

# 专题名称： 服饰搭配的流行趋势预测

姓名 常晟 学号 2017080064 班号 计73  
同组姓名 林志儒 学号 2017080075 班号 计72  
指导老师 贾珈 辅导老师 陈雨兰

## 1. 文献综述

### 1.1引言

随着潮流趋势的演变，人们对于服饰的偏好也有所变化。对于特定区域、特定时间的服装搭配通常会有极高的相似性，比如一些体育赛事亦或是重大节日。我们的项目是服饰搭配的流行趋势预测，希望能通过对往年数据的分析预测未来可能出现的服装搭配。

### 1.2预测模型的研究现状

首先说说流行色，在参考文献[8]中，作者采用的是马尔科夫预测法，他将 2007 年年度 流行色彩占比作为初始状态，并基于国际委员会发布的流行色彩占比值做出状态转移方程，通过代入初始状态预测未来流行颜色的年度占比，并将预测的结果与实际结果做比对，并以 相对误差作为判定依据。最后发现仅有蓝色的占比准确度达到 90%，其余的颜色则都有一定的偏差，总的来说效果不是很理想。我猜测以年作为预测的单位可能会有一定的局限性，时间跨度或许能小一些。所以后续实验我们可能会尝试复现该方法，并试着以月作为单位研究会有什么样的影响。

参考文献[7]作者同样是做流行色的预测，但则是选择从 CIFLab 系统的角度对颜色做 考量。他以色相、色调做综合预测，并采用 BP 神经网络模型。最终的预测结果也更加逼近 实际值，总的来说效果略优于马尔科夫预测法。

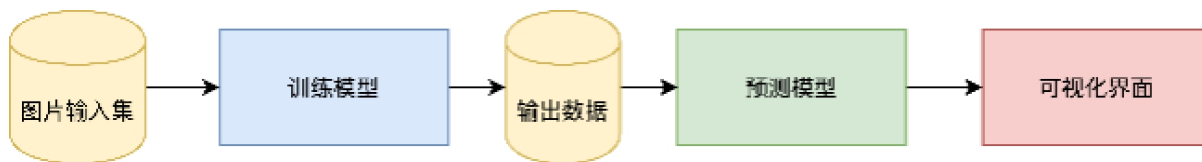
参考文献[9]作者尝试将回归分析法应用于服饰流行色的预测。在本篇文献中作者仅以 黄色服饰为例子，通过权重的方式表示出颜色的流行程度，之后将一个颜色的权重代入回归 方程计算出未来该颜色的权重，并以此来判断该颜色的流行程度。我认为该方法缺乏严谨，关于年度一个颜色的权重该如何判断，人为主观定义颜色的权重可能不准确，而且我们同样 也缺乏合适的手段来量化颜色的权重。

参考文献[4]中作者没有给出一个完整有效的预测模型，他是以统计分析为手段对多个 特征分析并进行人工的预测。作者将服饰的每个特征分类并做统计，然后从每类特征中选取 出现次数最多的元素作为预测结果。虽然作者没给出一个有效的预测模型，但我们可以尝试 沿着这个思路探索，特征的出现次数或许也可以作为预测考量的方法之一。

## 2. 实验方案设计报告

### 2.1整体框架

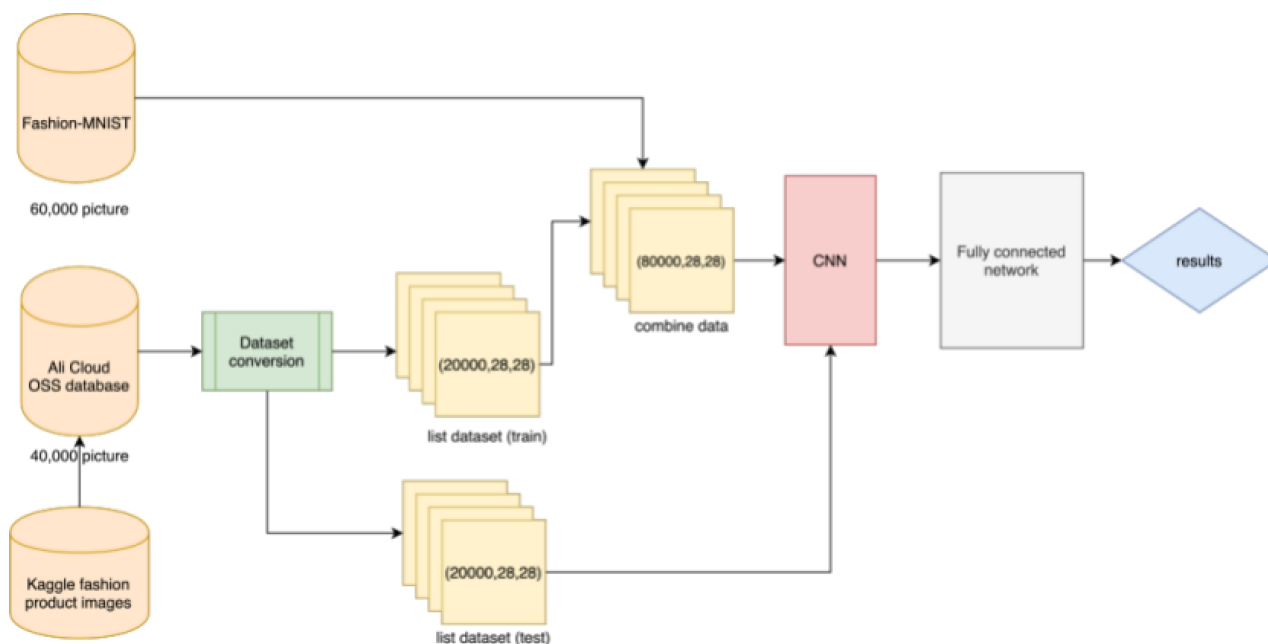
由于本项目的内容为“服饰搭配的流行趋势预测”，所以我们首先需要能够鉴别出不同服饰特征的一个模型，也需要一个能够预测将来趋势的模型。期望就是将一张照片喂给模型之后，能够提供服饰的两种特征，款式和风格。初步的流程框架如下：



首先将一系列的照片集喂给训练集进行训练，再将另一批数据喂给训练模型来进行一个历史数据统计的过程。之后，将这些历史的数据封装成某种数据结构之后，喂给预测模型来统计将来某种风格的服饰将走向的趋势。最后，将结构使用可视化的效果展示出来。

### 2.2训练模型

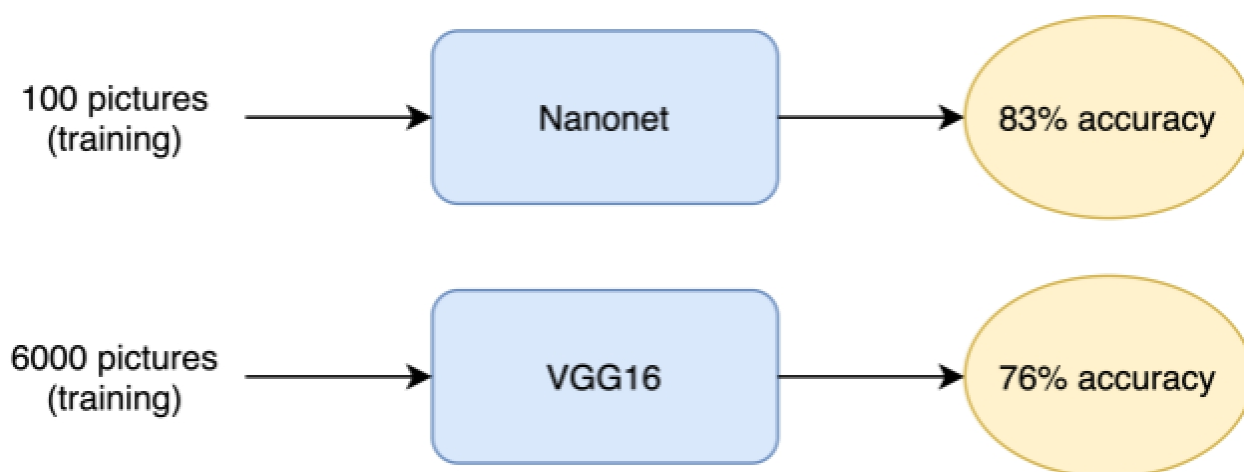
首先第一种训练模型是比较经典的CNN模型。CNN在图像处理方面有不少关注，而且很多文献里面提到过的模型。整体的框架如下图所示：



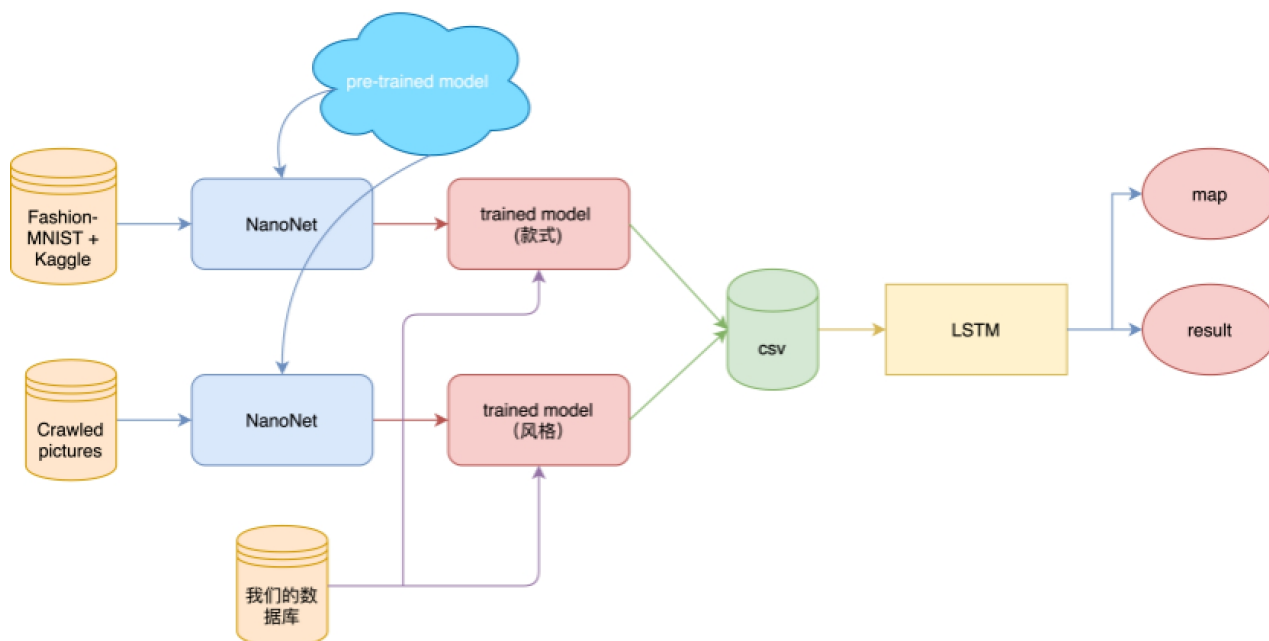
我们选用了两组数据集来检测服饰的款式，那就是谷歌提供的Fashion-MNIST训练集，还有Kaggle的训练集。里面有上千张服饰的图片，但是Fashion-MNIST训练集的缺点就是每张照片比较干净，很容易就能够辨别出照片中的款式。所以我们加入了Kaggle数据集，原因就是Kaggle里面的照片比较便生活化一点。意味着相比Fashion-MNIST，它的照片比较丰富，训练集的照片会变的稍微复杂一些。由于Fashion-MNIST的数据可以直接使用谷歌提供的load函数就能调用，但是Kaggle没有任何这种功能，所以要自己手动写照片分割，层次化，数字化等符合CNN输入的矩阵化脚本。由于Kaggle的照片偏多，在项目中保存的话不是很理想，所以将照片都存放在了阿里云OSS对象存储当中。将2万张照片提取出来之后，把每一张转换成2828的矩阵，矩阵里面的数字对应着照片的每一个像素的颜色。将这些矩阵叠加起来就是200002828的三维矩阵。将这些数据跟Fashion-MNIST的数据合并起来，喂给CNN模型。实现了这么多之后就出现了一系列的问题，首先就是我们实现的模型的准确率太低，准确率只有百分之50左右。

这个原因就是照片的质量不够高，而且照片数量过低。还有就是出现了严重的过拟合现象，导致把另一批数据喂进去之后，在效率和准确率方面都特别的低，而且也特别难解决。第三就是模型使用的时间过久，训练时间以及测试时间都要花将近1个多小时才能完成。第四就是代码的复杂度越来越高，逻辑过于复杂。所以由于这些原因，我们打算使用其他的模型来实现。其中在参考文献【1】当中有提到，***“While several excellent alternatives exist (such as VGG [Simonyan and Zisserman 2014]), GoogLeNet offers a good tradeoff between accuracy on tasks such as image classification [Russakovsky et al\*\*\*. \*2015] and speed. As with other moderate-sized datasets (fewer than 1 million training examples), it is difficult to train a CNN on our attribute prediction task from\****

***\*scratch without overfitting.\**** 意思就是如果是自己写的CNN模型的话将会很容易出现非常严重的过拟合现象，而且会比较难解决这一类问题。根据参考文献【2】，其中有提到使用预处理过的模型将会大大提高测试的准确率，而且不会出现过拟合的现象。里面提到的预处理模型有ResNet50, VGG16, InceptionV3, 以及GoogLeNet。最终由于我们的数据量较小，而且数据的质量不高，所以决定使用Nanonet进行训练。Nanonet包含了许多种不同的预处理神经网络模型，而且会根据提供的训练数据选择一个最符合这种数据的模型。在参考文献【3】里面有提到一个例子，就是作者对比了Nanonet和VGG16的准确率，作者给了Nanonet训练100张数据，且给VGG16训练6000张数据，最后把一样数量的测试集喂给两个模型之后，发现Nanonet的准确率要比VGG16的赵品霖要搞7%，这是因为Nanonet包含了更多的预处理模型。而且会选择一个最佳的参数，还会进行一个数据的扩充。



由于没有找到任何现有的服装风格的照片，所以只能自己写一份爬虫脚本，从bing爬去上千张不同风格的服饰照片。其中风格有“嘻哈风”，“欧美风”，“街头风”等等。最终我们的流程图结构如下：

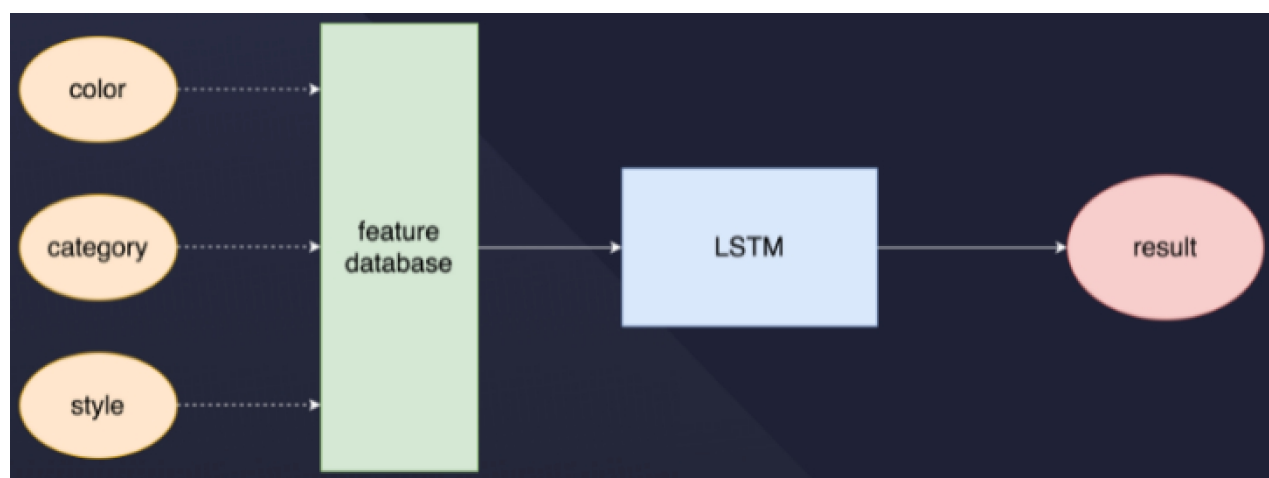


从左边开始，上面有Fashion-MNIST以及Kaggle的照片，用来训练检测服装款式的数据集，下面是我们爬出来的照片，有不同风格的照片。使用这两个数据集训练出两种不同的训练模型。最下面有我们找到的照片集，其中有4万张照片且每张照片拥有年份和地点。将这些照片喂给训练完的模型之后，将会得出一个历史数据统计集。将这些信息封装成一个csv的数据结构直接喂给LSTM进行预测。

## 2.3预测模型

我们采用LSTM模型对服饰特征的趋势进行预测。LSTM是一种特殊的RNN，相比于普通的RNN，LSTM在更长的序列中有更好的表现，并且它能比较好的解决梯度消失和梯度爆炸问题。

深入查阅资料后发现服饰的预测与时间序列预测有很大的相似性，它们都是希望能根据之前的内容来对未来的内容做预测，而服饰预测有很大的一部分内容都属于统计，即将特定特征的数据归纳，然后用这些归纳的数据做出预测。具体的预测框架图如下：



首先给定需要预测的color、category、style，然后LSTM会从数据模块中提取出我们指定的内容，并将这些内容按年份分类。本次课题中我们选取了2009-2018年的数据用于训练并对2019年的结果做预测，其中为了衡量我们模型的准确程度，我们选取了数据模块中2019年的统计结果作为正确值。

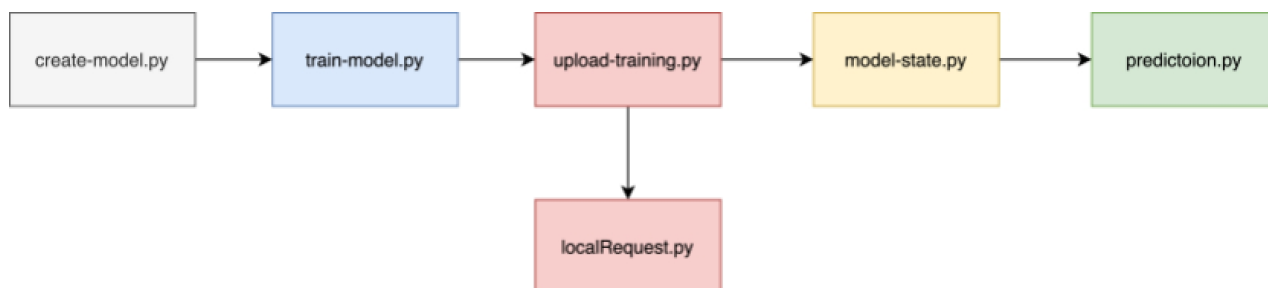
简单举个预测的例子，比如我要预测嘻哈风格黑色T恤的流行趋势，那么color就给定black，category给定Tshirts，style给定嘻哈服饰风格，之后LSTM会从数据模块中找出符合条件的所有数据，并按照年份将所有数据分类，之后我们就把它当成是一个时间序列预测的问题，对2019年的结果做预测。

由于我们做的是针对地区趋势预测，所以在提取数据时同样会将相同国家的数据作为一类。值得一提的是预测结果并不是我们最终返回的，因为我们希望能在世界地图上能比较直观的看出流行的趋势，也就是能直观的找出哪个国家在该选定的特征下会更流行，所以我们将预测结果与前一年的数值作为比值返回。该比值是我们对流行趋势的衡量，比值越高我们就认定这些特征的服饰在该地区流行的可能性会更高。

## 3. 程序说明与系统性能报告

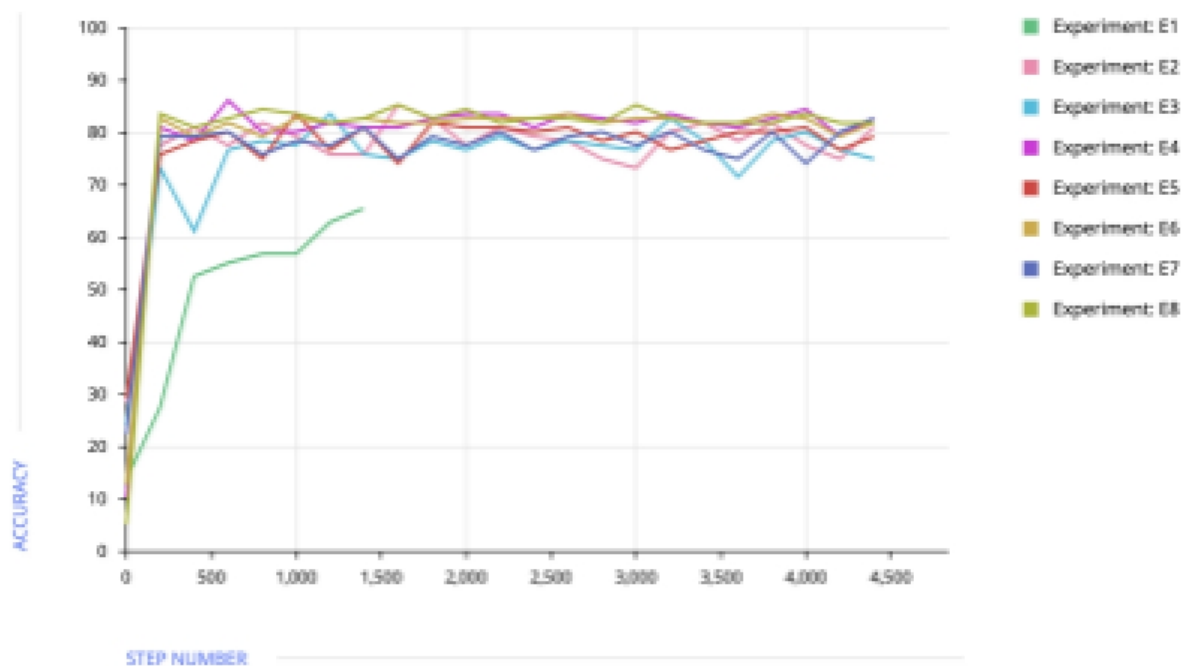
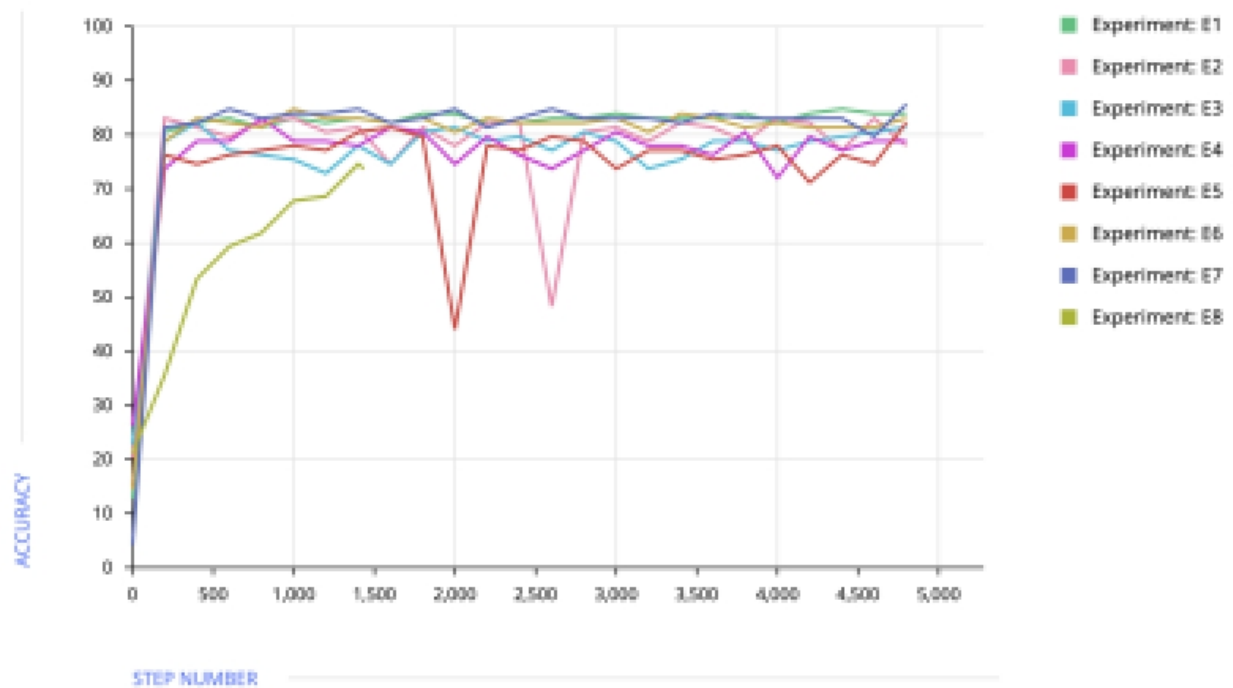
### 3.1 训练模型程序说明与性能

由于我们使用的Nanonet训练模型要传到云端，所以编译训练模型的时候如下：



在Nanonet/code路径下面有以上所示的python代码。首先就是运行create-model.py，它的功能就是能够初始化一个空的神经网络模型。之后运行train-model.py这个就是一个上传自己的数据集的一个过程。后面的upload-training.py就是会依据我们的数据，在云端找到一个最佳的预处理模型以及会帮我们进行一个数据的扩充。之后model-state.py会返回训练的状态以及prediction.py可以从本地上传照片到训练模型进行检测。下面的localRequest.py的作用就是使用POST请求的方式，能够给我们云端的训练模型提供照片，然后会用POST的返回结果来得出每一个类型的打分。

Nanonet的准确率比较高，也没有出现很严重的过拟合现象。服装风格的准确率可以保持在83%左右，且服装款式的准确率可以达到85%左右。



## 3.2预测模型程序说明与性能

预测模块包含predict.py和worldMap.py。predict.py是预测算法的主体，worldMap是用于对趋势可视化。其中predict.py最后会生成一个fashion\_statisc.csv，这个文件存储了每个国家对于特定特征的趋势预测结果，worldMap.py会读取该文件并可视化，然后生成一个html文件。该html文件绘制的是一张世界地图，图中每个国家均会被上色，黑色表示缺少预测结果的国家，其他的国家则是颜色越深，则代表某特定特征下来年的趋势会更流行，颜色越浅则是相反。

```
LSTM.py >>> entrance()
```

算法入口，在该部分指定需要预测的特征以及针对的国家，然后会对指定的各个国家分别做预测，并将预测结果汇总。

```
LSTM.py>>> LS()
```

预测算法，需要指定数据的训练年份。这里我们选取的是2009-2018年的数据做训练，对2019年的结果做预测。该函数会在entrance()中调用，并会传入选取的数据(进入该函数时已经获取了选取好的数据集，即此时已经知道要对什么特征做预测，以及对应预测的国家是什么)。目前模型的搭建如下图所示：

```
model = Sequential()
model.add(LSTM(4, input_shape=(None, 1)))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(trainX, trainY, epochs=50, batch_size=1, verbose=2)
```

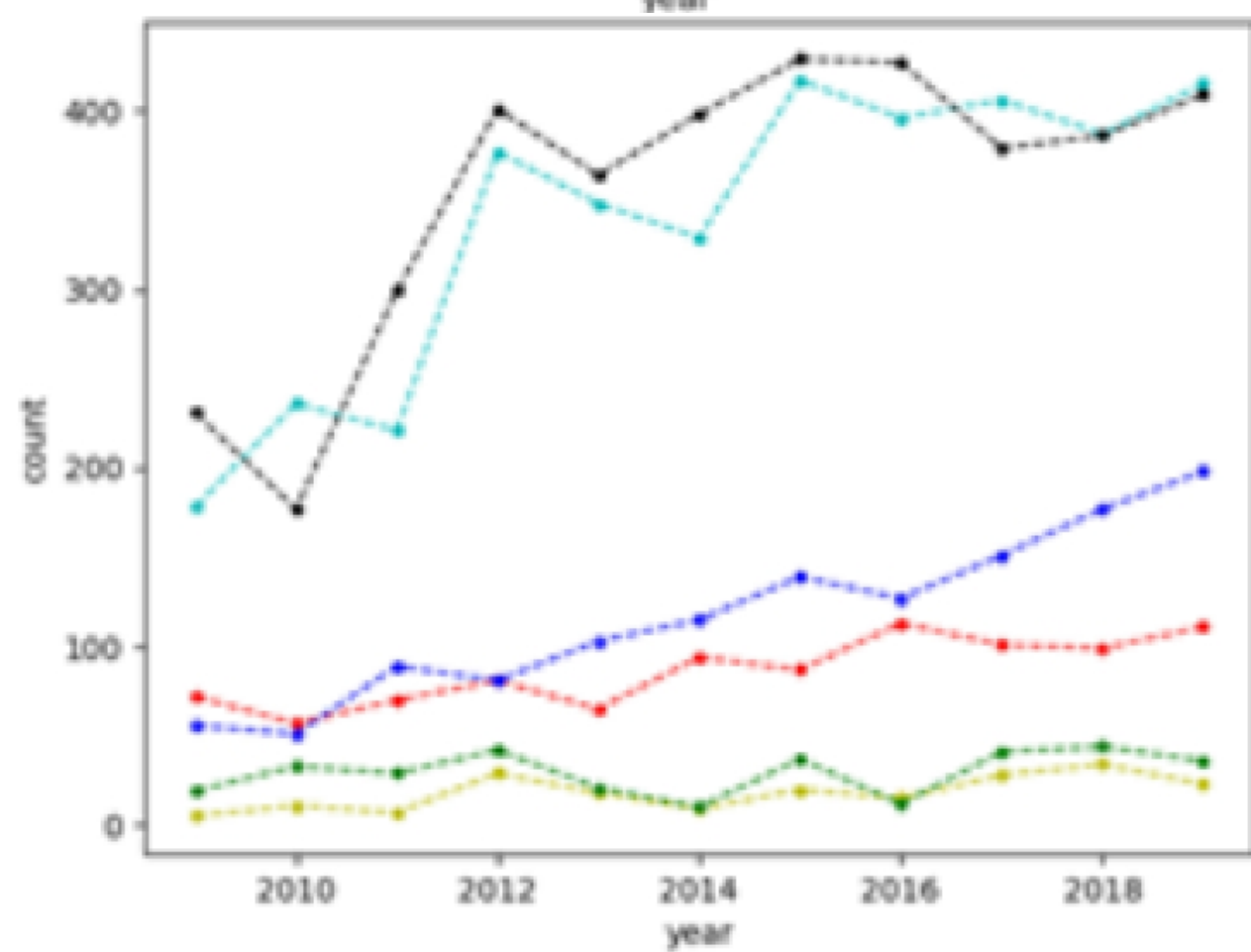
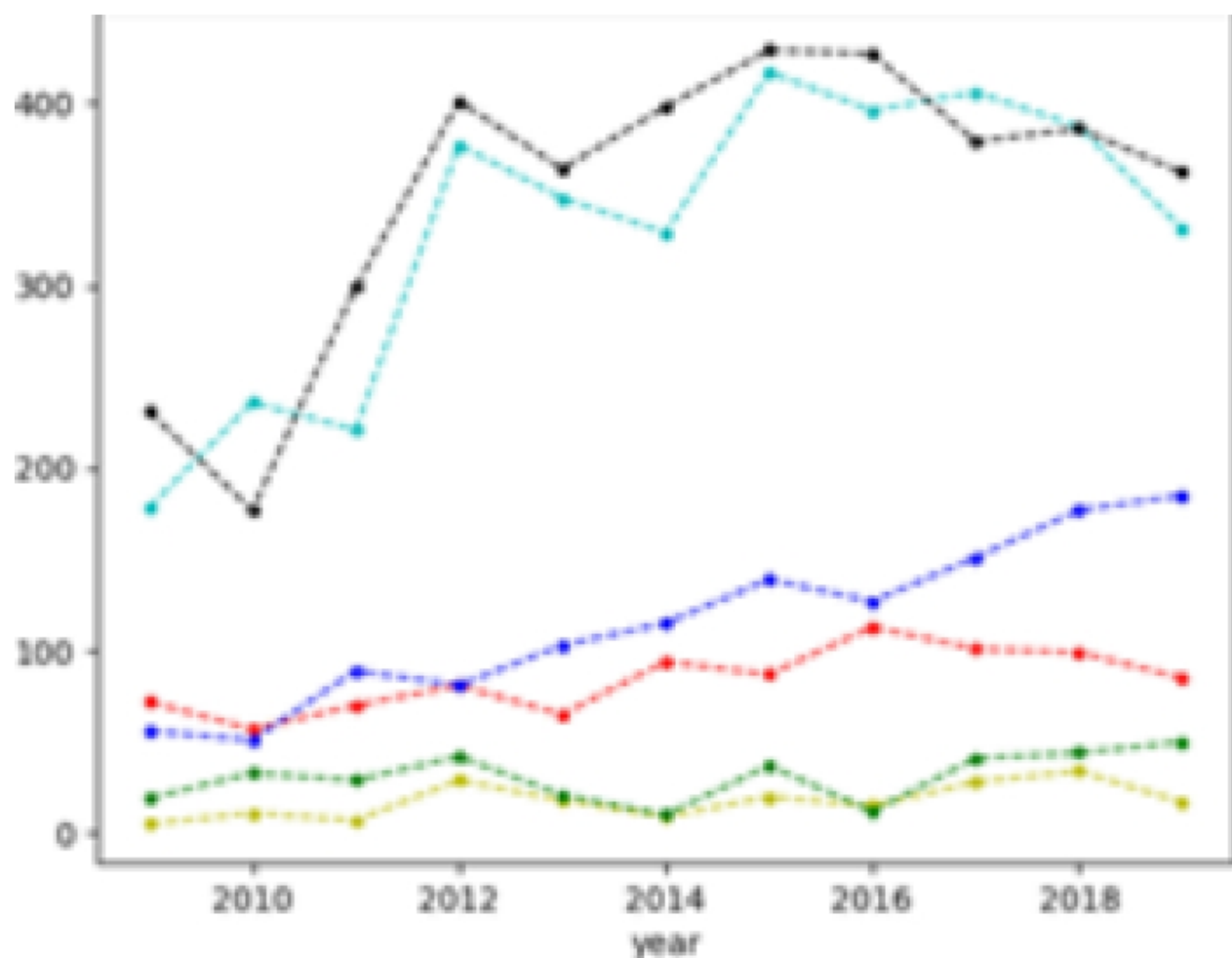
最终该函数会返回特定特征的预测结果，返回的形式是一个具体的数值。

```
LSTM.py>>> extract()
```

数据提取模块，调用时需指定服饰的具体特征(可以是多个)以及国家。目前支持三个特征：服饰主色调，服饰类型，服饰风格。然后该函数会读取df.csv，并返回我们需要的数据集。

首先是单个特征的预测，这里选取的是服饰主色调特征，具体预测的颜色为Red, Yellow, Blue, Green, White, Black，趋势线的颜色各自对应，除了白色对应青色趋势线。以下是预测的结果图。左图为正确结果，右图为预测结果，下图为预测的误差。这里我们只针对2019年做预测，所以2009至2018年这个区间内的线段趋势是相同的，两图仅是2019年这个点不同。

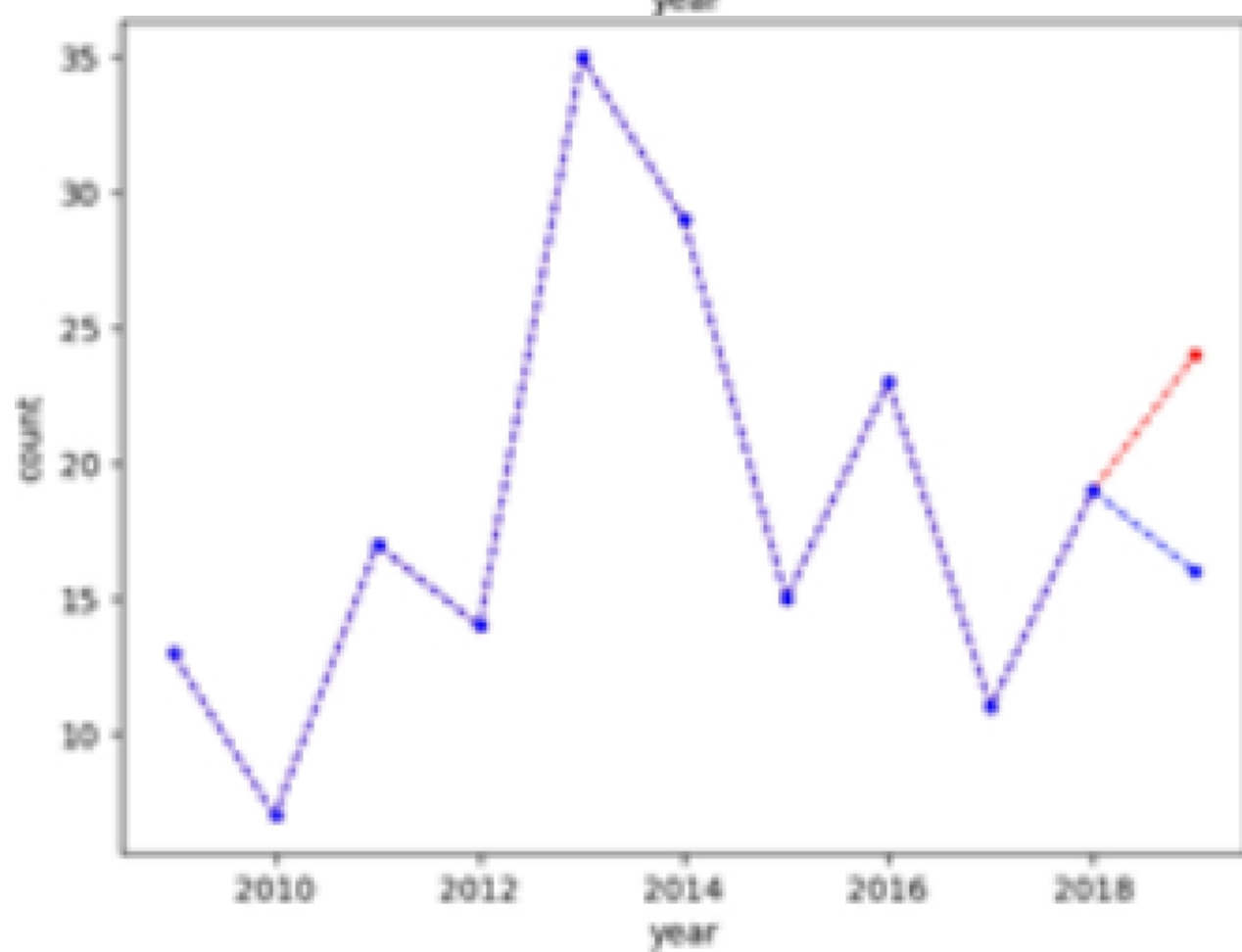
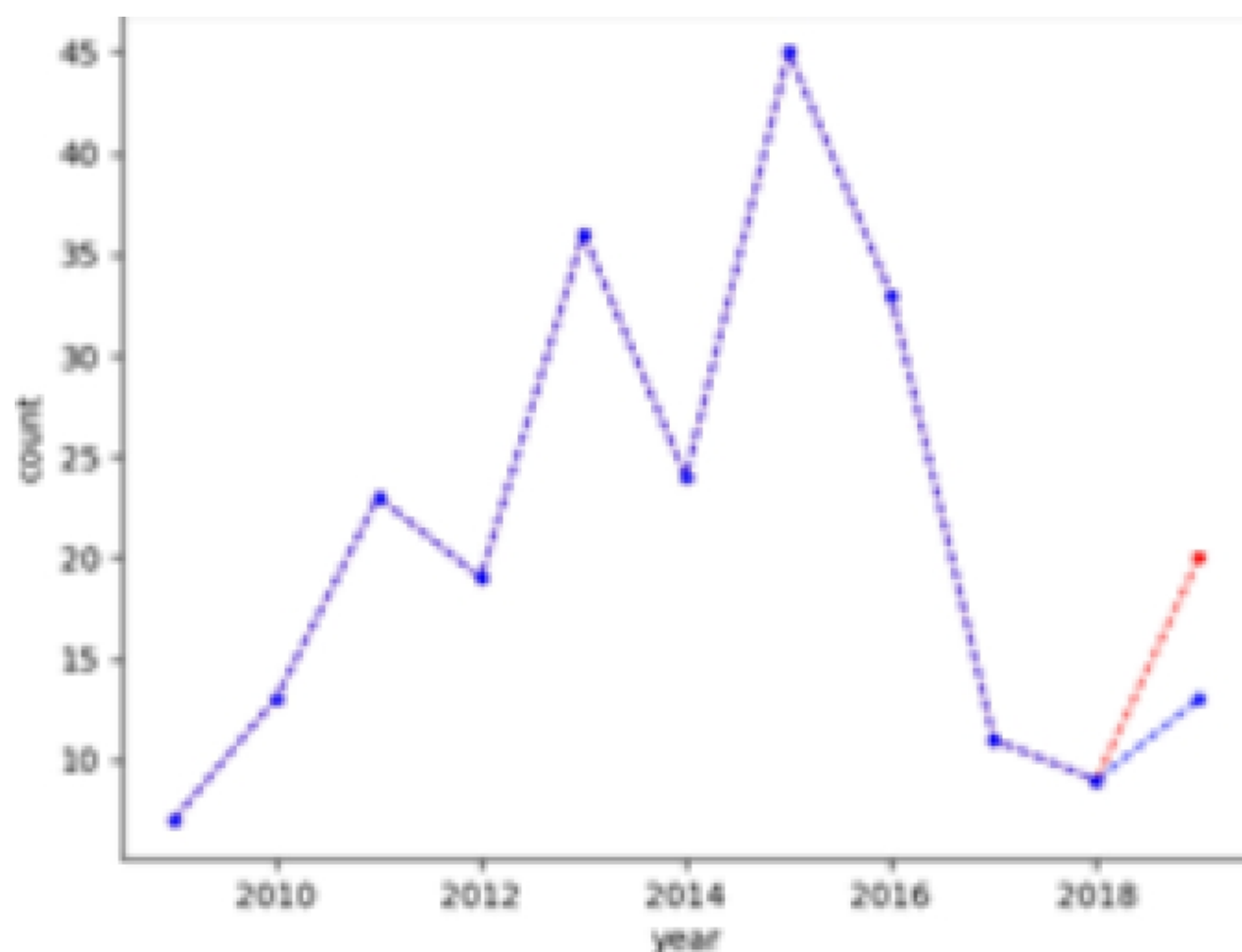






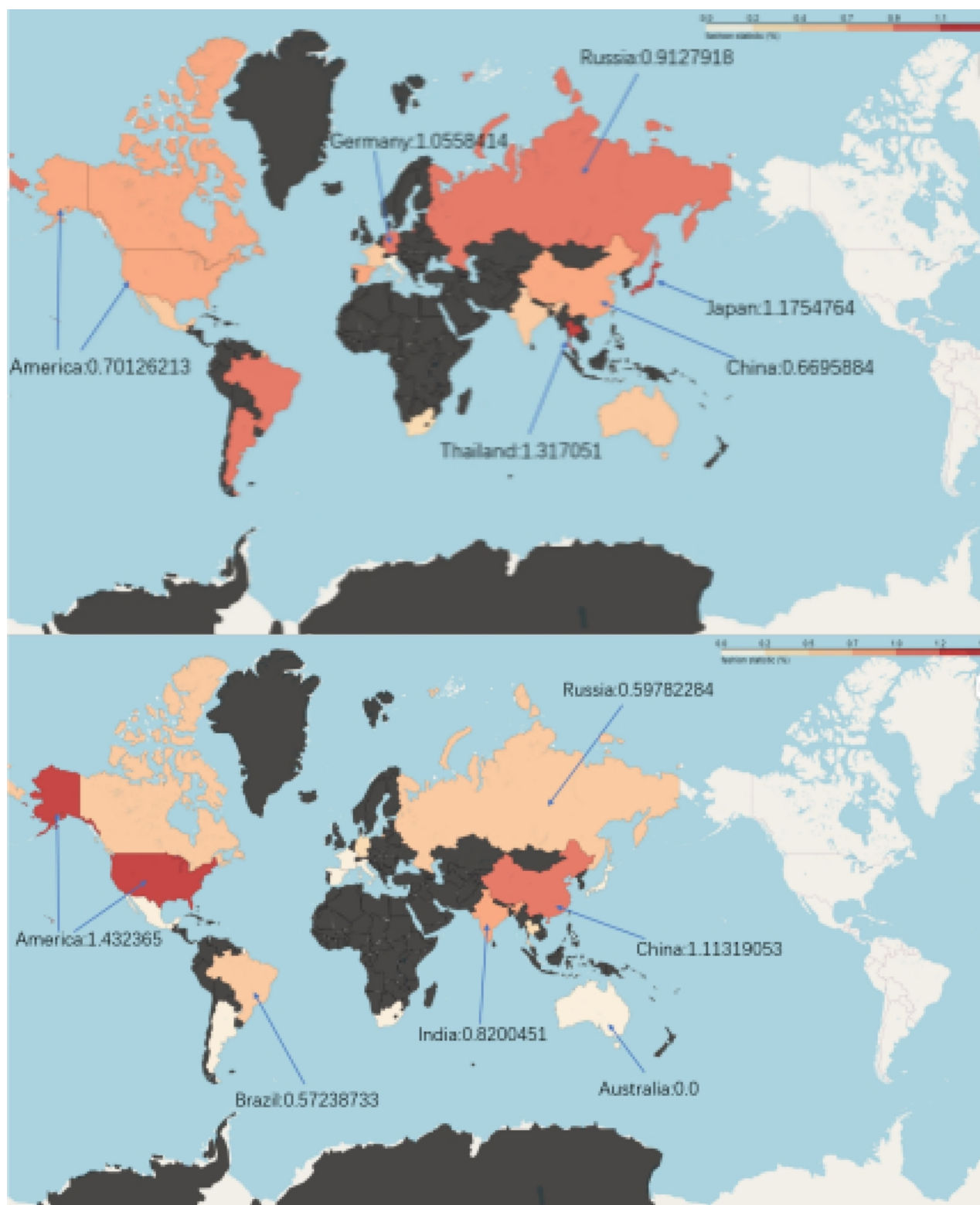
Color	Prediction	Answer	Error
Red	111	85	+26
Yellow	23	17	+6
Blue	198	185	+14
Green	36	50	-14
White	415	331	84
Black	409	362	47

这是三个特征的预测，这里选取了服饰主色调、服饰类型、服饰风格这三个特征。以下是预测的结果图。左图为黑色+T恤+嘻哈风格，右图为白色+T恤+嘻哈风格，下图为预测的误差。红色趋势线代表2019年的正确结果，蓝色趋势线代表2019年的预测结果。2009-2018年两条趋势线结果是相同的。



	Prediction	Answer	Error
Black+Tshirts+hip hop	13	20	-7
White+Tshirts+hip hop	16	24	-8

从误差结果来看数值较小的预测它们的误差也会偏小，而对于大数值的预测则是不太理想。整体来看预测结果会比正确结果偏高。这可能与数据集有一定的关系，因为用于训练的点大概只有10个，这算是比较少的。预测出来的结果也可能会因此受到不同程度的影响。除此之外数据量也会带来一些负面的作用，比如训练点是0，这也是我们非常不愿意见到的。



最后这两张图就是我们的可视化结果。左图的测试样例是蓝色+T恤两个特征的预测，右图是黑色+T恤+嘻哈风的预测。在图中可以看到有几个国家被标记了对应的数值，这也就是我们前边提到的比值，其比值越高，对应的颜色也就越深。但这样的表示也会有一些不足，比如在数据量较小的情况下其给出的比值可能会偏大，又或者是缺失数据也就是右图Australia这样的值。

## 4.总结与展望

---

通过这次的小学期项目，深入理解了多媒体领域的图片处理模型以及预测模型的知识。其中也遇到了许多的困难，而且因为疫情的缘故所以只能远程交流使项目变的更难完成，尤其是在对接方面没有足够的交流，导致了不少的麻烦。但最后还是顺利的完成了助教还有老师给的任务。最后要非常感谢我的队友林志儒，以及我们助教陈雨兰学姐的帮忙。

## 参考文献

---

[1].StreetStyle: Exploring world-wide clothing styles from millions of photos

[2].Evaluation of CNN Models with Fashion MNIST Data

[3].Le, J. (2018, September 08). How to Classify Fashion Images easily using ConvNets. Retrieved July 16, 2020, from <https://medium.com/nanonets/how-to-classify-fashion-images-easily-using-convnets-81e1e0019ffe>

[4]. Van Dyk Lewis, Mengyun (David) Shi.Using Artificial Intelligence to Analyze Fashion Trends[J].Cornell University, Hearst Magazines,2016.

[5]. Kevin Matzen.StreetStyle: Exploring world-wide clothing styles from millions of photos[J].arXiv,2017

[6]. Choudhary, Anshu. (2018). FASHION TRENDS' IMPACT ON SOCIETY.

[7]. 狄宏静, 刘冬云, 吴志明, DI Hongjing, LIU Dongyun, & WU Zhiming. (2011). 基于 BP 神经网络的春夏女装流行色预测. 纺织学报, 32(7), 111-116.

[8]. 常丽霞, 高卫东, 张万琴, 马宝林, 吴增宝, CHAGN Li-xia, . . . WU Zeng-bao. (2012). 马尔可夫预测法在国际服装流行色预测中的应用. 毛纺科技, 40(7), 44-47.

[9]. 周琴, 吴志明, 高卫东, ZHOU Qin, WU Zhi-ming, & GAO Wei-dong. (2005). 用回归分析法预测服装流行色. 丝绸, (2), 35-37.