
地理信息系统实习报告

1. 视频处理和数据清洗

1.1. 处理已分类的数据

使用语言：Python。

读取每一种情感对应的.csv 文件，获取到图片文件名以及对应分数，再根据图片的文件名，使用 PIL 库读取图片并将图片转为 numpy 矩阵格式，在读取文件时将每一种情感的文件以 8:2 的比例分为训练集和测试集，最后通过调用 pickle 库将得到的图片的矩阵和分数以二进制的形式写到文件中。

1.2. 处理视频数据

使用语言：Python。

先读取和视频文件同级目录的文本文件，获取视频开始录制时间，再读取 CSV 文件，获取每一次采集的时间以及点位数据，用采集时间和开始时间转为时间戳之后相减，得到间隔时间。

再使用 OpenCv 库根据间隔时间来截取视频对应的每一帧，由于 OpenCv 读取图片的格式是 BGR，所以需要转为 RGB 格式之后，再将图片保存下来。

随后使用 pickle 库将图片以及对应的经纬度写入到文件中。

2. 神经网络搭建

与老师论文不同^[1]，本次实习选择精度更加高的神经网络解决问题。实习选择了深度学习的框架 pytorch，因为 pytorch 入门比较简单，对新手比较友好，比较适合解决这次实习的问题。

2.1. 数据处理

原来提供的数据是 300*400，为了提高网络的鲁棒性，需要做一定的处理。pytorch 在训练时，有一些图像处理函数，专门针对于 PIL 的图像文件进行处理，使用起来比较方便。本次实习采取的方法为训练时将图片随机切割 224*224 大小，并且随机将图片进行旋转，最终再作为输入。

2.2. 网络搭建

使用了 pytorch 网络当中的迁移学习来编写神经网络，使用 pytorch 框架已经预训练好的模型作为特征提取器，这样可以减少训练的时间并且提高特征提取的准确度。

2.2.1. Vgg 网络搭建

选择 vgg 模型是因为 vgg 模型是比较有名的神经网络模型，vgg 能够对图形数据进行很好的特征提取。本次实习使用了预训练好的 vgg16 模型，使用了模型前面的特征提取层，去除了后面的三层全连接层，改为了两层全连接层，减小模型的大小。然后将全连接层的最后一层的输出改为 1，其中参数设置为可以调整，允许模型参数的进行微调。

2.2.2. Resnet 网络搭建

ResNet 网络是为了解决这种退化问题而诞生的神经网络，也是比较有名的网络模型之一，本次实习也尝试使用 resnet 网络来解决实习当中的问题，使用的是预训练好的 resnet18 模型，使用了模型前面的特征提取层，将最后一层 fc 层改为了输出为改为 1，其中模型设置为可以调整，允许模型的参数进行微调。

2.3. 最终结果

最终将两个模型的六个感情训练时的 loss 曲线显示出来，最终结果图如图 1、图 2 所示：

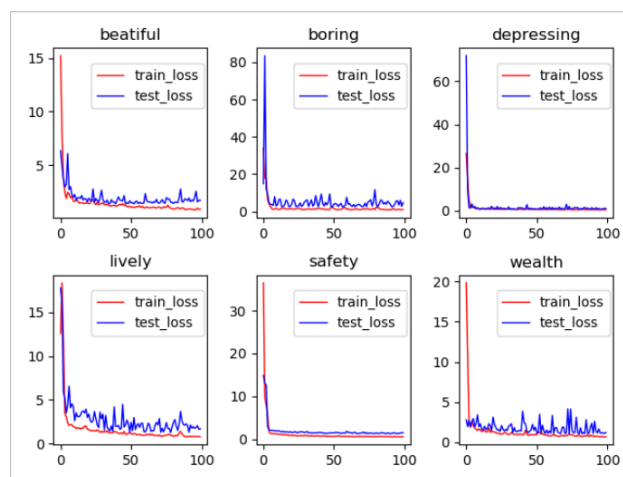


图 1: vgg 训练后的结果图，其中红色代表训练时的 loss 值，蓝色代表测试时的 loss 值，图从左到右，从上到下依次分别代表为 beautiful、boring、depressing、lively、safety、wealth。

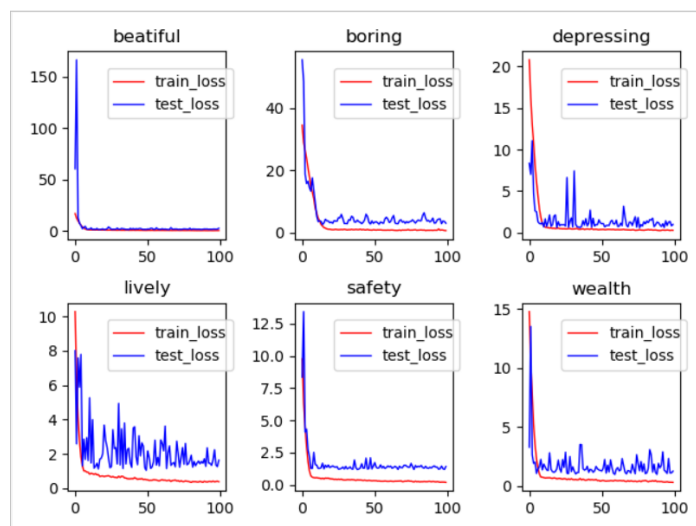


图 2: resnet 训练后的结果图, 其中红色代表训练时的 loss 值, 蓝色代表测试时的 loss 值, 图从左到右, 从上到下依次分别代表为 beautiful、boring、depressing、lively、safety、wealth。

3. 可视化

3.1. AntvL7 地图可视化

这一部分的可视化成果主要是可以选择点击右上角选择图层, 以及通过点击图中的要素展示要素的场景信息以及六维数据, 柱状高低和颜色深浅表示得分高低, 具体样式可以参考图 3。

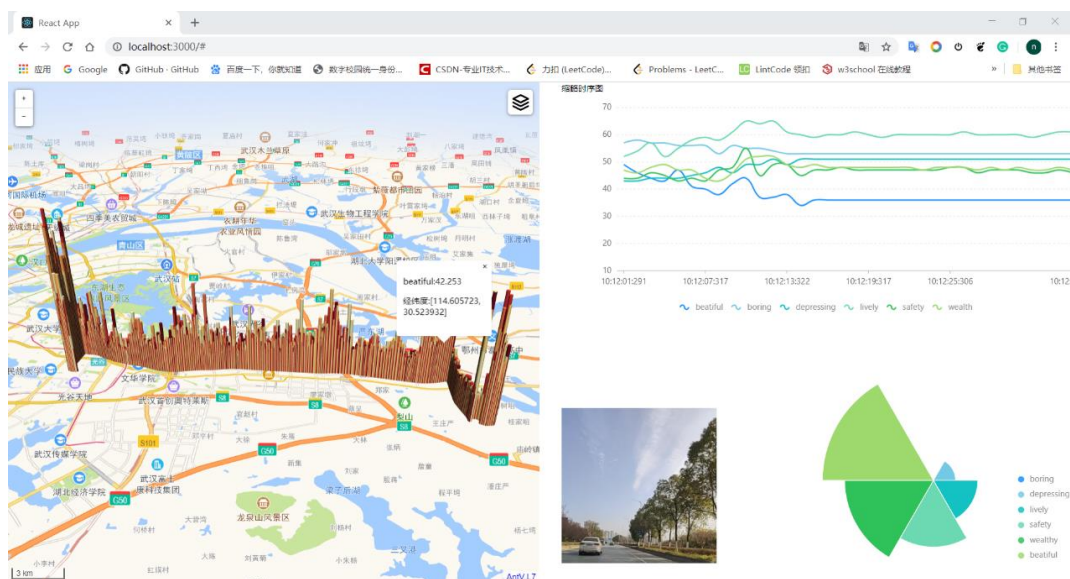


图 3: 可视化成功展示图, 左部为 17 制作的地图, 右下角左侧为被点击的图层要素所在点的场景, 右下角右侧为该点的六维数据制作的玫瑰图。

3.2. Antv G2 时序图展示

之所以选择时序图，是因为给的数据是一种视频数据，有很明显的时间连续性，各个数据之间应该是有一种关联的，通过查看一段时间内的数据的变化，再与视频数据相结合，更方便我们得出有效的结论。

3.2.1. 流动时序图

流动时序图的目的是可以更直观的查看一段区域内的情感的变化趋势，因为时序图变化时，同时下方的图片会随着时序图变化，这样可以结合视频查看对应情感分数变化趋势，流动时序图的样式可以参考图 4。

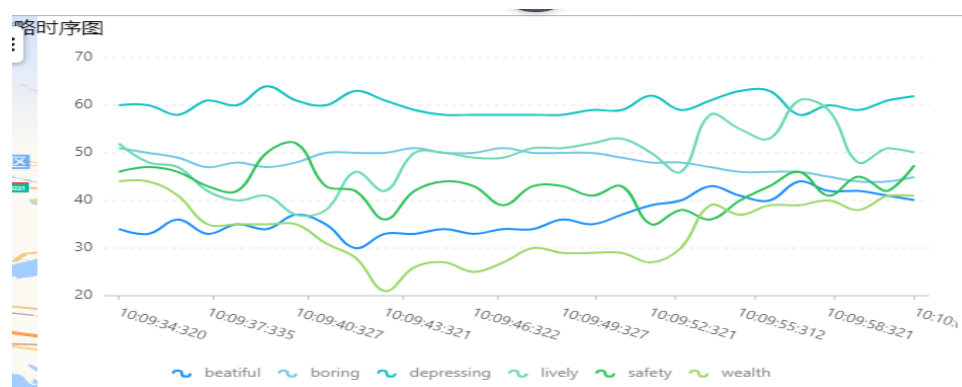


图 4：流动的时序图，流动时图 1 中右下的图片也会一起流动。

3.2.2. 带缩略轴的时序图

带缩略轴的时序图的目的是可以查看用户任意一段时间内的数据，可以让用户自由选择想要查看的时段的数据变化趋势，带缩略轴的时序图的样式可以参考图 5。

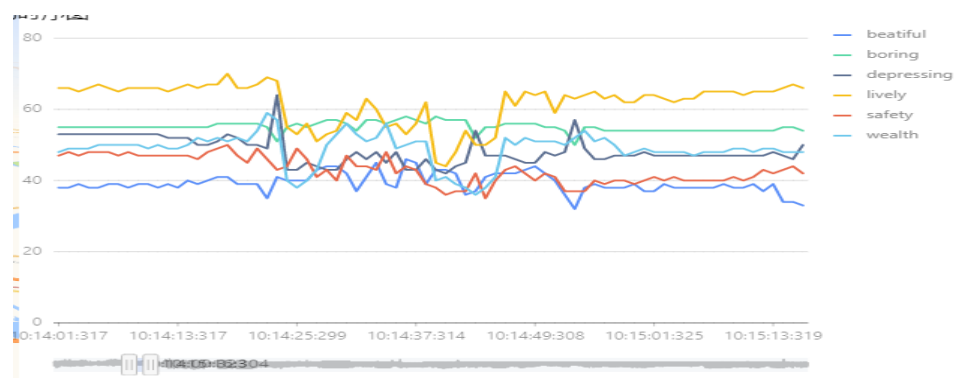


图 5：带缩略轴的时序图，可以通过滑动下方的缩略轴，查看用户任意一段时间内的数据。

3.3. 数据处理

我们对数据做了简单的处理，流动时序图和主地图中，将数据进行了将周围连续三个数据取平均值，缩略轴时序图中做了隔三个取一个值。这样可以避免直接将数据暴露给用户，也可以提高可视化的效率。

4. 采集视频和点位数据 APP

使用语言：Java。

开发软件：Android Studio。

使用 MediaRecorder API 进行录像，再使用高德 API 获取点位数据。

启动 APP 时就开始定位，当点击录制按钮时，先记录开始时间，然后每隔 1 秒采集一次经纬度和时间，判断是否处于录制状态，如果是就记录下经纬度和时间，否则不记录。当点击停止录制时，将采集到的数据存放到 CSV 文件中。



图 6：采集视频和点位数据的安卓 APP，可以录制视频，采集经纬度和时间信息，并保存到本地

Time	Longitude	Latitude	StartTime
2020-01-03 15:12:42:206	114.618076	30.455954	2020-01-03 15:12:40:386
2020-01-03 15:12:43:580	114.618076	30.455954	2020-01-03 15:12:40:386
2020-01-03 15:12:45:485	114.618076	30.455954	2020-01-03 15:12:40:386
2020-01-03 15:12:47:576	114.618076	30.455954	2020-01-03 15:12:40:386
2020-01-03 15:12:49:572	114.618076	30.455954	2020-01-03 15:12:40:386
2020-01-03 15:12:50:789	114.6177107	30.4556714	2020-01-03 15:12:40:386
2020-01-03 15:12:51:775	114.6176474	30.4556547	2020-01-03 15:12:40:386
2020-01-03 15:12:52:783	114.6175756	30.4556346	2020-01-03 15:12:40:386
2020-01-03 15:12:54:279	114.6173722	30.4557495	2020-01-03 15:12:40:386
2020-01-03 15:12:55:284	114.6173622	30.4557595	2020-01-03 15:12:40:386
2020-01-03 15:12:56:281	114.6173572	30.4557562	2020-01-03 15:12:40:386
2020-01-03 15:12:57:275	114.6173522	30.4557528	2020-01-03 15:12:40:386

图 7：APP 采集到的数据，保存在 CSV 文件中，分别是采集信息时间、经度、纬度和开始录制时间，取得的经纬度是火星坐标

5. 组内成员工作量说明

周宁：神经网络部分 35%

王纪峰：可视化部分 35%

王文旭：数据处理+App 编写 30%

6. 参考文献

[1] Yao Y, Liang Z, Yuan Z, et al. A human-machine adversarial scoring framework for urban perception assessment using street-view images[J]. International Journal of Geographical Information Science, 2019,33(12):2363-2384.