**Deep Neural Networks (dnn module)**

We use the DNN module in OpenCV when we want to load and run pre-trained deep learning models directly inside our code. We can take models trained in frameworks like TensorFlow, Caffe, or ONNX and run them for inference without having to write the whole training code again. It supports CPU and GPU acceleration, so we can run it fast depending on our system. We just give it the model file, load it, prepare the input blob, and get the output predictions. This is useful for tasks like image classification, object detection, segmentation, and even pose estimation.

Basically,

* It supports models for classification, object detection, segmentation, and image recognition.
* We can use formats like .pb (TensorFlow), .caffemodel (Caffe), or .onnx.
* It supports CPU and GPU (CUDA), so things are faster if hardware allows.
* It integrates easily with preprocessing functions (resizing, blobFromImage) and postprocessing.

1. **Neural network**

Neural networks are like a way for computers to learn from data by passing information through layers of connected “neurons”, kind of inspired by how our brain works. We give it inputs, the neurons do some math, and slowly the system learns patterns to give correct outputs.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Tensorflow**

TensorFlow is a big open-source library from Google that helps in building and training these neural networks. It supports both CPUs and GPUs so we can train faster, and it has many ready-made tools so we don’t have to code everything from scratch.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Caffe**

Caffe is another deep learning framework, but it’s more famous for computer vision tasks. It’s super fast for image processing because it’s written in C++ with a Python interface. Many researchers used Caffe before TensorFlow became popular, and even now it’s used for quick prototyping in vision projects.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### **Halide**

Halide is a low-level image processing language that can speed up DNN operations on CPUs. Once OpenCV is compiled with Halide, we can switch the DNN backend to it and get better performance for inference, especially for image-based models.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### **OpenCV usage with OpenVINO**

OpenCV works with OpenVINO so we can convert models and run them at real-time speeds on Intel hardware for things like object detection or segmentation.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### **YOLO DNNs**

OpenCV supports YOLO (You Only Look Once) models like YOLOv3, YOLOv4, and more recent ones in ONNX format. It's easy to detect objects in real-time videos or images, just load the YOLO model and directly get bounding boxes.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### **Custom OCR model**

OpenCV’s DNN module can load an OCR model (e.g.,Tesseract) to detect and recognize text in images. That’s useful for scanning receipts, number plates, or printed documents.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### **High-Level API: TextDetectionModel and TextRecognitionModel**

These are higher-level helpers that make it simpler to detect and read text. OpenCV handles both locating the text and recognizing the letters—so code is much cleaner.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### **DNN-based Face Detection and Recognition**

OpenCV has deep learning face detectors and recognition tools like ResNet. These are way more accurate than old-school Haar Cascades and work better in tricky lighting or angles.