

tf.batch_to_space

```
batch_to_space(  
    input,  
    crops,  
    block_size,  
    name=None  
)
```

Defined in [tensorflow/python/ops/array_ops.py](#).

See the guide: [Tensor Transformations > Slicing and Joining](#)

BatchToSpace for 4-D tensors of type T.

This is a legacy version of the more general BatchToSpaceND.

Rearranges (permutes) data from batch into blocks of spatial data, followed by cropping. This is the reverse transformation of SpaceToBatch. More specifically, this op outputs a copy of the input tensor where values from the **batch** dimension are moved in spatial blocks to the **height** and **width** dimensions, followed by cropping along the **height** and **width** dimensions.

Args:

- input**: A **Tensor**. 4-D tensor with shape **[batch*block_size*block_size, height_pad/block_size, width_pad/block_size, depth]**. Note that the batch size of the input tensor must be divisible by **block_size * block_size**.
- crops**: A **Tensor**. Must be one of the following types: **int32**, **int64**. 2-D tensor of non-negative integers with shape **[2, 2]**. It specifies how many elements to crop from the intermediate result across the spatial dimensions as follows:

```
crops = [[crop_top, crop_bottom], [crop_left, crop_right]]
```

- block_size**: An **int** that is **>= 2**.
- name**: A name for the operation (optional).

Returns:

A **Tensor**. Has the same type as **input**. 4-D with shape **[batch, height, width, depth]**, where:

```
height = height_pad - crop_top - crop_bottom  
width = width_pad - crop_left - crop_right
```

The attr **block_size** must be greater than one. It indicates the block size.

Some examples:

(1) For the following input of shape **[4, 1, 1, 1]** and **block_size** of 2:

```
[[[1]], [[2]], [[3]], [[4]]]
```

The output tensor has shape `[1, 2, 2, 1]` and value:

```
x = [[[[1], [2]], [[3], [4]]]]
```

(2) For the following input of shape `[4, 1, 1, 3]` and `block_size` of 2:

```
[[[1, 2, 3]], [[4, 5, 6]], [[7, 8, 9]], [[10, 11, 12]]]
```

The output tensor has shape `[1, 2, 2, 3]` and value:

```
x = [[[[1, 2, 3], [4, 5, 6]],  
      [[7, 8, 9], [10, 11, 12]]]]
```

(3) For the following input of shape `[4, 2, 2, 1]` and `block_size` of 2:

```
x = [[[[1], [3]], [[9], [11]]],  
      [[2], [4]], [[10], [12]]],  
      [[5], [7]], [[13], [15]]],  
      [[6], [8]], [[14], [16]]]]
```

The output tensor has shape `[1, 4, 4, 1]` and value:

```
x = [[[[1], [2], [3], [4]],  
      [[5], [6], [7], [8]],  
      [[9], [10], [11], [12]],  
      [[13], [14], [15], [16]]]]
```

(4) For the following input of shape `[8, 1, 2, 1]` and `block_size` of 2:

```
x = [[[[1], [3]]], [[9], [11]]], [[2], [4]], [[10], [12]]],  
      [[5], [7]]], [[13], [15]]], [[6], [8]], [[14], [16]]]]
```

The output tensor has shape `[2, 2, 4, 1]` and value:

```
x = [[[[1], [3]], [[5], [7]]],  
      [[2], [4]], [[10], [12]]],  
      [[5], [7]], [[13], [15]]],  
      [[6], [8]], [[14], [16]]]]
```

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated November 2, 2017.

Stay Connected

[Blog](#)

[GitHub](#)

[Twitter](#)

Support

[Issue Tracker](#)

[Release Notes](#)

English

[Terms](#) | [Privacy](#)