JavaScript is disabled on your browser.

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Tree](http://docs.google.com/package-tree.html)
* [Index](http://docs.google.com/index-all.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/org/opencv/dnn/DictValue.html)
* [Next Class](http://docs.google.com/org/opencv/dnn/Layer.html)
* [Frames](http://docs.google.com/index.html?org/opencv/dnn/Dnn.html)
* [No Frames](http://docs.google.com/Dnn.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Field](#3znysh7) |
* [Constr](#2et92p0) |
* [Method](#tyjcwt)
* Detail:
* [Field](#1t3h5sf) |
* [Constr](#2jxsxqh) |
* [Method](#3j2qqm3)

org.opencv.dnn

## Class Dnn

* java.lang.Object
  + org.opencv.dnn.Dnn
* public class Dnn  
  extends java.lang.Object

### Field SummaryFields

| Modifier and Type | Field and Description |
| --- | --- |
| static int | [**DNN\_BACKEND\_DEFAULT**](http://docs.google.com/org/opencv/dnn/Dnn.html#DNN_BACKEND_DEFAULT) |
| static int | [**DNN\_BACKEND\_HALIDE**](http://docs.google.com/org/opencv/dnn/Dnn.html#DNN_BACKEND_HALIDE) |
| static int | [**DNN\_BACKEND\_INFERENCE\_ENGINE**](http://docs.google.com/org/opencv/dnn/Dnn.html#DNN_BACKEND_INFERENCE_ENGINE) |
| static int | [**DNN\_BACKEND\_OPENCV**](http://docs.google.com/org/opencv/dnn/Dnn.html#DNN_BACKEND_OPENCV) |
| static int | [**DNN\_TARGET\_CPU**](http://docs.google.com/org/opencv/dnn/Dnn.html#DNN_TARGET_CPU) |
| static int | [**DNN\_TARGET\_FPGA**](http://docs.google.com/org/opencv/dnn/Dnn.html#DNN_TARGET_FPGA) |
| static int | [**DNN\_TARGET\_MYRIAD**](http://docs.google.com/org/opencv/dnn/Dnn.html#DNN_TARGET_MYRIAD) |
| static int | [**DNN\_TARGET\_OPENCL**](http://docs.google.com/org/opencv/dnn/Dnn.html#DNN_TARGET_OPENCL) |
| static int | [**DNN\_TARGET\_OPENCL\_FP16**](http://docs.google.com/org/opencv/dnn/Dnn.html#DNN_TARGET_OPENCL_FP16) |

### Constructor SummaryConstructors

| Constructor and Description |
| --- |
| [**Dnn**](http://docs.google.com/org/opencv/dnn/Dnn.html#Dnn())() |

### Method SummaryMethods

| Modifier and Type | Method and Description |
| --- | --- |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**blobFromImage**](http://docs.google.com/org/opencv/dnn/Dnn.html#blobFromImage(org.opencv.core.Mat))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image) Creates 4-dimensional blob from image. |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**blobFromImage**](http://docs.google.com/org/opencv/dnn/Dnn.html#blobFromImage(org.opencv.core.Mat,%20double))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, double scalefactor) Creates 4-dimensional blob from image. |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**blobFromImage**](http://docs.google.com/org/opencv/dnn/Dnn.html#blobFromImage(org.opencv.core.Mat,%20double,%20org.opencv.core.Size))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size) Creates 4-dimensional blob from image. |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**blobFromImage**](http://docs.google.com/org/opencv/dnn/Dnn.html#blobFromImage(org.opencv.core.Mat,%20double,%20org.opencv.core.Size,%20org.opencv.core.Scalar))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size, [Scalar](http://docs.google.com/org/opencv/core/Scalar.html) mean) Creates 4-dimensional blob from image. |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**blobFromImage**](http://docs.google.com/org/opencv/dnn/Dnn.html#blobFromImage(org.opencv.core.Mat,%20double,%20org.opencv.core.Size,%20org.opencv.core.Scalar,%20boolean))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size, [Scalar](http://docs.google.com/org/opencv/core/Scalar.html) mean, boolean swapRB) Creates 4-dimensional blob from image. |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**blobFromImage**](http://docs.google.com/org/opencv/dnn/Dnn.html#blobFromImage(org.opencv.core.Mat,%20double,%20org.opencv.core.Size,%20org.opencv.core.Scalar,%20boolean,%20boolean))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size, [Scalar](http://docs.google.com/org/opencv/core/Scalar.html) mean, boolean swapRB, boolean crop) Creates 4-dimensional blob from image. |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**blobFromImage**](http://docs.google.com/org/opencv/dnn/Dnn.html#blobFromImage(org.opencv.core.Mat,%20double,%20org.opencv.core.Size,%20org.opencv.core.Scalar,%20boolean,%20boolean,%20int))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size, [Scalar](http://docs.google.com/org/opencv/core/Scalar.html) mean, boolean swapRB, boolean crop, int ddepth) Creates 4-dimensional blob from image. |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**blobFromImages**](http://docs.google.com/org/opencv/dnn/Dnn.html#blobFromImages(java.util.List))(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> images) Creates 4-dimensional blob from series of images. |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**blobFromImages**](http://docs.google.com/org/opencv/dnn/Dnn.html#blobFromImages(java.util.List,%20double))(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> images, double scalefactor) Creates 4-dimensional blob from series of images. |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**blobFromImages**](http://docs.google.com/org/opencv/dnn/Dnn.html#blobFromImages(java.util.List,%20double,%20org.opencv.core.Size))(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> images, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size) Creates 4-dimensional blob from series of images. |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**blobFromImages**](http://docs.google.com/org/opencv/dnn/Dnn.html#blobFromImages(java.util.List,%20double,%20org.opencv.core.Size,%20org.opencv.core.Scalar))(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> images, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size, [Scalar](http://docs.google.com/org/opencv/core/Scalar.html) mean) Creates 4-dimensional blob from series of images. |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**blobFromImages**](http://docs.google.com/org/opencv/dnn/Dnn.html#blobFromImages(java.util.List,%20double,%20org.opencv.core.Size,%20org.opencv.core.Scalar,%20boolean))(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> images, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size, [Scalar](http://docs.google.com/org/opencv/core/Scalar.html) mean, boolean swapRB) Creates 4-dimensional blob from series of images. |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**blobFromImages**](http://docs.google.com/org/opencv/dnn/Dnn.html#blobFromImages(java.util.List,%20double,%20org.opencv.core.Size,%20org.opencv.core.Scalar,%20boolean,%20boolean))(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> images, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size, [Scalar](http://docs.google.com/org/opencv/core/Scalar.html) mean, boolean swapRB, boolean crop) Creates 4-dimensional blob from series of images. |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**blobFromImages**](http://docs.google.com/org/opencv/dnn/Dnn.html#blobFromImages(java.util.List,%20double,%20org.opencv.core.Size,%20org.opencv.core.Scalar,%20boolean,%20boolean,%20int))(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> images, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size, [Scalar](http://docs.google.com/org/opencv/core/Scalar.html) mean, boolean swapRB, boolean crop, int ddepth) Creates 4-dimensional blob from series of images. |
| static java.util.List<java.lang.Integer> | [**getAvailableTargets**](http://docs.google.com/org/opencv/dnn/Dnn.html#getAvailableTargets(int))(int be) |
| static java.lang.String | [**getInferenceEngineBackendType**](http://docs.google.com/org/opencv/dnn/Dnn.html#getInferenceEngineBackendType())() Returns Inference Engine internal backend API. |
| static java.lang.String | [**getInferenceEngineCPUType**](http://docs.google.com/org/opencv/dnn/Dnn.html#getInferenceEngineCPUType())() Returns Inference Engine CPU type. |
| static java.lang.String | [**getInferenceEngineVPUType**](http://docs.google.com/org/opencv/dnn/Dnn.html#getInferenceEngineVPUType())() Returns Inference Engine VPU type. |
| static void | [**imagesFromBlob**](http://docs.google.com/org/opencv/dnn/Dnn.html#imagesFromBlob(org.opencv.core.Mat,%20java.util.List))([Mat](http://docs.google.com/org/opencv/core/Mat.html) blob\_, java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> images\_) Parse a 4D blob and output the images it contains as 2D arrays through a simpler data structure (std::vector<cv::Mat>). |
| static void | [**NMSBoxes**](http://docs.google.com/org/opencv/dnn/Dnn.html#NMSBoxes(org.opencv.core.MatOfRect2d,%20org.opencv.core.MatOfFloat,%20float,%20float,%20org.opencv.core.MatOfInt))([MatOfRect2d](http://docs.google.com/org/opencv/core/MatOfRect2d.html) bboxes, [MatOfFloat](http://docs.google.com/org/opencv/core/MatOfFloat.html) scores, float score\_threshold, float nms\_threshold, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) indices) Performs non maximum suppression given boxes and corresponding scores. |
| static void | [**NMSBoxes**](http://docs.google.com/org/opencv/dnn/Dnn.html#NMSBoxes(org.opencv.core.MatOfRect2d,%20org.opencv.core.MatOfFloat,%20float,%20float,%20org.opencv.core.MatOfInt,%20float))([MatOfRect2d](http://docs.google.com/org/opencv/core/MatOfRect2d.html) bboxes, [MatOfFloat](http://docs.google.com/org/opencv/core/MatOfFloat.html) scores, float score\_threshold, float nms\_threshold, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) indices, float eta) Performs non maximum suppression given boxes and corresponding scores. |
| static void | [**NMSBoxes**](http://docs.google.com/org/opencv/dnn/Dnn.html#NMSBoxes(org.opencv.core.MatOfRect2d,%20org.opencv.core.MatOfFloat,%20float,%20float,%20org.opencv.core.MatOfInt,%20float,%20int))([MatOfRect2d](http://docs.google.com/org/opencv/core/MatOfRect2d.html) bboxes, [MatOfFloat](http://docs.google.com/org/opencv/core/MatOfFloat.html) scores, float score\_threshold, float nms\_threshold, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) indices, float eta, int top\_k) Performs non maximum suppression given boxes and corresponding scores. |
| static void | [**NMSBoxesRotated**](http://docs.google.com/org/opencv/dnn/Dnn.html#NMSBoxesRotated(org.opencv.core.MatOfRotatedRect,%20org.opencv.core.MatOfFloat,%20float,%20float,%20org.opencv.core.MatOfInt))([MatOfRotatedRect](http://docs.google.com/org/opencv/core/MatOfRotatedRect.html) bboxes, [MatOfFloat](http://docs.google.com/org/opencv/core/MatOfFloat.html) scores, float score\_threshold, float nms\_threshold, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) indices) |
| static void | [**NMSBoxesRotated**](http://docs.google.com/org/opencv/dnn/Dnn.html#NMSBoxesRotated(org.opencv.core.MatOfRotatedRect,%20org.opencv.core.MatOfFloat,%20float,%20float,%20org.opencv.core.MatOfInt,%20float))([MatOfRotatedRect](http://docs.google.com/org/opencv/core/MatOfRotatedRect.html) bboxes, [MatOfFloat](http://docs.google.com/org/opencv/core/MatOfFloat.html) scores, float score\_threshold, float nms\_threshold, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) indices, float eta) |
| static void | [**NMSBoxesRotated**](http://docs.google.com/org/opencv/dnn/Dnn.html#NMSBoxesRotated(org.opencv.core.MatOfRotatedRect,%20org.opencv.core.MatOfFloat,%20float,%20float,%20org.opencv.core.MatOfInt,%20float,%20int))([MatOfRotatedRect](http://docs.google.com/org/opencv/core/MatOfRotatedRect.html) bboxes, [MatOfFloat](http://docs.google.com/org/opencv/core/MatOfFloat.html) scores, float score\_threshold, float nms\_threshold, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) indices, float eta, int top\_k) |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNet**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNet(java.lang.String))(java.lang.String model) Read deep learning network represented in one of the supported formats. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNet**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNet(java.lang.String,%20org.opencv.core.MatOfByte))(java.lang.String framework, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferModel) Read deep learning network represented in one of the supported formats. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNet**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNet(java.lang.String,%20org.opencv.core.MatOfByte,%20org.opencv.core.MatOfByte))(java.lang.String framework, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferModel, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferConfig) Read deep learning network represented in one of the supported formats. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNet**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNet(java.lang.String,%20java.lang.String))(java.lang.String model, java.lang.String config) Read deep learning network represented in one of the supported formats. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNet**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNet(java.lang.String,%20java.lang.String,%20java.lang.String))(java.lang.String model, java.lang.String config, java.lang.String framework) Read deep learning network represented in one of the supported formats. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromCaffe**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromCaffe(org.opencv.core.MatOfByte))([MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferProto) Reads a network model stored in Caffe model in memory. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromCaffe**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromCaffe(org.opencv.core.MatOfByte,%20org.opencv.core.MatOfByte))([MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferProto, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferModel) Reads a network model stored in Caffe model in memory. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromCaffe**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromCaffe(java.lang.String))(java.lang.String prototxt) Reads a network model stored in <a href="http://caffe.berkeleyvision.org">Caffe</a> framework's format. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromCaffe**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromCaffe(java.lang.String,%20java.lang.String))(java.lang.String prototxt, java.lang.String caffeModel) Reads a network model stored in <a href="http://caffe.berkeleyvision.org">Caffe</a> framework's format. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromDarknet**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromDarknet(org.opencv.core.MatOfByte))([MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferCfg) Reads a network model stored in <a href="https://pjreddie.com/darknet/">Darknet</a> model files. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromDarknet**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromDarknet(org.opencv.core.MatOfByte,%20org.opencv.core.MatOfByte))([MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferCfg, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferModel) Reads a network model stored in <a href="https://pjreddie.com/darknet/">Darknet</a> model files. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromDarknet**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromDarknet(java.lang.String))(java.lang.String cfgFile) Reads a network model stored in <a href="https://pjreddie.com/darknet/">Darknet</a> model files. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromDarknet**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromDarknet(java.lang.String,%20java.lang.String))(java.lang.String cfgFile, java.lang.String darknetModel) Reads a network model stored in <a href="https://pjreddie.com/darknet/">Darknet</a> model files. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromModelOptimizer**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromModelOptimizer(org.opencv.core.MatOfByte,%20org.opencv.core.MatOfByte))([MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferModelConfig, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferWeights) Load a network from Intel's Model Optimizer intermediate representation. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromModelOptimizer**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromModelOptimizer(java.lang.String,%20java.lang.String))(java.lang.String xml, java.lang.String bin) Load a network from Intel's Model Optimizer intermediate representation. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromONNX**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromONNX(org.opencv.core.MatOfByte))([MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) buffer) Reads a network model from <a href="https://onnx.ai/">ONNX</a> in-memory buffer. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromONNX**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromONNX(java.lang.String))(java.lang.String onnxFile) Reads a network model <a href="https://onnx.ai/">ONNX</a>. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromTensorflow**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromTensorflow(org.opencv.core.MatOfByte))([MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferModel) Reads a network model stored in <a href="https://www.tensorflow.org/">TensorFlow</a> framework's format. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromTensorflow**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromTensorflow(org.opencv.core.MatOfByte,%20org.opencv.core.MatOfByte))([MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferModel, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferConfig) Reads a network model stored in <a href="https://www.tensorflow.org/">TensorFlow</a> framework's format. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromTensorflow**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromTensorflow(java.lang.String))(java.lang.String model) Reads a network model stored in <a href="https://www.tensorflow.org/">TensorFlow</a> framework's format. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromTensorflow**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromTensorflow(java.lang.String,%20java.lang.String))(java.lang.String model, java.lang.String config) Reads a network model stored in <a href="https://www.tensorflow.org/">TensorFlow</a> framework's format. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromTorch**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromTorch(java.lang.String))(java.lang.String model) Reads a network model stored in <a href="http://torch.ch">Torch7</a> framework's format. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromTorch**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromTorch(java.lang.String,%20boolean))(java.lang.String model, boolean isBinary) Reads a network model stored in <a href="http://torch.ch">Torch7</a> framework's format. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readNetFromTorch**](http://docs.google.com/org/opencv/dnn/Dnn.html#readNetFromTorch(java.lang.String,%20boolean,%20boolean))(java.lang.String model, boolean isBinary, boolean evaluate) Reads a network model stored in <a href="http://torch.ch">Torch7</a> framework's format. |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**readTensorFromONNX**](http://docs.google.com/org/opencv/dnn/Dnn.html#readTensorFromONNX(java.lang.String))(java.lang.String path) Creates blob from .pb file. |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**readTorchBlob**](http://docs.google.com/org/opencv/dnn/Dnn.html#readTorchBlob(java.lang.String))(java.lang.String filename) Loads blob which was serialized as torch.Tensor object of Torch7 framework. |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**readTorchBlob**](http://docs.google.com/org/opencv/dnn/Dnn.html#readTorchBlob(java.lang.String,%20boolean))(java.lang.String filename, boolean isBinary) Loads blob which was serialized as torch.Tensor object of Torch7 framework. |
| static void | [**resetMyriadDevice**](http://docs.google.com/org/opencv/dnn/Dnn.html#resetMyriadDevice())() Release a Myriad device (binded by OpenCV). |
| static java.lang.String | [**setInferenceEngineBackendType**](http://docs.google.com/org/opencv/dnn/Dnn.html#setInferenceEngineBackendType(java.lang.String))(java.lang.String newBackendType) Specify Inference Engine internal backend API. |
| static void | [**shrinkCaffeModel**](http://docs.google.com/org/opencv/dnn/Dnn.html#shrinkCaffeModel(java.lang.String,%20java.lang.String))(java.lang.String src, java.lang.String dst) Convert all weights of Caffe network to half precision floating point. |
| static void | [**shrinkCaffeModel**](http://docs.google.com/org/opencv/dnn/Dnn.html#shrinkCaffeModel(java.lang.String,%20java.lang.String,%20java.util.List))(java.lang.String src, java.lang.String dst, java.util.List<java.lang.String> layersTypes) Convert all weights of Caffe network to half precision floating point. |
| static void | [**writeTextGraph**](http://docs.google.com/org/opencv/dnn/Dnn.html#writeTextGraph(java.lang.String,%20java.lang.String))(java.lang.String model, java.lang.String output) Create a text representation for a binary network stored in protocol buffer format. |

### Methods inherited from class java.lang.Objectequals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Field Detail

#### DNN\_BACKEND\_DEFAULT public static final int DNN\_BACKEND\_DEFAULTSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.dnn.Dnn.DNN_BACKEND_DEFAULT)

#### DNN\_BACKEND\_HALIDE public static final int DNN\_BACKEND\_HALIDESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.dnn.Dnn.DNN_BACKEND_HALIDE)

#### DNN\_BACKEND\_INFERENCE\_ENGINE public static final int DNN\_BACKEND\_INFERENCE\_ENGINESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.dnn.Dnn.DNN_BACKEND_INFERENCE_ENGINE)

#### DNN\_BACKEND\_OPENCV public static final int DNN\_BACKEND\_OPENCVSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.dnn.Dnn.DNN_BACKEND_OPENCV)

#### DNN\_TARGET\_CPU public static final int DNN\_TARGET\_CPUSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.dnn.Dnn.DNN_TARGET_CPU)

#### DNN\_TARGET\_FPGA public static final int DNN\_TARGET\_FPGASee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.dnn.Dnn.DNN_TARGET_FPGA)

#### DNN\_TARGET\_MYRIAD public static final int DNN\_TARGET\_MYRIADSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.dnn.Dnn.DNN_TARGET_MYRIAD)

#### DNN\_TARGET\_OPENCL public static final int DNN\_TARGET\_OPENCLSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.dnn.Dnn.DNN_TARGET_OPENCL)

#### DNN\_TARGET\_OPENCL\_FP16 public static final int DNN\_TARGET\_OPENCL\_FP16See Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.dnn.Dnn.DNN_TARGET_OPENCL_FP16)

### Constructor Detail

#### Dnn public Dnn()

### Method Detail

#### blobFromImage public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) blobFromImage([Mat](http://docs.google.com/org/opencv/core/Mat.html) image) Creates 4-dimensional blob from image. Optionally resizes and crops image from center, subtract mean values, scales values by scalefactor, swap Blue and Red channels.Parameters:image - input image (with 1-, 3- or 4-channels). to be in (mean-R, mean-G, mean-B) order if image has BGR ordering and swapRB is true. in 3-channel image is necessary. if crop is true, input image is resized so one side after resize is equal to corresponding dimension in size and another one is equal or larger. Then, crop from the center is performed. If crop is false, direct resize without cropping and preserving aspect ratio is performed. Returns:4-dimensional Mat with NCHW dimensions order.

#### blobFromImage public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) blobFromImage([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, double scalefactor) Creates 4-dimensional blob from image. Optionally resizes and crops image from center, subtract mean values, scales values by scalefactor, swap Blue and Red channels.Parameters:image - input image (with 1-, 3- or 4-channels). to be in (mean-R, mean-G, mean-B) order if image has BGR ordering and swapRB is true.scalefactor - multiplier for image values. in 3-channel image is necessary. if crop is true, input image is resized so one side after resize is equal to corresponding dimension in size and another one is equal or larger. Then, crop from the center is performed. If crop is false, direct resize without cropping and preserving aspect ratio is performed. Returns:4-dimensional Mat with NCHW dimensions order.

#### blobFromImage public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) blobFromImage([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size) Creates 4-dimensional blob from image. Optionally resizes and crops image from center, subtract mean values, scales values by scalefactor, swap Blue and Red channels.Parameters:image - input image (with 1-, 3- or 4-channels).size - spatial size for output image to be in (mean-R, mean-G, mean-B) order if image has BGR ordering and swapRB is true.scalefactor - multiplier for image values. in 3-channel image is necessary. if crop is true, input image is resized so one side after resize is equal to corresponding dimension in size and another one is equal or larger. Then, crop from the center is performed. If crop is false, direct resize without cropping and preserving aspect ratio is performed. Returns:4-dimensional Mat with NCHW dimensions order.

#### blobFromImage public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) blobFromImage([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size, [Scalar](http://docs.google.com/org/opencv/core/Scalar.html) mean) Creates 4-dimensional blob from image. Optionally resizes and crops image from center, subtract mean values, scales values by scalefactor, swap Blue and Red channels.Parameters:image - input image (with 1-, 3- or 4-channels).size - spatial size for output imagemean - scalar with mean values which are subtracted from channels. Values are intended to be in (mean-R, mean-G, mean-B) order if image has BGR ordering and swapRB is true.scalefactor - multiplier for image values. in 3-channel image is necessary. if crop is true, input image is resized so one side after resize is equal to corresponding dimension in size and another one is equal or larger. Then, crop from the center is performed. If crop is false, direct resize without cropping and preserving aspect ratio is performed. Returns:4-dimensional Mat with NCHW dimensions order.

#### blobFromImage public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) blobFromImage([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size, [Scalar](http://docs.google.com/org/opencv/core/Scalar.html) mean, boolean swapRB) Creates 4-dimensional blob from image. Optionally resizes and crops image from center, subtract mean values, scales values by scalefactor, swap Blue and Red channels.Parameters:image - input image (with 1-, 3- or 4-channels).size - spatial size for output imagemean - scalar with mean values which are subtracted from channels. Values are intended to be in (mean-R, mean-G, mean-B) order if image has BGR ordering and swapRB is true.scalefactor - multiplier for image values.swapRB - flag which indicates that swap first and last channels in 3-channel image is necessary. if crop is true, input image is resized so one side after resize is equal to corresponding dimension in size and another one is equal or larger. Then, crop from the center is performed. If crop is false, direct resize without cropping and preserving aspect ratio is performed. Returns:4-dimensional Mat with NCHW dimensions order.

#### blobFromImage public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) blobFromImage([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size, [Scalar](http://docs.google.com/org/opencv/core/Scalar.html) mean, boolean swapRB, boolean crop) Creates 4-dimensional blob from image. Optionally resizes and crops image from center, subtract mean values, scales values by scalefactor, swap Blue and Red channels.Parameters:image - input image (with 1-, 3- or 4-channels).size - spatial size for output imagemean - scalar with mean values which are subtracted from channels. Values are intended to be in (mean-R, mean-G, mean-B) order if image has BGR ordering and swapRB is true.scalefactor - multiplier for image values.swapRB - flag which indicates that swap first and last channels in 3-channel image is necessary.crop - flag which indicates whether image will be cropped after resize or not if crop is true, input image is resized so one side after resize is equal to corresponding dimension in size and another one is equal or larger. Then, crop from the center is performed. If crop is false, direct resize without cropping and preserving aspect ratio is performed. Returns:4-dimensional Mat with NCHW dimensions order.

#### blobFromImage public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) blobFromImage([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size, [Scalar](http://docs.google.com/org/opencv/core/Scalar.html) mean, boolean swapRB, boolean crop, int ddepth) Creates 4-dimensional blob from image. Optionally resizes and crops image from center, subtract mean values, scales values by scalefactor, swap Blue and Red channels.Parameters:image - input image (with 1-, 3- or 4-channels).size - spatial size for output imagemean - scalar with mean values which are subtracted from channels. Values are intended to be in (mean-R, mean-G, mean-B) order if image has BGR ordering and swapRB is true.scalefactor - multiplier for image values.swapRB - flag which indicates that swap first and last channels in 3-channel image is necessary.crop - flag which indicates whether image will be cropped after resize or notddepth - Depth of output blob. Choose CV\_32F or CV\_8U. if crop is true, input image is resized so one side after resize is equal to corresponding dimension in size and another one is equal or larger. Then, crop from the center is performed. If crop is false, direct resize without cropping and preserving aspect ratio is performed. Returns:4-dimensional Mat with NCHW dimensions order.

#### blobFromImages public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) blobFromImages(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> images) Creates 4-dimensional blob from series of images. Optionally resizes and crops images from center, subtract mean values, scales values by scalefactor, swap Blue and Red channels.Parameters:images - input images (all with 1-, 3- or 4-channels). to be in (mean-R, mean-G, mean-B) order if image has BGR ordering and swapRB is true. in 3-channel image is necessary. if crop is true, input image is resized so one side after resize is equal to corresponding dimension in size and another one is equal or larger. Then, crop from the center is performed. If crop is false, direct resize without cropping and preserving aspect ratio is performed. Returns:4-dimensional Mat with NCHW dimensions order.

#### blobFromImages public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) blobFromImages(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> images, double scalefactor) Creates 4-dimensional blob from series of images. Optionally resizes and crops images from center, subtract mean values, scales values by scalefactor, swap Blue and Red channels.Parameters:images - input images (all with 1-, 3- or 4-channels). to be in (mean-R, mean-G, mean-B) order if image has BGR ordering and swapRB is true.scalefactor - multiplier for images values. in 3-channel image is necessary. if crop is true, input image is resized so one side after resize is equal to corresponding dimension in size and another one is equal or larger. Then, crop from the center is performed. If crop is false, direct resize without cropping and preserving aspect ratio is performed. Returns:4-dimensional Mat with NCHW dimensions order.

#### blobFromImages public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) blobFromImages(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> images, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size) Creates 4-dimensional blob from series of images. Optionally resizes and crops images from center, subtract mean values, scales values by scalefactor, swap Blue and Red channels.Parameters:images - input images (all with 1-, 3- or 4-channels).size - spatial size for output image to be in (mean-R, mean-G, mean-B) order if image has BGR ordering and swapRB is true.scalefactor - multiplier for images values. in 3-channel image is necessary. if crop is true, input image is resized so one side after resize is equal to corresponding dimension in size and another one is equal or larger. Then, crop from the center is performed. If crop is false, direct resize without cropping and preserving aspect ratio is performed. Returns:4-dimensional Mat with NCHW dimensions order.

#### blobFromImages public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) blobFromImages(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> images, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size, [Scalar](http://docs.google.com/org/opencv/core/Scalar.html) mean) Creates 4-dimensional blob from series of images. Optionally resizes and crops images from center, subtract mean values, scales values by scalefactor, swap Blue and Red channels.Parameters:images - input images (all with 1-, 3- or 4-channels).size - spatial size for output imagemean - scalar with mean values which are subtracted from channels. Values are intended to be in (mean-R, mean-G, mean-B) order if image has BGR ordering and swapRB is true.scalefactor - multiplier for images values. in 3-channel image is necessary. if crop is true, input image is resized so one side after resize is equal to corresponding dimension in size and another one is equal or larger. Then, crop from the center is performed. If crop is false, direct resize without cropping and preserving aspect ratio is performed. Returns:4-dimensional Mat with NCHW dimensions order.

#### blobFromImages public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) blobFromImages(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> images, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size, [Scalar](http://docs.google.com/org/opencv/core/Scalar.html) mean, boolean swapRB) Creates 4-dimensional blob from series of images. Optionally resizes and crops images from center, subtract mean values, scales values by scalefactor, swap Blue and Red channels.Parameters:images - input images (all with 1-, 3- or 4-channels).size - spatial size for output imagemean - scalar with mean values which are subtracted from channels. Values are intended to be in (mean-R, mean-G, mean-B) order if image has BGR ordering and swapRB is true.scalefactor - multiplier for images values.swapRB - flag which indicates that swap first and last channels in 3-channel image is necessary. if crop is true, input image is resized so one side after resize is equal to corresponding dimension in size and another one is equal or larger. Then, crop from the center is performed. If crop is false, direct resize without cropping and preserving aspect ratio is performed. Returns:4-dimensional Mat with NCHW dimensions order.

#### blobFromImages public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) blobFromImages(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> images, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size, [Scalar](http://docs.google.com/org/opencv/core/Scalar.html) mean, boolean swapRB, boolean crop) Creates 4-dimensional blob from series of images. Optionally resizes and crops images from center, subtract mean values, scales values by scalefactor, swap Blue and Red channels.Parameters:images - input images (all with 1-, 3- or 4-channels).size - spatial size for output imagemean - scalar with mean values which are subtracted from channels. Values are intended to be in (mean-R, mean-G, mean-B) order if image has BGR ordering and swapRB is true.scalefactor - multiplier for images values.swapRB - flag which indicates that swap first and last channels in 3-channel image is necessary.crop - flag which indicates whether image will be cropped after resize or not if crop is true, input image is resized so one side after resize is equal to corresponding dimension in size and another one is equal or larger. Then, crop from the center is performed. If crop is false, direct resize without cropping and preserving aspect ratio is performed. Returns:4-dimensional Mat with NCHW dimensions order.

#### blobFromImages public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) blobFromImages(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> images, double scalefactor, [Size](http://docs.google.com/org/opencv/core/Size.html) size, [Scalar](http://docs.google.com/org/opencv/core/Scalar.html) mean, boolean swapRB, boolean crop, int ddepth) Creates 4-dimensional blob from series of images. Optionally resizes and crops images from center, subtract mean values, scales values by scalefactor, swap Blue and Red channels.Parameters:images - input images (all with 1-, 3- or 4-channels).size - spatial size for output imagemean - scalar with mean values which are subtracted from channels. Values are intended to be in (mean-R, mean-G, mean-B) order if image has BGR ordering and swapRB is true.scalefactor - multiplier for images values.swapRB - flag which indicates that swap first and last channels in 3-channel image is necessary.crop - flag which indicates whether image will be cropped after resize or notddepth - Depth of output blob. Choose CV\_32F or CV\_8U. if crop is true, input image is resized so one side after resize is equal to corresponding dimension in size and another one is equal or larger. Then, crop from the center is performed. If crop is false, direct resize without cropping and preserving aspect ratio is performed. Returns:4-dimensional Mat with NCHW dimensions order.

#### getAvailableTargets public static java.util.List<java.lang.Integer> getAvailableTargets(int be)

#### getInferenceEngineBackendType public static java.lang.String getInferenceEngineBackendType() Returns Inference Engine internal backend API. See values of CV\_DNN\_BACKEND\_INFERENCE\_ENGINE\_\* macros. Default value is controlled through OPENCV\_DNN\_BACKEND\_INFERENCE\_ENGINE\_TYPE runtime parameter (environment variable).Returns:automatically generated

#### getInferenceEngineCPUType public static java.lang.String getInferenceEngineCPUType() Returns Inference Engine CPU type. Specify OpenVINO plugin: CPU or ARM.Returns:automatically generated

#### getInferenceEngineVPUType public static java.lang.String getInferenceEngineVPUType() Returns Inference Engine VPU type. See values of CV\_DNN\_INFERENCE\_ENGINE\_VPU\_TYPE\_\* macros.Returns:automatically generated

#### imagesFromBlob public static void imagesFromBlob([Mat](http://docs.google.com/org/opencv/core/Mat.html) blob\_, java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> images\_) Parse a 4D blob and output the images it contains as 2D arrays through a simpler data structure (std::vector<cv::Mat>).Parameters:blob\_ - 4 dimensional array (images, channels, height, width) in floating point precision (CV\_32F) from which you would like to extract the images.images\_ - array of 2D Mat containing the images extracted from the blob in floating point precision (CV\_32F). They are non normalized neither mean added. The number of returned images equals the first dimension of the blob (batch size). Every image has a number of channels equals to the second dimension of the blob (depth).

#### NMSBoxes public static void NMSBoxes([MatOfRect2d](http://docs.google.com/org/opencv/core/MatOfRect2d.html) bboxes, [MatOfFloat](http://docs.google.com/org/opencv/core/MatOfFloat.html) scores, float score\_threshold, float nms\_threshold, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) indices) Performs non maximum suppression given boxes and corresponding scores.Parameters:bboxes - a set of bounding boxes to apply NMS.scores - a set of corresponding confidences.score\_threshold - a threshold used to filter boxes by score.nms\_threshold - a threshold used in non maximum suppression.indices - the kept indices of bboxes after NMS.

#### NMSBoxes public static void NMSBoxes([MatOfRect2d](http://docs.google.com/org/opencv/core/MatOfRect2d.html) bboxes, [MatOfFloat](http://docs.google.com/org/opencv/core/MatOfFloat.html) scores, float score\_threshold, float nms\_threshold, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) indices, float eta) Performs non maximum suppression given boxes and corresponding scores.Parameters:bboxes - a set of bounding boxes to apply NMS.scores - a set of corresponding confidences.score\_threshold - a threshold used to filter boxes by score.nms\_threshold - a threshold used in non maximum suppression.indices - the kept indices of bboxes after NMS.eta - a coefficient in adaptive threshold formula: \(nms\\_threshold\_{i+1}=eta\cdot nms\\_threshold\_i\).

#### NMSBoxes public static void NMSBoxes([MatOfRect2d](http://docs.google.com/org/opencv/core/MatOfRect2d.html) bboxes, [MatOfFloat](http://docs.google.com/org/opencv/core/MatOfFloat.html) scores, float score\_threshold, float nms\_threshold, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) indices, float eta, int top\_k) Performs non maximum suppression given boxes and corresponding scores.Parameters:bboxes - a set of bounding boxes to apply NMS.scores - a set of corresponding confidences.score\_threshold - a threshold used to filter boxes by score.nms\_threshold - a threshold used in non maximum suppression.indices - the kept indices of bboxes after NMS.eta - a coefficient in adaptive threshold formula: \(nms\\_threshold\_{i+1}=eta\cdot nms\\_threshold\_i\).top\_k - if >0, keep at most top\_k picked indices.

#### NMSBoxesRotated public static void NMSBoxesRotated([MatOfRotatedRect](http://docs.google.com/org/opencv/core/MatOfRotatedRect.html) bboxes, [MatOfFloat](http://docs.google.com/org/opencv/core/MatOfFloat.html) scores, float score\_threshold, float nms\_threshold, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) indices)

#### NMSBoxesRotated public static void NMSBoxesRotated([MatOfRotatedRect](http://docs.google.com/org/opencv/core/MatOfRotatedRect.html) bboxes, [MatOfFloat](http://docs.google.com/org/opencv/core/MatOfFloat.html) scores, float score\_threshold, float nms\_threshold, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) indices, float eta)

#### NMSBoxesRotated public static void NMSBoxesRotated([MatOfRotatedRect](http://docs.google.com/org/opencv/core/MatOfRotatedRect.html) bboxes, [MatOfFloat](http://docs.google.com/org/opencv/core/MatOfFloat.html) scores, float score\_threshold, float nms\_threshold, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) indices, float eta, int top\_k)

#### readNet public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNet(java.lang.String model) Read deep learning network represented in one of the supported formats.Parameters:model - Binary file contains trained weights. The following file extensions are expected for models from different frameworks: \* \*.caffemodel (Caffe, http://caffe.berkeleyvision.org/) \* \*.pb (TensorFlow, https://www.tensorflow.org/) \* \*.t7 | \*.net (Torch, http://torch.ch/) \* \*.weights (Darknet, https://pjreddie.com/darknet/) \* \*.bin (DLDT, https://software.intel.com/openvino-toolkit) \* \*.onnx (ONNX, https://onnx.ai/) file with the following extensions: \* \*.prototxt (Caffe, http://caffe.berkeleyvision.org/) \* \*.pbtxt (TensorFlow, https://www.tensorflow.org/) \* \*.cfg (Darknet, https://pjreddie.com/darknet/) \* \*.xml (DLDT, https://software.intel.com/openvino-toolkit) Returns:Net object. This function automatically detects an origin framework of trained model and calls an appropriate function such REF: readNetFromCaffe, REF: readNetFromTensorflow, REF: readNetFromTorch or REF: readNetFromDarknet. An order of model and config arguments does not matter.

#### readNet public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNet(java.lang.String framework, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferModel) Read deep learning network represented in one of the supported formats. This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.Parameters:framework - Name of origin framework.bufferModel - A buffer with a content of binary file with weights Returns:Net object.

#### readNet public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNet(java.lang.String framework, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferModel, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferConfig) Read deep learning network represented in one of the supported formats. This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.Parameters:framework - Name of origin framework.bufferModel - A buffer with a content of binary file with weightsbufferConfig - A buffer with a content of text file contains network configuration. Returns:Net object.

#### readNet public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNet(java.lang.String model, java.lang.String config) Read deep learning network represented in one of the supported formats.Parameters:model - Binary file contains trained weights. The following file extensions are expected for models from different frameworks: \* \*.caffemodel (Caffe, http://caffe.berkeleyvision.org/) \* \*.pb (TensorFlow, https://www.tensorflow.org/) \* \*.t7 | \*.net (Torch, http://torch.ch/) \* \*.weights (Darknet, https://pjreddie.com/darknet/) \* \*.bin (DLDT, https://software.intel.com/openvino-toolkit) \* \*.onnx (ONNX, https://onnx.ai/)config - Text file contains network configuration. It could be a file with the following extensions: \* \*.prototxt (Caffe, http://caffe.berkeleyvision.org/) \* \*.pbtxt (TensorFlow, https://www.tensorflow.org/) \* \*.cfg (Darknet, https://pjreddie.com/darknet/) \* \*.xml (DLDT, https://software.intel.com/openvino-toolkit) Returns:Net object. This function automatically detects an origin framework of trained model and calls an appropriate function such REF: readNetFromCaffe, REF: readNetFromTensorflow, REF: readNetFromTorch or REF: readNetFromDarknet. An order of model and config arguments does not matter.

#### readNet public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNet(java.lang.String model, java.lang.String config, java.lang.String framework) Read deep learning network represented in one of the supported formats.Parameters:model - Binary file contains trained weights. The following file extensions are expected for models from different frameworks: \* \*.caffemodel (Caffe, http://caffe.berkeleyvision.org/) \* \*.pb (TensorFlow, https://www.tensorflow.org/) \* \*.t7 | \*.net (Torch, http://torch.ch/) \* \*.weights (Darknet, https://pjreddie.com/darknet/) \* \*.bin (DLDT, https://software.intel.com/openvino-toolkit) \* \*.onnx (ONNX, https://onnx.ai/)config - Text file contains network configuration. It could be a file with the following extensions: \* \*.prototxt (Caffe, http://caffe.berkeleyvision.org/) \* \*.pbtxt (TensorFlow, https://www.tensorflow.org/) \* \*.cfg (Darknet, https://pjreddie.com/darknet/) \* \*.xml (DLDT, https://software.intel.com/openvino-toolkit)framework - Explicit framework name tag to determine a format. Returns:Net object. This function automatically detects an origin framework of trained model and calls an appropriate function such REF: readNetFromCaffe, REF: readNetFromTensorflow, REF: readNetFromTorch or REF: readNetFromDarknet. An order of model and config arguments does not matter.

#### readNetFromCaffe public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromCaffe([MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferProto) Reads a network model stored in Caffe model in memory.Parameters:bufferProto - buffer containing the content of the .prototxt file Returns:Net object.

#### readNetFromCaffe public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromCaffe([MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferProto, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferModel) Reads a network model stored in Caffe model in memory.Parameters:bufferProto - buffer containing the content of the .prototxt filebufferModel - buffer containing the content of the .caffemodel file Returns:Net object.

#### readNetFromCaffe public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromCaffe(java.lang.String prototxt) Reads a network model stored in <a href="http://caffe.berkeleyvision.org">Caffe</a> framework's format.Parameters:prototxt - path to the .prototxt file with text description of the network architecture. Returns:Net object.

#### readNetFromCaffe public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromCaffe(java.lang.String prototxt, java.lang.String caffeModel) Reads a network model stored in <a href="http://caffe.berkeleyvision.org">Caffe</a> framework's format.Parameters:prototxt - path to the .prototxt file with text description of the network architecture.caffeModel - path to the .caffemodel file with learned network. Returns:Net object.

#### readNetFromDarknet public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromDarknet([MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferCfg) Reads a network model stored in <a href="https://pjreddie.com/darknet/">Darknet</a> model files.Parameters:bufferCfg - A buffer contains a content of .cfg file with text description of the network architecture. Returns:Net object.

#### readNetFromDarknet public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromDarknet([MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferCfg, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferModel) Reads a network model stored in <a href="https://pjreddie.com/darknet/">Darknet</a> model files.Parameters:bufferCfg - A buffer contains a content of .cfg file with text description of the network architecture.bufferModel - A buffer contains a content of .weights file with learned network. Returns:Net object.

#### readNetFromDarknet public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromDarknet(java.lang.String cfgFile) Reads a network model stored in <a href="https://pjreddie.com/darknet/">Darknet</a> model files.Parameters:cfgFile - path to the .cfg file with text description of the network architecture. Returns:Network object that ready to do forward, throw an exception in failure cases.

#### readNetFromDarknet public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromDarknet(java.lang.String cfgFile, java.lang.String darknetModel) Reads a network model stored in <a href="https://pjreddie.com/darknet/">Darknet</a> model files.Parameters:cfgFile - path to the .cfg file with text description of the network architecture.darknetModel - path to the .weights file with learned network. Returns:Network object that ready to do forward, throw an exception in failure cases.

#### readNetFromModelOptimizer public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromModelOptimizer([MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferModelConfig, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferWeights) Load a network from Intel's Model Optimizer intermediate representation.Parameters:bufferModelConfig - Buffer contains XML configuration with network's topology.bufferWeights - Buffer contains binary data with trained weights. Returns:Net object. Networks imported from Intel's Model Optimizer are launched in Intel's Inference Engine backend.

#### readNetFromModelOptimizer public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromModelOptimizer(java.lang.String xml, java.lang.String bin) Load a network from Intel's Model Optimizer intermediate representation.Parameters:xml - XML configuration file with network's topology.bin - Binary file with trained weights. Returns:Net object. Networks imported from Intel's Model Optimizer are launched in Intel's Inference Engine backend.

#### readNetFromONNX public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromONNX([MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) buffer) Reads a network model from <a href="https://onnx.ai/">ONNX</a> in-memory buffer.Parameters:buffer - in-memory buffer that stores the ONNX model bytes. Returns:Network object that ready to do forward, throw an exception in failure cases.

#### readNetFromONNX public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromONNX(java.lang.String onnxFile) Reads a network model <a href="https://onnx.ai/">ONNX</a>.Parameters:onnxFile - path to the .onnx file with text description of the network architecture. Returns:Network object that ready to do forward, throw an exception in failure cases.

#### readNetFromTensorflow public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromTensorflow([MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferModel) Reads a network model stored in <a href="https://www.tensorflow.org/">TensorFlow</a> framework's format.Parameters:bufferModel - buffer containing the content of the pb file Returns:Net object.

#### readNetFromTensorflow public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromTensorflow([MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferModel, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferConfig) Reads a network model stored in <a href="https://www.tensorflow.org/">TensorFlow</a> framework's format.Parameters:bufferModel - buffer containing the content of the pb filebufferConfig - buffer containing the content of the pbtxt file Returns:Net object.

#### readNetFromTensorflow public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromTensorflow(java.lang.String model) Reads a network model stored in <a href="https://www.tensorflow.org/">TensorFlow</a> framework's format.Parameters:model - path to the .pb file with binary protobuf description of the network architecture Resulting Net object is built by text graph using weights from a binary one that let us make it more flexible. Returns:Net object.

#### readNetFromTensorflow public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromTensorflow(java.lang.String model, java.lang.String config) Reads a network model stored in <a href="https://www.tensorflow.org/">TensorFlow</a> framework's format.Parameters:model - path to the .pb file with binary protobuf description of the network architectureconfig - path to the .pbtxt file that contains text graph definition in protobuf format. Resulting Net object is built by text graph using weights from a binary one that let us make it more flexible. Returns:Net object.

#### readNetFromTorch public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromTorch(java.lang.String model) Reads a network model stored in <a href="http://torch.ch">Torch7</a> framework's format.Parameters:model - path to the file, dumped from Torch by using torch.save() function. Returns:Net object. **Note:** Ascii mode of Torch serializer is more preferable, because binary mode extensively use long type of C language, which has various bit-length on different systems. The loading file must contain serialized <a href="https://github.com/torch/nn/blob/master/doc/module.md">nn.Module</a> object with importing network. Try to eliminate a custom objects from serialazing data to avoid importing errors. List of supported layers (i.e. object instances derived from Torch nn.Module class): - nn.Sequential - nn.Parallel - nn.Concat - nn.Linear - nn.SpatialConvolution - nn.SpatialMaxPooling, nn.SpatialAveragePooling - nn.ReLU, nn.TanH, nn.Sigmoid - nn.Reshape - nn.SoftMax, nn.LogSoftMax Also some equivalents of these classes from cunn, cudnn, and fbcunn may be successfully imported.

#### readNetFromTorch public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromTorch(java.lang.String model, boolean isBinary) Reads a network model stored in <a href="http://torch.ch">Torch7</a> framework's format.Parameters:model - path to the file, dumped from Torch by using torch.save() function.isBinary - specifies whether the network was serialized in ascii mode or binary. Returns:Net object. **Note:** Ascii mode of Torch serializer is more preferable, because binary mode extensively use long type of C language, which has various bit-length on different systems. The loading file must contain serialized <a href="https://github.com/torch/nn/blob/master/doc/module.md">nn.Module</a> object with importing network. Try to eliminate a custom objects from serialazing data to avoid importing errors. List of supported layers (i.e. object instances derived from Torch nn.Module class): - nn.Sequential - nn.Parallel - nn.Concat - nn.Linear - nn.SpatialConvolution - nn.SpatialMaxPooling, nn.SpatialAveragePooling - nn.ReLU, nn.TanH, nn.Sigmoid - nn.Reshape - nn.SoftMax, nn.LogSoftMax Also some equivalents of these classes from cunn, cudnn, and fbcunn may be successfully imported.

#### readNetFromTorch public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readNetFromTorch(java.lang.String model, boolean isBinary, boolean evaluate) Reads a network model stored in <a href="http://torch.ch">Torch7</a> framework's format.Parameters:model - path to the file, dumped from Torch by using torch.save() function.isBinary - specifies whether the network was serialized in ascii mode or binary.evaluate - specifies testing phase of network. If true, it's similar to evaluate() method in Torch. Returns:Net object. **Note:** Ascii mode of Torch serializer is more preferable, because binary mode extensively use long type of C language, which has various bit-length on different systems. The loading file must contain serialized <a href="https://github.com/torch/nn/blob/master/doc/module.md">nn.Module</a> object with importing network. Try to eliminate a custom objects from serialazing data to avoid importing errors. List of supported layers (i.e. object instances derived from Torch nn.Module class): - nn.Sequential - nn.Parallel - nn.Concat - nn.Linear - nn.SpatialConvolution - nn.SpatialMaxPooling, nn.SpatialAveragePooling - nn.ReLU, nn.TanH, nn.Sigmoid - nn.Reshape - nn.SoftMax, nn.LogSoftMax Also some equivalents of these classes from cunn, cudnn, and fbcunn may be successfully imported.

#### readTensorFromONNX public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) readTensorFromONNX(java.lang.String path) Creates blob from .pb file.Parameters:path - to the .pb file with input tensor. Returns:Mat.

#### readTorchBlob public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) readTorchBlob(java.lang.String filename) Loads blob which was serialized as torch.Tensor object of Torch7 framework. WARNING: This function has the same limitations as readNetFromTorch().Parameters:filename - automatically generated Returns:automatically generated

#### readTorchBlob public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) readTorchBlob(java.lang.String filename, boolean isBinary) Loads blob which was serialized as torch.Tensor object of Torch7 framework. WARNING: This function has the same limitations as readNetFromTorch().Parameters:filename - automatically generatedisBinary - automatically generated Returns:automatically generated

#### resetMyriadDevice public static void resetMyriadDevice() Release a Myriad device (binded by OpenCV). Single Myriad device cannot be shared across multiple processes which uses Inference Engine's Myriad plugin.

#### setInferenceEngineBackendType public static java.lang.String setInferenceEngineBackendType(java.lang.String newBackendType) Specify Inference Engine internal backend API. See values of CV\_DNN\_BACKEND\_INFERENCE\_ENGINE\_\* macros.Parameters:newBackendType - automatically generated Returns:previous value of internal backend API

#### shrinkCaffeModel public static void shrinkCaffeModel(java.lang.String src, java.lang.String dst) Convert all weights of Caffe network to half precision floating point.Parameters:src - Path to origin model from Caffe framework contains single precision floating point weights (usually has .caffemodel extension).dst - Path to destination model with updated weights. By default, converts only Convolutional and Fully-Connected layers' weights. **Note:** Shrinked model has no origin float32 weights so it can't be used in origin Caffe framework anymore. However the structure of data is taken from NVidia's Caffe fork: https://github.com/NVIDIA/caffe. So the resulting model may be used there.

#### shrinkCaffeModel public static void shrinkCaffeModel(java.lang.String src, java.lang.String dst, java.util.List<java.lang.String> layersTypes) Convert all weights of Caffe network to half precision floating point.Parameters:src - Path to origin model from Caffe framework contains single precision floating point weights (usually has .caffemodel extension).dst - Path to destination model with updated weights.layersTypes - Set of layers types which parameters will be converted. By default, converts only Convolutional and Fully-Connected layers' weights. **Note:** Shrinked model has no origin float32 weights so it can't be used in origin Caffe framework anymore. However the structure of data is taken from NVidia's Caffe fork: https://github.com/NVIDIA/caffe. So the resulting model may be used there.

#### writeTextGraph public static void writeTextGraph(java.lang.String model, java.lang.String output) Create a text representation for a binary network stored in protocol buffer format.Parameters:model - A path to binary network.output - A path to output text file to be created. **Note:** To reduce output file size, trained weights are not included.

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Tree](http://docs.google.com/package-tree.html)
* [Index](http://docs.google.com/index-all.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/org/opencv/dnn/DictValue.html)
* [Next Class](http://docs.google.com/org/opencv/dnn/Layer.html)
* [Frames](http://docs.google.com/index.html?org/opencv/dnn/Dnn.html)
* [No Frames](http://docs.google.com/Dnn.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Field](#3znysh7) |
* [Constr](#2et92p0) |
* [Method](#tyjcwt)
* Detail:
* [Field](#1t3h5sf) |
* [Constr](#2jxsxqh) |
* [Method](#3j2qqm3)

Generated on 2021-04-02 03:15:03 / OpenCV 3.4.14