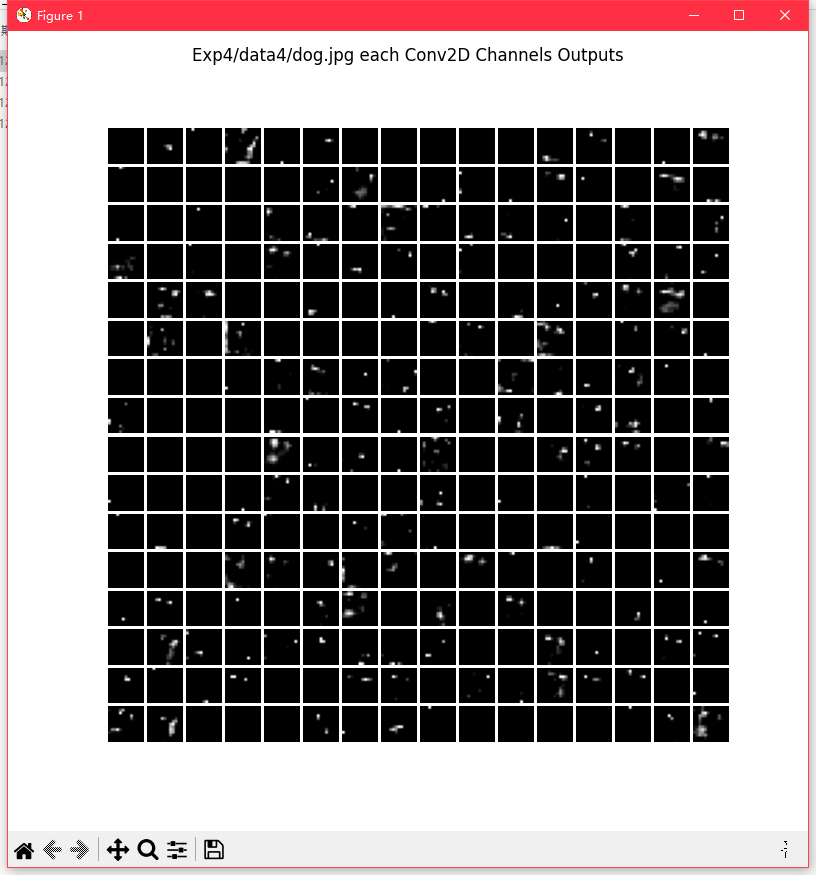
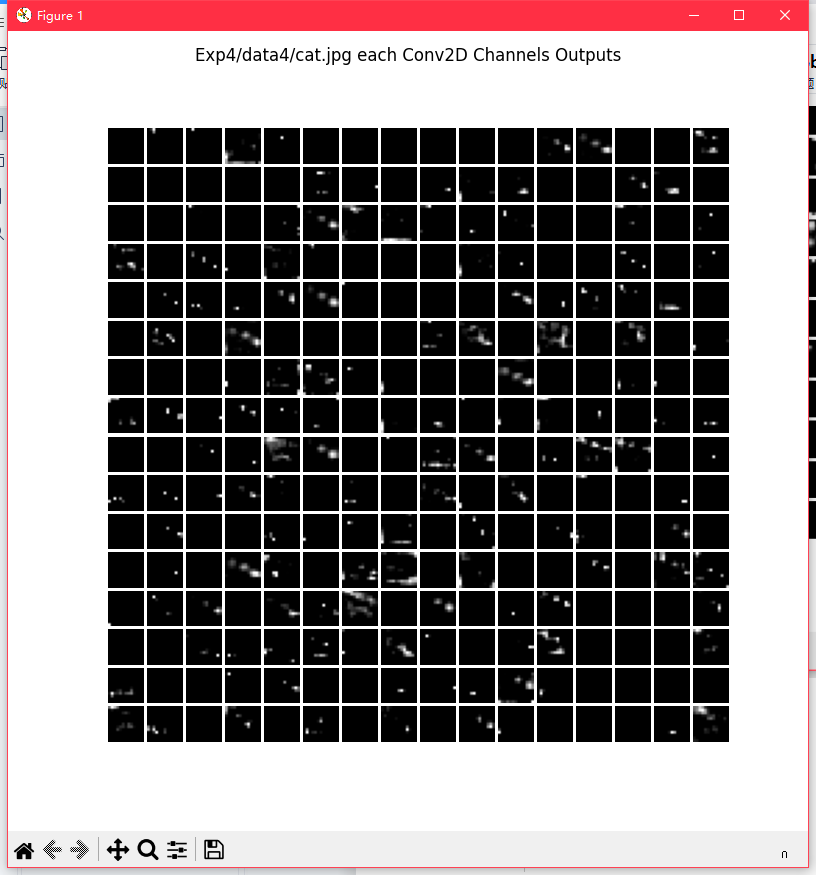
# 卷积层每一通道的可视化结果

下列分别是对三张输入图片在最后一层卷积层[10]层的每一个通道特征图的输出，每张图共用256个通道，采用16x16的方法展示

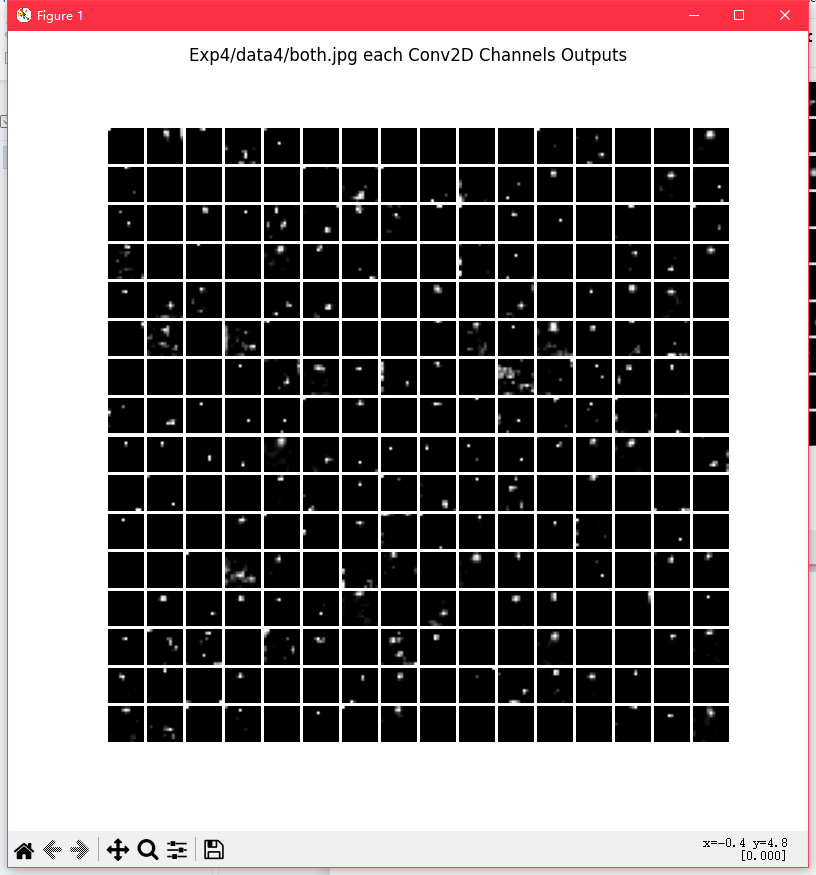
Dog.jpg



Cat.jpg



Both.jpg



## 实现过程

加载模型，首先是对使用的torch\_alex.pth预训练模型进行打印，得到模型的结构，以确定要对模型哪一层进行特征图输出、Grad-CAM和LayerCAM可视化分析

AlexNet(

(features): Sequential(

(0): Conv2d(3, 64, kernel\_size=(11, 11), stride=(4, 4), padding=(2, 2))

(1): ReLU(inplace=True)

(2): MaxPool2d(kernel\_size=3, stride=2, padding=0, dilation=1, ceil\_mode=False)

(3): Conv2d(64, 192, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(4): ReLU(inplace=True)

(5): MaxPool2d(kernel\_size=3, stride=2, padding=0, dilation=1, ceil\_mode=False)

(6): Conv2d(192, 384, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1))

(7): ReLU(inplace=True)

(8): Conv2d(384, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1))

(9): ReLU(inplace=True)

(10): Conv2d(256, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1))

(11): ReLU(inplace=True)

(12): MaxPool2d(kernel\_size=3, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(avgpool): AdaptiveAvgPool2d(output\_size=(6, 6))

(classifier): Sequential(

(0): Dropout(p=0.5, inplace=False)

(1): Linear(in\_features=9216, out\_features=4096, bias=True)

(2): ReLU(inplace=True)

(3): Dropout(p=0.5, inplace=False)

(4): Linear(in\_features=4096, out\_features=4096, bias=True)

(5): ReLU(inplace=True)

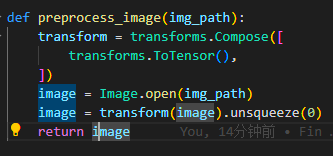
(6): Linear(in\_features=4096, out\_features=2, bias=True)

)

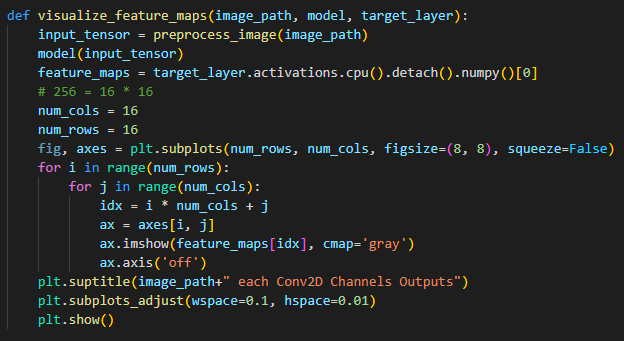
)

通过阅读以上模型结构，不难发现，最后一层卷积层为标红的层所示，即(10): Conv2d(256, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1))，第10层，所以对该层进行特征图输出、Grad-CAM和LayerCAM解释。

简单的预处理函数如下图，仅将图片转为四维向量

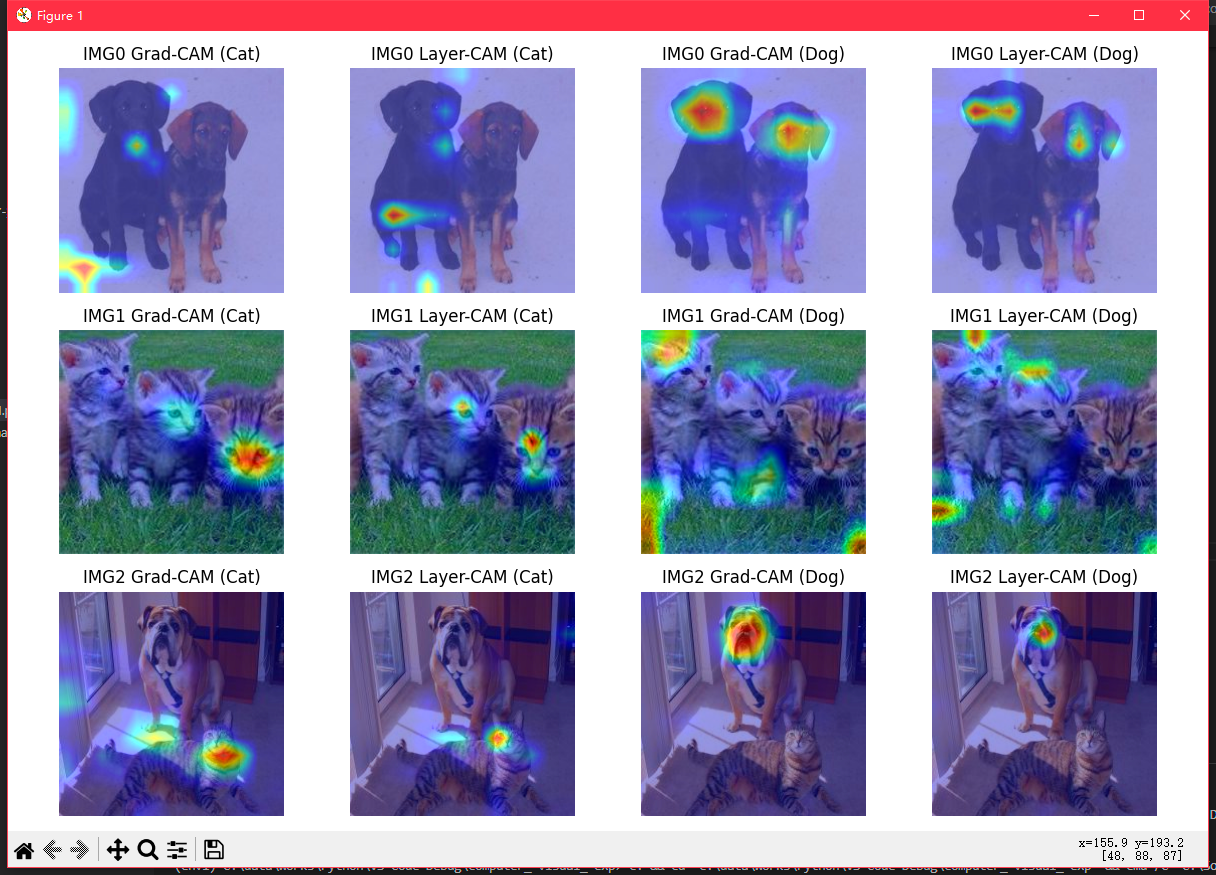


接下来将图片转换为四维向量输入后，直接读取经过目标层时每一个通道的激活度，将其转换为灰度图显示即可



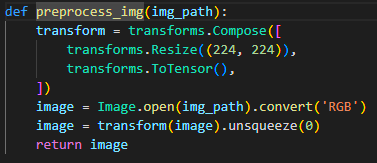
# GradCAM与LayerCAM可视化实现过程

得到的GradCAM与LayerCAM可解释方法结果如下图所示

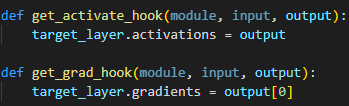


## 实现过程

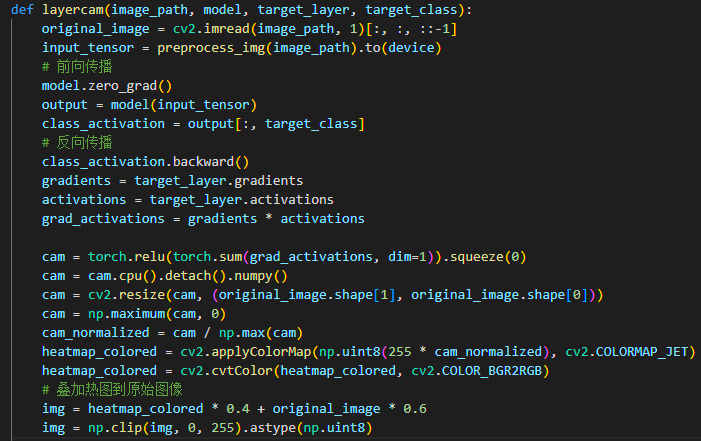
首先是对图像进行预处理，将图像大小调整为224x224像素，转换为4维的Tensor张量并返回，如所示



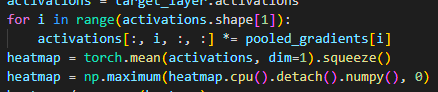
定义获取激活和梯度的钩子函数，如下图所示，以获取正向传播的激活和反向传播时的梯度



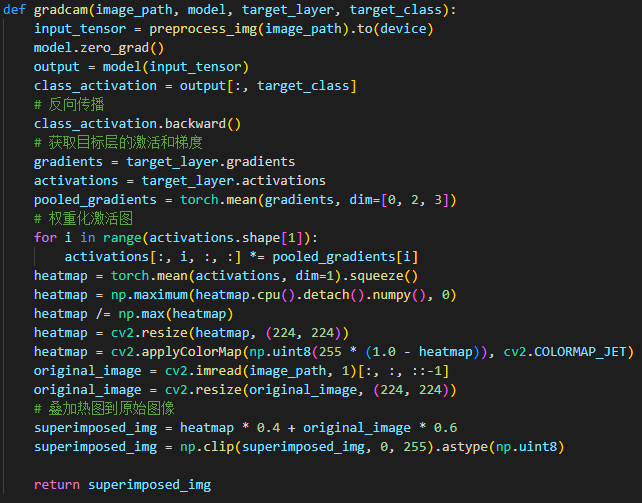
之后便可以计算LayerCAM，首先先对图像转换为4维Tensor，然后进行前向和反向传播，获取图像目标类别的激活度和反向传播中的梯度，然后计算cam，进行逐像素相乘求和，然后通过ReLU函数去除负值，并且归一化，然后将生成的彩色热力图叠加到原图像即可，代码如下图所示。

，

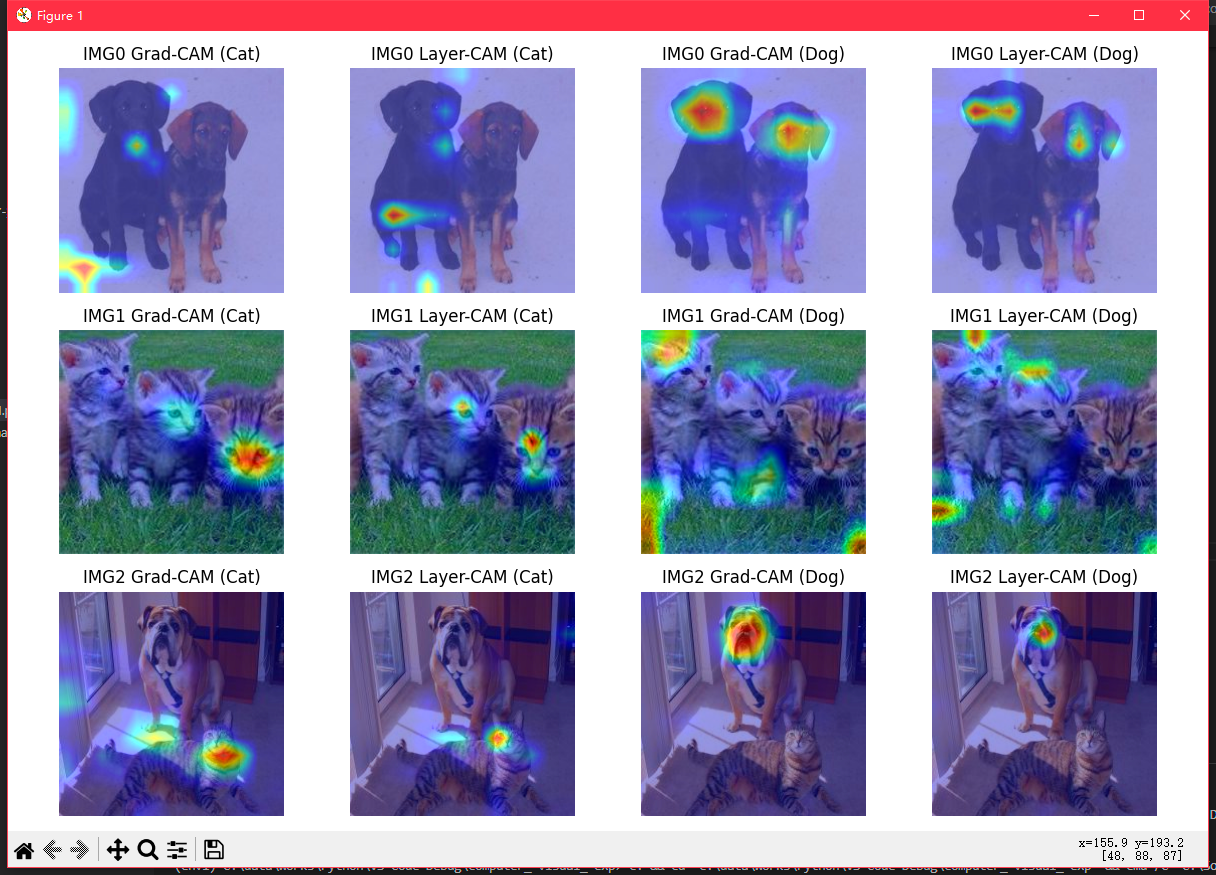
对GradCAM方法，与LayerCAM方法的区别主要在cam的计算方法上，Gradcam在创建钩子后，获取目标层维度的梯度，然后对于目标层的每个通道（channel），都将梯度乘以相应的激活，以突出对目标类别预测的贡献度，如下图所示



完整代码如下图所示



## 结果分析



对上图结果，采用Grad-CAM和LayerCAM对图像进行可解释化分析得到的结果略有不同。可以看出，在图中含有对应类别生物时，二者都能准确找出其位置。其中LayerCAM对图片中含有指定类别物体时的贡献划分得更为精准，更为聚焦。第三行猫和狗同时存在时，对Cat的预测结果GradCAM和LayerCAM都有一些偏差，推测是模型导致的。

原因：

Grad-CAM与LayerCAM的计算方法不相同，Grad-CAM是一种基于梯度的方法，通过计算损失相对于模型中某一层的特征图的梯度来确定激活区域，依赖于反向传播过程中的梯度信息。而LayerCAM直接在目标层的特征图上进行激活映射，而不涉及梯度计算，避免了对应层梯度信息的依赖，更为精准和聚焦。