How do you use Image Augmentation in TensorFLow?

- With the tf.augment API
- You have to write a plugin to extend tf.layers
- **Using parameters to the ImageDataGenerator**
- With the keras.augment API

If my training data only has people facing left, but I want to classify people facing right, how would I avoid overfitting?

- Use the ‘flip’ parameter and set ‘horizontal’
- Use the ‘flip_vertical’ parameter around the Y axis
- Use the ‘flip’ parameter
- **Use the ‘horizontal_flip’ parameter**

After adding data augmentation and using the same batch size and steps per epoch, you noticed that each training epoch became a little slower than when you trained without it. Why?

- Because the augmented data is bigger
- Because there is more data to train on
- **Because the image preprocessing takes cycles**
- Because the training is making more mistakes

What does the fill_mode parameter do?

- There is no fill_mode parameter
- It creates random noise in the image
- **It attempts to recreate lost information after a transformation like a shear**
- It masks the background of an image

When using Image Augmentation with the ImageDataGenerator, what happens to your raw image data on-disk?

- It gets overwritten, so be sure to make a backup
- A copy is made and the augmentation is done on the copy
- **Nothing, all augmentation is done in-memory**
- It gets deleted

How does Image Augmentation help solve overfitting?

- It slows down the training process
- **It manipulates the training set to generate more scenarios for features in the images**
- It manipulates the validation set to generate more scenarios for features in the images

- It automatically fits features to images by finding them through image processing techniques

When using Image Augmentation my training gets...

- **Slower**
- Faster
- Stays the Same
- Much faster

Using Image Augmentation effectively simulates having a larger data set for training.

- False
- **True**