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/Layer.html)
* Next Class
* [Frames](http://docs.google.com/index.html?org/opencv/dnn/Net.html)
* [No Frames](http://docs.google.com/Net.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#3znysh7) |
* [Method](#2et92p0)
* Detail:
* Field |
* [Constr](#3dy6vkm) |
* [Method](#4d34og8)

org.opencv.dnn

## Class Net

* java.lang.Object
  + org.opencv.dnn.Net
* public class Net  
  extends java.lang.Object  
  This class allows to create and manipulate comprehensive artificial neural networks. Neural network is presented as directed acyclic graph (DAG), where vertices are Layer instances, and edges specify relationships between layers inputs and outputs. Each network layer has unique integer id and unique string name inside its network. LayerId can store either layer name or layer id. This class supports reference counting of its instances, i. e. copies point to the same instance.

### Constructor SummaryConstructors

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

### Method SummaryMethods

| Modifier and Type | Method and Description |
| --- | --- |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**\_\_fromPtr\_\_**](http://docs.google.com/org/opencv/dnn/Net.html#__fromPtr__(long))(long addr) |
| void | [**connect**](http://docs.google.com/org/opencv/dnn/Net.html#connect(java.lang.String,%20java.lang.String))(java.lang.String outPin, java.lang.String inpPin) Connects output of the first layer to input of the second layer. |
| java.lang.String | [**dump**](http://docs.google.com/org/opencv/dnn/Net.html#dump())() Dump net to String |
| void | [**dumpToFile**](http://docs.google.com/org/opencv/dnn/Net.html#dumpToFile(java.lang.String))(java.lang.String path) Dump net structure, hyperparameters, backend, target and fusion to dot file |
| boolean | [**empty**](http://docs.google.com/org/opencv/dnn/Net.html#empty())() Returns true if there are no layers in the network. |
| void | [**enableFusion**](http://docs.google.com/org/opencv/dnn/Net.html#enableFusion(boolean))(boolean fusion) Enables or disables layer fusion in the network. |
| [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**forward**](http://docs.google.com/org/opencv/dnn/Net.html#forward())() Runs forward pass to compute output of layer with name outputName. |
| void | [**forward**](http://docs.google.com/org/opencv/dnn/Net.html#forward(java.util.List))(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> outputBlobs) Runs forward pass to compute output of layer with name outputName. |
| void | [**forward**](http://docs.google.com/org/opencv/dnn/Net.html#forward(java.util.List,%20java.util.List))(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> outputBlobs, java.util.List<java.lang.String> outBlobNames) Runs forward pass to compute outputs of layers listed in outBlobNames. |
| void | [**forward**](http://docs.google.com/org/opencv/dnn/Net.html#forward(java.util.List,%20java.lang.String))(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> outputBlobs, java.lang.String outputName) Runs forward pass to compute output of layer with name outputName. |
| [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**forward**](http://docs.google.com/org/opencv/dnn/Net.html#forward(java.lang.String))(java.lang.String outputName) Runs forward pass to compute output of layer with name outputName. |
| long | [**getFLOPS**](http://docs.google.com/org/opencv/dnn/Net.html#getFLOPS(int,%20java.util.List))(int layerId, java.util.List<[MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html)> netInputShapes) |
| long | [**getFLOPS**](http://docs.google.com/org/opencv/dnn/Net.html#getFLOPS(int,%20org.opencv.core.MatOfInt))(int layerId, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) netInputShape) |
| long | [**getFLOPS**](http://docs.google.com/org/opencv/dnn/Net.html#getFLOPS(java.util.List))(java.util.List<[MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html)> netInputShapes) Computes FLOP for whole loaded model with specified input shapes. |
| long | [**getFLOPS**](http://docs.google.com/org/opencv/dnn/Net.html#getFLOPS(org.opencv.core.MatOfInt))([MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) netInputShape) |
| [Layer](http://docs.google.com/org/opencv/dnn/Layer.html) | [**getLayer**](http://docs.google.com/org/opencv/dnn/Net.html#getLayer(org.opencv.dnn.DictValue))([DictValue](http://docs.google.com/org/opencv/dnn/DictValue.html) layerId) Returns pointer to layer with specified id or name which the network use. |
| int | [**getLayerId**](http://docs.google.com/org/opencv/dnn/Net.html#getLayerId(java.lang.String))(java.lang.String layer) Converts string name of the layer to the integer identifier. |
| java.util.List<java.lang.String> | [**getLayerNames**](http://docs.google.com/org/opencv/dnn/Net.html#getLayerNames())() |
| int | [**getLayersCount**](http://docs.google.com/org/opencv/dnn/Net.html#getLayersCount(java.lang.String))(java.lang.String layerType) Returns count of layers of specified type. |
| void | [**getLayerTypes**](http://docs.google.com/org/opencv/dnn/Net.html#getLayerTypes(java.util.List))(java.util.List<java.lang.String> layersTypes) Returns list of types for layer used in model. |
| void | [**getMemoryConsumption**](http://docs.google.com/org/opencv/dnn/Net.html#getMemoryConsumption(int,%20java.util.List,%20long%5B%5D,%20long%5B%5D))(int layerId, java.util.List<[MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html)> netInputShapes, long[] weights, long[] blobs) |
| void | [**getMemoryConsumption**](http://docs.google.com/org/opencv/dnn/Net.html#getMemoryConsumption(int,%20org.opencv.core.MatOfInt,%20long%5B%5D,%20long%5B%5D))(int layerId, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) netInputShape, long[] weights, long[] blobs) |
| void | [**getMemoryConsumption**](http://docs.google.com/org/opencv/dnn/Net.html#getMemoryConsumption(org.opencv.core.MatOfInt,%20long%5B%5D,%20long%5B%5D))([MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) netInputShape, long[] weights, long[] blobs) |
| long | [**getNativeObjAddr**](http://docs.google.com/org/opencv/dnn/Net.html#getNativeObjAddr())() |
| [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**getParam**](http://docs.google.com/org/opencv/dnn/Net.html#getParam(org.opencv.dnn.DictValue))([DictValue](http://docs.google.com/org/opencv/dnn/DictValue.html) layer) Returns parameter blob of the layer. |
| [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**getParam**](http://docs.google.com/org/opencv/dnn/Net.html#getParam(org.opencv.dnn.DictValue,%20int))([DictValue](http://docs.google.com/org/opencv/dnn/DictValue.html) layer, int numParam) Returns parameter blob of the layer. |
| long | [**getPerfProfile**](http://docs.google.com/org/opencv/dnn/Net.html#getPerfProfile(org.opencv.core.MatOfDouble))([MatOfDouble](http://docs.google.com/org/opencv/core/MatOfDouble.html) timings) Returns overall time for inference and timings (in ticks) for layers. |
| [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) | [**getUnconnectedOutLayers**](http://docs.google.com/org/opencv/dnn/Net.html#getUnconnectedOutLayers())() Returns indexes of layers with unconnected outputs. |
| java.util.List<java.lang.String> | [**getUnconnectedOutLayersNames**](http://docs.google.com/org/opencv/dnn/Net.html#getUnconnectedOutLayersNames())() Returns names of layers with unconnected outputs. |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readFromModelOptimizer**](http://docs.google.com/org/opencv/dnn/Net.html#readFromModelOptimizer(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) Create a network from Intel's Model Optimizer in-memory buffers with intermediate representation (IR). |
| static [Net](http://docs.google.com/org/opencv/dnn/Net.html) | [**readFromModelOptimizer**](http://docs.google.com/org/opencv/dnn/Net.html#readFromModelOptimizer(java.lang.String,%20java.lang.String))(java.lang.String xml, java.lang.String bin) Create a network from Intel's Model Optimizer intermediate representation (IR). |
| void | [**setHalideScheduler**](http://docs.google.com/org/opencv/dnn/Net.html#setHalideScheduler(java.lang.String))(java.lang.String scheduler) Compile Halide layers. |
| void | [**setInput**](http://docs.google.com/org/opencv/dnn/Net.html#setInput(org.opencv.core.Mat))([Mat](http://docs.google.com/org/opencv/core/Mat.html) blob) Sets the new input value for the network |
| void | [**setInput**](http://docs.google.com/org/opencv/dnn/Net.html#setInput(org.opencv.core.Mat,%20java.lang.String))([Mat](http://docs.google.com/org/opencv/core/Mat.html) blob, java.lang.String name) Sets the new input value for the network |
| void | [**setInput**](http://docs.google.com/org/opencv/dnn/Net.html#setInput(org.opencv.core.Mat,%20java.lang.String,%20double))([Mat](http://docs.google.com/org/opencv/core/Mat.html) blob, java.lang.String name, double scalefactor) Sets the new input value for the network |
| void | [**setInput**](http://docs.google.com/org/opencv/dnn/Net.html#setInput(org.opencv.core.Mat,%20java.lang.String,%20double,%20org.opencv.core.Scalar))([Mat](http://docs.google.com/org/opencv/core/Mat.html) blob, java.lang.String name, double scalefactor, [Scalar](http://docs.google.com/org/opencv/core/Scalar.html) mean) Sets the new input value for the network |
| void | [**setInputShape**](http://docs.google.com/org/opencv/dnn/Net.html#setInputShape(java.lang.String,%20org.opencv.core.MatOfInt))(java.lang.String inputName, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) shape) Specify shape of network input. |
| void | [**setInputsNames**](http://docs.google.com/org/opencv/dnn/Net.html#setInputsNames(java.util.List))(java.util.List<java.lang.String> inputBlobNames) Sets outputs names of the network input pseudo layer. |
| void | [**setParam**](http://docs.google.com/org/opencv/dnn/Net.html#setParam(org.opencv.dnn.DictValue,%20int,%20org.opencv.core.Mat))([DictValue](http://docs.google.com/org/opencv/dnn/DictValue.html) layer, int numParam, [Mat](http://docs.google.com/org/opencv/core/Mat.html) blob) Sets the new value for the learned param of the layer. |
| void | [**setPreferableBackend**](http://docs.google.com/org/opencv/dnn/Net.html#setPreferableBackend(int))(int backendId) Ask network to use specific computation backend where it supported. |
| void | [**setPreferableTarget**](http://docs.google.com/org/opencv/dnn/Net.html#setPreferableTarget(int))(int targetId) Ask network to make computations on specific target device. |

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

### Constructor Detail

#### Net public Net()

### Method Detail

#### \_\_fromPtr\_\_ public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) \_\_fromPtr\_\_(long addr)

#### connect public void connect(java.lang.String outPin, java.lang.String inpPin) Connects output of the first layer to input of the second layer.Parameters:outPin - descriptor of the first layer output.inpPin - descriptor of the second layer input. Descriptors have the following template <DFN><layer\_name>[.input\_number]</DFN>: - the first part of the template <DFN>layer\_name</DFN> is string name of the added layer. If this part is empty then the network input pseudo layer will be used; - the second optional part of the template <DFN>input\_number</DFN> is either number of the layer input, either label one. If this part is omitted then the first layer input will be used. SEE: setNetInputs(), Layer::inputNameToIndex(), Layer::outputNameToIndex()

#### dump public java.lang.String dump() Dump net to StringReturns:String with structure, hyperparameters, backend, target and fusion Call method after setInput(). To see correct backend, target and fusion run after forward().

#### dumpToFile public void dumpToFile(java.lang.String path) Dump net structure, hyperparameters, backend, target and fusion to dot fileParameters:path - path to output file with .dot extension SEE: dump()

#### empty public boolean empty() Returns true if there are no layers in the network.Returns:automatically generated

#### enableFusion public void enableFusion(boolean fusion) Enables or disables layer fusion in the network.Parameters:fusion - true to enable the fusion, false to disable. The fusion is enabled by default.

#### forward public [Mat](http://docs.google.com/org/opencv/core/Mat.html) forward() Runs forward pass to compute output of layer with name outputName.Returns:blob for first output of specified layer. By default runs forward pass for the whole network.

#### forward public void forward(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> outputBlobs) Runs forward pass to compute output of layer with name outputName.Parameters:outputBlobs - contains all output blobs for specified layer. If outputName is empty, runs forward pass for the whole network.

#### forward public void forward(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> outputBlobs, java.util.List<java.lang.String> outBlobNames) Runs forward pass to compute outputs of layers listed in outBlobNames.Parameters:outputBlobs - contains blobs for first outputs of specified layers.outBlobNames - names for layers which outputs are needed to get

#### forward public void forward(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> outputBlobs, java.lang.String outputName) Runs forward pass to compute output of layer with name outputName.Parameters:outputBlobs - contains all output blobs for specified layer.outputName - name for layer which output is needed to get If outputName is empty, runs forward pass for the whole network.

#### forward public [Mat](http://docs.google.com/org/opencv/core/Mat.html) forward(java.lang.String outputName) Runs forward pass to compute output of layer with name outputName.Parameters:outputName - name for layer which output is needed to get Returns:blob for first output of specified layer. By default runs forward pass for the whole network.

#### getFLOPS public long getFLOPS(int layerId, java.util.List<[MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html)> netInputShapes)

#### getFLOPS public long getFLOPS(int layerId, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) netInputShape)

#### getFLOPS public long getFLOPS(java.util.List<[MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html)> netInputShapes) Computes FLOP for whole loaded model with specified input shapes.Parameters:netInputShapes - vector of shapes for all net inputs. Returns:computed FLOP.

#### getFLOPS public long getFLOPS([MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) netInputShape)

#### getLayer public [Layer](http://docs.google.com/org/opencv/dnn/Layer.html) getLayer([DictValue](http://docs.google.com/org/opencv/dnn/DictValue.html) layerId) Returns pointer to layer with specified id or name which the network use.Parameters:layerId - automatically generated Returns:automatically generated

#### getLayerId public int getLayerId(java.lang.String layer) Converts string name of the layer to the integer identifier.Parameters:layer - automatically generated Returns:id of the layer, or -1 if the layer wasn't found.

#### getLayerNames public java.util.List<java.lang.String> getLayerNames()

#### getLayersCount public int getLayersCount(java.lang.String layerType) Returns count of layers of specified type.Parameters:layerType - type. Returns:count of layers

#### getLayerTypes public void getLayerTypes(java.util.List<java.lang.String> layersTypes) Returns list of types for layer used in model.Parameters:layersTypes - output parameter for returning types.

#### getMemoryConsumption public void getMemoryConsumption(int layerId, java.util.List<[MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html)> netInputShapes, long[] weights, long[] blobs)

#### getMemoryConsumption public void getMemoryConsumption(int layerId, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) netInputShape, long[] weights, long[] blobs)

#### getMemoryConsumption public void getMemoryConsumption([MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) netInputShape, long[] weights, long[] blobs)

#### getNativeObjAddr public long getNativeObjAddr()

#### getParam public [Mat](http://docs.google.com/org/opencv/core/Mat.html) getParam([DictValue](http://docs.google.com/org/opencv/dnn/DictValue.html) layer) Returns parameter blob of the layer.Parameters:layer - name or id of the layer. SEE: Layer::blobs Returns:automatically generated

#### getParam public [Mat](http://docs.google.com/org/opencv/core/Mat.html) getParam([DictValue](http://docs.google.com/org/opencv/dnn/DictValue.html) layer, int numParam) Returns parameter blob of the layer.Parameters:layer - name or id of the layer.numParam - index of the layer parameter in the Layer::blobs array. SEE: Layer::blobs Returns:automatically generated

#### getPerfProfile public long getPerfProfile([MatOfDouble](http://docs.google.com/org/opencv/core/MatOfDouble.html) timings) Returns overall time for inference and timings (in ticks) for layers. Indexes in returned vector correspond to layers ids. Some layers can be fused with others, in this case zero ticks count will be return for that skipped layers.Parameters:timings - vector for tick timings for all layers. Returns:overall ticks for model inference.

#### getUnconnectedOutLayers public [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) getUnconnectedOutLayers() Returns indexes of layers with unconnected outputs.Returns:automatically generated

#### getUnconnectedOutLayersNames public java.util.List<java.lang.String> getUnconnectedOutLayersNames() Returns names of layers with unconnected outputs.Returns:automatically generated

#### readFromModelOptimizer public static [Net](http://docs.google.com/org/opencv/dnn/Net.html) readFromModelOptimizer([MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferModelConfig, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) bufferWeights) Create a network from Intel's Model Optimizer in-memory buffers with intermediate representation (IR).Parameters:bufferModelConfig - buffer with model's configuration.bufferWeights - buffer with model's trained weights. Returns:Net object.

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

#### setHalideScheduler public void setHalideScheduler(java.lang.String scheduler) Compile Halide layers.Parameters:scheduler - Path to YAML file with scheduling directives. SEE: setPreferableBackend Schedule layers that support Halide backend. Then compile them for specific target. For layers that not represented in scheduling file or if no manual scheduling used at all, automatic scheduling will be applied.

#### setInput public void setInput([Mat](http://docs.google.com/org/opencv/core/Mat.html) blob) Sets the new input value for the networkParameters:blob - A new blob. Should have CV\_32F or CV\_8U depth. SEE: connect(String, String) to know format of the descriptor. If scale or mean values are specified, a final input blob is computed as: \(input(n,c,h,w) = scalefactor \times (blob(n,c,h,w) - mean\_c)\)

#### setInput public void setInput([Mat](http://docs.google.com/org/opencv/core/Mat.html) blob, java.lang.String name) Sets the new input value for the networkParameters:blob - A new blob. Should have CV\_32F or CV\_8U depth.name - A name of input layer. SEE: connect(String, String) to know format of the descriptor. If scale or mean values are specified, a final input blob is computed as: \(input(n,c,h,w) = scalefactor \times (blob(n,c,h,w) - mean\_c)\)

#### setInput public void setInput([Mat](http://docs.google.com/org/opencv/core/Mat.html) blob, java.lang.String name, double scalefactor) Sets the new input value for the networkParameters:blob - A new blob. Should have CV\_32F or CV\_8U depth.name - A name of input layer.scalefactor - An optional normalization scale. SEE: connect(String, String) to know format of the descriptor. If scale or mean values are specified, a final input blob is computed as: \(input(n,c,h,w) = scalefactor \times (blob(n,c,h,w) - mean\_c)\)

#### setInput public void setInput([Mat](http://docs.google.com/org/opencv/core/Mat.html) blob, java.lang.String name, double scalefactor, [Scalar](http://docs.google.com/org/opencv/core/Scalar.html) mean) Sets the new input value for the networkParameters:blob - A new blob. Should have CV\_32F or CV\_8U depth.name - A name of input layer.scalefactor - An optional normalization scale.mean - An optional mean subtraction values. SEE: connect(String, String) to know format of the descriptor. If scale or mean values are specified, a final input blob is computed as: \(input(n,c,h,w) = scalefactor \times (blob(n,c,h,w) - mean\_c)\)

#### setInputShape public void setInputShape(java.lang.String inputName, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) shape) Specify shape of network input.Parameters:inputName - automatically generatedshape - automatically generated

#### setInputsNames public void setInputsNames(java.util.List<java.lang.String> inputBlobNames) Sets outputs names of the network input pseudo layer. Each net always has special own the network input pseudo layer with id=0. This layer stores the user blobs only and don't make any computations. In fact, this layer provides the only way to pass user data into the network. As any other layer, this layer can label its outputs and this function provides an easy way to do this.Parameters:inputBlobNames - automatically generated

#### setParam public void setParam([DictValue](http://docs.google.com/org/opencv/dnn/DictValue.html) layer, int numParam, [Mat](http://docs.google.com/org/opencv/core/Mat.html) blob) Sets the new value for the learned param of the layer.Parameters:layer - name or id of the layer.numParam - index of the layer parameter in the Layer::blobs array.blob - the new value. SEE: Layer::blobs **Note:** If shape of the new blob differs from the previous shape, then the following forward pass may fail.

#### setPreferableBackend public void setPreferableBackend(int backendId) Ask network to use specific computation backend where it supported.Parameters:backendId - backend identifier. SEE: Backend If OpenCV is compiled with Intel's Inference Engine library, DNN\_BACKEND\_DEFAULT means DNN\_BACKEND\_INFERENCE\_ENGINE. Otherwise it equals to DNN\_BACKEND\_OPENCV.

#### setPreferableTarget public void setPreferableTarget(int targetId) Ask network to make computations on specific target device.Parameters:targetId - target identifier. SEE: Target List of supported combinations backend / target: | | DNN\_BACKEND\_OPENCV | DNN\_BACKEND\_INFERENCE\_ENGINE | DNN\_BACKEND\_HALIDE | |------------------------|--------------------|------------------------------|--------------------| | DNN\_TARGET\_CPU | + | + | + | | DNN\_TARGET\_OPENCL | + | + | + | | DNN\_TARGET\_OPENCL\_FP16 | + | + | | | DNN\_TARGET\_MYRIAD | | + | | | DNN\_TARGET\_FPGA | | + | |

* [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/Layer.html)
* Next Class
* [Frames](http://docs.google.com/index.html?org/opencv/dnn/Net.html)
* [No Frames](http://docs.google.com/Net.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#3znysh7) |
* [Method](#2et92p0)
* Detail:
* Field |
* [Constr](#3dy6vkm) |
* [Method](#4d34og8)

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