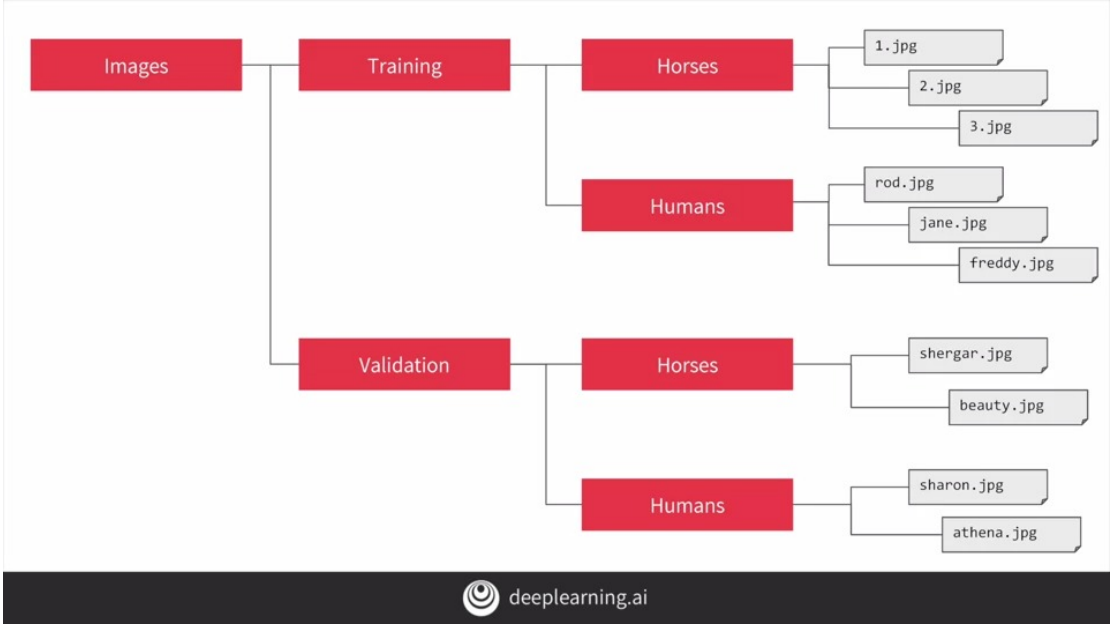
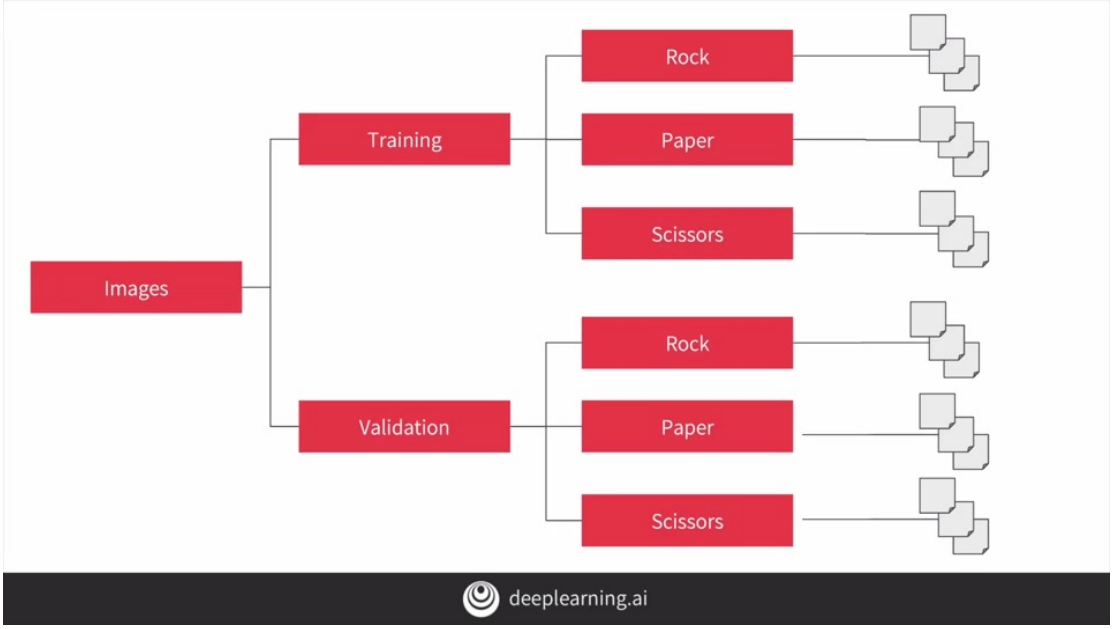
4.1 Moving from binary to multi-class classification



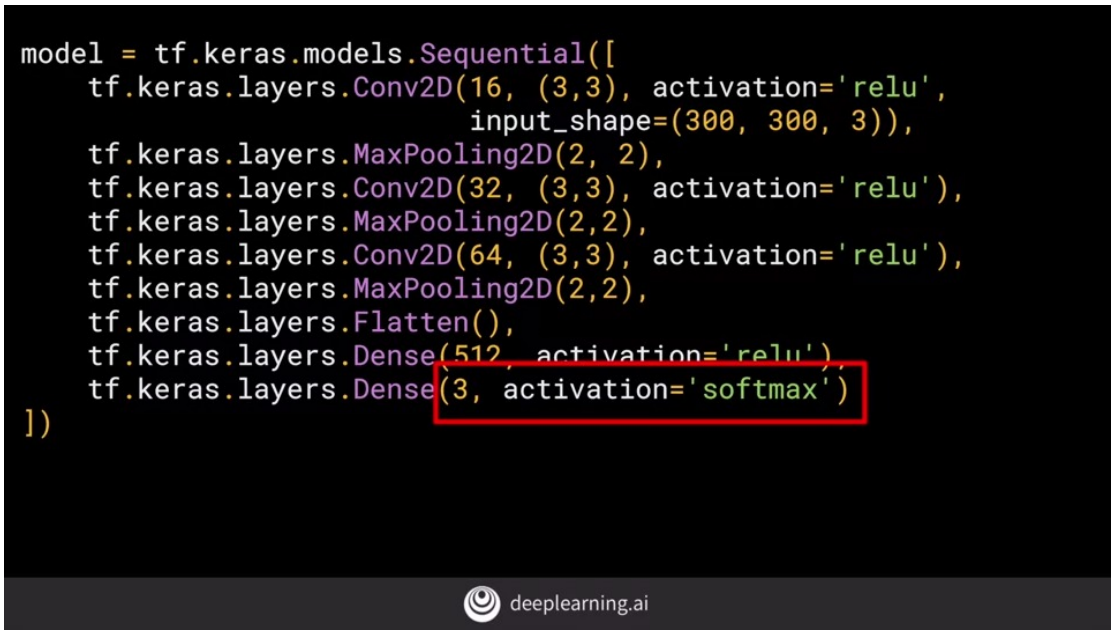


Remember when we were classifying horses or human, we had a file structure like this. There were subdirectories for each class, where in this case we only had two. The first thing that you'll need to do is replicate this for multiple classes like this. It's very similar and here you can see that both the training and validation have three subdirectories. One for Rock, one for Paper, and one for Scissors. In these, we can put training and validation images for Rock, Paper, and Scissors.

http://www.laurencemoroney.com/rock-paper-scissors-dataset/
Rock Paper Scissors is a dataset containing 2,892 images of diverse hands in Rock/Paper/Scissors poses. It is licensed CC By 2.0 and available for all purposes, but it's intent

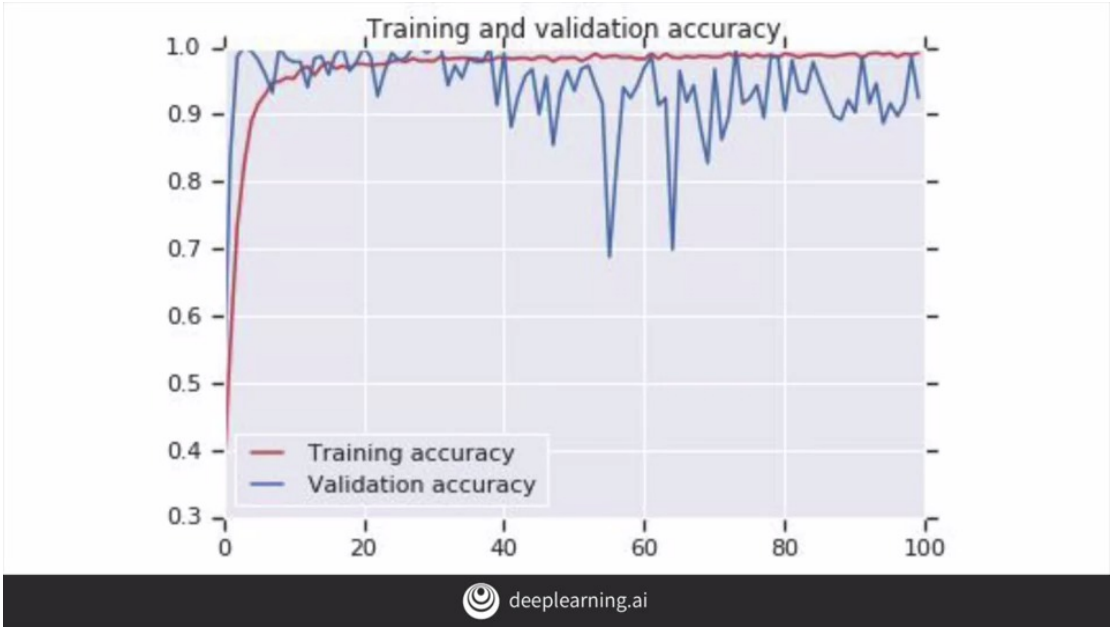
is primarily for learning and research.
Rock Paper Scissors contains images from a variety of different hands, from different races, ages and genders, posed into Rock / Paper or Scissors and labelled as such. You can download the training set here, and the test set here. These images have all been generated using CGI techniques as an experiment in determining if a CGI-based dataset can be used for classification against real images. I also generated a few images that you can use for predictions. You can find them here.
Note that all of this data is posed against a white background.
Each image is 300×300 pixels in 24-bit color
You'll see how this dataset can be used to build a multi-class classifier in the next video

4.2 Explore multi-class with Rock Paper Scissors dataset




```
from tensorflow.keras.optimizers import RMSprop

model.compile(loss='categorical_crossentropy',
              optimizer=RMSprop(lr=0.001),
              metrics=['acc'])
```



4.3 Train a classifier with Rock Paper Scissors

rock_paper_scissors.ipynb

Note that while the images are 300 by 300, we are setting up the image generators to give us 150 by 150 variance. It will resize them on the fly and augment the ones in the training directory.
I'm only training for 25 epochs, based on the chart you saw in the last lesson. But by the time we reach the 10th epoch, we're already doing quite well. By the time we finish, the training data is above 98% and the validation data is at 95 percent accuracy. This is highly specialized data that's optimized for this lesson and not a great real-world scenario for Rock, Paper, and Scissors.
We can see the training improving and trending towards one.
The validation zig-zags a bit, but it's always between 0.9 and one after the first few epochs.

4.4 Test the Rock Paper Scissors classifier

rock_paper_scissors.ipynb

You're coming to the end of Course 2, and you've come a long way! From first principles in understanding how ML works, to using a DNN to do basic computer vision, and then beyond into Convolutions.
With Convolutions, you then saw how to extract features from an image, and you saw the tools in TensorFlow and Keras to build with Convolutions and Pooling as well as handling complex, multi-sized images.
Through this you saw how overfitting can have an impact on your classifiers, and explored some strategies to avoid it, including Image Augmentation, Dropouts, Transfer Learning and more. To wrap things up, this week you've looked at the considerations in your code that you need for moving towards multi-class classification!