

What does `flow_from_directory` give you on the `ImageDataGenerator`?

- The ability to easily load images for training
- The ability to pick the size of training images
- The ability to automatically label images based on their directory name
- **All of the Above**

If my Image is sized 150x150, and I pass a 3x3 Convolution over it, what size is the resulting image?

- 153x153
- **148x148**
- 150x150
- 450x450

If my data is sized 150x150, and I use Pooling of size 2x2, what size will the resulting image be?

- 149x149
- 148x148
- **75x75**
- 300x300

If I want to view the history of my training, how can I access it?

- Use a `model.fit_generator`
- Pass the parameter `'history=true'` to the `model.fit`
- Download the model and inspect it
- **Create a variable 'history' and assign it to the return of `model.fit` or `model.fit_generator`**

What's the name of the API that allows you to inspect the impact of convolutions on the images?

- The model.images API
- The model.pools API
- **The model.layers API**
- The model.convolution API

When exploring the graphs, the loss levelled out at about .75 after 2 epochs, but the accuracy climbed close to 1.0 after 15 epochs. What's the significance of this?

- There was no point training after 2 epochs, as we overfit to the validation data
- **There was no point training after 2 epochs, as we overfit to the training data**
- A bigger training set would give us better validation accuracy
- A bigger validation set would give us better training accuracy

Why is validation accuracy a better indicator of model performance than training accuracy?

- It isn't, they're equally valuable
- There's no relationship between them
- **The validation accuracy is based on images that the model hasn't been trained with, and thus a better indicator of how the model will perform with new images.**
- The validation dataset is smaller, and thus less accurate at measuring accuracy, so its performance isn't as important

Why is overfitting more likely to occur on smaller datasets?

- Because in a smaller dataset, your validation data is more likely to look like your training data
- Because there isn't enough data to activate all the convolutions or neurons
- Because with less data, the training will take place more quickly, and some features may be missed
- **Because there's less likelihood of all possible features being encountered in the training process**