```
            ax.set_xticks([])
            ax.set_yticks([])
            # plot filter channel in grayscale
            pyplot.imshow(fmap[0, :, :, ix-1], cmap='gray')
            ix += 1
    # show the figure
    pyplot.show()
```

Tying these changes together, we can now create five separate plots for each of the five blocks in the VGG16 model for our bird photograph. The complete listing is provided below.

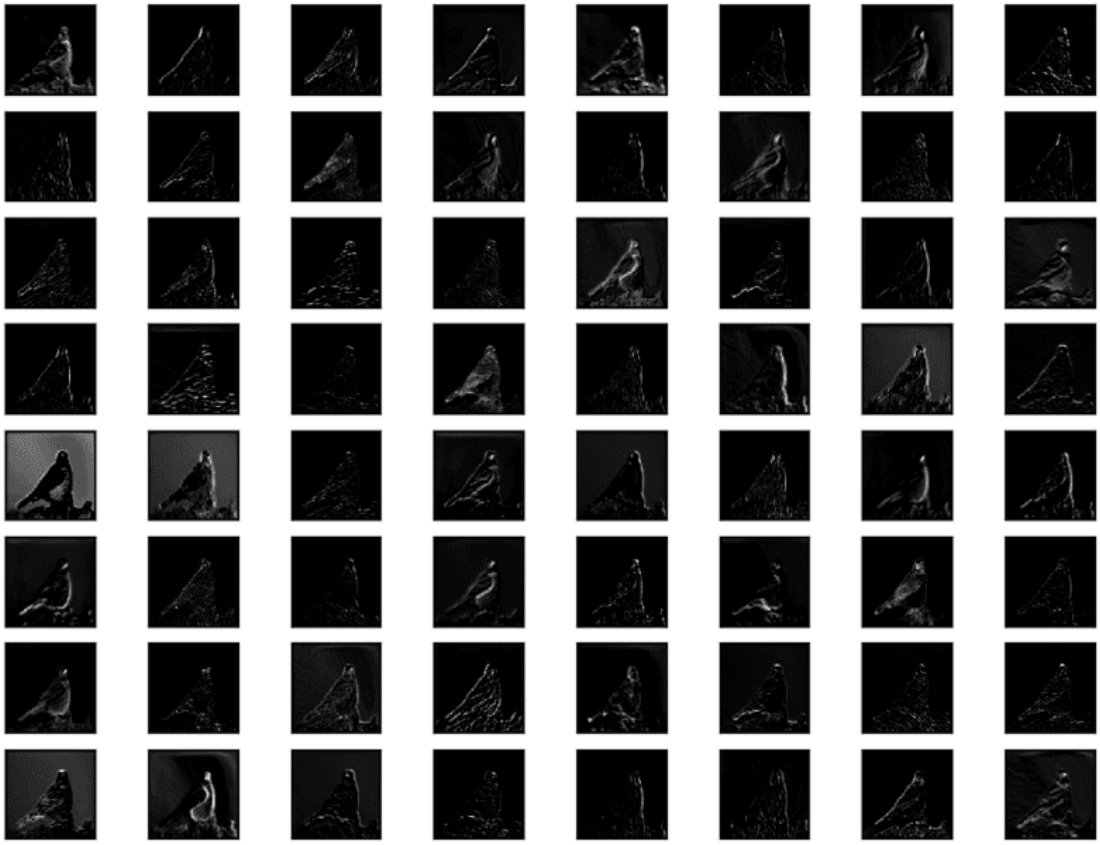
```
# visualize feature maps output from each block in the vgg model
from keras.applications.vgg16 import VGG16
from keras.applications.vgg16 import preprocess_input
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.models import Model
from matplotlib import pyplot
from numpy import expand_dims
# load the model
model = VGG16()
# redefine model to output right after the first hidden layer
ixs = [2, 5, 9, 13, 17]
outputs = [model.layers[i].output for i in ixs]
model = Model(inputs=model.inputs, outputs=outputs)
# load the image with the required shape
img = load_img('bird.jpg', target_size=(224, 224))
# convert the image to an array
img = img_to_array(img)
# expand dimensions so that it represents a single 'sample'
img = expand_dims(img, axis=0)
# prepare the image (e.g. scale pixel values for the vgg)
img = preprocess_input(img)
# get feature map for first hidden layer
feature_maps = model.predict(img)
# plot the output from each block
square = 8
for fmap in feature_maps:
    # plot all 64 maps in an 8x8 squares
    ix = 1
    for _ in range(square):
        for _ in range(square):
            # specify subplot and turn of axis
            ax = pyplot.subplot(square, square, ix)
            ax.set_xticks([])
            ax.set_yticks([])
            # plot filter channel in grayscale
            pyplot.imshow(fmap[0, :, :, ix-1], cmap='gray')
            ix += 1
    # show the figure
    pyplot.show()
```


Running the example results in five plots showing the feature maps from the five main blocks of the VGG16 model.

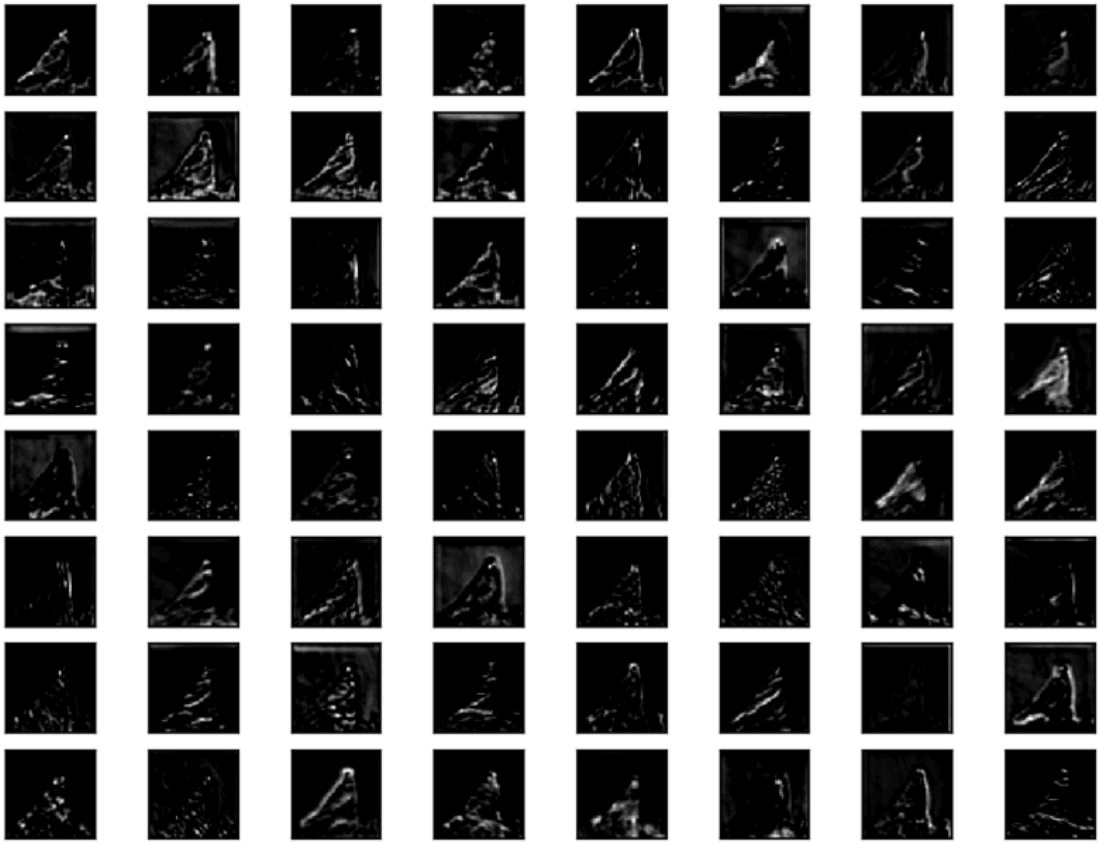
We can see that the feature maps closer to the input of the model capture a lot of fine detail in the image and that as we progress deeper into the model, the feature maps show less and less detail.

This pattern was to be expected, as the model abstracts the features from the image into more general concepts that can be used to make a classification. Although it is not clear from the final image that the model saw a bird, we generally lose the ability to interpret these deeper feature maps.

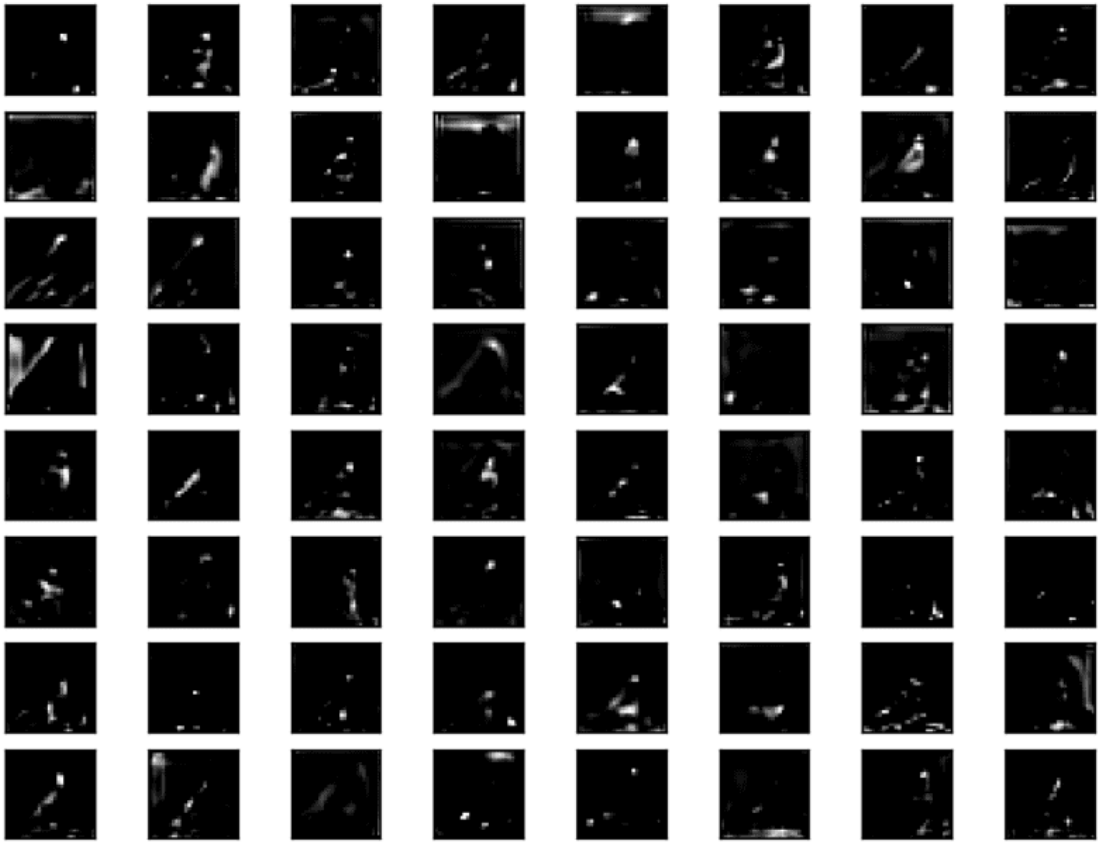
Visualization of the Feature Maps Extracted From Block 1 in the VGG16 Model



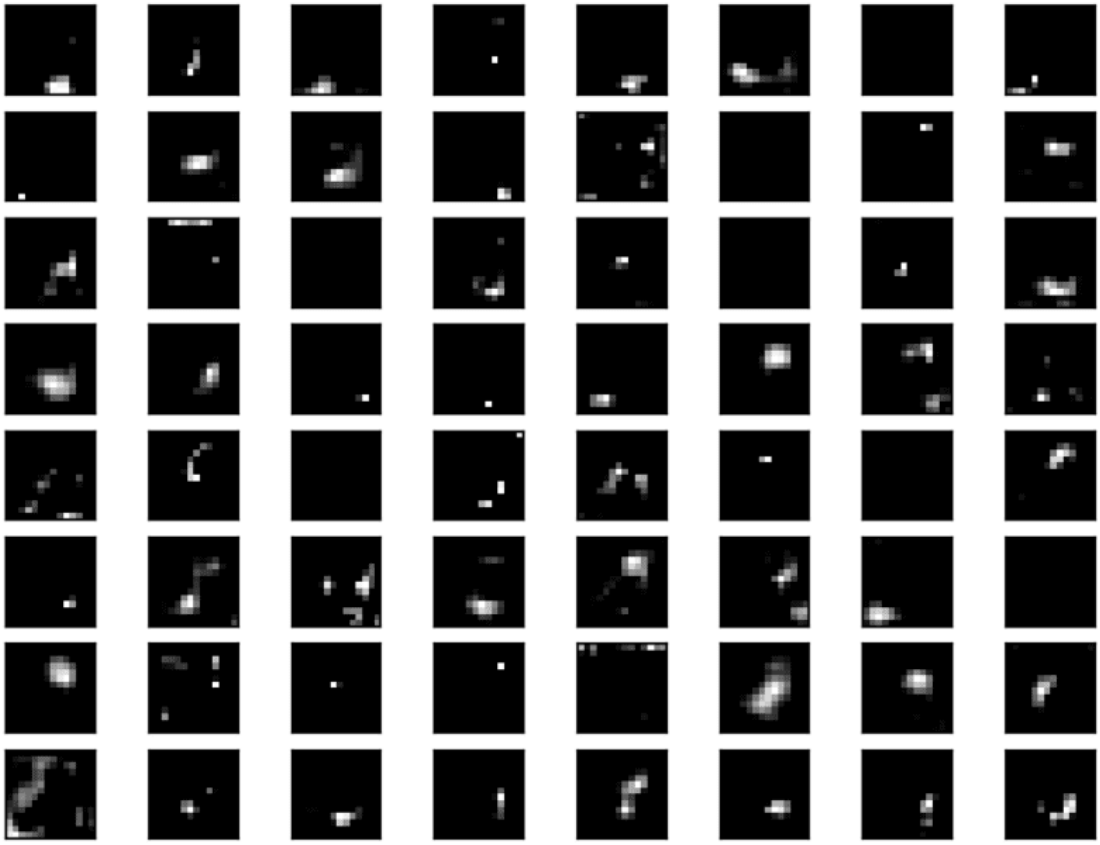
Visualization of the Feature Maps Extracted From Block 2 in the VGG16 Model



Visualization of the Feature Maps Extracted From Block 3 in the VGG16 Model



Visualization of the Feature Maps Extracted From Block 4 in the VGG16 Model



Visualization of the Feature Maps Extracted From Block 5 in the VGG16 Model

Further Reading

This section provides more resources on the topic if you are looking to go deeper.

Books

- Chapter 9: Convolutional Networks, Deep Learning, 2016.
- Chapter 5: Deep Learning for Computer Vision, Deep Learning with Python, 2017.

API

- Keras Applications API
- Visualization of the filters of VGG16, Keras Example.

Articles

- Lecture 12 | Visualizing and Understanding, CS231n: Convolutional Neural Networks for
- Visualizing what ConvNets learn, CS231n: Convolutional Neural Networks for Visual Rec
- How convolutional neural networks see the world, 2016.

Summary

In this tutorial, you discovered how to develop simple visualizations for filters and feature maps in a convolutional neural network.

Specifically, you learned:

- How to develop a visualization for specific filters in a convolutional neural network.
- How to develop a visualization for specific feature maps in a convolutional neural network.
- How to systematically visualize feature maps for each block in a deep convolutional neural network.

Do you have any questions?
Ask your questions in the comments below and I will do my best to answer.

More On This Topic