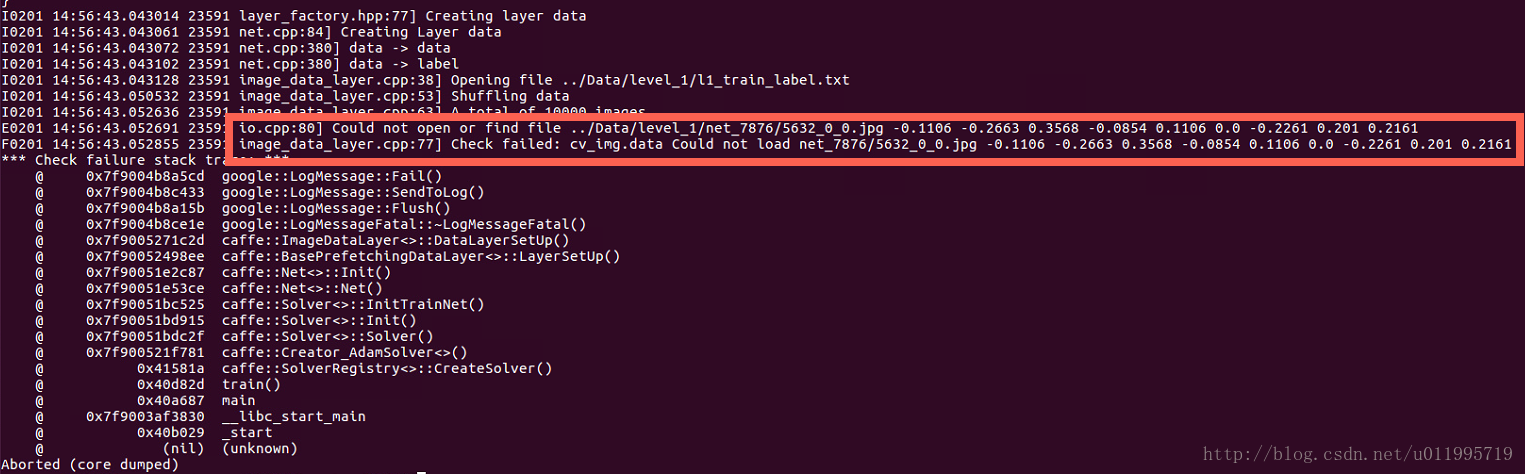
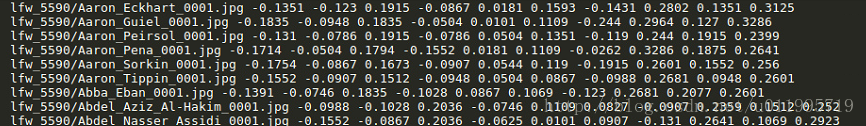
[**修改caffe源码--支持多标签--关键点检测**](http://blog.csdn.net/u011995719/article/details/79231671)

原版caffe不支持多标签，会报错，如下：   


注：这里读取数据的method是ImageData，即 type：ImageData

此方法直接从txt中获取图片路径和label，进行读取，txt如下所示：   


——————————————————————–正文————————————————————-   
需要更改两个文件分别是   
第一：caffe-master/include/caffe/layers 下的 image\_data\_layer.hpp   
第二：caffe-master/src/caffe/layers 下的image\_data\_layer.cpp

**第一步： image\_data\_layer.hpp**

头文件 image\_data\_layer.hpp，改动较少，仅一行；   
将：

vector<std::pair<std::string, int> > lines\_；

改成：

vector<std:pair<std::string, vector<float>>> lines\_;

原本label是用int类型，但我的任务是回归任务，因此修改为vector

**第二步： image\_data\_layer.cpp**

改动有三处，分别是   
1. 读取label；   
2. label\_shape;   
3. top\_shape

**1.读取label**   
原版label定义为整型 ，并且只有一个label值，原版label读取是这样的：   
第42行起

int label;

while (std::getline(infile, line)) {

pos = line.find\_last\_of(' ');

label = atoi(line.substr(pos + 1).c\_str());

lines\_.push\_back(std::make\_pair(line.substr(0, pos), label));

}

infile就是txt文件，line就是txt当中的一行，pos为了寻找标签的位置，lines\_存储读取的txt数据

从42行开始更改，更改为如下：

int pos; // int pos ;

int label\_dim = 0 ;

bool gfirst = true;

while (std::getline(infile, line)) {

if(line.find\_last\_of(' ')==line.size()-2) line.erase(line.find\_last\_not\_of(' ')-1);

pos = line.find\_first\_of(' ');

string img\_path = line.substr(0, pos);

int p0 = pos + 1;

vector<float> label\_vec;

while (pos != -1){

pos = line.find\_first\_of(' ', p0);

float v = atof(line.substr(p0, pos).c\_str());

label\_vec.push\_back(v);

p0 = pos + 1;

}

if (gfirst){

label\_dim = label\_vec.size();

gfirst = false;

LOG(INFO) << "label dim: " << label\_dim;

}

CHECK\_EQ(label\_vec.size(), label\_dim) << "label dim not match in: " << lines\_.size()<<", "<<lines\_[lines\_.size()-1].first;

lines\_.push\_back(std::make\_pair(img\_path, label\_vec));

}

label不再是int类型，而定义为vector<float>类型 ，并在 while(pos!=-1)里面循环读取

**2.label\_shape**   
在94行开始，原版是这样的

vector<int> label\_shape(1, batch\_size);

top[1]->Reshape(label\_shape);

改为

vector<int> label\_shape(2, batch\_size);

label\_shape[1] = label\_dim;

top[1]->Reshape(label\_shape);

**3. top\_shape**   
在134行，原版是这样

// Reshape batch according to the batch\_size.

top\_shape[0] = batch\_size;

batch->data\_.Reshape(top\_shape);

修改为

// Reshape batch according to the batch\_size.

top\_shape[0] = batch\_size;

batch->data\_.Reshape(top\_shape);///////////////////////////////////////////////

vector<int> top\_shape1(4);

top\_shape1[0] = batch\_size;

top\_shape1[1] = lines\_[0].second.size();

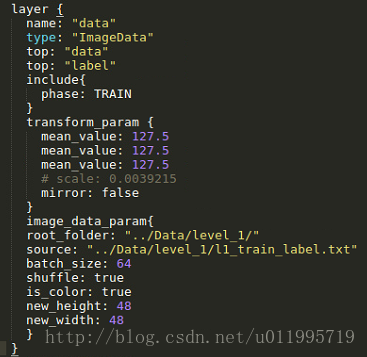
top\_shape1[2] = 1;

top\_shape1[3] = 1;

batch->data\_.Reshape(top\_shape);

重新 make 就可以使用多标签输入了；

prototxt中的data\_layer是这样的：



image\_data\_layer.hpp   
image\_data\_layer.cpp   
上传到了：   
<http://download.csdn.net/download/u011995719/10235806>