tf.nn.conv2d_backprop_filter

```
conv2d_backprop_filter(
    input,
    filter_sizes,
    out_backprop,
    strides,
    padding,
    use_cudnn_on_gpu=True,
    data_format='NHWC',
    name=None
)
```

Defined in `tensorflow/python/ops/gen_nn_ops.py`.

See the guide: Neural Network > Convolution

Computes the gradients of convolution with respect to the filter.

Args:

- `input`: A `Tensor`. Must be one of the following types: `half`, `float32`. 4-D with shape `[batch, in_height, in_width, in_channels]`.
- `filter_sizes`: A `Tensor` of type `int32`. An integer vector representing the tensor shape of `filter`, where `filter` is a 4-D `[filter_height, filter_width, in_channels, out_channels]` tensor.
- `out_backprop`: A `Tensor`. Must have the same type as `input`. 4-D with shape `[batch, out_height, out_width, out_channels]`. Gradients w.r.t. the output of the convolution.
- `strides`: A list of `ints`. The stride of the sliding window for each dimension of the input of the convolution. Must be in the same order as the dimension specified with format.
- `padding`: A `string` from: `"SAME", "VALID"`. The type of padding algorithm to use.
- `use_cudnn_on_gpu`: An optional `bool`. Defaults to `True`.
- `data_format`: An optional `string` from: `"NHWC", "NCHW"`. Defaults to `"NHWC"`. Specify the data format of the input and output data. With the default format "NHWC", the data is stored in the order of: [batch, in_height, in_width, in_channels]. Alternatively, the format could be "NCHW", the data storage order of: [batch, in_channels, in_height, in_width].
- `name`: A name for the operation (optional).

Returns:

A `Tensor`. Has the same type as `input`. 4-D with shape `[filter_height, filter_width, in_channels, out_channels]`. Gradient w.r.t. the `filter` input of the convolution.

Last updated November 2, 2017.

Stay Connected

Blog

GitHub

Twitter

Support

Issue Tracker

Release Notes

Stack Overflow

English

Terms | **Privacy**