

# What the heck does it mean to make an image a tensor?



NOVEMBER 29, 2021

So we've learned about [what a tensor is](#), and we know a little bit about [what you can do with them](#), but why the heck would we want to turn an image into a tensor? Well, if we want to use them with TensorFlow.js or for Deep Learning, then we change them to tensors to put them in a useable form.

When we think about images, we know they're made of pixels. Together, those small units create an image. Each of these pixels holds a position in the image, and each pixel can be represented by an array of three numbers that correspond with an [RGB \(red, green, blue\) value](#) that represents a color.

So, for example, if we look at the rgb value (0, 255, 255), we'll see aqua. Now, you might see a fourth value, and that stands for the transparency of the pixel. That fourth value—the alpha parameter—allows you to set a number between 0.0 (transparent) and 1.0 (opaque). Take that aqua color. If we add an alpha parameter of 0, it will be fully transparent and we won't be able to see it. If we add .5, then it will be at a middle point for transparency.

When we transform an image into a tensor, each of those pixels in the position they hold in the image, get transformed into tensors. Once we transform these tensors, we can train our models with this data, among other things. We can mirror images, resize, crop, and manipulate in other ways that are useful to us.

Let's say that we're [Air BnB](#). We want to be able to quickly look through the images our hosts have submitted but haven't labeled, so we can provide labels for those images. We know that *most* of the properties will have bathrooms. What's the easiest way to identify a bathroom? If you said by toilets, I would say that you're right. So we need our image classification to know what a toilet looks like. But not everyone has the same toilet, in the same position, and/or has taken the picture from the same angle. To increase accuracy, we need data taken from different angles, with different size images of toilets, etc.

One way to augment your data is to take an image, like a toilet, and apply a method like [.reverse\(\)](#) to the image. By reversing the image to add to our dataset, we have doubled our dataset and made it more reliable.

Let's simplify the steps you need to get there. Let's imagine that we have an image of a toilet that we have stored in the variable toiletImg.

First, we'll convert that image into a tensor—for the example, we'll grab it from the browser.

```
const toiletTensor = tf.browser.fromPixels(toiletImg)
```

Now we have a tensor, but we want to reverse the image:

```
tf.reverse(toiletImg, 1)
```

You'll notice that I have a 1 there. What's that for? That second parameter indicates that we want to flip the axis that holds the pixels for the *width* of the image.

This is one example of what you can do, *but* I wasn't making up the AirBnB example. They have [used image classification to help them identify images](#). Can you imagine how much time that saved them? There's a lot of power in Machine Learning, image classification, and possibilities that have been explored and remain to be explored. But you always start with the basics. Let yourself be inspired and then take it one day at a time. And if you're interested in learning more, check out my [livestreams on TFJS on Tuesday afternoons at 1pm est](#) or let me know if you want to join our study group on Friday afternoons at 1pm est!

SHARE:

[Next](#)

Moms, you're carrying a lot of plates

NOVEMBER 30, 2021

[Previous](#)

Dear Senior Devs, Use your privilege

NOVEMBER 28, 2021

