Deep Learning is the most popular and the fastest growing area in Computer Vision nowadays. Since OpenCV 3.1 there is DNN module in the library that implements forward pass (inferencing) with deep networks, pre-trained using some popular deep learning frameworks, such as Caffe. In OpenCV 3.3 the module has been promoted from opencv_contrib repository to the main repository (https://github.com/opencv/opencv/tree/master/modules/dnn) and has been accelerated significantly.

The module has no any extra dependencies, except for libprotobuf, and libprotobuf is now included into OpenCV.

The supported frameworks:

- Caffe
- TensorFlow
- Torch
- Darknet
- Models in ONNX format (as the main method to import models from PyTorch and Keras for some cases)

The supported layers:

- AbsVal
- Accum
- AveragePooling
- BatchNormalization
- BNLL
- Concatenation
- Convolution (1d, 2d, including dilated convolution, 3d)
- Crop
- CropAndResize (RCNN-specific layer)
- Deconvolution, a.k.a. transposed convolution or full convolution
- DetectionOutput (SSD-specific layer)
- Dropout
- Eltwise (+, *, max)
- ELU
- Expand
- Flatten
- FullyConnected
- FlowWarp
- Gather
- Interpolation
- LRN
- LSTM
- MaxPooling
- MaxUnpooling
- Mish
- MVN
- NormalizeBBox (SSD-specific layer)
- Padding
- Permute
- Power
- PReLU (including ChannelPReLU with channel-specific slopes)
- PriorBox (SSD-specific layer)
- ReLU
- ReduceL1

- ReduceL2
- ReduceLogSum
- ReduceLogSumExp
- ReduceMax
- ReduceMean
- ReduceMin
- ReduceProd
- ReduceSum
- ReduceSumSquare
- Region (for DarkNet models)
- Reorg
- Resize
- RNN
- ROI Pooling (RCNN-specific layer)
- Scale
- Shift
- ShuffleChannel
- Sigmoid
- Slice
- Softmax
- Split
- Swish
- TanH

You also can write your own Custom layer.

The module includes some SSE, AVX, AVX2 and NEON acceleration of the performance-critical layers as well as support of CUDA for the most of the layers. There is also constantly-improved Halide backend. OpenCL (libdnn-based) backend is being developed and should be integrated after OpenCV 3.3 release. Here you may find the up-to-date benchmarking results: [[DNN Efficiency|DNN-Efficiency]]

The provided API (for C++ and Python) is very easy to use, just load the network and run it. Multiple inputs/outputs are supported. Here are the examples: https://github.com/opencv/opencv/tree/master/samples/dnn.

There is Habrahabr article describing the module: https://habrahabr.ru/company/intel/blog/333612/ (in Russian).

The following networks have been tested and known to work:

Image classification

Caffe

- AlexNet
- GoogLeNet
- VGG
- ResNet
- SqueezeNet
- DenseNet
- ShuffleNet

TensorFlow

- Inception
- Inception, MobileNet
- EfficientNet

Darknet: https://pjreddie.com/darknet/imagenet/
ONNX: https://github.com/onnx/models

- AlexNet
- GoogleNet
- CaffeNet
- RCNN_ILSVRC13
- ZFNet512
- VGG16, VGG16_bn
- ResNet-18v1, ResNet-50v1
- CNN Mnist
- MobileNetv2
- LResNet100E-IR
- Emotion FERPlus
- Squeezenet
- DenseNet121
- Inception v1, v2
- Shufflenet

Torchvision: https://github.com/pytorch/vision
PyTorch Image Models: https://github.com/rwightman/pytorch-image-models
PyTorch EfficientNet

Object detection

Caffe

- SSD VGG
- MobileNet-SSD
- Faster-RCNN
- R-FCN
- OpenCV face detector

TensorFlow

- SSD, Faster-RCNN and Mask-RCNN from TensorFlow Object Detection API
- EAST: An Efficient and Accurate Scene Text Detector, paper: https://arxiv.org/abs/1704.03155v2
- EfficientDet from AutoML, details, paper: https://arxiv.org/abs/1911.09070

Darknet

- YOLOv2, tiny YOLO, YOLOv3, Tiny YOLOv3, YOLOv4, Tiny Yolov4 original repo

ONNX: https://github.com/onnx/models

- TinyYolov2
- PyTorch YOLOv3 and YOLOv3-tiny
- PyTorch SSD VGG (using this pull request):

Semantic segmentation

- FCN (Caffe)
- ENet (Torch)
- ResNet101_DUC_HDC (ONNX: https://github.com/onnx/models)
- DeepLab (TensorFlow)
- UNet, DeepLabV3, FPN from Segmentation Models PyTorch
- Unet, UNetPlus, BiSeNet from Human Segmenation PyTorch

Pose estimation

- OpenPose body and hands pose estimation (Caffe)
- AlphaPose (PyTorch)

Image processing

- Colorization (Caffe)
- Fast-Neural-Style (Torch)
- Style Transfer (ONNX)

Person identification

- OpenFace (Torch)
- Torchreid (PyTorch)
- Face Verification with MobileFaceNet (TensorFlow)

Text detection and Recognition

- EasyOCR (PyTorch)
- CRNN (PyTorch)

Depth Estimation

- Monodepth2 (PyTorch) details