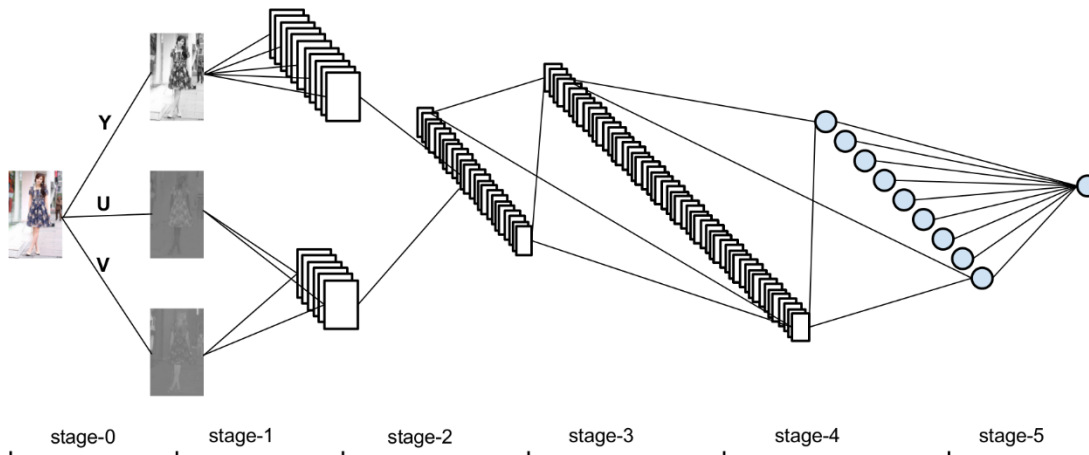


## 一、模型简介

在本次比赛中, 我们没有采用任何人工图像特征, 而是利用深度学习算法, 直接以像素作为模型输入, 中间经过 3 层可训练的卷积层和 1 个隐层进行变换, 最后 logistic 回归函数输出每个像素的概率, 训练后的效果为: 本地 i5 四核配置下平均每张图片处理时间为 1.14s, 交并比为 61.96%。该算法模型图如下:



Stage-0: 原图裁剪、缩小到 320\*240, 转换到 YUV 空间, 并对每个通道进行局部归一化;  
Stage-1: Y 图通过“卷积-池化-tanh”, 得到 10 张映射, UV 图同样得到 6 张卷积后的映射;  
Stage-2: 将 stage-1 得到的 16 张特征映射图通过“卷积-池化-tanh”得到 64 张特征映射图;  
Stage-3: 将 stage-2 得到的 64 张特征映射图通过“卷积”得到 256 张 80\*60 的特征映射图;  
Stage-4: 将每个像素对应位置的 256 维的向量作为全连接隐层 (384 个节点) 的输入;  
Stage-5: sigmoid 函数输出像素被划分为人物区域的概率。

## 二、训练方法

对于非线性层数较多的模型训练, 主要思想是采用逐层贪心进行监督训练, 共分三个阶段:

**阶段一:** 训练卷积层。由上图的 stage-0,1,2,3 + logistic regression, 代价函数为 cross-entropy, 采用 Mini-batch SGD 优化方法, 通过监督训练得到卷积层参数, 此时 logistic 可以去掉;

**阶段二:** 训练 MLP。将【阶段一】得到的卷积层参数固定, 在模型后面添加 hidden layer 和 logistic layer, 同样, 通过监督训练得到 stage-4 和 stage-5 的参数;

**阶段三:** Fine-tuning。在前面阶段所得参数初始值的基础上, 对整个神经网络进行微调。

训练代码用 python 的深度学习库 theano 完成, 在本地利用 GPU GTX 650 对 4900 张和 488 张图片进行训练和验证, 大概需要 5 天的时间。

## 三、总结与拓展

在上述模型训练中, 没有采用任何 regularization terms 或 early-stop 来防止过拟合, 得到的 training error 和 test error 差距约为 0.6%, 这说明了这个模型对于该任务来说还是偏于简单。另外, 该模型得到的剪影出现中空的现象, 这说明了卷积范围偏小, 某些区域无法利用边缘特征, 模型需要学习更大范围的局部特征, 针对以上问题, 提出以下几点拓展的建议:

- 1) 搭建多尺度图片的卷积神经网络, 充分利用局部特征和全局特征;
- 2) 采用稍复杂的图片后处理分割方法 (如 super-pixel, CRF 等), 结合神经网络的对每个像素的分类概率, 对原图实现更加精准的分割。