介绍的文章主要是 building high-level...这篇，因为最后实现的 sysytem 需要并行计算，不太懂。首先文章的目的，训练未标记的数据，一千万张 200*200 的 pixel 图片，从一千万的 youtube 视频里面每个截一张彩色图片。训练系统之后，用来检测两个已知的图片集中的人脸/猫脸/人身体。达到了 17.9%，比其他的方法好了 70%。

接下来是系统构造，共有三个大层，每个大层包括了三个小层，左边是基本结构，如上图所示，三个小层分别为过滤层（filtering），池化层（pooling），局部对比归一化层（local contrast normalization）。输入图片大小 200*200，经过预处理形成三个 channel（三原色？），感知区域（receptive fields）大小为 20*20，filter blocks 大小为 4*4，8 个 channel，其中 filter blocks 权重在三个输入 channel 之间共用（黑色箭头表示权重在三个 channel 中共享），但是不同的 4*4 区域不公用，这个地方就跟传统的卷积不同，传统的卷积只能够得到平移不变性，平移不变性只指某个特征无论放在哪里都能被提取出来，传统的权重共享就能做到，而不 recpetive fields 不公用权重来提取不同的特征，当图片特征被缩放，旋转，同样能被提取出来。卷积的功能使得特征增强（将有取值的方组合叠加起来得到了更大的值），噪声降低（取值较小的地方变得更小），因此选择 filtering，从而一张输入图在第一层参数个数大概为 4*4*8*5*5*181*181。

然后进行 l2 pooling，这个 pooling 的意义在于子采样（计算某个区域的最大值或者平均值），2 的意思是不重叠的长度为 2，l2 pooling 就是周围的元素平方和再取根号，本文中 pooling size 为 5*5，大小变为 89*89，这个部分权重值固定（1 或者 0），每一个 feature map，生成一个 pooling 层结果。pooling 是先加权，再取平方和的根号。

第一层第二层表达式就是传统的方法，参见（Tiled cnns）（插图），V 表示 pooling 层参数固定，它先与 filtering 后的图片每一个节点相成加权然后再取元素平方和取根号，W 表示 filter 层参数，在本文中，使用了 RTICA 来构建，因此表达方式为下图（此处涉及到自编码，Wx 编码，W 转置为解码）,自编码它尝试逼近一个恒等函数，从而使得输出 y 接近于输入 x，当隐藏节点少于可视节点，它会迫使系统来从压缩的隐藏节点重构出输入，假如输入之间有一定关系，那么这个算法就能够发现这些关系，因此可以求出重构损失，然后之前的 RICA 论文中是证明了在输入数据白化后重构损失是等于正交损失，第一二层的结构是 TICA 的结构，卷积只是作为求第一层的一个思想，因此用 RICA 来进行优化。

$$p_i(x^{(t)}; W, V) = \sqrt{\sum_{k=1}^{m} V_{ik} (\sum_{j=1}^{n} W_{kj} x_j^{(t)})^2}$$

$$\underset{W,\alpha}{\text{minimize}} \sum_i ||W^\top(\alpha W x^{(i)}) - x^{(i)}||_2^2 + \qquad (1)$$

$$\lambda \sum_j \sqrt{V_j (\alpha W x^{(i)})^2}$$

$$\text{subject to } ||W^{(k)}||_2 = 1, \forall k.$$

第三层 LCD 层，它会迫使在特征 map 中的相邻特征进行局部竞争，还会迫使在不同特征 maps 的同一空间位置的特征进行竞争，（就是谁的值大就会留下谁的值，谁的特征更显著留下谁的值。）局部做减法/除法归一化，取该地方的值减去周围各点加权后的值（加权是为了区分位置不同影响不同，高斯权重窗口，和为 1），然后计算 8 个 feature maps 同一位置的这个值，除以最大的一个。此处的权重只有一个 5*5（高斯权重窗口）。大小仍为 89*89。首先移除周围节点的平均值，表达式如下：

$$g_{i,j,k} = h_{i,j,k} - \sum_{iuv} G_{uv} h_{i,j+u,i+v}$$

除法归一化：

$$y_{i,j,k} = g_{i,j,k} / \max\{c, (\sum_{iuv} G_{uv} g^2_{i,j+u,i+v})^{0.5}\}$$

j，k 表示某一个节点（论文这个地方应该是写错了，参考论文里面后面的 i 是 k），uv 取值就得到了周围 5*5 的节点，G 是高斯窗口，h 是上一层的输出，i 表示是第 i 个 channel。

第二个大层的个数，70*70，34*34

第三大层，15*15，6*6

优化过程中，V，G 是保持不变的，更新的是 W。也就是上面的公式（2），求得这个表达式的最小化，第一项的求导结果如下：

$$\begin{aligned}\nabla_W F &= \nabla_W F + (\nabla_{W^T} F)^T \\ &= (W^T)(2(W^T W x - x))x^T + 2(Wx)(W^T W x - x)^T\end{aligned}$$

其中 F 没有将所有的 xi 叠加起来。第二项求导结果手写：