Operator List

Issue 01

Date 2020-05-30

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Parameter Description

Parameter	Description
ni	Batch size
ci/co	Channel count
hi/ho/	Height
wi/wo	Width
sh/sw	Stride
kh/kw	Size of the convolution filter
window_h(window_y)/ window_w(window_x)	Window size
dh(dilation_h)/ dw(dilation_w)	Convolution dilation coefficient
FilterHDilation/ FilterWDilation	H/W dimension of the dilated filter
FilterH/FilterW	H/W dimension of the convolution weight
padWHead/padHHead	Pad head of the H/W dimension
PadWTail/padHTail	Pad tail of the H/W dimension
dilationsize	User-defined dilation coefficient
FilterSize	User-defined filter count
INT32_MAX	Maximum value that can be represented by data type int32
ALIGN	Roundup alignment
CEIL	Mapping to the least succeeding integer

2 Operator Boundaries

2.1 General Restrictions

Under the Caffe framework, if the input dimension count of an operator is not 4 and the **axis** parameter is available, its value cannot be negative.

2.2 Caffe Operator Boundaries

No.	Operator	Description	Boundary
1	Absval	Computes the absolute value of the input.	[Inputs] One input [Arguments] **engine**: (optional) enum, default to **0**, CAFFE = 1, CUDNN = 2 [Restrictions] None [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
2	Argmax	Computes the index of the maximum values.	[Inputs] One input [Arguments] • **out_max_val**: (optional) bool, default to **false** • **top_k**: (optional) unit32, default to **1** • **axis**: (optional) int32 [Restrictions] None [Quantization tool supporting] No
3	BatchNorm	Normalizes the input: variance of [(x – avg(x))/x]	[Inputs] One input [Arguments] • **use_global_stats**: bool, must be **true** • **moving_average_fraction**: (optional) float, default to **0.999** • **eps**: (optional) float, default to **1e – 5** [Restrictions] Only the C dimension can be normalized. [Quantization tool supporting] Yes
4	Concat	Concatenates the input along the given dimension.	[Inputs] Multiple inputs [Arguments] • **concat_dim**: (optional) uint32, default to **1**, greater than **0** • **axis**: (optional) int32, default to **1**, exclusive with **concat_dim**. When axis is **–1**, four input dimensions are required. Otherwise, the result may be incorrect. [Restrictions] • The number of dimensions of the input tensors must match, and all dimensions except **axis** must be equal. • The range of the input Tensor count is [1, 1000]. [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
5	ConvolutionDepthwise	Convolution depthwise	[Inputs] One input, with a constant filter and four dimensions [Arguments] • **num_output**: (optional) uint32 • **bias_term**: (optional) bool, default to **true** • **pad**: uint32, default to **0**, array • **kernel_size**: uint32, array • **stride**: uint32, default to **1**, array • **dilation**: uint32, only dilation=**1** is supported, array • **pad_h**: (optional) uint32, default to **0** (2D only) • **pad_w**: (optional) uint32, default to **0** (2D only) • **kernel_h**: (optional) uint32 (2D only) • **kernel_w**: (optional) uint32 (2D only) • **stride_h**: (optional) uint32 (2D only) • **stride_w**: (optional) uint32 (2D only) • **group**: (optional) uint32, default to **1** • **weight_filler**: This parameter is not supported. • **bias_filler**: This parameter is not supported. • **engine**: (optional) enum, default to **0**, CAFFE = 1, CUDNN = 2 • **force_nd_im2col**:(optional) bool, default to **false** • **axis**: (optional) int32, default to **1** [Restrictions] filterN = inputC = group (W + 15)/16 * 16) * filter.W * 32 ≤ 32 * 1024, where, W is W of the operator input and filter.W is W of the filter. [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
6	Convolution	Convolves the input.	[Inputs] One input, with a constant filter and four dimensions [Arguments] • **num_output**: (optional) uint32 • **bias_term**: (optional) bool, default to **true** • **pad**: uint32, default to **0**, array • **kernel_size**: uint32, array • **stride**: uint32, default to **1**, array • **dilation**: uint32, default to **1**, array • **pad_h**: (optional) uint32, default to **0** (2D only) • **pad_w**: (optional) uint32, default to **0** (2D only) • **kernel_h**: (optional) uint32 (2D only) • **kernel_w**: (optional) uint32 (2D only) • **stride_h**: (optional) uint32 (2D only) • **stride_w**: (optional) uint32 (2D only) • **group**: (optional) uint32, default to **1** • **weight_filler**: This parameter is not supported. • **bias_filler**: This parameter is not supported. • **engine**: (optional) enum, default to **0**, CAFFE = 1, CUDNN = 2 • **force_nd_im2col**:(optional) bool, default to **false** • **axis**: (optional) int32, default to **1** [Restrictions] • (inputW + padWHead + padWTail) ≥ (((FilterW – 1) * dilationW) + 1) • (inputW + padWHead + padWTail)/StrideW + 1 ≤ 2147483647 • (inputH + padHHead + padHTail) ≥ (((FilterH – 1) * dilationH) + 1) • (inputH + padHHead + padHTail)/StrideH + 1 ≤ 2147483647 • 0 ≤ Pad < 256, 0 < FilterSize < 256, 0 < Stride < 64, 1 ≤ dilationsize < 256 [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
7	Crop	Crops the input.	[Inputs] Two inputs [Arguments] • **axis**: (optional) int32, default to **2**. When axis is **–1**, four input dimensions are required. • **offset**: uint32, array [Restrictions] None [Quantization tool supporting] No

No.	Operator	Description	Boundary
8	Deconvolution	Deconvolution	[Inputs] One input, with a constant filter and four dimensions [Arguments] • **num_output**: (optional) uint32 • **bias_term**: (optional) bool, default to **true** • **pad**: uint32, default to **0**, array • **kernel_size**: uint32, array • **stride**: uint32, default to **1**, array • **dilation**: uint32, default to **1**, array • **pad_h**: (optional) uint32, default to **0** (2D only) • **pad_w**: (optional) uint32, default to **0** (2D only) • **kernel_h**: (optional) uint32 (2D only) • **kernel_w**: (optional) uint32 (2D only) • **stride_h**: (optional) uint32 (2D only) • **stride_w**: (optional) uint32 (2D only) • **group**: (optional) uint32, default to **1** • **weight_filler**: This parameter is not supported. • **bias_filler**: This parameter is not supported. • **engine**: (optional) enum, default to **0**, CAFFE = 1, CUDNN = 2 • **force_nd_im2col**:(optional) bool, default to **false** • **axis**: (optional) int32, default to **1** [Restrictions] • group = 1 • dilation = 1 • filterH – padHHead – 1 ≥ 0 • filterW – padWHead – 1 ≥ 0 • Restrictions involving intermediate variables: 1. a = ALIGN(filter_num, 16) * ALIGN(filter_c, 16) * filter_h * filter_w * 2 If ALIGN(filter_c, 16)%32 = 0, a = a/2 2. conv_input_width = (deconvolution input W – 1) * strideW + 1

No.	Operator	Description	Boundary
			3. b = (conv_input_width) * filter_h * ALIGN(filter_num, 16) * 2 * 2 4. a + b ≤ 1024 * 1024 [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
9	DetectionOutput	Generates detection results and outputs FSR.	[Inputs] Three inputs [Arguments] • **num_classes**: (mandatory) int32, indicating the number of classes to be predicted • **share_location**: (optional) bool, default to **true**, indicating that classes share one BBox • **background_label_id**: (optional) int32, default to **0** • **nms_param**: (optional) indicating non-maximum suppression (NMS) • **save_output_param**: (optional) indicating whether to save the detection result • **code_type**: (optional) default to **CENTER_SIZE** • **variance_encoded_in_target**: (optional) bool, default to **true**. The value **true** indicates that the variance is encoded in the target, otherwise the prediction offset needs to be adjusted accordingly. • **keep_top_k**: (optional) int32, indicating the total number of BBoxes to be reserved for each image after NMS • **confidence_threshold**: (optional) float, indicating that only the detection whose confidence is above the threshold is considered. If this parameter is not set, all boxes are considered. • **nms_threshold**: (optional) float • **top_k**: (optional) int32 • **boxes**: (optional) int32, default to **1** • **relative**: (optional) bool, default to **true** • **objectness_threshold**: (optional) float, default to **0.5** • **class_threshold**: (optional) float, default to **0.5** • **biases**: array • **general_nms_param**: (optional) [Restrictions] • Used for Faster R-CNN • Non-maximum suppression (NMS) ratio **nmsThreshold** is of range (0, 1).

No.	Operator	Description	Boundary
			• Probability threshold **postConfThreshold** is of range (0, 1). • At least two classes • Input box count ≤ 1024 • Output **W** dimension = 16 [Quantization tool supporting] Yes
10	Eltwise	Compute element-wise operations (PROD, MAX, and SUM).	[Inputs] At least two inputs [Arguments] • **operation**: (optional) enum, (PROD = 0; SUM = 1; MAX = 2), default to SUM • **coeff**: array, float • **stable_prod_grad**: (optional) bool, default to **true** [Restrictions] • Up to four inputs • Compared with the native operator, this operator does not support the **stable_prod_grad** parameter. • PROD, MAX, and SUM operations are supported. [Quantization tool supporting] Yes
11	Elu	Activation function	[Inputs] One input [Arguments] **alpha**: (optional) float, default to **1** [Restrictions] None [Quantization tool supporting] No

No.	Operator	Description	Boundary
12	Exp	Applies **e** as the base and **x** as the exponent.	[Inputs] One input [Arguments] • **base**: (optional) float, default to **–1.0** • **scale**: (optional) float, default to **1.0** • **shift**: (optional) float, default to **0.0** [Restrictions] None [Quantization tool supporting] No
13	Flatten	Converts an input n * c * h * w into a vector n * (c * h * w).	[Inputs] One input top_size ≠ bottom_size ≠ 1 When **axis** is **–1**, four input dimensions are required. [Arguments] • **axis**: (optional) int32, default to **1** • **end_axis**: (optional) int32, default to **–1** [Restrictions] axis < end axis [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
14	FullConnection	Computes an inner product.	[Inputs] One input [Arguments] • **num_output**: (optional) uint32 • **bias_term**: (optional) bool, default to **true** • **weight_filler**: This parameter is not supported. • **bias_filler**: This parameter is not supported. • **axis**: (optional) int32, default to **1** • **transpose**: (optional) bool, default to **false** [Restrictions] • transpose = false, axis = 1 • In the quantization scenario, Bais_C <= 59136; In non-quantified scenarios, Bais_C <= 118272 • To quantify the model, the following dimension restrictions must be satisfied: • When N = 1: 2 * CEIL(C, 16) * 16 * xH * xW ≤ 1024 * 1024; • When N > 1: 2 * 16 * CEIL(C, 16) * 16 * xH * xW ≤ 1024 * 1024. [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
15	Interp	Interpolation layer	[Inputs] One input [Arguments] • **height**: (optional) int32, default to **0** • **width**: (optional) int32, default to **0** • **zoom_factor**: (optional) int32, default to **1** • **shrink_factor**: (optional) int32, default to **1** • **pad_beg**: (optional) int32, default to **0** • **pad_end**: (optional) int32, default to **0** Note: • **zoom_factor** and **shrink_factor** are exclusive. • **height** and **zoom_factor** are exclusive. • **height** and **shrink_factor** are exclusive. [Restrictions] (outputH * outputW)/(inputH * inputW) > 1/30 [Quantization tool supporting] No
16	Log	Performs logarithmic operation on the input.	[Inputs] One input [Arguments] • **base**: (optional) float, default to **–1.0** • **scale**: (optional) float, default to **1.0** • **shift**: (optional) float, default to **0.0** [Restrictions] None [Quantization tool supporting] No

No.	Operator	Description	Boundary
17	LRN	Normalizes the input in a local region.	[Inputs] One non-constant input [Arguments] • **local_size**: (optional) uint32, default to **5** • **alpha**: (optional) float, default to **1** • **beta**: (optional) float, default to **0.75** • **norm_region**: (optional) enum, default to **ACROSS_CHANNELS** (ACROSS_CHANNELS = 0, WITHIN_CHANNEL = 1) • **lrnk**: (optional) float, default to **1** • **engine**: (optional) enum, default to **0**, CAFFE = 1, CUDNN = 2 [Restrictions] • **local_size** is an odd number greater than 0. • Inter-channel: If **local_size** is of range [1, 15]: lrnK > 0.00001 and beta > 0.01; Otherwise, **lrnK** and **beta** are any values. **lrnK** and **alpha** are not **0** at the same time. When the **C** dimension is greater than **1776**, local_size < 1728. • Intra-channel: lrnK = 1, **local_size** is of range [1, 15], beta > 0.01 [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
18	LSTM	Long and short term memory network (LSTM)	[Inputs] Two or three inputs • **X**: time sequence data (T * B * Xt). According to the NCHW format of 4D, ensure that the following conditions are met: **N** is the time sequence length T, **C** is the batch number B, **H** is the input data Xt at the t moment, and **W** is 1. • **Cont**: sequence continuity flag (T * B) • **Xs**: (optional) static data (B * Xt) [Arguments] • **num_output**: (optional) uint32, default to **0** • **weight_filler**: This parameter is not supported. • **bias_filler**: This parameter is not supported. • **debug_info**: (optional) bool, default to **false** • **expose_hidden**: (optional) bool, default to **false** [Restrictions] • Restrictions involving intermediate variables, **ht** and **output** are the argument **num_output**: a = (ALIGN(xt,16) + ALIGN(output,16)) * 16 * 2 * 2 b = (ALIGN(xt,16) + ALIGN(output,16)) * 16 * 4 * 2 * 2 d = 16 * ALIGN(ht,16) * 2 e = B * 4 That is: a + b ≤ 1024 * 1024 d ≤ 256 * 1024/8 e ≤ 256*1024/32 • B<=16, T<=768 [Quantization tool supporting] No

No.	Operator	Description	Boundary
19	Normalize	Normalization layer	[Inputs] One input [Arguments] • **across_spatial**: (optional) bool, default to **true** • **scale_filler**: This parameter is not supported. • **channel_shared**: (optional) bool, default to **true** • **eps**: (optional) float, default to **1e – 10** [Restrictions] • 1e – 7 < eps ≤ 0.1 + (1e – 6) • **across_spatial** must be **true** for Caffe, indicating normalization by channel [Quantization tool supporting] Yes
20	Permute	Permutes the input dimensions according to a given mode.	[Inputs] One input [Arguments] **order**: uint32, array [Restrictions] None [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
21	Pooling	Pools the input.	[Inputs] One input [Arguments] • **pool**: (optional) enum, indicating the pooling method, MAX = 0, AVE = 1, and STOCHASTIC = 2, default to **MAX** • **pad**: (optional) uint32, default to **0** • **pad_h**: (optional) uint32, default to **0** • **pad_w**: (optional) uint32, default to **0** • **kernel_size**: (optional) uint32, exclusive with **kernel_h/kernel_w** • **kernel_h**: (optional) uint32 • **kernel_w**: (optional) uint32, used in pair with **kernel_h** • **stride**: (optional) uint32, default to **1** • **stride_h**: (optional) uint32 • **stride_w**: (optional) uint32 • engine: (optional) enum, default to **0**, CAFFE = 1, CUDNN = 2 • **global_pooling**: (optional) bool, default to **false** • **ceil_mode**: (optional) bool, default to **true** • **round_mode**: (optional) enum, CEIL = 0, FLOOR = 1, default to **CEIL** [Restrictions] • kernelH ≤ inputH + padTop + padBottom • kernelW ≤ inputW + padLeft + padRight • padTop < windowH • padBottom < windowH • padLeft < windowW • padRight < windowW • Only the global pooling mode is supported. The following restrictions must be satisfied: 1) outputH == 1 && outputW == 1 && kernelH ≥ inputH && kernelW ≥ inputW 2) inputH * inputW ≤ 10000 [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
22	Power	y = (scale * x + shift)^power	[Inputs] One input [Arguments] • **power**: (optional) float, default to **1.0** • **scale**: (optional) float, default to **1.0** • **shift**: (optional) float, default to **0.0** [Restrictions] • power! = 1 • scale*x + shift > 0 [Quantization tool supporting] Yes
23	Prelu	Activation function	[Inputs] One input [Arguments] • **filler**: This parameter is not supported. • **channel_shared**: (optional) bool, indicating whether to share slope parameters across channels, default to **false** [Restrictions] None [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
24	PriorBox	Obtains the real location of the target from the box proposals.	[Inputs] Two inputs: • **Original input image of model,** data format: NCHW; • **FeatureMap,** data format: NCHW. [Arguments] • **min_size**: (mandatory) indicating the minimum frame size (in pixels) • **max_size**: (mandatory) indicating the maximum frame size (in pixels) • **aspect_ratio**: array, float. A repeated ratio is ignored. If no aspect ratio is provided, the default ratio **1** is used. • **flip**: (optional) bool, default to true. The value true indicates that each aspect ratio is reversed. For example, for aspect ratio r, the aspect ratio **1.0/r** is generated. • **clip**: (optional) bool, default to **false**. The value true indicates that the previous value is clipped to the range [0, 1]. • **variance**: array, used to adjust the variance of the BBoxes • **img_size**: (optional) uint32. exclusive with **img_h/img_w** • **img_h**: (optional) uint32 • **img_w**: (optional) uint32 • **step**: (optional) float. **step_h** and **step_w** are exclusive. • **step_h**: (optional) float • **step_w**: (optional) float • **offset**: (optional) float, default to **0.5** [Restrictions] Used for the SSD network only Output dimensions: [n, 2, detection frame * 4, 1] [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
25	Proposal	Sorts the box proposals by (proposal, score) and obtains the top N proposals by using the NMS.	[Inputs] Three inputs: scores, bbox_pred, im_info [Arguments] • **feat_stride**: (optional) float • **base_size**: (optional) float • **min_size**: (optional) float • **ratio**: array (optional), float • **scale**: array (optional), float • **pre_nms_topn**: (optional) int32 • **post_nms_topn**: (optional) int32 • **nms_thresh**: (optional) float [Restrictions] Used only for Faster R-CNN • **ProposalParameter** and **PythonParameter** are exclusive. 1. Value range of **preTopK**: 1–6144 2. Value range of **postTopK**: 1–1024 3. scaleCnt * ratioCnt ≤ 64 4. 0 < nmsTresh ≤ 1 (threshold for box filtering) 5. **minSize**: minimum edge length of a proposal. A box with any side smaller than **minSize** is removed. 6. **featStride**: H/W stride between the two adjacent boxes used in default box generation 7. **baseSize**: base box size used in default box generation 8. **ratio** and **scale**: used in default box generation 9. **imgH** and **imgW**: height and width of the image input to the network. The values must be greater than **0**. • Restrictions on the input dimensions: clsProb: C = 2 * scaleCnt * ratioCnt bboxPred: C = 4 * scaleCnt * ratioCnt bboxPrior: N = clsProb.N, C = 4 * scaleCnt * ratioCnt **imInfo**: N = clsProb.N, C = 3 [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
26	PSROIPooling	Position-sensitive region-of-interest pooling (PSROIPooling)	[Inputs] Two inputs [Arguments] • **spatial_scale**: (mandatory) float • **output_dim**: (mandatory) int32, indicating the number of output channels • **group_size**: (mandatory) int32, indicating the number of groups to encode position-sensitive score maps [Restrictions] Used for the Region-based Fully Convolutional Network (R-FCN) • ROI coordinates [roiN, roiC, roiH, roiW]: 1 ≤ roiN ≤ 65535, roiC == 5, roiH == 1, roiW == 1 • Dimensions of the input feature map: [xN, xC, xH, xW] • pooledH == pooledW == groupSize ≤ 128 **pooledH** and **pooledW** indicate the length and width of the pooled ROI. Output format: y [yN, yC, yH, yW] • poolingMode == avg pooling, pooledH == pooledW == groupSize, pooledH ≤ 128, spatialScale > 0, groupSize > 0, outputDim > 0 • 1 ≤ xN ≤ 65535, roisN % xN == 0 • HW_LIMIT is the limit of **xH** and **xW**. xHW = xH * xW pooledHW = pooledH * pooledW HW_LIMIT = (64 * 1024 – 8 * 1024)/32, xH ≥ pooledH, xW ≥ pooledW xHW ≥ pooledHW xHW/pooledHW ≤ HW_LIMIT • In multi-batch scenarios, the ROIs are allocated equally to the batches. In addition, the batch sequence of the ROIs is the same as the feature. [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
27	Relu	Activation function, including common ReLU and Leaky ReLU, which can be specified by parameters	[Inputs] One input [Arguments] • **negative_slope**: (optional) float, default to **0** • **engine**: (optional) enum, default to **0**, CAFFE = 1, CUDNN = 2 [Restrictions] None [Quantization tool supporting] Yes
28	Reshape	Reshapes the input.	[Inputs] One input [Arguments] • **shape**: constant, int64 or int • **axis**: (optional) int32, default to **0** • **num_axes**: (optional) int32, default to **–1** [Restrictions] None [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
29	ROIAlign	Aggregates features using ROIs.	[Inputs] At least two inputs [Arguments] • **pooled_h**: (optional) uint32, default to **0** • **pooled_w**: (optional) uint32, default to **0** • **spatial_scale**: (optional) float, default to **1** • **sampling_ratio**: (optional) int32, default to **–1** [Restrictions] Mainly used for Mask R-CNN Restrictions on the feature map: • H * W ≤ 5248 or W * C < 40960 • C ≤ 1280 • ((C – 1)/128+1) * pooledW ≤ 216 Restrictions on the ROI: • C = 5 (Caffe), H = 1, W = 1 • samplingRatio * pooledW ≤ 128, samplingRatio * pooledH ≤ 128 • H ≥ pooledH, W ≥ pooledW [Quantization tool supporting] Yes
30	ROIPooling	Maps ROI proposals to a feature map.	[Inputs] At least two inputs [Arguments] • **pooled_h**: (mandatory) uint32, default to **0** • **pooled_w**: (mandatory) uint32, default to **0** • **spatial_scal**e: (mandatory) float, default to **1**. The multiplication spatial scale factor is used to convert ROI coordinates from the input scale to the pool scale. [Restrictions] Mainly used for Faster R-CNN • Input dimensions: H * W ≤ 8160, H ≤ 120, W ≤ 120 • Output dimensions: pooledH ≤ 20, pooledW ≤ 20 [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
31	Scale	out = alpha*Input +beta	[Inputs] Two inputs, each with four dimensions [Arguments] • **axis**: (optional) int32, **1** (default) or **–3** • **num_axes**: (optional) int32, default to **1** • **filler**: This parameter is not supported. • **bias_term**: (optional) bool, default to **false**, indicating whether to learn a bias (equivalent to ScaleLayer + BiasLayer, but may be more efficient). • **bias_filler**: This parameter is not supported. [Restrictions] **shape** of **scale** and **bias**: (n, c, 1, 1), with the **C** dimension equal to that of the input [Quantization tool supporting] Yes
32	ShuffleChannel	Shuffles information cross the feature channels.	[Inputs] One input [Arguments] **group**: (optional) uint32, default to **1** [Restrictions] None [Quantization tool supporting] Yes
33	Sigmoid	Activation function	[Inputs] One input [Arguments] **engine**: (optional) enum, default to **0**, CAFFE = 1, CUDNN = 2 [Restrictions] None [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
34	Slice	Slices an input into multiple outputs.	[Inputs] One input [Arguments] • **slice_dim**: (optional) uint32, default to **1**, exclusive with **axis** • **slice_point**: array, uint32 • **axis**: (optional) int32, default to **1**, indicating concatenation along the channel dimension [Restrictions] None [Returns] No restrictions [Quantization tool supporting] Yes
35	Softmax	Normalization logic function	[Inputs] One input [Arguments] • **engine**: (optional) default to **0**, CAFFE = 1, CUDNN = 2 • **axis**: (optional) int32, default to **1**, indicating the axis along which softmax is performed [Restrictions] If the input contains four dimensions, softmax is performed on each of them. According to **axis**: • When axis = 1: C ≤ ((256 * 1024/4) – 8 * 1024 – 256)/2 • When axis = 0: N ≤ (56 * 1024 – 256)/2 • When axis = 2: W = 1, 0 < H < (1024 * 1024/32) • When axis = 3: 0 < W < (1024 * 1024/32) If the input contains fewer than four dimensions, softmax is performed only on the last dimension, with the last dimension ≤ 46080. [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
36	Tanh	Activation function	[Inputs] One input [Arguments] **engine**: (optional) enum, default to **0**, CAFFE = 1, CUDNN = 2 [Restrictions] The number of tensor elements cannot exceed **INT32_MAX**. [Quantization tool supporting] Yes
37	Upsample	Backward propagation of max pooling	[Inputs] Two inputs [Arguments] **scale**: (optional) int32, default to **1** [Restrictions] None [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
38	SSDDetectionOutput	SSD network detection output	[Inputs] Three inputs [Arguments] • **num_classes**: (mandatory) int32, indicating the number of classes to be predicted • **share_location**: (optional) bool, default to **true**, indicating that classes share one BBox • **background_label_id**: (optional) int32, default to **0** • **nms_param**: (optional) indicating non-maximum suppression (NMS) • **save_output_param**: (optional) indicating whether to save the detection result • **code_type**: (optional) default to **CENTER_SIZE** • **variance_encoded_in_target**: (optional) bool, default to **true**. The value **true** indicates that the variance is encoded in the target, otherwise the prediction offset needs to be adjusted accordingly. • **keep_top_k**: (optional) int32, indicating the total number of BBoxes to be reserved for each image after NMS • **confidence_threshold**: (optional) float, indicating that only the detection whose confidence is above the threshold is considered. If this parameter is not set, all boxes are considered. • **nms_threshold**: (optional) float • **top_k**: (optional) int32 • **boxes**: (optional) int32, default to **1** • **relative**: (optional) bool, default to **true** • **objectness_threshold**: (optional) float, default to **0.5** • **class_threshold**: (optional) float, default to **0.5** • **biases**: array • **general_nms_param**: (optional) [Restrictions] • Used for the SSD network • Value range of **preTopK** and **postTopK**: 1–1024

No.	Operator	Description	Boundary
			• shareLocation = true • nmsEta = 1 • Value range of **numClasses**: 1–2048 • code_type = CENTER_SIZE • Value range of nms_threshold and confidence_threshold: 0.0–1.0 [Quantization tool supporting] Yes
39	Reorg	Real-time object detection	[Inputs] One input [Arguments] • **stride**: (optional) uint32, default to **2** • **reverse**: (optional) bool, default to **false** [Restrictions] Used only for YOLOv2 [Quantization tool supporting] No
40	Reverse	Reversion	[Inputs] One input [Arguments] **axis**: (optional) int32, default to **1**. Controls the axis to be reversed. The content layout will not be reversed. [Restrictions] None [Quantization tool supporting] No
41	LeakyRelu	LeakyRelu activation function	[Inputs] One input [Arguments] Same as Relu [Restrictions] None [Quantization tool supporting] Yes

No.	Operator	Description	Boundary
42	YOLODetectionOutput	YOLO network detection output	[Inputs] Four inputs [Arguments] • **num_classes**: (mandatory) int32, indicating the number of classes to be predicted • **share_location**: (optional) bool, default to **true**, indicating that classes share one BBox • **background_label_id**: (optional) int32, default to **0** • **nms_param**: (optional) indicating non-maximum suppression (NMS) • **save_output_param**: (optional) indicating whether to save the detection result • **code_type**: (optional) default to **CENTER_SIZE** • **variance_encoded_in_target**: (optional) bool, default to **true**. The value **true** indicates that the variance is encoded in the target, otherwise the prediction offset needs to be adjusted accordingly. • **keep_top_k**: (optional) int32, indicating the total number of BBoxes to be reserved for each image after NMS • **confidence_threshold**: (optional) float, indicating that only the detection whose confidence is above the threshold is considered. If this parameter is not set, all boxes are considered. • **nms_threshold**: (optional) float • **top_k**: (optional) int32 • **boxes**: (optional) int32, default to **1** • **relative**: (optional) bool, default to **true** • **objectness_threshold**: (optional) float, default to **0.5** • **class_threshold**: (optional) float, default to **0.5** • **biases**: array • **general_nms_param**: (optional) [Restrictions] • Used only for YOLOv2 • classNUm < 10240, anchorBox < 5 • W ≤ 1536

No.	Operator	Description	Boundary
			• The upper layer of **yolodetectionoutput** must be the **yoloregion** operator. [Quantization tool supporting] No

2.3 TensorFlow Operator Boundaries

No.	Python API	C++ API	Boundary
1	tf.nn.avg_pool	AvgPool Type: Pooling	[Arguments] • **value**: 4D tensor of float32 type, with shape [batch, height, width, channels] • **ksize**: list or tuple of four integers, each value corresponding to the window size for each dimension of the input tensor. • **strides**: list or tuple of four integers, each value corresponding to the stride of the sliding window for each dimension of the input tensor • **padding**: string, either **VALID** or **SAME** • **data_format**: string, either **NHWC** (default) or **NCHW** • **name**: (optional) operation name, string [Restrictions] • kernelH ≤ inputH + padTop + padBottom • kernelW ≤ inputW + padLeft + padRight • padTop < windowH • padBottom < windowH • padLeft < windowW • padRight < windowW • Only the global pooling mode is supported. The following restrictions must be satisfied: 1) outputH == 1 && outputW == 1 && kernelH ≥ inputH && kernelW ≥ inputW 2) inputH * inputW ≤ 10000 [Returns] Tensor of the identical data type as **value** [Quantization tool supporting] Yes

No.	Python API	C++ API	Boundary
2	tf.math.reduce_mean	Mean	[Arguments] Same as **tf.nn.avg_pool** [Restrictions] The Mean operator adapts to the AvgPool operator. The restrictions are as follows: • keep_dims=true. If keep_dims=false, the subsequent operators must be Reshape. • The average value can be calculated only for the HW dimension, that is, when format=NCHW, axis = [2,3]. When format=NHWC, axis=[1,2]. [Returns] Tensor of the identical data type as **value** [Quantization tool supporting] Yes
3	tf.nn.max_pool	MaxPool	Same as **tf.nn.avg_pool**

No.	Python API	C++ API	Boundary
4	tf.nn.conv2d	Conv2D	[Arguments] • **value**: 4D tensor of float32 type, with shape [batch, height, width, channels] • **filter**: constant Tensor, with same data type and dimensions as **value**, with shape [filter_height, filter_width, in_channels, out_channels] • **strides**: non-null list or tuple of four integers, each value corresponding to the stride of the sliding window for each dimension of the input tensor • **padding**: non-null string, either **VALID** or **SAME** • **use_cudnn_on_gpu**: bool, default to **True** • **data_format**: non-null, string, either **NHWC** (default) or **NCHW** • **dilations**: (optional) list of four integers, default to [1,1,1,1], each value corresponding to a dimension. If k > 1, k – 1 units are skipped at the corresponding dimension in filtering. The dimension sequence is determined by **data_format**. The values of **batch** and **depth** of **dilations** must be **1**. • **name**: (optional) operation name, string [Restrictions] • (inputW + padWHead + padWTail) ≥ (((FilterW – 1) * dilationW) + 1) • (inputW + padWHead + padWTail)/StrideW + 1 ≤ INT32_MAX • (inputH + padHHead + padHTail) ≥ (((FilterH – 1) * dilationH) + 1) • (inputH + padHHead + padHTail)/StrideH + 1 ≤ INT32_MAX • 0 ≤ Pad < 256, 0 < FilterSize < 256, 0 < Stride < 64, 1 ≤ dilationsize < 256 [Returns] Tensor of the identical data type as **value** [Quantization tool supporting] Yes

No.	Python API	C++ API	Boundary
5	tf.concat	Concat	[Arguments] • **values**: list of Tensor objects or a single Tensor. The values of dimensions must be the same except the dimensions to be concatenated. • **axis**: 0D Tensor of type int32, specifying the dimension to be concatenated. The value range is [–rank(values), rank(values)]. As in Python, indexing for **axis** is 0-based. Positive **axis** in the range [0, rank(values)) refers to **axis**-th dimension, while negative **axis** refers to [axis + rank(values)]-th dimension. [Restrictions] • The number of dimensions of the input tensors must match, and all dimensions except **axis** must be equal. The range of the input Tensor count is [1, 1000]. [Returns] Tensor, resulting from concatenation of the input Tensors [Quantization tool supporting] Yes

No.	Python API	C++ API	Boundary
6	tf.matmul	MatMul	[Arguments] • **a**: non-constant Tensor of type float32, rank ≥ 2 • **b**: constant Tensor with the same data type and rank as **a** • **transpose_a**: The value **true** indicates that a is transposed before multiplication. • **transpose_b**: The value **true** indicates that b is transposed before multiplication. If **transpose_a** is **false**, **transpose_b** is also **false**. • **adjoint_a**: The value **true** indicates that **a** is conjugated and transposed before multiplication. • **adjoint_b**: The value **true** indicates that **b** is conjugated and transposed before multiplication. • **a_is_sparse**: The value **true** indicates that **a** is treated as a sparse matrix. • **b_is_sparse**: The value **true** indicates that **b** is treated as a sparse matrix. • **name**: (optional) operation name [Restrictions] • The transposing property of weight is **false**. • The multiplication of two Tensors is not supported. Only one Tensor by one constant is supported. [Returns] Tensor of the identical data type as **a** and **b** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
7	tf.nn.fused_batch_norm	FusedBatchNorm	[Arguments] • **x**: 4D Tensor of type float32 • **scale**: 1D Tensor for scaling • **offset**: 1D Tensor for bias • **mean**: 1D Tensor for population mean used for inference • **variance**: 1D Tensor for population variance used for inference • **epsilon**: small float number added to the variance of **x** • **data_format**: data format for **x**, either **NHWC** (default) or **NCHW** • **is_training**: bool, specifying whether the operation is used for training or inference • **name**: (optional) operation name [Restrictions] The shape of **scale**, **bias**, **mean**, and **var** must be (1, C, 1, 1), with the same **C** dimension as **input**. [Returns] • **y**: 4D Tensor for the normalized, scaled, offset **x** • **batch_mean**: 1D Tensor for the mean of **x** • **batch_var**: 1D Tensor for the variance of **x** [Quantization tool supporting] No
8	tf.abs	Abs	[Arguments] • **x**: Tensor or SparseTensor of type float32 • **name**: (optional) operation name [Restrictions] None [Returns] Returns the absolute value of **x**, Tensor or SparseTensor. The size and type are the same as those of **x**. [Quantization tool supporting] Yes

No.	Python API	C++ API	Boundary
9	tf.image.resize_nearest_neighbor	ResizeNearestNeighbor	[Arguments] • **images**: 4D Tensor of type float32, with shape [batch, height, width, channels] or 3D Tensor of type float32 with shape [height, width, channels] • **size**: 1D 2-element constant Tensor, indicating the new size for the images • **method**: **ResizeMethod.NEARESTNEIGHBOR** • **align_corners**: bool, default to **False**. The value true indicates that the centers of the 4 corner pixels of the input and output tensors are aligned, preserving the values at the corner pixels. [Restrictions] None [Returns] Tensor of type float, with the identical shape as the **input** [Quantization tool supporting] No
10	tf.image.resize_bilinear	ResizeBilinear	[Arguments] • **images**: 4D non-constant Tensor with shape [batch, height, width, channels] of type float32 • **size**: 1D 2-element constant Tensor, indicating the new size for the images • **method**: **ResizeMethod.BILINEAR** • **align_corners**: bool, default to **False**. The value true indicates that the centers of the 4 corner pixels of the input and output tensors are aligned, preserving the values at the corner pixels. [Restrictions] (outputH * outputW)/(inputH * inputW) > 1/7 [Returns] Tensor of type float, with the identical shape as the **input** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
11	tf.cast	Cast	[Inputs] • Data type: float32, int32, bool, int64, int16, int8, uint8, uint16, double [Arguments] • **x**: Tensor, SparseTensor, or IndexedSlices • **dtype**: destination type, same as the data type of **x** • **name**: (optional) operation name [Restrictions] None [Returns] Tensor, SparseTensor, or IndexedSlices, same **dtype** and **shape** as the input [Quantization tool supporting] N/A

No.	Python API	C++ API	Boundary
12	tf.nn.depthwise_conv2d	DepthwiseConv2dNative	[Arguments] • **input**: 4D • **filter**: 4D constant, with shape [filter_height, filter_width, in_channels, channel_multiplier] • **strides**: non-null list of four integers, each value corresponding to the stride of the sliding window for each dimension of the input tensor • **padding**: string, either **VALID** or **SAME** • **rate**: 1D of size 2. The dilation rate in which we sample input values across the height and width dimensions in atrous convolution. If it is greater than 1, then all values of strides must be 1. • **data_format**: data format for input, either **NHWC** (default) or **NCHW** • **name**: (optional) operation name [Restrictions] (W + 15)/16 * 16) * filter.W * 32 ≤ 32 * 1024, where, W is W of the operator input and filter.W is W of the filter. [Returns] 4D Tensor, with shape according to **data_format**. For example, for format NHWC, shape = [batch, out_height, out_width, in_channels * channel_multiplier] for the NHWC format [Quantization tool supporting] Yes
13	tf.reshape	Reshape	[Arguments] • **tensor**: Tensor • **shape**: output shape, constant Tensor of type int64 or int • **name**: (optional) operation name [Restrictions] None [Returns] Tensor of the identical data type as **input** [Quantization tool supporting] Yes

No.	Python API	C++ API	Boundary
14	tf.squeeze	Squeeze	[Arguments] • **input**: non-constant Tensor • **axis**: list of ints, specifying the dimensions to be squeezed, default to []. It is an error to squeeze a dimension that is not **1**. • **name**: (optional) operation name • **squeeze_dims**: (deprecated) exclusive with **axis** [Restrictions] None [Returns] Tensor, with the same data and type as **input**, but has one or more dimensions of size **1** removed. [Quantization tool supporting] Yes
15	tf.expand_dims	ExpandDims	[Arguments] • **input**: Tensor • **axis**: 0D (scalar), specifying the dimension index of the extended **input** shape • **name**: name of the output Tensor • **dim**: (deprecated) 0D (scalar), equivalent to **axis** [Restrictions] None [Returns] Tensor with the same data as **input**, but its shape has an additional dimension of size **1** added [Quantization tool supporting] Yes

No.	Python API	C++ API	Boundary
16	tf.greater	Greater	[Inputs] Two inputs • One input is a constant tensor. The data format requirements are as follows: NC1HWC0 (Huawei-developped format 5D) and 4-dimensional input data (for example, NCHW) In the NC1HWC0 data format, the C0 is closely related to the micro architecture, and the value is equal to the size of the cube unit, for example, 16. C1 divides the C dimension by C0, C1=C/C0. If the result is not divided, the last data needs to be padding to C0. • Another input is a constant scalar. [Arguments] **name**: (optional) operation name [Restrictions] • Broadcasting is supported, so the shape of **x** and shape of **y** are compared. For a right-aligned dimension, if the values of **xdim[i]** and **ydim[i]** are not the same, one of them must be **1** or missing. • The next layer operator of the tf.greater can have only two inputs, one of which is the output of the tf.greater operator and the other is a constant. [Returns] Constant tensor of type bool [Quantization tool supporting] N/A
17	tf.nn.relu	Relu	[Arguments] • **features**: non-constant Tensor • **name**: (optional) operation name [Restrictions] None [Returns] Tensor of the identical data type as **features** [Quantization tool supporting] Yes

No.	Python API	C++ API	Boundary
18	tf.nn.relu6	Relu6	[Arguments] • **features**: non-constant Tensor • **name**: (optional) operation name [Restrictions] None [Returns] Tensor of the identical data type as **features** [Quantization tool supporting] Yes
19	tf.nn.leaky_relu	/	[Arguments] • **features**: non-constant Tensor representing pre-activation values • **alpha**: slope of the activation function at x < 0 • **name**: (optional) operation name [Restrictions] None [Returns] Activation value [Quantization tool supporting] Yes
20	tf.exp	exp	[Arguments] • **x**: Tensor of type float32 or double • **name**: (optional) operation name [Restrictions] None [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
21	tf.nn.conv2d_transpose	Conv2DBackpropInput	[Inputs] • **value**: 4D Tensor with shape [batch, height, width, in_channels] for NHWC data format or [batch, in_channels, height, width] for NCHW data format • **filter**: 4D constant Tensor with shape [height, width, output_channels, in_channels] • input_sizes: 1D constant Tensor [Arguments] • **output_shape**: 1D Tensor, indicating the output shape • **strides**: non-null list of integers, each value corresponding to the stride of the sliding window for each dimension of the input tensor • **padding**: non-null, string, either **VALID** or **SAME** • **data_format**: non-null string, either **NHWC** or **NCHW** • **name**: (optional) output name [Restrictions] • group = 1 • dilation = 1 • filterH – padHHead – 1 ≥ 0 • filterW – padWHead – 1 ≥ 0 • Restrictions involving intermediate variables: 1. a = ALIGN(filter_num, 16) * ALIGN(filter_c, 16) * filter_h * filter_w * 2 If ALIGN(filter_c, 16)%32 = 0, a = a/2 2. conv_input_width = (deconvolution input W – 1) * strideW + 1 3. b = (conv_input_width) * filter_h * ALIGN(filter_num, 16) * 2 * 2 4. a + b ≤ 1024 * 1024 [Returns] Tensor of the identical data type as **value** [Quantization tool supporting] Yes

No.	Python API	C++ API	Boundary
22	tf.sigmoid	Sigmoid	[Arguments] • **x**: Tensor • **name**: (optional) operation name [Restrictions] None [Returns] Tensor of the identical data type as **value** [Quantization tool supporting] Yes
23	tf.add	Add	[Arguments] • **x**: Tensor of type float32 or int32 • **y**: Tensor of the identical data type as **x**. For two constant inputs, one of them is a scalar. • **name**: (optional) operation name [Restrictions] If the two inputs have inconsistent dimensions, broadcasting (that is, dimension padding) is performed. Broadcasting is supported only in the following scenarios: NHWC+ NHWC, NHWC+scalar NHWC + 1 1 1 1 NHWC + W, HWC + W, HW + W (W-based broadcasting) NCHW + NH1C, HWC + H1C, HW + H1 HWC + 1 WC (H-based broadcasting) Note: The input sequence of the two Tensors is not fixed. [Returns] Tensor of the identical data type as **y** [Quantization tool supporting] Yes
24	tf.add_n	-	The **tf.add_n** operator internally invokes the **Eltwise** operator to add all input tensors by element-wise. For details about the input, constraints, and attributes, see the **Eltwise** operator in the **2.2 Caffe Operator Boundaries**.

No.	Python API	C++ API	Boundary
25	tf.multiply	Multiply Type: Mul	[Arguments] • **x**: Tensor of type float32 or int32 • **y**: Tensor of the identical data type as **x**. If two constants are input, their dimensions must be identical (scalar or 1D Tensor) • **name**: (optional) operation name [Restrictions] If the two inputs have inconsistent dimensions, broadcasting (that is, dimension padding) is performed. Broadcasting is supported only in the following scenarios: NHWC+ NHWC, NHWC+scalar NHWC + 1 1 1 1 NHWC + W, HWC + W, HW + W (W-based broadcasting) NCHW + NH1C, HWC + H1C, HW + H1 HWC + 1 WC (H-based broadcasting) Note: The input sequence of the two Tensors is not fixed. [Returns] Tensor [Quantization tool supporting] Yes

No.	Python API	C++ API	Boundary
26	tf.subtract	Subtract Type: Sub	[Arguments] • **x**: Tensor • **y**: Tensor of the identical data type as **x**, constant or non-constant • **name**: (optional) operation name [Restrictions] If the two inputs have inconsistent dimensions, broadcasting (that is, dimension padding) is performed. Broadcasting is supported only in the following scenarios: NHWC+ NHWC, NHWC+scalar NHWC + 1 1 1 1 NHWC + W, HWC + W, HW + W (W-based broadcasting) NCHW + NH1C, HWC + H1C, HW + H1 HWC + 1 WC (H-based broadcasting) Note: The input sequence of the two Tensors is not fixed. [Returns] Tensor [Quantization tool supporting] Yes

No.	Python API	C++ API	Boundary
27	tf.nn.bias_add	BiasAdd	[Arguments] • **value**: non-constant Tensor • **bias**: 1D constant Tensor, with size matching the last dimension of **value**, of the same type as **value** unless **value** is a quantized type • **data_format**: string, either **NHWC** or **NCHW** • **name**: (optional) operation name [Restrictions] • C < 10000 • **input** and **bias** must have the same data layout. • When **bias** is added to the **C** dimensions, the **C** dimensions of **input** and **bias** must be the same. [Returns] Tensor of the identical data type as **value** [Quantization tool supporting] Yes

No.	Python API	C++ API	Boundary
28	tf.nn.lrn	Local response normalization (LRN)	[Arguments] • **x**: 4D Tensor of type float32 • **depth_radius**: 0D of type int, default to **5**, indicating the half-width of the 1D normalization window • **bias**: (optional) float, default to **1**, indicating the offset (usually positive to avoid dividing by **0**) • **alpha**: (optional) float, default to **1**, indicating the scale factor, usually positive • **beta**: (optional) float, default to **0.5**, indicating an exponent • **name**: (optional) operation name [Restrictions] • **depth_radius** is an odd number greater than 0. • Inter-channel: When **depth_radius** is of range [1,15], alpha > 0.00001 and beta > 0.01; Otherwise, **alpha** and **beta** are any values. When C > 1776, depth_radius < 1728. [Returns] Tensor of the identical data type as **input** [Quantization tool supporting] Yes
29	tf.nn.elu	Elu	[Arguments] • **features**: non-constant Tensor • **name**: (optional) operation name [Restrictions] None [Returns] Tensor of the identical data type as **features** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
30	tf.rsqrt	Rsqrt	[Arguments] • **x**: Tensor • **name**: (optional) operation name [Restrictions] None [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] Yes
31	tf.log	Log	[Arguments] • **x**: Tensor of type float32 • **name**: (optional) operation name [Restrictions] None [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] No
32	tf.tanh	Tanh	[Arguments] • **x**: non-constant Tensor • **name**: (optional) operation name [Restrictions] None [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] Yes

No.	Python API	C++ API	Boundary
33	tf.slice	Slice	[Arguments] • **input_**: Tensor • **begin**: Tensor of type int32 or int64 • **size**: Tensor of type int32 or int64 • **name**: (optional) operation name [Restrictions] The number of tensor elements cannot exceed **INT32_MAX**. [Returns] Tensor of the identical data type as **input_** [Quantization tool supporting] Yes
34	tf.split	Split	[Arguments] • **value**: Tensor • **num_or_size_splits**: not supported • **axis**: integer, specifying the dimension along which to split • **name**: (optional) operation name, string [Restrictions] None [Returns] List of Tensor objects resulting from splitting [Quantization tool supporting] Yes
35	tf.nn.softplus	Softplus	[Arguments] • **features**: Tensor of type float32 • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of the identical data type as **features** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
36	tf.nn.softsign	Softsign	[Arguments] • **features**: Tensor of type float32 • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of the identical data type as **features** [Quantization tool supporting] No
37	tf.pad	Pad/ MirrorPad/ PadV2	[Arguments] • **tensor**: 4D Tensor of type float32 or int32 • **paddings**: constant Tensor of type int32 • **mode**: string, one of **CONSTANT**, **REFLECT**, or **SYMMETRIC** • **name**: (optional) operation name, string • **constant_values**: scalar pad value to use, of the identical data type as **tensor** [Restrictions] In **CONSTANT** mode: $0 \leq PAD \leq 128$, $0 < W \leq 3000$ [Returns] Tensor of the identical data type as **tensor** [Quantization tool supporting] Yes

No.	Python API	C++ API	Boundary
38	tf.fake_quant_with_min_max_vars	FakeQuantWithMinMaxVars	[Arguments] • **inputs**: Tensor of type float32 • **min**: Tensor of type float32 • **max**: Tensor of type float32 • **num_bits**: scalar of type int, default to **8** • **narrow_range**: (optional) bool, default to **False** • **name**: (optional) operation name, string [Restrictions] –65504 ≤ min ≤ +65504, –65504 ≤ max ≤ +65504 [Returns] Tensor of type float32 [Quantization tool supporting] No
39	tf.reduce_max	Max	[Arguments] • **input_tensor**: Tensor, must be one of the following types: float32, int64, int32, uint8, uint16, int16, int8 • **axis**: 1D list or scalar of type integer • **keepdims**: bool, indicating whether to retain reduced dimensions with length **1** • **name**: (optional) operation name, string • **reduction_indices**: (deprecated) equivalent to **axis** • **keep_dims**: (deprecated) equivalent to **keepdims** [Restrictions] • When the input Tensor has four dimensions: input axis = {3,{1,2,3}}, keepDims = true, H * W * 16 * 2 ≤ 16 * 1024 • When the input Tensor has two dimensions: input axis = {1,{1}}, keepDims = true, H * W * CEIL(C, 16) * 16 * 2 ≤ 16 * 1024 [Returns] Reduced Tensor of the identical data type as **input_tensor** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
40	tf.strided_slice	StridedSlice	[Arguments] • **input_**: Tensor • **begin**: 1D Tensor of type int32 • **end**: 1D Tensor of type int32 • **strides**: 1D Tensor of type int32 • **begin_mask**: scalar of type int32 • **end_mask**: scalar of type int32 • **ellipsis_mask**: scalar of type int32 • **new_axis_mask**: scalar of type int32 • **shrink_axis_mask**: scalar of type int32 • **var**: variable corresponding to **input_** or **None** • **name**: (optional) operation name, string [Restrictions] strides ≠ 0 [Returns] Tensor of the identical data type as **input_** [Quantization tool supporting] No
41	tf.reverse	Reverse	[Arguments] • **tensor**: list of Tensor objects • **axis**: dimensions to reverse, of type int32 or int64 • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of the identical data type as **tensor** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
42	tf.realdiv	RealDiv	[Arguments] • **x**: Tensor of type float32 • **y**: Tensor of type float32 • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] Yes
43	tf.stack	Stack	[Arguments] • **values**: list of Tensor objects with the same shape and type (float32 or int32) • **axis**: (mandatory) integer, indicating the axis to stack along, default to the first dimension • **name**: (optional) operation name [Restrictions] None [Returns] Stacked Tensor of the identical data type as **values** Properties **T**, **N**, and **axis** are mandatory. [Quantization tool supporting] No
44	tf.transpose	Transpose	[Arguments] • **a**: Tensor • **perm**: permutation of the dimensions of **a** • **name**: (optional) operation name • **conjugate**: (optional) bool, default to **False**. Setting it to **True** is mathematically equivalent to **tf.conj(tf.transpose(input))**. [Restrictions] None [Returns] Transposed Tensor [Quantization tool supporting] Yes

No.	Python API	C++ API	Boundary
45	tf.space_to_batch_nd	SpaceToBatchND	[Arguments] • **input**: ND Tensor with shape input_shape = [batch] + spatial_shape + remaining_shape, where **spatial_shape** has M dimensions. The following data types are supported: uint8, int8, int16, uint16, int32, int64, and float32 • **block_shape**: 1D Tensor of type int32 or int64, with shape [M]. All values must be ≥ 1. • **paddings**: 2D Tensor of type int32 or int64, with shape [M, 2]. All values must be ≥ 0. [Restrictions] • When the tensor rank is 4: the length of **blockShape** must be 2, and the length of paddings must be 4. • Element value of **blockShape** ≥ 1; Element value of **paddings** ≥ 0 • The padded **H** dimension is a multiple of **blockShape[0]**, and the padded **W** dimension is a multiple of **blockShape[1]**. [Returns] Tensor of the identical data type as **input** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
46	tf.batch_to_space_nd	BatchToSpaceND	[Arguments] • **input**: ND Tensor with shape input_shape = [batch] + spatial_shape + remaining_shape, where **spatial_shape** has M dimensions. The following data types are supported: uint8, int8, int16, uint16, int32, int64, and float32 • **block_shape**: 1D Tensor of type int32 or int64, with shape [M]. All values must be ≥ 1. • **crops**: 2D Tensor of type int32 or int64, with shape [M, 2]. All values must be ≥ 0. [Restrictions] • The element data type of **blockShape** and **crops** must be int32. When the dimension count of the Tensor is **4**, the length of **blockShape** must be **2**, and the length of **crops** must be **4**. • Element value of **blockShape** ≥ 1; Element value of **crops** ≥ 0 crop_start[i] + crop_end[i] < block_shape[i] * input_shape[i + 1] [Returns] Tensor of the identical data type as **images** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
47	tf.extract_image_patches	ExtractImagePatches	[Arguments] • **images**: 4-D Tensor with shape [batch, in_rows, in_cols, depth], must be one of the following types: float32, int32, int64, uint8, int8, uint16, and int16; • **ksizes**: list of ints with length ≥ 4 • **strides**: list of ints, must be [1, stride_rows, stride_cols, 1] • **rate**: list of ints, must be [1, rate_rows, rate_cols, 1] • **padding**: string, either **VALID** or **SAME**. **VALID** indicates that the selected patch area must be completely included in the source image. **SAME** indicates that the part that exceeds the source image is padded with **0**. • **name**: (optional) operation name [Restrictions] None [Returns] Tensor of the identical data type as **images** [Quantization tool supporting] No
48	tf.floormod	FloorMod	[Arguments] • **x**: Tensor of type float32 or int32 • **y**: Tensor of the identical data type as **x** • **name**: (optional) operation name [Restrictions] Broadcasting is supported, so the shape of **x** and shape of **y** are compared. For a right-aligned dimension, if the values of **xdim[i]** and **ydim[i]** are not the same, one of them must be **1** or missing. [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
49	tf.nn.softmax	Softmax	[Arguments] • **logits**: non-null Tensor of type float32 • **axis**: dimension softmax to be performed on, default to **–1**, indicating the last dimension. The value cannot be greater than the rank of **logits**. • **name**: (optional) operation name • **dim**: (deprecated) equivalent to **axis** [Restrictions] • If the input contains four dimensions, softmax is performed on each of them. According to **axis**: When axis = 1: C ≤ ((256 * 1024/4) – 8 * 1024 – 256)/2 When axis = 0: N ≤ (56 * 1024 – 256)/2 When axis = 2: W = 1, 0 < H < (1024 * 1024/32) When axis = 3: 0 < W < (1024 * 1024/32) • If the input contains fewer than four dimensions, softmax is performed only on the last dimension, with the last dimension ≤ 46080. [Returns] Tensor of the identical type and shape as **logits** [Quantization tool supporting] Yes
50	tf.math.pow	Power	[Arguments] • **x**: Tensor of type float32 • **y**: Tensor of type float32 • **name**: (optional) operation name [Restrictions] power! = 1 scale * x + shift > 0 [Returns] Tensor [Quantization tool supporting] Yes

No.	Python API	C++ API	Boundary
51	tf.nn.leaky_relu	LeakyRelu	[Arguments] • **features**: Tensor of type float32 • **alpha**: slope of the activation function at x < 0 • **name**: (optional) operation name [Restrictions] None [Returns] Activation value [Quantization tool supporting] Yes
52	tf.placeholder	-	[Arguments] • **dype**: (mandatory) data type • **shape**: (mandatory) shape of the tensor to be fed [Restrictions] None [Returns] Tensor [Quantization tool supporting] No
53	tf.shape	Shape	[Arguments] • **input**: Tensor or SparseTensor • **name**: (optional) operation name, string • **out_type**: data type for the output Tensor, either int32 (default) or int64 [Restrictions] None [Returns] Tensor of the data type specified by **out_type** [Quantization tool supporting] N/A

No.	Python API	C++ API	Boundary
54	tf.math.argmax	ArgMax	[Arguments] • **input**: Tensor, must be one of the following types: int8, uint8, int16, uint16, int32, int64, float32 • **axis**: Tensor of type int32 or int64 • **out_type**: data type for the output Tensor, either int32 or int64 (default) • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of the data type specified by **out_type** [Quantization tool supporting] No
55	tf.gather	Gather GatherV2	[Arguments] • **params**: Tensor, must be at least rank **axis + 1** • **indices**: Tensor of type float32 or int64, must be in range [0, params.shape[axis]) • **axis**: output Tensor of type float32 or int64, specifying the axis in params to gather indices from, rank = 0 • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of the identical data type as **params** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
56	tf.gather_nd	GatherNd	[Arguments] • **params**: Tensor, must be at least rank **axis + 1** • **axis**: Tensor of type int32 or int64 • **name**: (optional) operation name, string [Restrictions] **indices**: The last dimension of **indices** can be at most the rank of **params**. The elements in the last dimension of **indices** correspond to the coordinates along a dimension of **params**. Therefore, the coordinate rules must be met. The coordinates along the corresponding dimension of **indices** cannot exceed the dimension size. [Returns] Tensor of the identical data type as **params** [Quantization tool supporting] Yes
57	tf.math.floordiv	FloorDiv	[Arguments] • **x**: Tensor of type float32 or int32 • **y**: Tensor, denominator of type float32 or int32 • **name**: (optional) operation name, string [Restrictions] Broadcasting is supported, so the shape of x and shape of y are compared. For a right-aligned dimension, if the values of xdim[i] and ydim[i] are not the same, one of them must be 1 or missing. [Returns] Tensor, floor (x/y) [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
58	tf.range	Range	[Arguments] • **start**: start constant scalar of type float32 or int32 • **limit**: end constant scalar of type float32 or int32 • **delta**: stride constant scalar of type float32 or int32 • **dtype**: data type of the resulting Tensor • **name**: (optional) operation name, string [Restrictions] None [Returns] 1D Tensor [Quantization tool supporting] No
59	tf.tile	Tile	[Arguments] • **input**: Tensor, must be one of the following types: int8, uint8, int16, uint16, int32, int64, float32 • **multiples**: 1D constant Tensor of type int32. The length must be the same as that of **input**. • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor [Quantization tool supporting] Yes

No.	Python API	C++ API	Boundary
60	tf.size	Size	[Arguments] • **input**: Tensor of type float32 • **name**: (optional) operation name, string • **out_type**: data type for the output Tensor, default to int32 [Restrictions] None [Returns] Tensor of the data type specified by **out_type** [Quantization tool supporting] No
61	tf.fill	Fill	[Arguments] • **dims**: 1D Tensor of type int32 • **value**: variable of type int32 or float32 • **name**: (optional) operation name, string [Restrictions] The following padding modes are supported: Constant, GivenTensor, Range, Diagonal, Gaussian, MSRA, Uniform, UniformInt, UniqueUniform, and XavierFill. When the Uniform, UniformInt, UniqueUniform, and xavier padding modes are used, the value range of the generated value is [min, max). [Returns] Tensor of the identical data type as **value** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
62	tf.concat	Concat	[Arguments] • **value**: list of Tensor objects of type int32 or float32 • **axis**: Tensor of type int32, specifying the dimension to be concatenated • **name**: (optional) operation name, string [Restrictions] The number of dimensions of the input tensors must match, and all dimensions except **axis** must be equal. The range of the input Tensor count is [1, 1000]. [Returns] Tensor [Quantization tool supporting] Yes
63	tf.reverse	Reverse	[Arguments] • **tensor**: list of Tensor objects • **axis**: dimensions to reverse, of type int32 • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of the identical data type as **tensor** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
64	tf.reduce_sum	sum	[Arguments] • **input_tensor**: Tensor • **axis**: dimensions to sum up, int32 • **keepdims**: bool, indicating whether to retain reduced dimensions • **name**: (optional) operation name, string • **reduction_indices**: (deprecated) string, equivalent to **axis** • **keep_dims**: (deprecated) equivalent to **keepdims** [Restrictions] None [Returns] Tensor of the identical data type as **tensor** [Quantization tool supporting] No
65	tf.math.maximum	Maximum	[Arguments] • **x**: Tensor, must be one of the following types: int32, int64, float32 • **y**: Tensor of the identical data type as **x** • **name**: (optional) operation name, string [Restrictions] Broadcasting is supported, so the shape of x and shape of y are compared. For a right-aligned dimension, if the values of xdim[i] and ydim[i] are not the same, one of them must be 1 or missing. [Returns] Tensor. Returns the max of x and y (x > y ? x: y). Identical data type as **x** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
66	tf.math.minimum	Minimum	[Arguments] • **x**: Tensor, must be one of the following types: int32, int64, float32 • **y**: Tensor of the identical data type as **x** • **name**: (optional) operation name [Restrictions] Broadcasting is supported in the following two scenarios: NHWC+scaler, NHWC+NHWC [Returns] Tensor. Returns the min of x and y (x < y ? x : y). Identical data type as **x** [Quantization tool supporting] No
67	tf.clip_by_value	ClipByValue	[Arguments] • **t**: Tensor • **clip_value_min**: minimum value to clip by • **clip_value_max**: maximum value to clip by • **name**: (optional) operation name, string [Restrictions] The minimum value must be less than or equal to the maximum value. [Returns] Clipped Tensor. The return value range is [clip_value_min, clip_value_max]. [Quantization tool supporting] No
68	tf.math.logical_not	LogicalNot	[Arguments] • **x**: Tensor of type bool • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of type bool [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
69	tf.math.logical_and	LogicalAnd	[Arguments] • **x**: Tensor of type bool • **y**: Tensor of type bool • **name**: (optional) operation name, string [Restrictions] Broadcasting is supported in the following dimension scenarios: NHWC and [1,1,1,1], [N,C,H,W], [N,1,H,W], [1,C,H,W], [N,C,1,1] [Returns] Tensor of type bool [Quantization tool supporting] No
70	tf.equal	Equal	[Arguments] • **x**: Tensor • **y**: Tensor of the identical data type as **x** • **name**: (optional) operation name, string [Restrictions] Broadcasting is supported, so the shape of x and shape of y are compared. For a right-aligned dimension, if the values of **xdim[i]** and **ydim[i]** are not the same, one of them must be 1 or missing. [Returns] Tensor of type bool [Quantization tool supporting] No
71	tf.square	Square	[Arguments] • **x**: Tensor • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
72	tf.image.crop_and_resize	CropAndResize	[Arguments] • **image**: 4D Tensor, must be one of the following types: float32 and int8, int32, int64; with shape [num_boxes, 4] • boxes: 2D Tensor of type float32, with shape [num_boxes] • **box_ind**: 1D Tensor of type int32 • **crop_size**: 1D 2-element Tensor of type int32 • method: interpolation method string, options: **bilinear** (default) or **nearest** • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of type float32 [Quantization tool supporting] No
73	tf.math.top_k	TopKV2	[Arguments] • **input**: 1D Tensor or higher with the last dimension at least **k**, of type float32 (k: scalar of type int32, ≥ 1) • **sorted**: bool • **name**: (optional) operation name, string [Restrictions] **k**: constant [Returns] • **values**: Tensor, indicating k largest elements along each last dimensional slice • **indices**: Tensor, indicating the indices of **values** of **input** [Quantization tool supporting] • No

No.	Python API	C++ API	Boundary
74	tf.invert_permutation	InvertPermutation	[Arguments] • **x**: 1D Tensor of type int32 or int64 • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] No
75	tf.multinomial	Multinomial	[Arguments] • logits: 2D Tensor with shape [batch_size, num_classes] • **num_samples**: scalar, indicating the number of samples to draw • **seed**: int32 or int64, used to create a random seed • **name**: (optional) operation name, string • **output_dtype**: integer, data type for the output Tensor, default to int64 [Restrictions] When **seed** is **0**, the generated random is dynamic. The number of output data rows is the actual number of output data rows while the number of output data columns is **num_samples**. [Returns] Tensor of the data type specified by **output_dtype** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
76	tf.reverse_sequence	ReverseSequence	[Arguments] • **input**: Tensor • **seq_lengths**: 1D Tensor of type int32 or int64 • **seq_axis**: scalar of type integer • **batch_axis**: scalar of type integer • **name**: (optional) operation name, string [Restrictions] • The length of **seq_lengths** must be equal to the number of elements of input in **batchAxis**. • The maximum element in **seq_lengths** must be less than or equal to the number of elements in **seq_dim**. • **seqAxis**, **batchAxis**, **seqDim**, and **batchDim** must be of type int64. • **seqAxis** and **seqDim** are exclusive. **batchAxis** and **batchDim** are exclusive. • **batchAxis** and **batchDim** are optional. The default values are **0**. Only one weight is required. [Returns] Tensor of the identical data type as **input** [Quantization tool supporting] No
77	tf.math.reciprocal	Reciprocal	[Arguments] • **x**: Tensor of type float32 • **name**: (optional) operation name, string [Restrictions] The input data cannot contain **0**. [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
78	tf.nn.selu	Selu	[Arguments] • **features**: Tensor of type float32 • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of the identical data type as **features** [Quantization tool supporting] No
79	tf.math.acosh	Acosh	[Arguments] • **x**: Tensor of type float32 • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] No
80	tf.math.asinh	Asinh	[Arguments] • **x**: Tensor of type float32 • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
81	tf.math.reduce_prod	Prod	[Arguments] • **input_tensor**: Tensor • **axis**: dimension to reduce • **keepdims**: bool, indicating whether to retain reduced dimensions • **name**: (optional) operation name, string • **reduction_indices**: (deprecated) equivalent to **axis** • **keep_dims**: (deprecated) equivalent to keepdims [Restrictions] None [Returns] Tensor and reduced Tensor [Quantization tool supporting] No
82	tf.math.sqrt	Sqrt	[Arguments] • **x**: Tensor of type float32 • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
83	tf.math.reduce_all	All	[Arguments] • **input_tensor**: Tensor of type bool • **axis**: dimension to reduce • **keepdims**: bool • **name**: (optional) operation name, string • **reduction_indices**: (deprecated) equivalent to **axis** • **keep_dims**: (deprecated) equivalent to **keepdims** [Restrictions] None [Returns] Tensor of the identical data type as **input_tensor** [Quantization tool supporting] No
84	tf.nn.l2_normalize	L2Normalize	[Arguments] • **x**: Tensor of type bool • **axis**: dimension along which to normalize For format NCHW, **axis** must be set to **1**. For format NHWC, **axis** must be set to **3**. • **epsilon**: lower bound value for the norm. Value range: (1e-7, 0.1]. If norm < sqrt(epsilon), **sqrt(epsilon)** is used as the divisor. • **name**: (optional) operation name, string • **dim**: (deprecated) equivalent to **axis** [Restrictions] H * W * 2 < 256* 1024/4 [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
85	tf.keras.backend.hard_sigmoid	Hardsigmoid	[Arguments] **x**: Tensor [Restrictions] None [Returns] Output Tensor If x < –2.5, **0** is returned. If x > 2.5, **1** is returned. If –2.5 ≤ x ≤ 2.5, **0.2 * x + 0.5** is returned. [Quantization tool supporting] No
86	tf.keras.layers.ThresholdedReLU	ThresholdedReLU	[Arguments] **theta**: scalar ≥ 0 of type float32 [Restrictions] None [Returns] Tensor [Quantization tool supporting] No
87	tf.math.acos	Acos	[Arguments] • **x**: Tensor, must be one of the following types: float32, int32, int64 • **name**: (optional) operation name, string [Restrictions] The input data range is (–1 ≤ x ≤ +1), and the output data range is (0 ≤ y ≤ π). [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
88	tf.math.atan	Arctan	[Arguments] • **x**: Tensor, must be one of the following types: float32, int32, int64 • **name**: (optional) operation name, string [Restrictions] The input data range is (–65504 ≤ x ≤ +65504), and the output data range is (–π/2 ≤ y ≤ +π/2). [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] No
89	tf.math.asin	Asin	[Arguments] • **x**: Tensor, must be one of the following types: float32, int32, int64 • **name**: (optional) operation name, string [Restrictions] The input data range is (–1 ≤ x ≤ +1), and the output data range is (–π/2 ≤ y ≤ +π/2). [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] No
90	tf.math.atanh	Atanh	[Arguments] • **x**: Tensor, must be one of the following types: float32, int32, int64 • **name**: (optional) operation name, string [Restrictions] Input data range: **x** is of range (–1, +1) [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
91	tf.math.tan	Tan	[Arguments] • **x**: Tensor, must be one of the following types: float32, int32, int64 • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] No
92	tf.math.logical_or	LogicalOr	[Arguments] • **x**: Tensor of type bool • **y**: Tensor of type bool • **name**: (optional) operation name, string [Restrictions] Broadcasting is supported, so the shape of x and shape of y are compared. For a right-aligned dimension, if the values of **xdim[i]** and **ydim[i]** are not the same, one of them must be **1** or missing. [Returns] Tensor of type bool [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
93	tf.math.reduce_min	ReduceMin	[Arguments] • **input_tensor**: Tensor, must be one of the following types: float32, int64, int32, uint8, uint16, int8, int16 • **axis**: dimension to reduce • **keepdims**: scalar of type bool • **name**: (optional) operation name, string • **reduction_indices**: (deprecated) equivalent to axis • **keep_dims**: (deprecated) equivalent to **keepdims** [Restrictions] • When the input Tensor has four dimensions: input axis = {3,{1,2,3}}, keepDims = true, H * W * 16 * 2 ≤ 16 * 1024 • When the input Tensor has two dimensions: input axis = {1,{1}}, keepDims = true, H * W * CEIL(C, 16) * 16 * 2 ≤ 16 * 1024 [Returns] Tensor of the identical data type as **input_tensor** [Quantization tool supporting] No
94	tf.math.negative	Neg	[Arguments] • **x**: Tensor, must be one of the following types: float32, int64, int32 • **name**: (optional) operation name, string [Restrictions] The input data range is –65504 ≤ x ≤ +65504, and the output data range is –65504 ≤ y ≤ +65504. [Returns] Tensor. Returns **–x**. [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
95	tf.math.greater_equal	Greaterequal	[Arguments] • **x**: Tensor, must be one of the following types: float32, int64, int32, uint8, uint16, int8, int16 • **y**: Tensor of the identical data type as **x** • **name**: (optional) operation name, string [Restrictions] Input data range: –65504 ≤ x ≤ +65504 [Returns] Tensor of type bool [Quantization tool supporting] No
96	tf.space_to_depth	SpaceToDepth	[Arguments] • **input**: Tensor, must be one of the following types: float32, int64, int32, uint8, int8 • **block_size**: scalar of type integer, ≥ 2 • **data_format**: string, default to **NHWC** (options: **NHWC**, **NCHW**, **NCHW_VECT_C**) • **name**: (optional) operation name, string [Restrictions] blockSize ≥ 1 and **blockSize** must be a divisor of both the input height and width. [Returns] Tensor of the identical data type as **input** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
97	tf.depth_to_space	DepthToSpace	[Arguments] • **input**: Tensor, must be one of the following types: float32, int64, int32, uint8, int8 • **block_size**: scalar of type integer, ≥ 2 • **data_format**: string, default to **NHWC** (options: **NHWC**, **NCHW**, **NCHW_VECT_C**) • **name**: (optional) operation name, string [Restrictions] **blockSize** must be greater than or equal to **1**, and **blockSize * blockSize** must be exactly divided by **C**. [Returns] Tensor of the identical data type as **input** [Quantization tool supporting] No
98	tf.math.round	Round	[Arguments] • **x**: Tensor, must be one of the following types: float32, int64, int32 • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of the identical data type and shape as **x** [Quantization tool supporting] No
99	tf.math.rint	Rint	[Arguments] • **x**: Tensor, must be one of the following types: float32, int64, int32, uint8, int8 • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of the identical data type and shape as **x** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
100	tf.math.less	Less	[Arguments] • **x**: Tensor, must be one of the following types: float32, int64, int32, uint8, uint16, int8, int16 • **y**: Tensor of the identical data type as **x** • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of type bool [Quantization tool supporting] No
101	tf.math.sinh	Sinh	[Arguments] • **x**: Tensor of type float32 • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] No
102	tf.math.cosh	Cosh	[Arguments] • **x**: Tensor of type float32 • **name**: (optional) operation name, string [Restrictions] None [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] No

No.	Python API	C++ API	Boundary
103	tf.math.squared_difference	Squared_difference	[Arguments] • **x**: Tensor, must be one of the following types: float32, int64, int32 • **y**: Tensor of the identical data type as **x** • **name**: (optional) operation name, string [Restrictions] Broadcasting is supported only in the following scenarios: One NCHW Tensor and one Tensor of the following format: dim{} = [1,1,1,1], [N,C,H,W], [N,1,H,W], [1,C,H,W], [N,C,1,1], [1,C,1,1], [1,1,H,W], or [N,1,1,1] [Returns] Tensor of the identical data type as **x** [Quantization tool supporting] No

3 Appendix

3.1 Change History

Release Date	Description
2020-05-30	This issue is the first official release.

