Using Image Generator, how do you label images?

- TensorFlow figures it out from the contents
- You have to manually do it
- It's based on the file name
- **It's based on the directory the image is contained in**

What method on the Image Generator is used to normalize the image?

- Rescale_image
- normalize
- normalize_image
- **rescale**

How did we specify the training size for the images?

- The target_size parameter on the validation generator
- **The target_size parameter on the training generator**
- The training_size parameter on the training generator
- The training_size parameter on the validation generator

When we specify the input_shape to be (300, 300, 3), what does that mean?

- Every Image will be 300x300 pixels, and there should be 3 Convolutional Layers
- **Every Image will be 300x300 pixels, with 3 bytes to define color**
- There will be 300 horses and 300 humans, loaded in batches of 3
- There will be 300 images, each size 300, loaded in batches of 3

If your training data is close to 1.000 accuracy, but your validation data isn't, what's the risk here?

- You're underfitting on your validation data
- No risk, that's a great result
- **You're overfitting on your training data**
- You're overfitting on your validation data

Convolutional Neural Networks are better for classifying images like horses and humans because:

- **There's a wide variety of horses**
- **In these images, the features may be in different parts of the frame**
- **There's a wide variety of humans**

After reducing the size of the images, the training results were different. Why?

- The training was faster
- There was less information in the images
- **We removed some convolutions to handle the smaller images**
- There was more condensed information in the images