Mask2Former, YOLOv8-seg, and the Segment Anything Model (SAM).
- Build and train generative models including DCGANs, CycleGAN, VAEs, diffusion models, and fine-tune Stable Diffusion with DreamBooth and ControlNet.
- Apply Vision-Language Models (CLIP, LLaVA, InstructBLIP) for zero-shot classification, visual question answering, image captioning, and document understanding.
- Work with advanced video understanding techniques including multi-object tracking, action recognition, and video segmentation with SAM 2.
- Explore 3D vision topics including monocular depth estimation, NeRF, and 3D Gaussian Splatting for novel view synthesis.
- Optimize, containerize, and deploy CV models to production using ONNX, TensorRT, FastAPI, Docker, and cloud platforms.
- Manage the full MLOps lifecycle for computer vision — from data annotation with Roboflow and Label Studio to experiment tracking with MLflow and model monitoring in production.

## Course Outline

**Module 1: Machine Learning Foundations**

- Introduction to Machine Learning
- Linear Regression
- Logistic Regression (Vectorization)
- Bias, Variance and Learning Curves
- SVM & Kernel Trick

**Module 2: Neural Networks & Deep Learning**

- Neural Networks
- Deep Learning
- Forward & Backward Propagation
- Neural Network Vectorization
- Neural Networks Hyper-Parameters Tuning
- Types of Activation Functions

**Module 3: Convolutional Neural Networks (CNN)**

- Convolutional Neural Network (CNN)
- Classic CNNs
- CNN Applications

**Module 4 (Practicum): Python for Machine Learning & Neural Networks NumPy Quick Revision**

- Introduction
    - Vectors and Matrices
    - NumPy Basic Operations

**Image Preprocessing**

- Loading
    - Visualization
    - Vectorization
    - Normalization

**Single Perceptron**

- Initializing Weights
    - Propagation
    - Backward Propagation

**Single Layer Neural Networks**

- Initializing Weights
- Forward Propagation
- Activation Functions
- Cost Functions
- Backward Propagation
- Optimization

**Hyper Parameters Tuning**

- Regularization
  - Ridge and Lasso Regression
  - Dropouts
- Optimization
  - Momentum
  - RMS Prop
  - Adam Optimizer

**Keras and CNN**

- Keras
  - Introduction
  - Applying Vanilla Neural Networks
  - Parameter Tuning in Keras
- MNIST Data Classification
  - Introduction
  - Applying Vanilla Neural Networks
  - Parameter Tuning in Keras

**CNN**

- Object Detection
- Residual Nets
- Classification with Localization

**Transfer Learning and COLAB**

- Introduction to Transfer Learning
- Use Cases on Transfer Learning
- Using CNN Architecture using Keras API (VGG / GoogleNet / Residual Network)
- Applying Transfer Learning on MNIST Dataset using VGG with Keras API
- Introduction to COLAB
- Using VGG16 Transfer Learning on CIFAR-10 Dataset using COLAB GPU Resources

**Module 5: Advanced CNN Architectures & Transfer Learning Deep Dive**

**5.1 Revisiting CNN Foundations**

- Receptive fields, feature maps, and spatial hierarchies
- Depthwise separable convolutions (MobileNet insight)
- Dilated / atrous convolutions for dense prediction
- Deformable convolutions

**5.2 Modern CNN Architectures**

- ResNet family — residual connections, identity shortcuts, ResNeXt
- DenseNet — dense connectivity and feature reuse
- EfficientNet — compound scaling and NAS
- ConvNeXt — modernizing CNNs with Transformer design principles
- RegNet — designing network design spaces

### 5.3 Transfer Learning Strategies

- Feature extraction vs fine-tuning vs full retraining
- Layer freezing strategies and learning rate scheduling
- Domain adaptation techniques
- Hands-on: Fine-tuning EfficientNet on a custom classification dataset

## Module 6: Vision Transformers (ViT) & Attention in Vision

### 6.1 From NLP Transformers to Vision

- Patch embedding — treating images as sequences of patches
- ViT (Vision Transformer) — architecture walkthrough
- Positional embeddings in 2D space
- CLS token and classification head
- Reference: Dosovitskiy et al. (2020) — "An Image is Worth 16x16 Words"

### 6.2 ViT Variants & Improvements

- DeiT — Data-efficient Image Transformers (distillation token)
- Swin Transformer — shifted window attention for hierarchical features
- BEiT — BERT pre-training for image Transformers
- MAE — Masked Autoencoders for self-supervised ViT pre-training
- CvT, PVT — hybrid CNN-Transformer designs

### 6.3 Self-Supervised Vision with DINO & DINOv2

- DINO — self-distillation with no labels
- Emergent segmentation properties in DINO attention maps
- DINOv2 — curated data and universal visual features
- Hands-on: Visualizing DINO self-attention maps on custom images

### 6.4 Segment Anything Model (SAM)

- SAM architecture — image encoder, prompt encoder, mask decoder
- Promptable segmentation: point, box, text, and mask prompts
- SAM 2 — real-time video segmentation extension
- Hands-on: Using SAM for zero-shot segmentation on custom data

## Module 7: Object Detection — From Anchors to Anchor-Free

### 7.1 Detection Framework Taxonomy

- Two-stage detectors vs one-stage detectors vs anchor-free
- Evaluation metrics: mAP, IoU, precision-recall curves
- Dataset formats: COCO, Pascal VOC, YOLO format

### 7.2 Two-Stage Detectors

- Faster R-CNN — RPN, RoI Pooling, end-to-end training
- Mask R-CNN — extending Faster R-CNN with instance segmentation
- Feature Pyramid Networks (FPN) — multi-scale feature fusion
- Cascade R-CNN — progressive IoU threshold refinement

**7.3 YOLO Family (Deep Dive)**

- YOLOv5 — architecture, training, and hyperparameter tuning
- YOLOv8 — unified detection, segmentation, pose, and classification
- YOLO-NAS — Neural Architecture Search for detection
- YOLOv9 / YOLOv10 — programmable gradient information and efficiency
- Hands-on: Training YOLOv8 on a custom dataset with Roboflow

**7.4 Transformer-Based Detection**

- DETR — Detection Transformer (end-to-end, no NMS)
- Deformable DETR — faster convergence with deformable attention
- DINO (detection) — improved DETR with contrastive denoising
- Grounding DINO — open-set detection with text queries

**7.5 Open-Vocabulary & Zero-Shot Detection**

- CLIP-based detection (OWL-ViT)
- Grounded SAM — combining Grounding DINO + SAM
- Hands-on: Zero-shot detection pipeline with Grounded SAM

**Module 8: Image Segmentation — Semantic, Instance & Panoptic**

**8.1 Semantic Segmentation**

- FCN — Fully Convolutional Networks
- U-Net — encoder-decoder with skip connections (medical imaging focus)
- DeepLab family — atrous convolutions and ASPP
- SegFormer — hierarchical Transformer for semantic segmentation
- Hands-on: Semantic segmentation with SegFormer on a custom dataset

**8.2 Instance Segmentation**

- Mask R-CNN revisited — instance-level masks
- YOLACT — real-time instance segmentation
- SOLOv2 — segmenting objects by locations
- Hands-on: Instance segmentation with YOLOv8-seg

**8.3 Panoptic Segmentation**

- Panoptic FPN — unified things and stuff segmentation
- Mask2Former — universal image segmentation architecture
- OneFormer — one model for all segmentation tasks
- Hands-on: Panoptic segmentation with Mask2Former via Hugging Face

**8.4 Medical & Specialized Segmentation**

- SAM for medical image segmentation (MedSAM)
- 3D segmentation concepts for volumetric data
- Weakly supervised and semi-supervised segmentation

**Module 9: Generative Models for Computer Vision**

**9.1 GANs — Generative Adversarial Networks**

- GAN fundamentals — generator, discriminator, minimax game
- DCGAN — deep convolutional GAN

- Progressive GAN and StyleGAN2/3 — high-fidelity face synthesis
- CycleGAN — unpaired image-to-image translation
- Pix2Pix — paired image-to-image translation
- Conditional GAN (cGAN) — class-conditional generation
- GAN training instability — mode collapse, Wasserstein loss, gradient penalty

**9.2 Variational Autoencoders (VAE)**

- VAE architecture and the reparameterization trick
- Latent space interpolation and disentanglement
- VQ-VAE — vector quantized representations

**9.3 Diffusion Models**

- DDPM — Denoising Diffusion Probabilistic Models (theory and math)
- Forward diffusion process and reverse denoising
- DDIM — accelerated deterministic sampling
- Score-based generative models and SDE framework
- Classifier-free guidance

**9.4 Stable Diffusion & Latent Diffusion Models**

- Latent Diffusion Models (LDM) — diffusion in latent space
- Stable Diffusion architecture: VAE + U-Net + CLIP text encoder
- Text-to-image generation pipeline
- ControlNet — conditioning on edges, depth, pose, segmentation maps
- IP-Adapter — image prompt adaptation
- DreamBooth and LoRA for personalized fine-tuning
- SDXL and SD3 — architectural improvements
- Hands-on: Fine-tuning Stable Diffusion with DreamBooth on custom subjects

**Module 10: Vision-Language Models (VLMs)**

**10.1 CLIP & Contrastive Vision-Language Pre-training**

- CLIP architecture — image encoder + text encoder + contrastive loss
- Zero-shot image classification with CLIP
- ALIGN, SigLIP — scaling contrastive pre-training
- Hands-on: Zero-shot classification and image retrieval with CLIP

**10.2 Multimodal Large Language Models (MLLMs)**

- LLaVA — visual instruction tuning for conversation
- InstructBLIP — instruction-following vision-language model
- Qwen-VL and InternVL — high-performance open-source MLLMs
- GPT-4V / GPT-4o — closed-source multimodal capabilities
- Gemini Vision — Google's multimodal frontier model

**10.3 Vision-Language Tasks**

- Visual Question Answering (VQA)
- Image captioning and dense captioning
- Visual grounding and referring expression comprehension
- Document understanding (LayoutLM, Donut, TrOCR)

- Chart and table understanding
- Hands-on: Building a VQA pipeline with LLaVA via Hugging Face

**10.4 Text-to-Image & Multimodal Generation**

- DALL-E 3 — text-to-image with ChatGPT integration
- Imagen and Parti — Google's diffusion and autoregressive approaches
- Unified generation: LLMs that output images (Emu, SEED-LLaMA)

**Module 11: Advanced OpenCV & Classical Vision Pipelines**

**11.1 Advanced Image Processing**

- Morphological operations, contour analysis, and shape descriptors
- Frequency domain processing — Fourier and Gabor transforms
- Image registration and homography estimation
- Optical flow — Lucas-Kanade, Farneback, RAFT

**11.2 Feature Matching & 3D Vision Fundamentals**

- SIFT, SURF, ORB — classical feature detectors and descriptors
- SuperPoint and SuperGlue — learned feature matching
- Camera calibration and lens distortion correction
- Stereo vision and depth estimation
- Structure from Motion (SfM) overview

**11.3 Video Understanding**

- Multi-object tracking: SORT, DeepSORT, ByteTrack
- Action recognition: 3D CNNs, SlowFast, Video Transformers
- Video object segmentation with SAM 2
- Anomaly detection in video streams

**Module 12: 3D Vision & Emerging Frontiers**

**12.1 Monocular Depth Estimation**

- Classical stereo vs monocular depth
- MiDaS — robust monocular depth estimation
- Depth Anything — foundation model for depth
- Hands-on: Depth estimation pipeline with Depth Anything v2

**12.2 3D Representations**

- Point clouds and voxel grids
- NeRF — Neural Radiance Fields for novel view synthesis
- Instant-NGP — hash encoding for fast NeRF training
- 3D Gaussian Splatting — real-time novel view synthesis
- Hands-on: 3D Gaussian Splatting demo on custom scene

**12.3 Pose Estimation**

- 2D human pose estimation: HRNet, ViTPose
- Whole-body and hand pose estimation
- 6D object pose estimation
- Hands-on: Human pose estimation with YOLOv8-pose

**Module 13: MLOps & Deployment for Computer Vision**

**13.1 Model Optimization for Edge & Production**
- Quantization: INT8, INT4 with TensorRT and ONNX Runtime
- Pruning and knowledge distillation for CV models
- ONNX export pipeline for cross-platform deployment
- TensorRT optimization for NVIDIA GPUs
- OpenVINO for Intel edge hardware

**13.2 Serving CV Models**
- FastAPI endpoints for image inference
- Triton Inference Server for batched GPU inference
- Hugging Face Inference Endpoints and Spaces (Gradio)
- Streaming video inference pipelines

**13.3 Data & Experiment Management**
- Roboflow — dataset management, augmentation, and annotation
- Label Studio — open-source annotation platform
- MLflow — experiment tracking for CV workloads
- DVC — data version control for large image datasets

**13.4 Containerization & Cloud Deployment**
- Dockerizing CV inference applications
- Deploying on AWS SageMaker, GCP Vertex AI, Azure ML
- Real-time inference with Kafka + model server pipelines
- Monitoring: data drift, label drift, and prediction confidence tracking

## Labs and Assignment

**Vision Transformer Mechanics from Scratch**
- Implementing patch embedding and multi-head self-attention in NumPy/PyTorch
  - Image-to-sequence patch extraction
  - Scaled dot-product attention for 2D inputs
  - Visualizing ViT attention maps

**Hugging Face Transformers for Vision — Hands-on Labs**
- Lab 1: Image Classification with ViT & Swin Transformer
  - Loading and fine-tuning pre-trained ViT on custom dataset
  - Data augmentation with torchvision and Albumentations
  - Training with Hugging Face Trainer and evaluation
- Lab 2: Object Detection with DETR / Grounding DINO
  - Loading DETR and running inference
  - Visualizing bounding boxes and attention maps
  - Open-vocabulary detection with Grounding DINO
- Lab 3: Image Segmentation with Mask2Former & SAM
  - Panoptic segmentation pipeline with Mask2Former
  - Interactive segmentation with SAM — point and box prompts
  - Automated mask generation and filtering

**YOLO Hands-on Labs**

- Lab 4: Custom Object Detection with YOLOv8
  - Dataset preparation and annotation with Roboflow
  - Training YOLOv8 on custom classes
  - Hyperparameter tuning and augmentation strategies
  - Exporting to ONNX and running inference

**Generative Model Labs**

- Lab 5: GAN Implementation — DCGAN from Scratch
  - Building generator and discriminator in PyTorch
  - Training loop, loss curves, and mode collapse diagnosis
  - Generating and visualizing synthetic images
- Lab 6: Stable Diffusion Fine-Tuning with DreamBooth
  - Setting up the diffusers library environment
  - Preparing a subject dataset (3–20 images)
  - DreamBooth training run on Google Colab / cloud GPU
  - Generating personalized images with text prompts

**Vision-Language Model Labs**

- Lab 7: Zero-Shot Vision Tasks with CLIP
  - Zero-shot image classification against custom label sets
  - Cross-modal image retrieval by text query
  - CLIP-based image similarity scoring
- Lab 8: Visual QA & Captioning with LLaVA
  - Loading LLaVA via Hugging Face Transformers
  - Image-conditioned dialogue and instruction following
  - Document and chart understanding pipeline

**Deployment Lab**

- Lab 9: Deploying a Real-Time CV Model as a FastAPI + Gradio Service
  - Wrapping a YOLOv8 inference pipeline in FastAPI
  - Building a Gradio front-end for image upload and annotation
  - Dockerizing the full application
  - Deploying to Hugging Face Spaces

## Capstone Projects

- Project 1: End-to-End Object Detection System — Train, optimize, and deploy a custom YOLO model for a real-world use case (e.g., manufacturing defect detection, retail shelf monitoring)
- Project 2: Medical Image Segmentation — Build a U-Net / MedSAM pipeline for organ or lesion segmentation on a public medical dataset
- Project 3: Generative Image Application — Fine-tune Stable Diffusion with DreamBooth or ControlNet for a domain-specific image generation task
- Project 4: Vision-Language Chatbot — Build a multimodal assistant using LLaVA or CLIP + LLM that answers questions about uploaded images
- Project 5: Real-Time Video Analytics Dashboard — Multi-object tracking + action recognition pipeline

served via FastAPI with live video feed

- Project 6: 3D Scene Reconstruction — Implement a NeRF or 3D Gaussian Splatting pipeline on a self-captured scene
- Project 7: Open-Vocabulary Detection & Segmentation — Production pipeline with Grounded SAM for zero-shot detection and segmentation
- Project 8: End-to-End CV MLOps Pipeline — Full workflow: data versioning (DVC), training (MLflow), optimization (ONNX/TensorRT), and cloud deployment

**Prerequisites:**

Basics of Python programming, fundamental understanding of linear algebra, calculus, and statistics.

HOTLINE
15200
One number to better serve you!

**Headquarters – Nasr City**
22 A, Dr. Anwar El Mofty St., Tiba 2000
P.O.Box 8164 Nasr City, Cairo, Egypt
Tel.: (+2) 02 24054472
Fax: (+2) 02 24054471

**Working hours:** 9:00 am - 5:00 pm
**www.ebi.gov.eg**

**Like us on**
facebook.com/EgyptianBankingInstitute

**Follow us on**
twitter.com/EBItweets

**Join us on**
linkedin.com/company/egyptian-banking-institute

**Watch us on**
YouTube Channel: Egyptian Banking Institute (EBI)