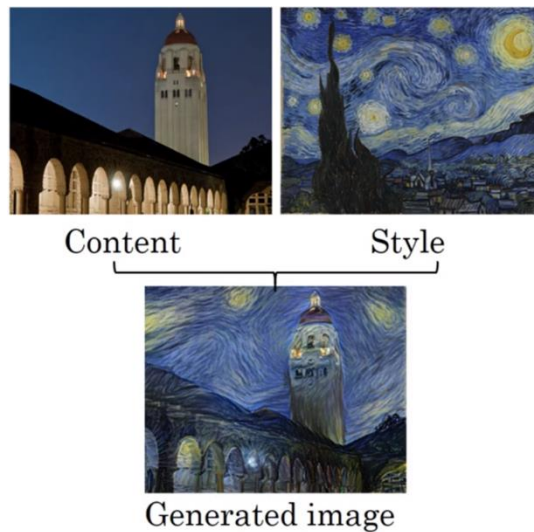


1、图像风格迁移 CSG

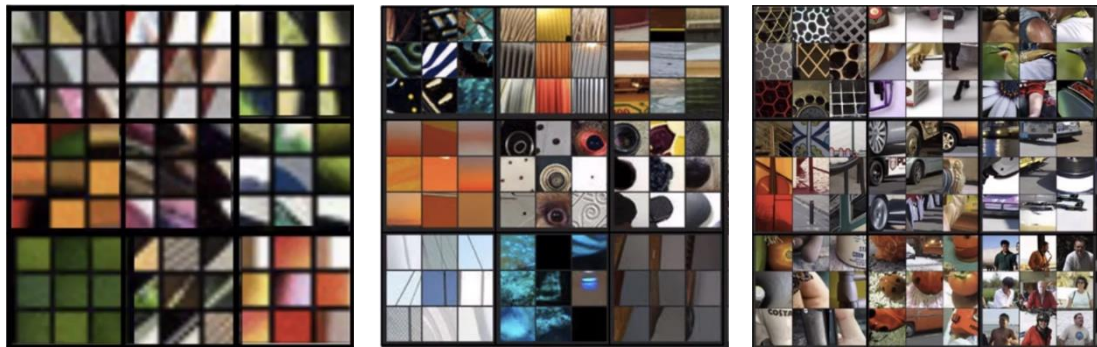
计算机视觉中其中一个很有趣的例子就是图像风格迁移，利用指定风格的图片 (S)



将自己提供的原图 (C) 转换为具有这样风格的图片 (G)

2、可视化神经网络浅层与深层

在学习风格迁移之前，需要搞明白一个原理，即神经网络浅层与深层究竟做了什么，或者是学习到了什么东西，哪些神经单元被激活了。不论是卷积层还是全连接层，每层都有一个激活函数，激活某些单元，然后传递到下一层。



来看一个简单的例子，上图从左至右是一个卷积神经网络由浅至深的学习可视化表现，在第一层中，我们可以看到，它似乎之学习到了简单的边缘特征，像是垂直边缘或者某些颜色块；第二层，它学到了更加复杂的特征，线条更加分明；第三层，它已经可以学习复杂成熟的图像了。可见神经网络的学习由浅入深逐渐提取更加高级的特征，至到最后的识别分类。

3、代价函数

要实现风格迁移，得定义一个合适的代价函数，这里直接给出定义， $J(G) = \alpha J_{\text{Content}}(C, G) + \beta J_{\text{Style}}(S, G)$ ，即含有内容代价和风格代价，然后赋予合适的权重，通过不断最小化代价 J 得到最终想要的图片。

内容代价函数： $J_{\text{Content}}(C, G) = \frac{1}{2} |a^{LC} - a^{LG}|^2$ ，在内容代价函数中，层数不能选

择太少，因为这只能识别到最初的几个简单特征，亦不能选择太深，否则新图与原图风格差别较大。

风格代价函数：

$$J_{style}^{[l]}(S, G) = \frac{1}{\left(2n_H^{[l]}n_W^{[l]}n_C^{[l]}\right)^2} \sum_k \sum_{k'} (G_{kk'}^{[l](S)} - G_{kk'}^{[l](G)})$$

之前讲到过，神经网络层中通常会有很多 channels（通道）每个通道专注于识别某个特征，譬如第一通道识别垂直边缘，第二通道识别水平边缘，而要想使得两张图片的风格类似，就要从被识别出的特征入手，因此，定义上方所示的风格代价函数，通道之间的相关系数越高，则表明如果有某一通道的特征，则有很大概率有另一通道的特征，如果不同层都要进行特征风格迁移的话，那么上式还要进行改造，在每一层都要计算代价函数，这样效果会更好。

最后，两者相加形成最终的 J 函数，在不断最小化这个代价函数的过程中，形成新的图片。由于这两个代价函数的计算需要两个图片作为输入，因此整个网络为两层同时计算，亦或先算完 C 和 S 的内容和风格，再计算 G 的内容和风格。