模型假设

1. 假设每张图片所有格式全部都是四通道格式，即（R，G，B，α）
2. 假设扫描得到图像偏角度为0⁰
3. 假设扫描的得到的图像不会受到扫描时环境因素的干扰，能够反应图像的原有信息
4. 假设所有图片的标签均是正确的

符号说明

|  |  |  |
| --- | --- | --- |
| 编号 | 符号 | 说明 |
| 1 | x | 标签为true样本值 |
| 2 | y | 改动后的样本值 |
| 3 | W | 权重矩阵 |
| 4 | b | 偏置矩阵 |
| 5 | i | 向量下标 |
| 6 | ai | 矩阵向量 |
| 7 | Wij | 权重矩阵的向量 |
| 8 | Z | 经过卷积或池化运算得到的矩阵 |

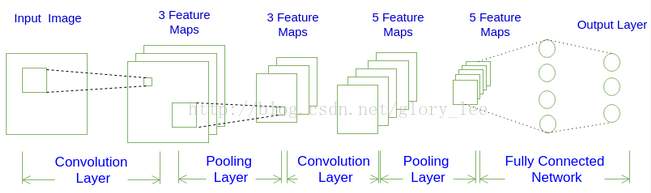
**模型建立与求解**

**基于cnn的图像二分类算法建立**

根据题中所给的附件，True集包含5888张真实图片，Fake集包含与之对应的被修改过的图片。将这些图片作为样本库，有这些样本库得到特征库，以此解决位数过高的计算量和数据冗杂问题。

**Cnn卷积神经网络模型**

卷积神经网络是一个多层的神经网络，每层由多个二维平面组成，而每个平面由多个独立神经元组成。下图是一个卷积神经网络的示意图



一个卷积神经网络由若干****卷积层、Pooling层、全连接层****组成。你可以构建各种不同的卷积神经网络，它的常用架构模式为：

INPUT -> [[CONV]\*N -> POOL?]\*M -> [FC]\*K

也就是N个卷积层叠加，然后(可选)叠加一个Pooling层，重复这个结构M次，最后叠加K个全连接层。

而我们所建立的卷积神经网络：

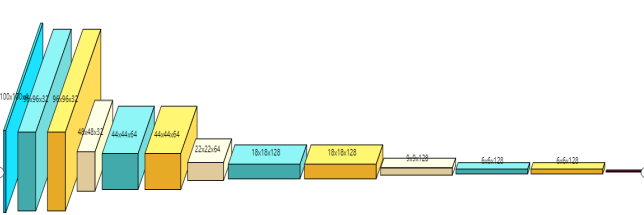
INPUT -> CONV -> POOL -> CONV -> POOL -> CONV -> POOL -> CONV -> POOL->FC-> FC-> FC

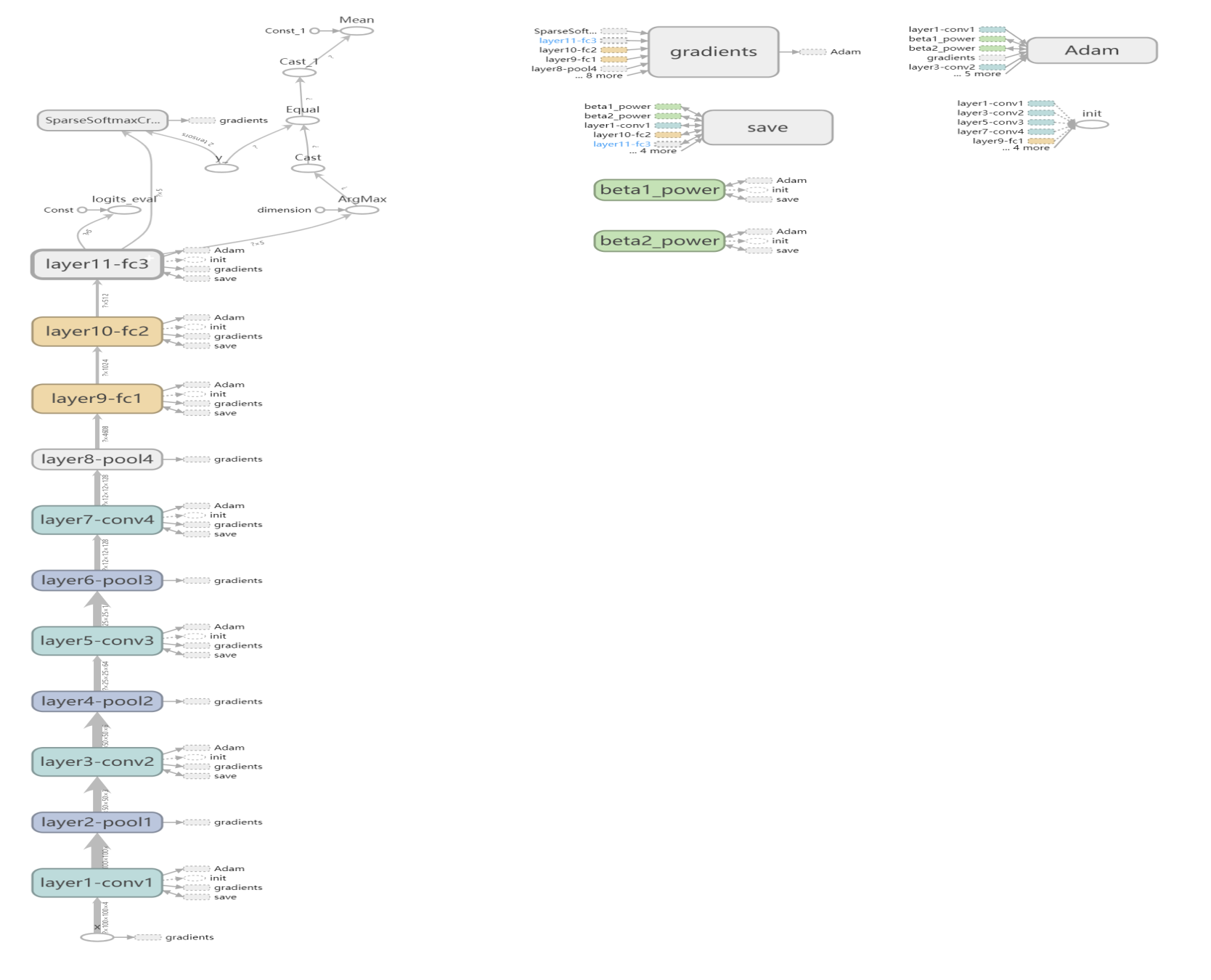
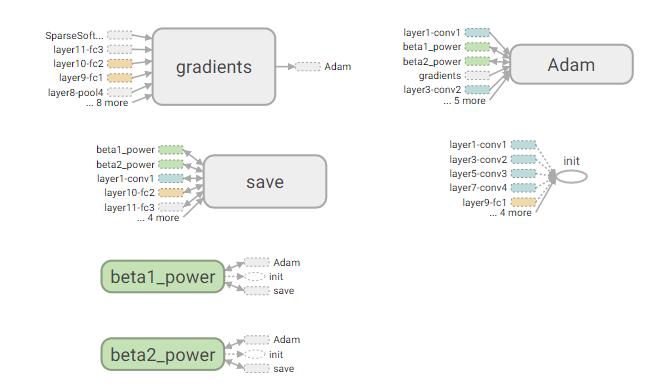
按照上述模式可以表示为：

INPUT -> [[CONV]\*1 -> POOL]\*4 -> [FC]\*3

也就是：N=1, M=4, K=3。

输入图像通过和32个可训练的滤波器和可加偏置进行卷积，卷积后在 C1 层产生32个特征映射图，然后特征映射图 中每组的4个像素再进行求和，加权值，加偏置，通过一个 relu（线性修正） 函数得到4个p1 层的特征映射图。这些映射图再经过最大池化（max\_pool)得到 C2 层。这个层级结构再和 p1 一样产生 p2得到C3，以此类推得到p4、C4，并将最终得到的128个特征映射图导入3个全连接层。 通过 softmax进行计算并产生输出。





卷积神经网络原理

参数量大小的计算，分为weights和biases：

1. 首先来计算weights的参数量：F×F×Cinput×K
2. 接着计算biases的参数量：K
3. 所以总参数量为：F×F×Cinput×K+K

输入层

对输入的特征图进行压缩，一方面使特征图变小，简化网络计算复杂度；一方面进行特征压缩，提取主要特征，

卷积层输入特征图的尺寸：Hinput×Winput×Cinput

1. Hinput表示输入特征图的高
2. Winput表示输入特征图的宽
3. Cinput表示输入特征图的通道数（如果是第一个卷积层则是输入图像的通道数，如果是中间的卷积层则是上一层的输出通道数）

假设输入 x = [a1, a2, a3，a4]，令第一层的输入 z (1) 和激活输出 a (1) 相等，即 z (1) = a (1) = x。

**隐藏层**

卷积层的参数有如下几个：

1. 输出通道数为K
2. 正方形卷积核的边长为F
3. 步幅(stride)为S
4. 补零的行数和列数(padding)为P

第二层的输入为：

Z1(2) = a1(1)w11(2) + a2(1)w12(2 )+ a3(1)w13(2 ) + a4(1)w14(2 ) + b1(2)

Z2(2) = a1(1)w21(2) + a2(1)w22(2 )+ a3(1)w23(2 ) + a4(1)w24(2 ) + b2(2)

Z3(2) = a1(1)w31(2) + a2(1)w32(2 )+ a3(1)w33(2 ) + a4(1)w34(2 ) + b3(2)

Z4(2) = a1(1)w41(2) + a2(1)w42(2 )+ a3(1)w43(2 ) + a4(1)w44(2 ) + b4(2)

即

Z(2) = a(1)w(2) + b(2)

Relu激活：

a(2) = f(Z(2))

其中f为激活函数，a(2)∈R3×1

**输出层**

输出特征图(output feature map)的尺寸为Houtput×Woutput×Coutput，其中每一个变量的计算方式如下

1. Houtput=(Hintput−F+2P)/S+1
2. Woutput=(Winput−F+2P)/S+1
3. Coutput=K

输入：

z(3) = w(3)a(2) + b(3) 其中，z(3) ∈ R1×1 , w(3) ∈ R1×3 , a(2) ∈ R3×1 , b(3) ∈ R1×1

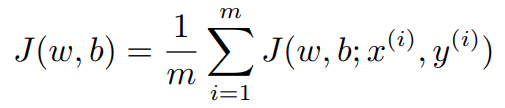
激活输出：

a (3) = f(z(3)) 其中，f 为激活函数，a(3) ∈ R1×1。特别地，记 a(3) 为 hw,b(x)

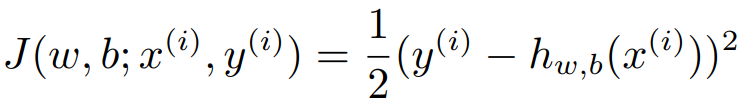
前向传导

前向传导可以表示为：z (l+1)= w (l+1)a(l) + b(l+1) a(l+1) = f(z(l+1)) 其中，l = 1, 2, ..., L − 1，L 为神经网络的层数。

假设神经网络的代价函数为

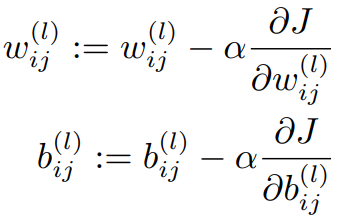


其中：

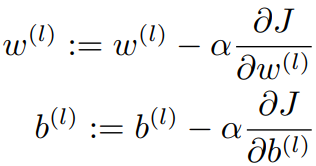


即，网络的整体代价为所有训练样例的平均代价。

根据梯度下降法的思想，对于每个IMG_256,我们只要往负梯度方向更新就可以了，即：



向量化表达即为：



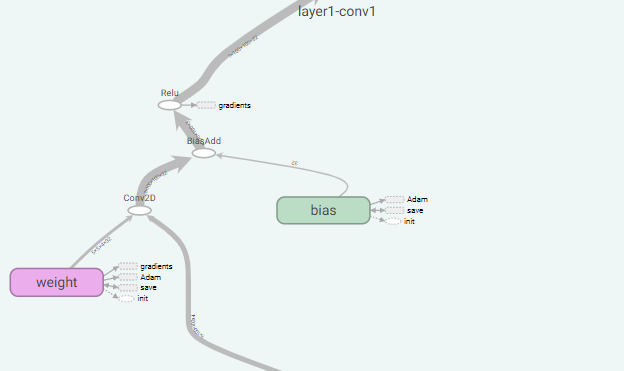
因此，只要能求出 w, b 的偏导数就能迭代更新，从而完成整个算法。看似简单，但却困难。因 为 J(w, b) 是很难写出显式表达式的，从而很难对每个 wij , bij 都求出偏导，主要原因是网络是分层 的进而 w, b 也是分层，这才导致了偏导的难求，从而才有了反向传播。

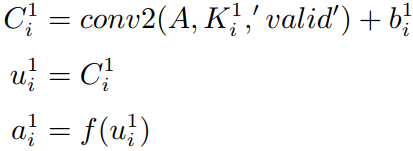
**卷积（即提取图片特征值）**

假设 A 的大小为 ha × wa，B 的大小为 hb × wb（其中 ha ≥ hb, wa ≥ wb），则 C 的大小 为 (ha − hb + 1) × (wa − wb + 1)，矩阵 B 称为卷积核或滤波器 (filter)，矩阵 C 称为特征图 (feature map)。B 卷积 A 的结果就是让 B 在矩阵 A 上滑动，换言之，就是 B 与 A 的所有 2×2 连续子 矩阵做“对应元素积之和”运算。上述运算称为窄卷积，若矩阵 A 预先上下各添加 hb − 1 行零向量，左右各添加 wb − 1 列 零向量，再与 B 卷积，则称为宽卷积。窄卷积和宽卷积分别用 conv2(A, B, ’valid’)和 conv2(A, B, ’full’)表示。

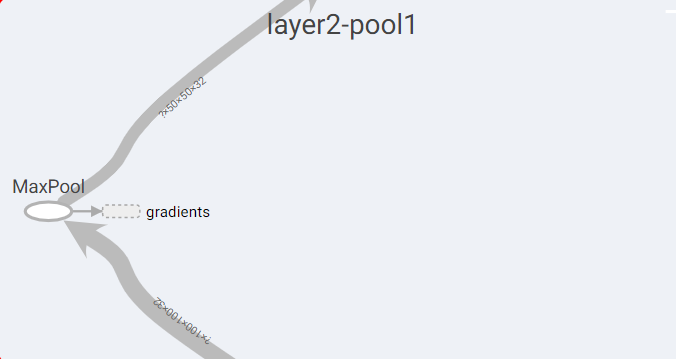
**模型整体描述**

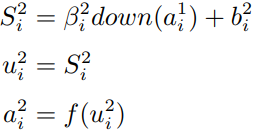
C1 层：卷积神经网络的输入是 100×100 的矩阵 A，经过 32 个 5×5 步长为1的卷积核 K1 i (i = 1, 2, ..., 32) 的卷积生成 F1 个 96 × 96 大小的 feature maps：



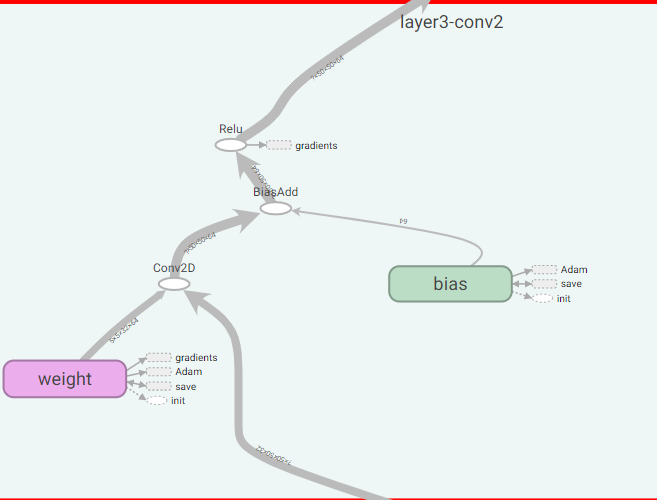


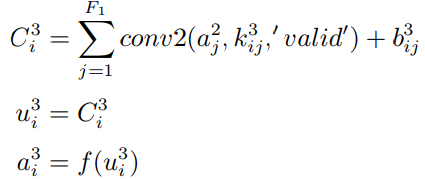
p1 层：接着池化，池化窗口为 2 × 2，步长为2 × 2，一个 96 × 96 的 feature map 池化成一个 48 × 48 大小的 pool map，共生成 32 个 pool maps：



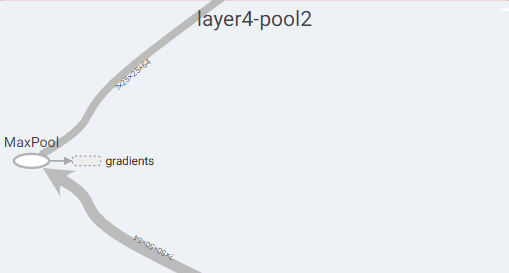


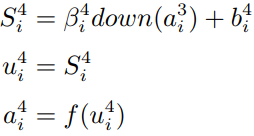
C2 层：接着再次卷积，C2 层中每个 48 × 48 的 feature map都由 S2 中的所有 32 个 pool maps 经过 32 个 5 × 5 的卷积核 K3 ij (j = 1, 2, ..., 32)，共生成 64 个 44 × 44的feature maps：





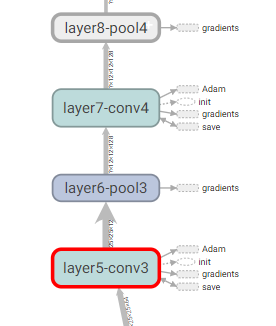
P2 层：再次池化，池化窗口为 2 × 2，步长为2 × 2，一个 48 × 48 的 feature map 池化成一个 22 × 22 大小的 pool map，共生成 64 个 pool maps：





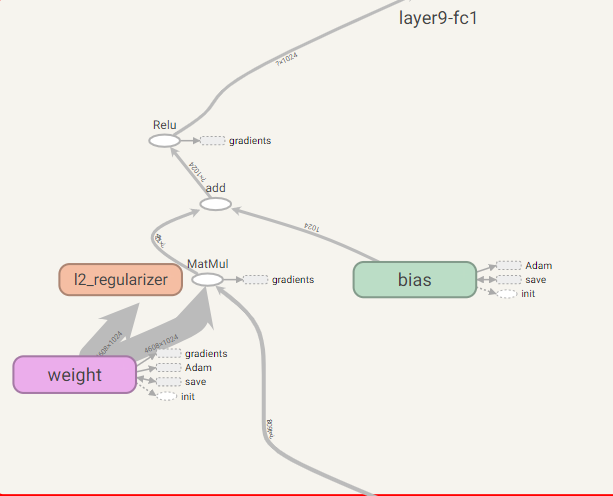
以此类推，

最后一个卷积层，共生成128 个6 × 6的feature map。



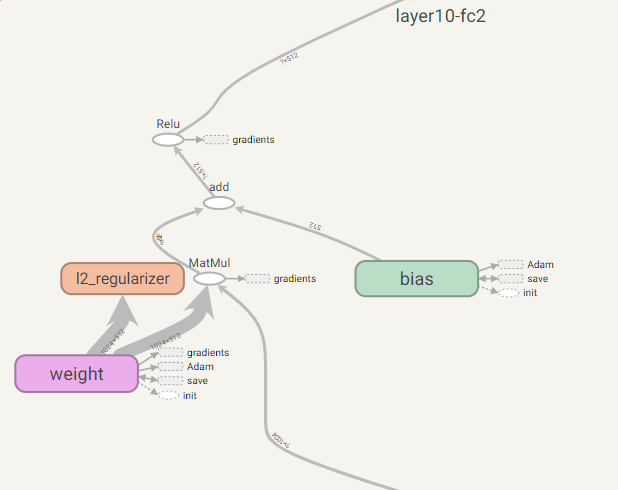
全连接层：

Fc1：再连接一个1×1×1024全连接层。则其对应的转换后的卷积层的参数为：

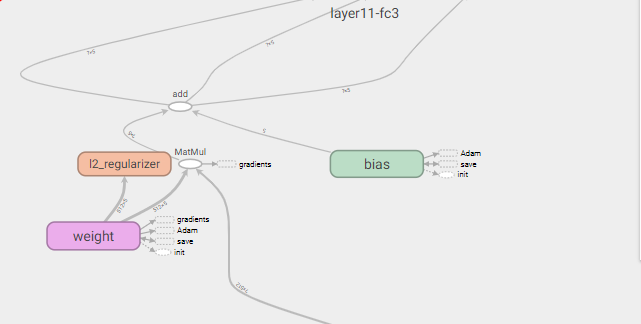


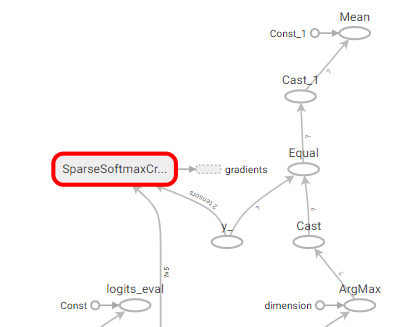
1.共有1024组滤波器  
2.每组滤波器含有128个卷积核  
3.每个卷积核的大小为6×6  
4.输出为1X1X1024

Fc2：再连接一个1×1×1024全连接层。则其对应的转换后的卷积层的参数为：



1.共有1024组滤波器  
2.每组滤波器含有512个卷积核  
3.每个卷积核的大小为1×1  
4.输出为1×1×512  
 Fc3：再连接一个1×1×512全连接层。则其对应的转换后的卷积层的参数为：



1.共有512组滤波器  
2.每组滤波器含有5个卷积核  
3.每个卷积核的大小为1×1  
4.输出为1×1×5  
 

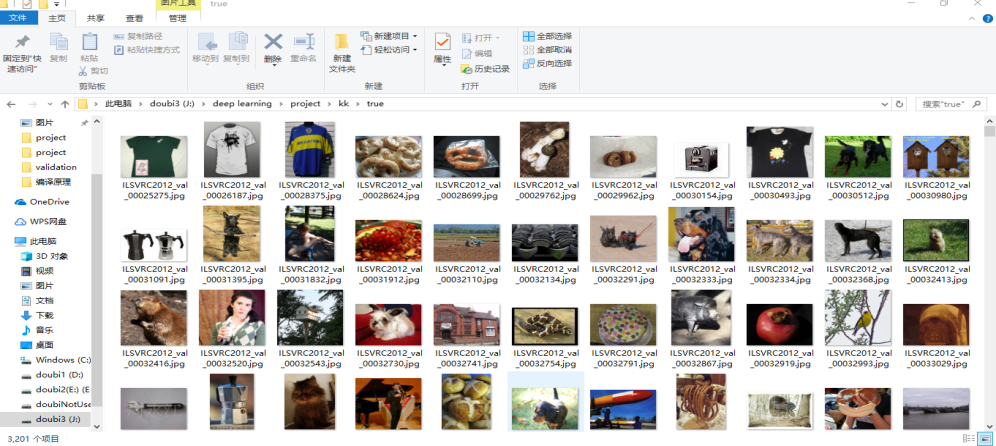
**模型检验**

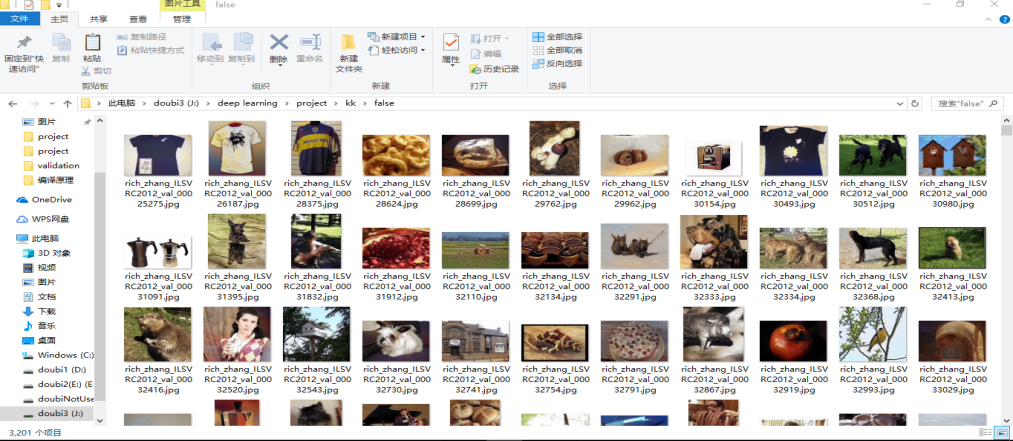
首先对图像集合进行预处理，即特征值提取和图像压缩。

将附件中的数据集通过matlab编写的k-l将所有的附带数据集的图片，并将图片统一转换为.jpg格式（。。。。。。。。。。插入k-l部分）

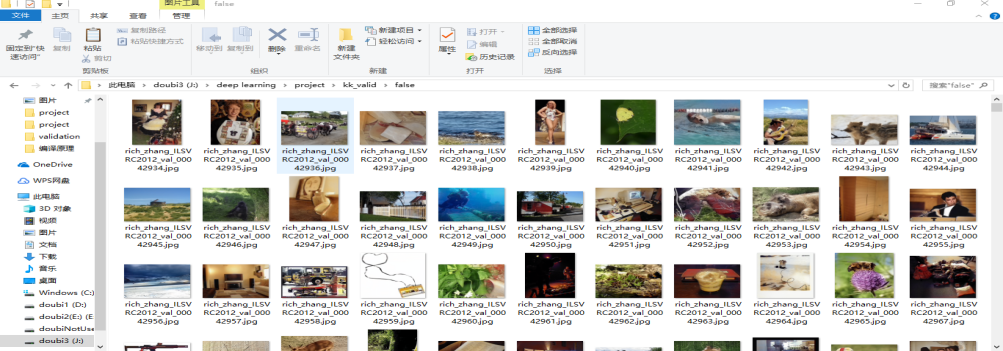
选取2个数据集的前3021张图片作为训练集，

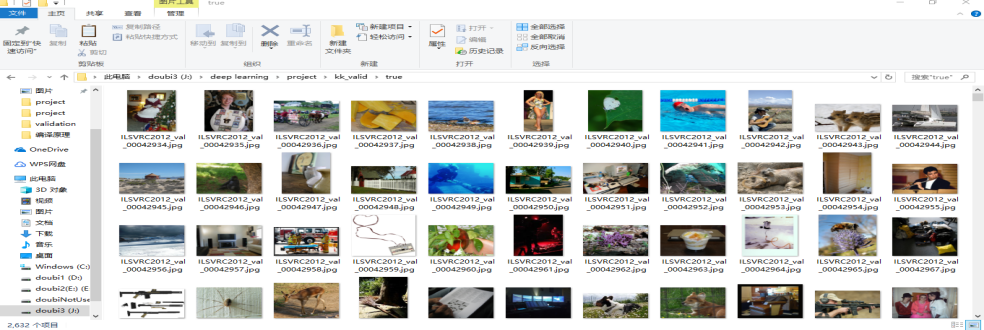
分别命名为kk\true 和 kk\false





选取2个训练集余下的图片作为验证集，用于验证模型训练时的准确性，同时依靠验证集的准确性防止训练集由于训练样本过少而产生过拟合现象



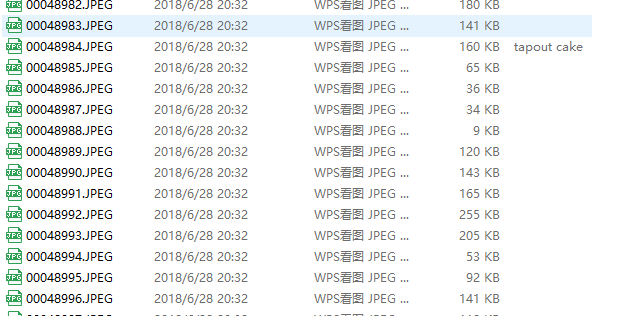


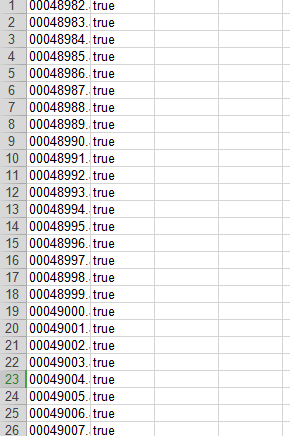
两个验证集分别命名为kk\_valid\true 和 kk\_valid\false。

将训练集和验证集导入，训练神经网络，得到训练结果



将测试集导入训练好的模型：





得到被修改的图片如下：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 00049184.JPEG |  | 00049210.JPEG |  | 00049236.JPEG |  | 00049262.JPEG |  |
| 00049185.JPEG |  | 00049211.JPEG |  | 00049237.JPEG |  | 00049263.JPEG |  |
| 00049186.JPEG |  | 00049212.JPEG |  | 00049238.JPEG |  | 00049264.JPEG |  |
| 00049187.JPEG |  | 00049213.JPEG |  | 00049239.JPEG |  | 00049265.JPEG |  |
| 00049188.JPEG |  | 00049214.JPEG |  | 00049240.JPEG |  | 00049266.JPEG |  |
| 00049189.JPEG |  | 00049215.JPEG |  | 00049241.JPEG |  | 00049267.JPEG |  |
| 00049190.JPEG |  | 00049216.JPEG |  | 00049242.JPEG |  | 00049268.JPEG |  |
| 00049191.JPEG |  | 00049217.JPEG |  | 00049243.JPEG |  | 00049269.JPEG |  |
| 00049192.JPEG |  | 00049218.JPEG |  | 00049244.JPEG |  | 00049270.JPEG |  |
| 00049193.JPEG |  | 00049219.JPEG |  | 00049245.JPEG |  | 00049271.JPEG |  |
| 00049194.JPEG |  | 00049220.JPEG |  | 00049246.JPEG |  | 00049272.JPEG |  |
| 00049195.JPEG |  | 00049221.JPEG |  | 00049247.JPEG |  | 00049273.JPEG |  |
| 00049196.JPEG |  | 00049222.JPEG |  | 00049248.JPEG |  | 00049274.JPEG |  |
| 00049197.JPEG |  | 00049223.JPEG |  | 00049249.JPEG |  | 00049275.JPEG |  |
| 00049198.JPEG |  | 00049224.JPEG |  | 00049250.JPEG |  | 00049276.JPEG |  |
| 00049199.JPEG |  | 00049225.JPEG |  | 00049251.JPEG |  | 00049277.JPEG |  |
| 00049200.JPEG |  | 00049226.JPEG |  | 00049252.JPEG |  | 00049278.JPEG |  |
| 00049201.JPEG |  | 00049227.JPEG |  | 00049253.JPEG |  | 00049279.JPEG |  |
| 00049202.JPEG |  | 00049228.JPEG |  | 00049254.JPEG |  | 00049280.JPEG |  |
| 00049203.JPEG |  | 00049229.JPEG |  | 00049255.JPEG |  | 00049281.JPEG |  |
| 00049204.JPEG |  | 00049230.JPEG |  | 00049256.JPEG |  | 00049282.JPEG |  |
| 00049205.JPEG |  | 00049231.JPEG |  | 00049257.JPEG |  | 00049283.JPEG |  |
| 00049206.JPEG |  | 00049232.JPEG |  | 00049258.JPEG |  | 00049284.JPEG |  |
| 00049207.JPEG |  | 00049233.JPEG |  | 00049259.JPEG |  | 00049285.JPEG |  |
| 00049208.JPEG |  | 00049234.JPEG |  | 00049260.JPEG |  | 00049286.JPEG |  |
| 00049209.JPEG |  | 00049235.JPEG |  | 00049261.JPEG |  | 00049287.JPEG |  |
| 00049288.JPEG |  | 00049314.JPEG |  | 00049340.JPEG |  | 00049367.JPEG |  |
| 00049289.JPEG |  | 00049315.JPEG |  | 00049341.JPEG |  | 00049368.JPEG |  |
| 00049290.JPEG |  | 00049316.JPEG |  | 00049342.JPEG |  | 00049369.JPEG |  |
| 00049291.JPEG |  | 00049317.JPEG |  | 00049343.JPEG |  | 00049370.JPEG |  |
| 00049292.JPEG |  | 00049318.JPEG |  | 00049344.JPEG |  | 00049371.JPEG |  |
| 00049293.JPEG |  | 00049319.JPEG |  | 00049345.JPEG |  | 00049372.JPEG |  |
| 00049294.JPEG |  | 00049320.JPEG |  | 00049346.JPEG |  | 00049373.JPEG |  |
| 00049295.JPEG |  | 00049321.JPEG |  | 00049347.JPEG |  | 00049374.JPEG |  |
| 00049296.JPEG |  | 00049322.JPEG |  | 00049349.JPEG |  | 00049375.JPEG |  |
| 00049297.JPEG |  | 00049323.JPEG |  | 00049350.JPEG |  | 00049376.JPEG |  |
| 00049298.JPEG |  | 00049324.JPEG |  | 00049351.JPEG |  | 00049377.JPEG |  |
| 00049299.JPEG |  | 00049325.JPEG |  | 00049352.JPEG |  | 00049378.JPEG |  |
| 00049300.JPEG |  | 00049326.JPEG |  | 00049353.JPEG |  | 00049379.JPEG |  |
| 00049301.JPEG |  | 00049327.JPEG |  | 00049354.JPEG |  | 00049380.JPEG |  |
| 00049302.JPEG |  | 00049328.JPEG |  | 00049355.JPEG |  | 00049382.JPEG |  |
| 00049303.JPEG |  | 00049329.JPEG |  | 00049356.JPEG |  | 00049383.JPEG |  |
| 00049304.JPEG |  | 00049330.JPEG |  | 00049357.JPEG |  | 00049384.JPEG |  |
| 00049305.JPEG |  | 00049331.JPEG |  | 00049358.JPEG |  | 00049385.JPEG |  |
| 00049306.JPEG |  | 00049332.JPEG |  | 00049359.JPEG |  | 00049592.JPEG |  |
| 00049307.JPEG |  | 00049333.JPEG |  | 00049360.JPEG |  | 00049593.JPEG |  |
| 00049308.JPEG |  | 00049334.JPEG |  | 00049361.JPEG |  | 00049594.JPEG |  |
| 00049309.JPEG |  | 00049335.JPEG |  | 00049362.JPEG |  | 00049595.JPEG |  |
| 00049310.JPEG |  | 00049336.JPEG |  | 00049363.JPEG |  | 00049596.JPEG |  |
| 00049311.JPEG |  | 00049337.JPEG |  | 00049364.JPEG |  | 00049597.JPEG |  |
| 00049312.JPEG |  | 00049338.JPEG |  | 00049365.JPEG |  | 00049598.JPEG |  |
| 00049313.JPEG |  | 00049339.JPEG |  | 00049366.JPEG |  | 00049599.JPEG |  |
| 00049600.JPEG |  | 00049627.JPEG |  | 00049654.JPEG |  | 00049680.JPEG |  |
| 00049601.JPEG |  | 00049628.JPEG |  | 00049655.JPEG |  | 00049681.JPEG |  |
| 00049602.JPEG |  | 00049629.JPEG |  | 00049656.JPEG |  | 00049682.JPEG |  |
| 00049603.JPEG |  | 00049630.JPEG |  | 00049657.JPEG |  | 00049683.JPEG |  |
| 00049604.JPEG |  | 00049631.JPEG |  | 00049658.JPEG |  | 00049684.JPEG |  |
| 00049605.JPEG |  | 00049632.JPEG |  | 00049659.JPEG |  | 00049685.JPEG |  |
| 00049606.JPEG |  | 00049633.JPEG |  | 00049660.JPEG |  | 00049686.JPEG |  |
| 00049607.JPEG |  | 00049634.JPEG |  | 00049661.JPEG |  | 00049687.JPEG |  |
| 00049608.JPEG |  | 00049635.JPEG |  | 00049662.JPEG |  | 00049688.JPEG |  |
| 00049609.JPEG |  | 00049637.JPEG |  | 00049663.JPEG |  | 00049689.JPEG |  |
| 00049610.JPEG |  | 00049638.JPEG |  | 00049664.JPEG |  | 00049690.JPEG |  |
| 00049611.JPEG |  | 00049639.JPEG |  | 00049665.JPEG |  | 00049691.JPEG |  |
| 00049612.JPEG |  | 00049640.JPEG |  | 00049666.JPEG |  | 00049692.JPEG |  |
| 00049613.JPEG |  | 00049641.JPEG |  | 00049667.JPEG |  | 00049693.JPEG |  |
| 00049615.JPEG |  | 00049642.JPEG |  | 00049668.JPEG |  | 00049694.JPEG |  |
| 00049616.JPEG |  | 00049643.JPEG |  | 00049669.JPEG |  | 00049695.JPEG |  |
| 00049617.JPEG |  | 00049644.JPEG |  | 00049670.JPEG |  | 00049696.JPEG |  |
| 00049618.JPEG |  | 00049645.JPEG |  | 00049671.JPEG |  | 00049697.JPEG |  |
| 00049619.JPEG |  | 00049646.JPEG |  | 00049672.JPEG |  | 00049698.JPEG |  |
| 00049620.JPEG |  | 00049647.JPEG |  | 00049673.JPEG |  | 00049699.JPEG |  |
| 00049621.JPEG |  | 00049648.JPEG |  | 00049674.JPEG |  | 00049700.JPEG |  |
| 00049622.JPEG |  | 00049649.JPEG |  | 00049675.JPEG |  | 00049701.JPEG |  |
| 00049623.JPEG |  | 00049650.JPEG |  | 00049676.JPEG |  | 00049702.JPEG |  |
| 00049624.JPEG |  | 00049651.JPEG |  | 00049677.JPEG |  | 00049703.JPEG |  |
| 00049625.JPEG |  | 00049652.JPEG |  | 00049678.JPEG |  | 00049704.JPEG |  |
| 00049626.JPEG |  | 00049653.JPEG |  | 00049679.JPEG |  | 00049705.JPEG |  |
| 00049706.JPEG |  | 00049732.JPEG |  | 00049759.JPEG |  | 00049786.JPEG |  |
| 00049707.JPEG |  | 00049733.JPEG |  | 00049760.JPEG |  | 00049787.JPEG |  |
| 00049708.JPEG |  | 00049734.JPEG |  | 00049761.JPEG |  | 00049788.JPEG |  |
| 00049709.JPEG |  | 00049735.JPEG |  | 00049762.JPEG |  | 00049789.JPEG |  |
| 00049710.JPEG |  | 00049736.JPEG |  | 00049763.JPEG |  | 00049790.JPEG |  |
| 00049711.JPEG |  | 00049737.JPEG |  | 00049764.JPEG |  | 00049791.JPEG |  |
| 00049712.JPEG |  | 00049738.JPEG |  | 00049765.JPEG |  | 00049792.JPEG |  |
| 00049713.JPEG |  | 00049739.JPEG |  | 00049766.JPEG |  | 00049793.JPEG |  |
| 00049714.JPEG |  | 00049740.JPEG |  | 00049767.JPEG |  | 00049794.JPEG |  |
| 00049715.JPEG |  | 00049741.JPEG |  | 00049768.JPEG |  | 00049795.JPEG |  |
| 00049716.JPEG |  | 00049742.JPEG |  | 00049770.JPEG |  |  |  |
| 00049717.JPEG |  | 00049743.JPEG |  | 00049771.JPEG |  |  |  |
| 00049718.JPEG |  | 00049744.JPEG |  | 00049772.JPEG |  |  |  |
| 00049719.JPEG |  | 00049745.JPEG |  | 00049773.JPEG |  |  |  |
| 00049720.JPEG |  | 00049746.JPEG |  | 00049774.JPEG |  |  |  |
| 00049721.JPEG |  | 00049747.JPEG |  | 00049775.JPEG |  |  |  |
| 00049722.JPEG |  | 00049748.JPEG |  | 00049776.JPEG |  |  |  |
| 00049723.JPEG |  | 00049749.JPEG |  | 00049777.JPEG |  |  |  |
| 00049724.JPEG |  | 00049750.JPEG |  | 00049778.JPEG |  |  |  |
| 00049725.JPEG |  | 00049752.JPEG |  | 00049779.JPEG |  |  |  |
| 00049726.JPEG |  | 00049753.JPEG |  | 00049780.JPEG |  |  |  |
| 00049727.JPEG |  | 00049754.JPEG |  | 00049781.JPEG |  |  |  |
| 00049728.JPEG |  | 00049755.JPEG |  | 00049782.JPEG |  |  |  |
| 00049729.JPEG |  | 00049756.JPEG |  | 00049783.JPEG |  |  |  |
| 00049730.JPEG |  | 00049757.JPEG |  | 00049784.JPEG |  |  |  |
| 00049731.JPEG |  | 00049758.JPEG |  | 00049785.JPEG |  |  |  |