Link to Cats and Dogs Dataset:

<https://www.microsoft.com/en-us/download/confirmation.aspx?id=54765>

1. The necessary libraries are imported, including os, shutil, matplotlib.pyplot, and various modules from Keras for building and training the neural network.
2. Paths to the data directories, including training, validation, and test sets, are defined.
3. The copy\_files() function is defined to copy the image files from a base data directory to their respective train, validation, and test directories for cats and dogs.
4. The display\_results() function is defined to visualize the training and validation accuracy and loss over epochs.
5. Image data generators are created using the ImageDataGenerator class from Keras. These generators will apply various data augmentation techniques to the images, such as rescaling, rotation, shifting, shearing, zooming, and horizontal flipping.
6. The generators are then used to load the images from the respective directories, applying the specified transformations and resizing the images to the target size of (150, 150).
7. The VGG16 model is instantiated with pre-trained weights from the 'imagenet' dataset. The include\_top argument is set to False to exclude the fully connected layers at the top of the network.
8. A new sequential model is created, and the VGG16 model is added as the base convolutional feature extractor.
9. Additional layers are added to the model, including a flatten layer, a dense layer with 216 units and ReLU activation, and a final dense layer with 1 unit and a sigmoid activation function.
10. The model is compiled with a binary cross-entropy loss function, an RMSprop optimizer with a learning rate of 1e-4, and accuracy as the evaluation metric.
11. The model summary is printed, displaying the architecture and the number of parameters.
12. The model is trained using the fit\_generator function, with the training and validation generators as inputs. The number of steps per epoch, number of epochs, and validation steps are specified.
13. The top layers of the VGG16 model are unfrozen, allowing them to be fine-tuned. A loop is used to iterate through the layers, and when the layer named 'block5\_conv1' is encountered, all subsequent layers are set to trainable.
14. The model is recompiled with a lower learning rate of 1e-5, and again trained using the fit\_generator function.
15. The trained model is saved to the specified model\_path.