

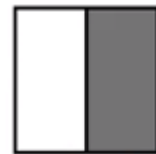
1、前言

在之前的学习中，已经可以使用简单的神经网络全连接层结构实现对图像的识别。可是，真正要运用在实际生活中，还有很长的一段路要走，首先就是图像的特征数量，比较直观的感受是，在 1080p 下的彩色图像，难以提供巨大的内存装载这么多特征数量，因此，首要解决的就是如何在最大限度的缩小特征数量同时又不丢失原本信息。接下来，就要谈到卷积神经网络了。不过，在那之前，先来学习一下计算机对图像处理的简单原理。

2、原图*过滤器=新图

这是一幅简单的 6*6pixel 的灰度图像，它可能看起来像右边这样，左右有一条明显

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0



的垂直边界。计算机要提取其中的垂直边界信息，不得不用到一个过滤器，我们称之为垂直过滤器，使用过滤器进行卷积，将得到这样一幅“新的图像”，可见计算机在利用

1	0	-1
1	0	-1
1	0	-1

为垂直过滤器，使用过滤器进行卷积，将得到这样一幅“新的图像”，可见计算机在利用

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0



垂直滤波器将垂直边界准确的定位了出来。关于其中的原理，其实很简单，颜色相同的属性经过一次垂直滤波，颜色特征将趋于0，否则不为0。类似的，还可以制作出水平边界过滤器。

3、边缘检测

在上一节例子中，如果将原图改为这样，那么输出特征如右图所示。这称之为边缘过渡。

0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10

0	-30	-30	0
0	-30	-30	0
0	-30	-30	0
0	-30	-30	0
0	-30	-30	0

在边缘检测中，科学家们经常思考用什么样的的过滤器才能很好的检测出边缘，历史上曾经有过

1	0	-1
2	0	-2
1	0	-1

Sobel 过滤器

3	0	-3
10	0	-10
3	0	-3

Scharr 滤波器

依据前面所学知识，我们可以自行设计过滤器，如参数滤波器，然后利用反向传播的思想计算出合适的参数，得到最终想要的过滤器。并且它可以实现任意角度边缘的检测，尽管利用反向传播计算要费劲许多。

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9