# tf.image.resize_images

```
resize_images(
    images,
    size,
    method=ResizeMethod.BILINEAR,
    align_corners=False
)
```

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

See the guide: Images > Resizing

Resize `images` to `size` using the specified `method`.

Resized images will be distorted if their original aspect ratio is not the same as `size`. To avoid distortions see `tf.image.resize_image_with_crop_or_pad`.

`method` can be one of:

- `ResizeMethod.BILINEAR`: Bilinear interpolation.
- `ResizeMethod.NEAREST_NEIGHBOR`: Nearest neighbor interpolation.
- `ResizeMethod.BICUBIC`: Bicubic interpolation.
- `ResizeMethod.AREA`: Area interpolation.

The return value has the same type as `images` if `method` is `ResizeMethod.NEAREST_NEIGHBOR`. It will also have the same type as `images` if the size of `images` can be statically determined to be the same as `size`, because `images` is returned in this case. Otherwise, the return value has type `float32`.

## Args:

- `images`: 4-D Tensor of shape `[batch, height, width, channels]` or 3-D Tensor of shape `[height, width, channels]`.
- `size`: A 1-D int32 Tensor of 2 elements: `new_height, new_width`. The new size for the images.
- `method`: ResizeMethod. Defaults to `ResizeMethod.BILINEAR`.
- `align_corners`: bool. If true, exactly align all 4 corners of the input and output. Defaults to `false`.

## Raises:

- `ValueError`: if the shape of `images` is incompatible with the shape arguments to this function
- `ValueError`: if `size` has invalid shape or type.
- `ValueError`: if an unsupported resize method is specified.

## Returns:

If `images` was 4-D, a 4-D float Tensor of shape `[batch, new_height, new_width, channels]`. If `images` was 3-D, a 3-D float Tensor of shape `[new_height, new_width, channels]`.

*Last updated November 2, 2017.*