

手绘草图分类问题

林宇欣: 220030910004, 胡屹垚: 220030910016, 王祎雯: 220030910002

- 1 项目简介
- 2 相关工作
- 3 数据集介绍
- 4 研究方法
 - 4.1 传统分类方法
 - 4.2 深度学习方法
 - 4.2.1 ResNet
 - 4.2.2 VGGNet
- 5 实验分析
 - 5.1 传统分类模型
 - 5.1.1 SVM
 - 5.1.2 KNN
 - 5.1.3 MLP
 - 5.1.4 PCA & TSNE
 - 5.2 深度学习模型
 - 5.2.1 ResNet
 - 5.2.2 VGGNet
- 6 总结与展望
- 7 参考文献

1 项目简介

手绘草图 (freehand sketch) 是一种形式简单但蕴含丰富信息的艺术形式, 通过简洁的线条能表现出强烈的风格以及传递鲜明的信息。虽然人类能够轻松地 from 简笔画中获得主要的信息, 但是这个任务对于机器来说非常有挑战。这种挑战主要来自于手绘草图含义的模糊性、同类对象在几何表达上的差异性, 以及模型的输入图片缺乏丰富的纹理特征。因此本项目旨在实验比较不同模型对手绘草图的分类效果。本项目实践了传统分类模型和深度神经网络模型, 其中传统模型包括了SVM (Support Vector Machine)、KNN (K Nearest Neighbor) 和MLP (Multi Layer Perceptron) 方法, 深度神经网络模型包括了ResNet和VGG网络, 其中VGG网络是本项目中表现最优的模型, 其准确率达到85.06%。

2 相关工作

手绘草图识别的难点有3个。首先, 对于相同类别的图片, 人对于该类别的关键信息理解不同, 并且根据直觉创造出内涵模糊的图形, 这种模糊性甚至对人的识别来说也是较为困难的。其次, 手绘草图根据绘画者的不同, 其艺术表达风格也不同, 因此同种类别下草图的表现形式差异也非常大。最后, 和照片的识别相比, 手绘草图没有颜色和纹理的信息, 同时画面中也存在大量相同的留白, 因此图像信息相对之下更稀少。

针对手绘草图识别任务的模型可以按照模型输入的不同分为三类。第一类是将手绘草图的分类视为图像分类任务, 即模型的输入是像素图片。像素图片的分类可以用传统的分类方法, 如SVM和KNN来实现。也可以用卷积神经网络进行学习和分类。早期使用卷积神经网络的方法效果并不出众的一个原因在于没有足够的数据集来支撑卷积网络的训练。而另外一个原因则是由于手绘图和照片的区别, 导致为照片分类设计的卷积网络在手绘草图的分类上表现并不突出。第二类是基于笔画顺序设计的分类模型, 这类模型的输入是人在绘制时按照时间先后顺序画出的线条序列。这类模型认为手绘草图的重要特性是笔画的线条和绘制的时序信息, 因此选择笔画顺序为模型输入。这类模型为了利用信息中的时序信息, 通常采用循环神经网络。而第三类模型是同时利用像素图像信息和笔画顺序信息进行分类, 希望充分挖掘和利用手绘草图中的信息。例如Sktech-a-Net[1], 是将传统的卷积神经网络进行了修改, 考虑到了人在绘制时倾向于先画总体特征, 再画细节特征的习惯, 因此对多层次的特征进行了提取和融合。而Sketch-R2CNN[2]则是采用了卷积神经网络和循环神经网络两个分支, 使用卷积神经网络将笔画顺序特征转换为多通道的点特征, 再使用卷积神经网络进行识别分类。

3 数据集介绍

本项目使用的数据集为QuickDraw数据集[3]。QuickDraw数据集收集了来自Quick, Draw! 游戏中用户在20秒内所画的指定类别的手绘草图。在这个数据集中, 每个草图样本是以笔画顺序的形式进行存储的。草图样本由一系列点组成, 每个点由5个特征($\Delta x, \Delta y, p_1, p_2, p_3$)构成, 这5个特征依次表示了当前笔画相对于上一笔画在画布上的偏移量($\Delta x, \Delta y$), 以及当前笔画的特征状态。其中 p_1 表示了当前笔画正在接触画布, 并且会与下一个点连接起来; p_2 表示在当前笔画之后笔将被抬起; 而 p_3 表示整个绘制已经结束。原始的QuickDraw数据集共有345个种类的手绘草图, 每个种类下均有70000张训练集样本, 2500验证集样本, 和2500个测试集样本。本项目使用的数据是QuickDraw原始数据集中的25个动物草图种类, 分别为牛、熊猫、狮子、老虎、浣熊、猴子、刺猬、斑马、马、猫头鹰、大象、松鼠、羊、狗、熊、袋鼠、鲸鱼、鳄鱼、犀牛、企鹅、骆驼、火烈鸟、长颈鹿、猪和猫。

对于原始数据的笔画序列, 我们选择将其转换为28*28像素大小的png格式图片作为模型的输入。序列到图片的转换我们采用了<https://github.com/CMACH508/RPCL-pix2seq> 中的方法。

4 研究方法

4.1 传统分类方法

传统的分类方法有SVM和KNN算法，这两个方法作为经典的有监督学习模型，应用场景非常广泛，因此本项目选择采用这两个算法进行实验。其中SVM算法作为本次实验的基线实验进行测试。而MLP模型虽然作为神经网络模型区别于以上提到的两个模型，但由于他的结构相对简单，并且网络的每层之间只采用了全连接的形式，因此在本项目中把它也归入了传统的分类方式。

同时，为了更好地了解数据集，我们还使用了TSNE对数据的分布进行了可视化。类比于同样维度，同样是黑白图片，同样是手写的MINST数据集，降维手段在MINST数据集上能帮助提高分类器的准确率，因此我们也选择在本项目的数据集上进行降维，探究降维对手绘草图分类任务是否有帮助。

4.2 深度学习方法

4.2.1 ResNet

直观来看，一个深度学习所采用的网络架构层数越多，就越适用于具有复杂结构的数据的特征提取。可是事实上，即便是在采用BatchNorm等手段削弱深度网络中梯度爆炸或梯度消失对训练造成的影响后，深度学习的过程中仍然可能会出现随着网络层数的增加，网络准确度达到顶峰乃至下降的困难。人们称这种难以解释的现象为深度网络的退化问题。本次大作业中，我们采用了深度残差网络（Deep Residual Network, ResNet）来进行手绘草图数据集的分类训练；而它的提出正是为了应对如上所述的退化问题，从而能够构建更深的网络习取更为精确的数据特征。

一个很朴素的想法是，若较深的网络层能够直接复制浅层网络习得的特征，那么即便深度网络没有办法在训练过程中获得更多的长进，也至少不会发生退化。基于这样的假设，ResNet网络在VGG网络的基础上做出了一些改进，也即通过引入了一个系列的残差单元来构造一种短路连接；它将深层网络的输出与浅层网络的输出（也即残差单元的输入）相加，让这个单元在输入特征基础上学习新的特征。残差单元的构造如下：

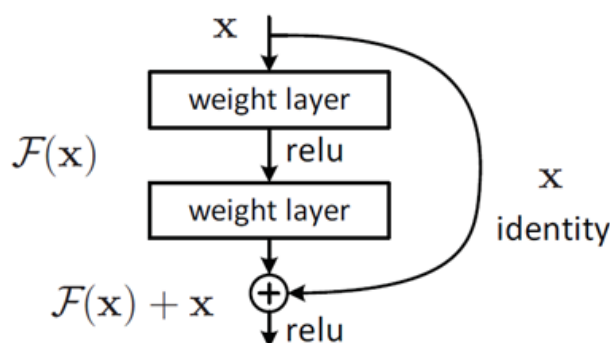


图4-1 残差单元

其中， x 代表残差单元的输入， $F(x)$ 代表单元中最深层网络的输出，也即残差；残差单元的输出为 $F(x) + x$ ，也即网络在该层上学习得到的特征。残差一般较小，在这个结构下传播的梯度也不容易消失，因此学习起来较为容易，即便构建很深的网络也不会轻易产生性能下降的问题。

考虑到在本次大作业中我们的数据集为大小仅有28*28的图片，我们分别采用了ResNet18，ResNet34和ResNet50这三个网络模型来进行上述手绘草图数据的特征学习和分类。没有采用ResNet101及层数更深的模型是因为数据结构复杂程度有限，使用太深的网络训练分类器会导致较为严重的性能过剩，缺乏实际意义。上述网络均包含5个卷积层组，每个卷积组中包含1次下采样操作和至少1个基本的卷积计算过程（即卷积计算，批量归一化，ReLU函数激活神经元）。其中，第2至第5个卷积层组由残差单元构建。

4.2.2 VGGNet

VGGNet是由牛津大学VGG组提出的深度神经网络，它的主要工作是在AlexNet的基础上，通过更小的卷积核和更多的卷积层代替较大的卷积核，从而增加网络的深度[5]。可以得到证明的是，多个堆积的小卷积核可以代替大卷积核，与此同时引入的更多非线性层可以保证学习更复杂的模式，而且参数数量可以得到控制。VGG神经网络的构造形式见表：

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64	conv3-64	conv3-64	conv3-64
maxpool					
conv3-128	conv3-128	conv3-128	conv3-128	conv3-128	conv3-128
maxpool					
conv3-256	conv3-256	conv3-256	conv3-256	conv3-256	conv3-256
conv3-256	conv3-256	conv3-256	conv3-256	conv3-256	conv3-256
maxpool					
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
maxpool					
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

在本次项目中，考虑到数据是 28×28 的灰度图片，我们使用了最常见的VGG16和VGG19网络（表中D、E结构），并在此基础上做了一些修改。由于VGG网络中有5个最大池化层，每经过一次池化层，输入的宽度和高度均会缩小到原来的一半，而 $2^5 > 28$ ，所以我们去掉了第1、3和第5个最大池化层来保证该网络可以处理小尺寸的图片。

此外，草图图片的数据分布差异较大，不同类别的样本数据存在着比较大的差异，所以我们在每一个卷积层的输出后面增加了一层批标准化层，希望使数据分布在较小范围内，使训练过程更加稳定。

$$\mu_B \leftarrow \frac{1}{m} \sum_{i=1}^m x_i$$

$$\sigma_B^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

使用批标准化层的部分代码如下：

```
def conv_layer(C_in, c_out, k, p):
    layer = tnn.Sequential(
        tnn.Conv2d(c_in, c_out, kernel_size=k, padding=p),
        tnn.BatchNorm2d(c_out),
        tnn.ReLU()
    )
    return layer
```

5 实验分析

5.1 传统分类模型

传统方法的实验中，我们选择将 28×28 的像素图片展平，将 784×1 维的数据作为模型的输入进行实验。

5.1.1 SVM

由于实验机器的内存和时间限制，本实验无法使用每种动物分类下的全部草图，因此本实验采取了进一步缩减训练集样本数量的策略，采取随机抽取再混合的方式生成新的训练集，并在测试得出最佳实验参数后，对内存承受范围内的不同大小的训练集也进行了测试。

首先我们从每个分类下的动物中抽取了500张图片进行混合生成了大小为5000的训练集，用于确定SVM模型的最佳参数。我们测试了不同的惩罚项系数 C ，和两种不同的核函数，其训练结果如下图所示。从图中可以看出，线性SVM在不同惩罚项系数下的准确率几乎没有差别，可以认为线性SVM在该分类任务中的效果非常差，而rbf核在 $C=1$ 时取得了最高的准确率0.476256。因此后续关于训练集大小的实验中我们采用了这组超参数。

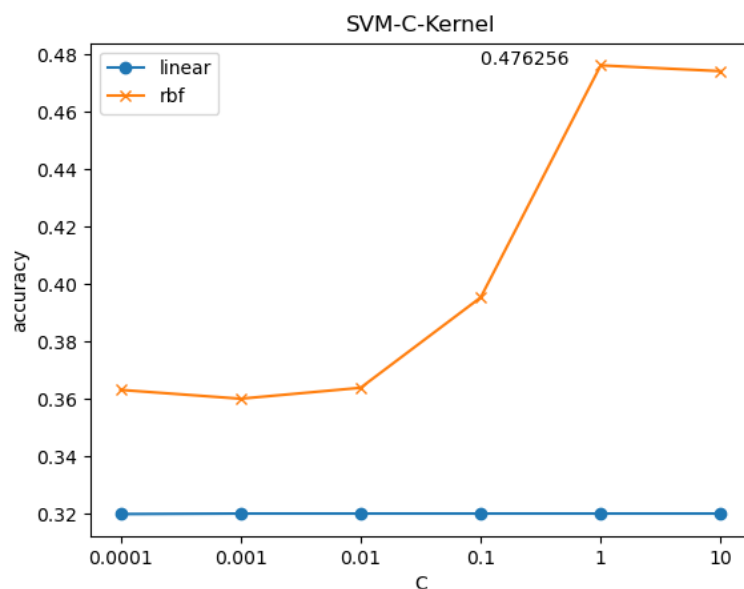


图5-1 SVM不同参数表现

其次，我们按照上述相同的方法，分别从每个分类下的动物中抽取了100，500，1000，2000，5000个样本混合组成了训练集进行训练和测试，其训练结果如下图所示。从图中可以看出，SVM的准确率在随着训练集的增大而增大，因此可以推测当内存和算力足够时，SVM的准确率可以达到高于59.74%的效果。但是需要注意的是，当训练集样本的增多，训练时长也在快速增长，并且这个增长并不是线性的。

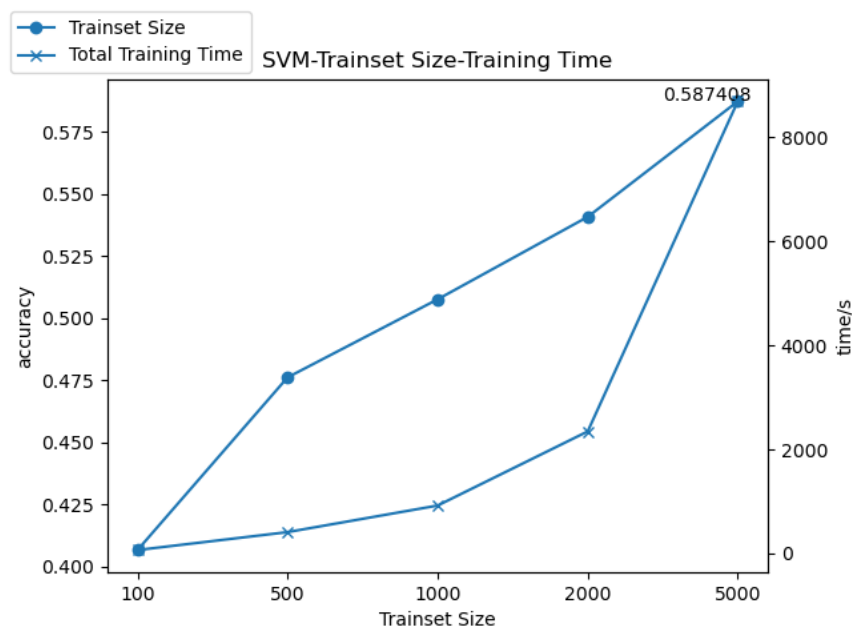


图5-2 SVM training time

5.1.2 KNN

对于KNN算法，由于本实验的场景下我们已知动物分类共有25种，因此我们选择直接指定超参数 $k=25$ 。对KNN算法，我们实验了三种距离度量方式。同时，由于机器内存和时间限制，我们也采取了减小训练集大小，测试不同大小训练集结果的方式进行实验。

首先我们从每个分类下的动物中抽取了10000张图片进行混合生成了大小为250000的训练集，用于确定KNN模型的最佳距离度量方式。实验结果如下表所示。从结果中可以看出，欧式距离度量是最适合本项目场景的，其拥有最高的准确率和最快的运行时间。因此之后关于训练集大小实验也采用与之相同的超参数。

距离度量	准确率	运行时长/s
manhattan	38.76%	9117.74
euclidean	45.48%	480.92
chebyshev	29.53%	5735.5

其次，我们按照上述相同的方法，分别从每个分类下的动物中抽取了100, 500, 1000, 2000, 5000, 10000个样本混合组成了训练集进行训练和测试，其训练结果如下图所示。从下图中可以看出，KNN的准确率在随着训练样本的增多而增大，因此可以合理推断，在内存足够时，KNN的准确率可以达到高于45.48%的效果。同时，与SVM相比，KNN的运行时间明显更短，并且其时间是随着训练集样本数量线性增长的。虽然KNN相比SVM的分类效果更差，但是在考虑运行时间成本上，KNN具有好的表现。

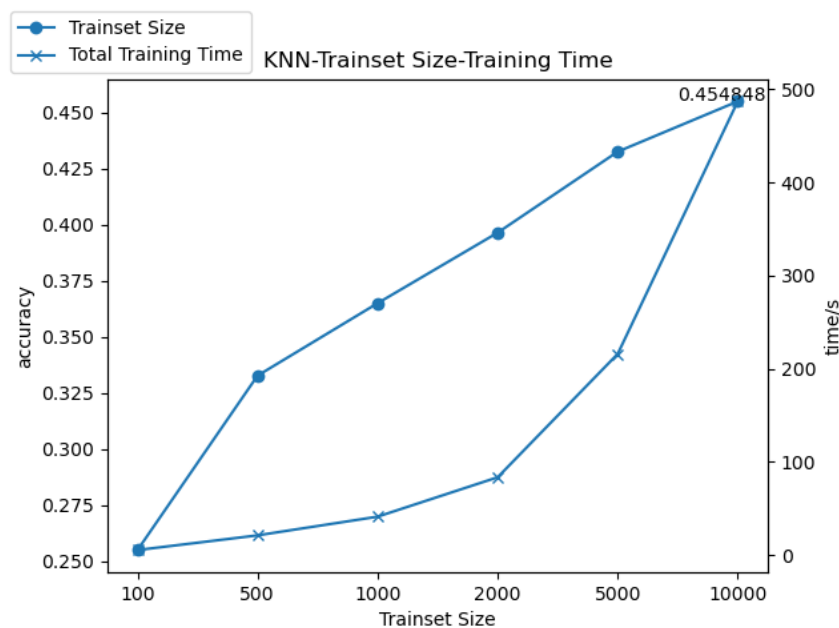


图5-3 KNN training time

5.1.3 MLP

对于MLP模型，我们测试了不同的网络宽度，深度以及迭代轮次。同时，由于机器内存和时间限制，我们也采取了减小训练集大小的策略，我们在每个种类的动物中选取了5000张草图样本组成了新的训练集进行训练。

首先，我们对单隐层不同宽度的MLP网络在不同迭代轮次下的性能进行了测试。考虑到MLP运行结果的不稳定性，我们的测试结果是重复3次运行后的平均准确率。测试结果如下所示。从图中可以看出，随着网络宽度的加深，MLP的准确率在提高，但是可以看出，每增加100个神经元，每次模型准确率的提升在减少。除此之外，可以看出模型的准确率随着迭代轮次的增加在上下震荡，因此后续实验中考虑采用最少的5000轮进行迭代。

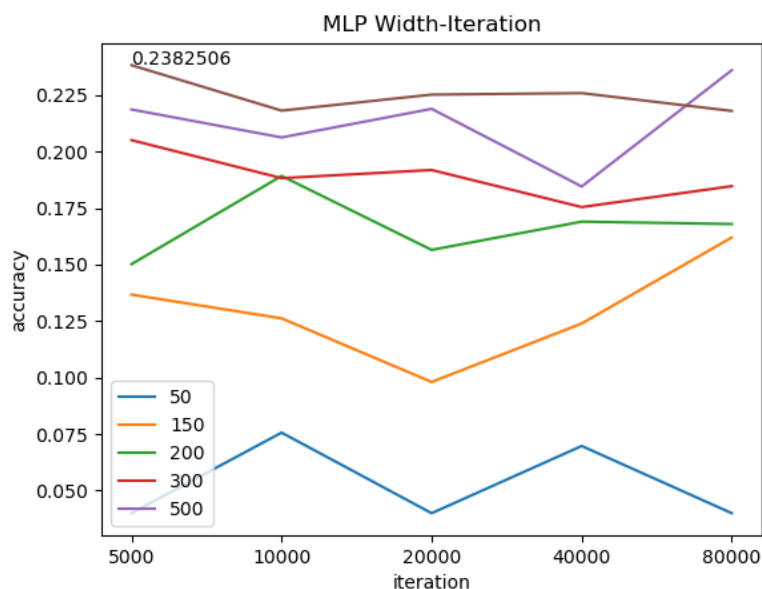


图5-4 MLP accuracy

其次我们测试了隐层每层500个神经元的情况下，存在1层，2层和3层隐层的网络效果，其结果如下表所示。可以看出在两个隐层为500时，准确率最高只有29.02%。MLP模型与SVM和KNN相比，效果更差。有部分原因是复杂的网络模型需要更大量的样本进行训练，但由于机器的限制，本项目只能支持MLP使用与SVM和KNN同等大小训练集进行训练。同时，MLP模型的输出结果并不稳定，经常会出现现在测试集上的表现为随机选择的期望大小的正确率。此外，MLP的全连接结构并不能获取图像中空间有关的信息，因此整体网络结构并不适合此场景。

网络结构	[500]	[500, 500]	[500, 500, 500]
准确率	23.82%	29.02%	11.13%

5.1.4 PCA & TSNE

首先，我们采用了TSNE对原始数据进行了可视化，其可视化结果如下图所示。可以看出，对于25类的分类任务，将数据拉至2维后无法直观地看出同类样本的聚集。

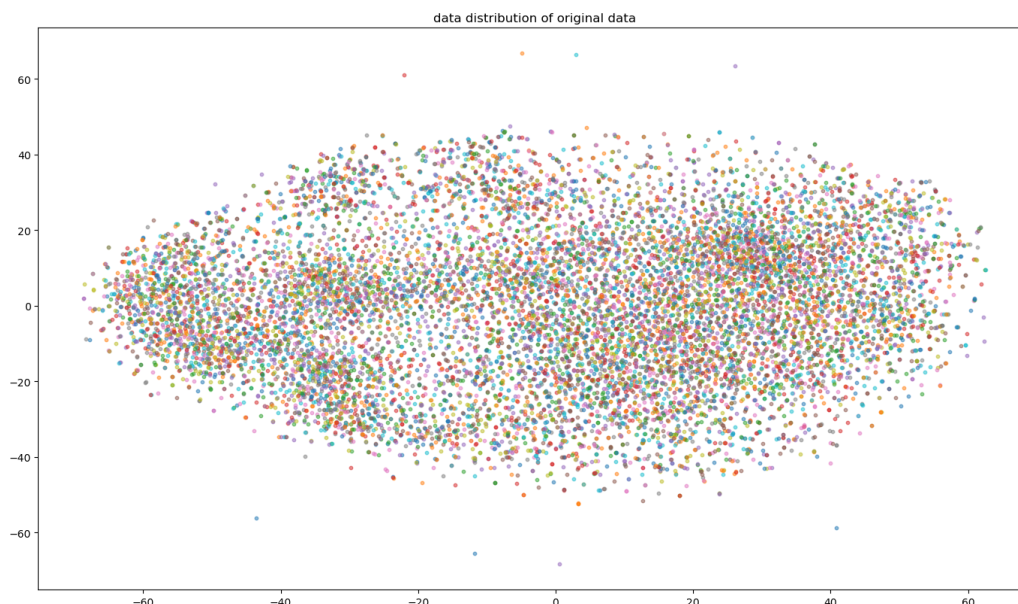


图5-5 data distribution of original data

其次，我们使用PCA，对数据进行降维，我们选择了0.999, 0.99, 0.9作为方差的阈值，对原始数据进行了降维，并使用分类结果较好的SVM和KNN进行分类，两个模型使用的参数均为上述实验中最优超参的参数，其分类结果如下表所示，PCA降维后的数据以及原始数据和PCA降维后数据的对比如下两图所示。可以看出，PCA通过筛选，能成功淘汰部分维度，但是降维后SVM和KNN的准确率均大幅度下降。同时，由TSNE的结果可以看出，PCA并未有效辅助分类，不同种类的样本依旧均匀地分散在空间中。因此可以认为，降维策略在手绘草图分类任务中并不适用。虽然手写数字和手绘草图在某些方面有一定的相似性，但是明显手写数字存在更高的规范性，样本之间的差异性更小，而手绘草图的样本差异性明显更大，分类数目也更多，因此使得PCA降维策略在两个数据集上起到了相反的作用。

更进一步来说，784*1维的特征对于这个分类任务来说是不足以支撑它进行准确分类的，因此在模型中，我们不仅不需要降维，反而还需要对模型进行升维，即通过卷积操作来获取图像中的空间信息，或者额外考虑笔画序列的时序信息。

方差阈值	数据维度	SVM准确率	KNN准确率
0.999	637	26.34%	12.56%
0.99	548	26.37%	11.43%
0.9	308	26.51%	12.34%

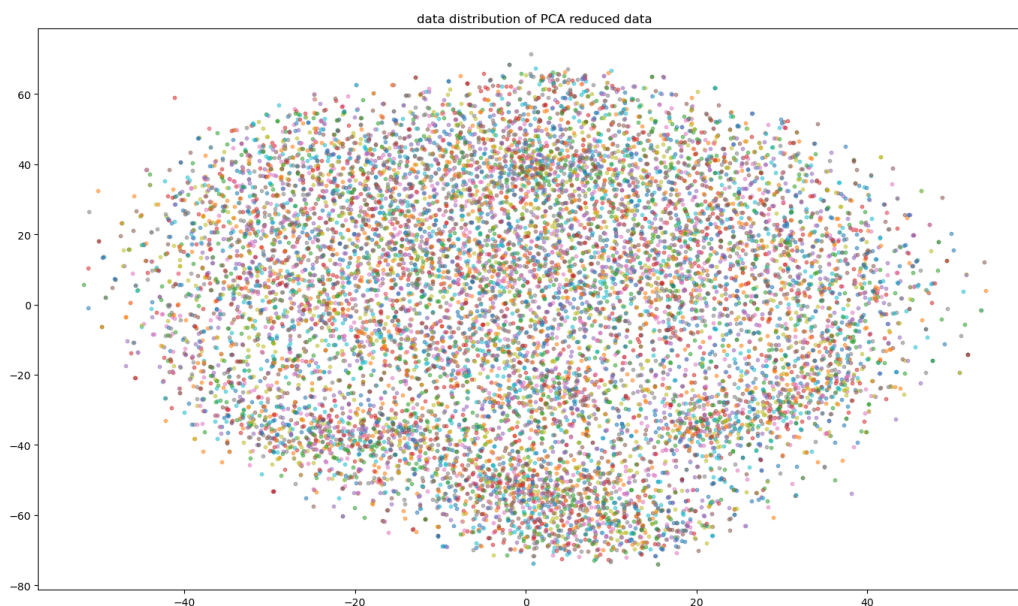


图5-6 data distribution of PCA reduced data

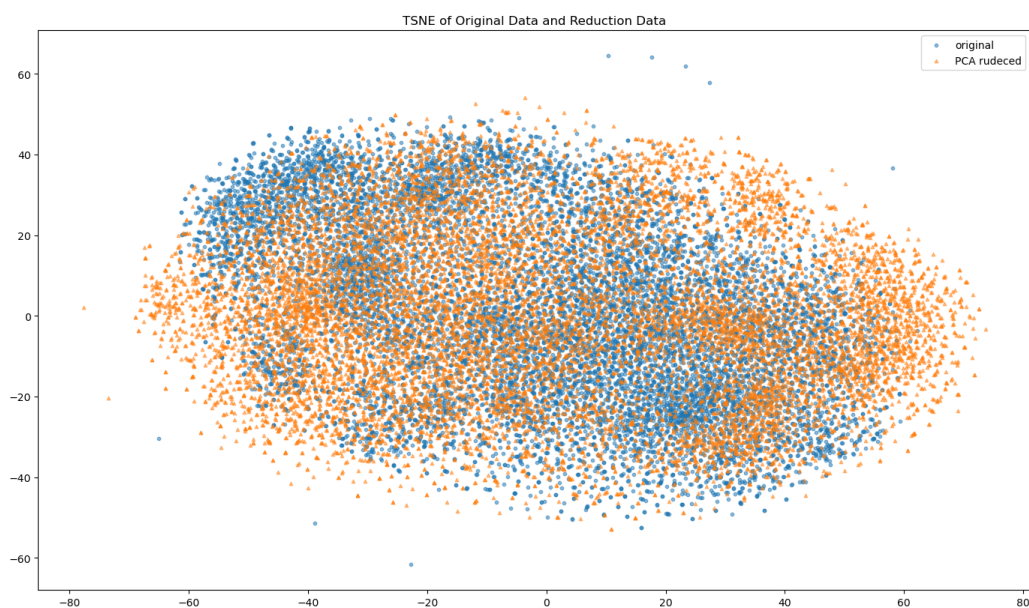


图5-7 TSNE of Original Data and Reduction Data

5.2 深度学习模型

5.2.1 ResNet

在本次实验中，我们基于Torchvision库所提供的ResNet18，ResNet34和ResNet50模型，采用pytorch框架分别构建、训练和测试了数个分类器。以ResNet18为例：

```
from torchvision import models

classifier = models.resnet18(pretrained=False)
classifier.load_state_dict(torch.load('../pretrain_models/resnet18-5c106cde.pth'))
```

由于基于不同初始化参数训练出来的分类器性能会有较大的差别，此处我们预先下载了已完成预训练的模型作为分类器训练的基准，以此保证分类器分类效果的稳定性。此外，直接从模型文件中导入参数使我们不必在训练开始之后再临时联网进行模型下载，从而影响到训练效率。

我们的手绘草图数据集一共包括25个类别，因此，我们需要通过 `classifier.fc = nn.Linear(classifier.fc.in_features, 25)` 来将输出的softmax层调整为对应的形状。在训练和测试的过程中，我们取概率最大的类别作为该分类器对于一张输入图片输出的分类类别。

对于ResNet18，ResNet34和ResNet50网络，我们首先以如下两种方式进行了分类器训练：

- 1. 保持学习率为0.0001，在预训练模型的基础上训练20轮，保存最终的模型参数；
- 2. 保持学习率为0.0001，在预训练模型的基础上训练50轮，保存最终的模型参数。

以20轮训练为例，我们记录了分类器在训练过程中的模型在训练集上的准确率和在评估集上的准确率，结果如下图所示：

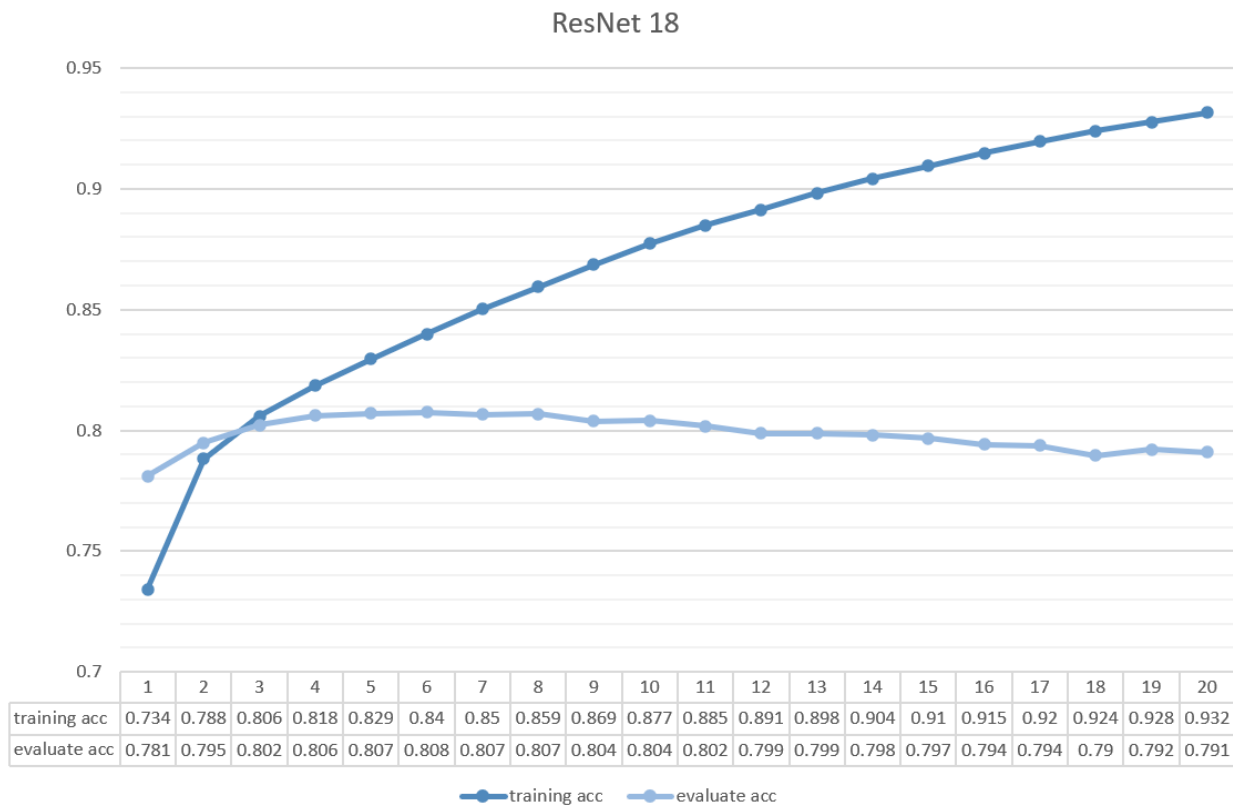


图5-8 ResNet18 Training Accuracy

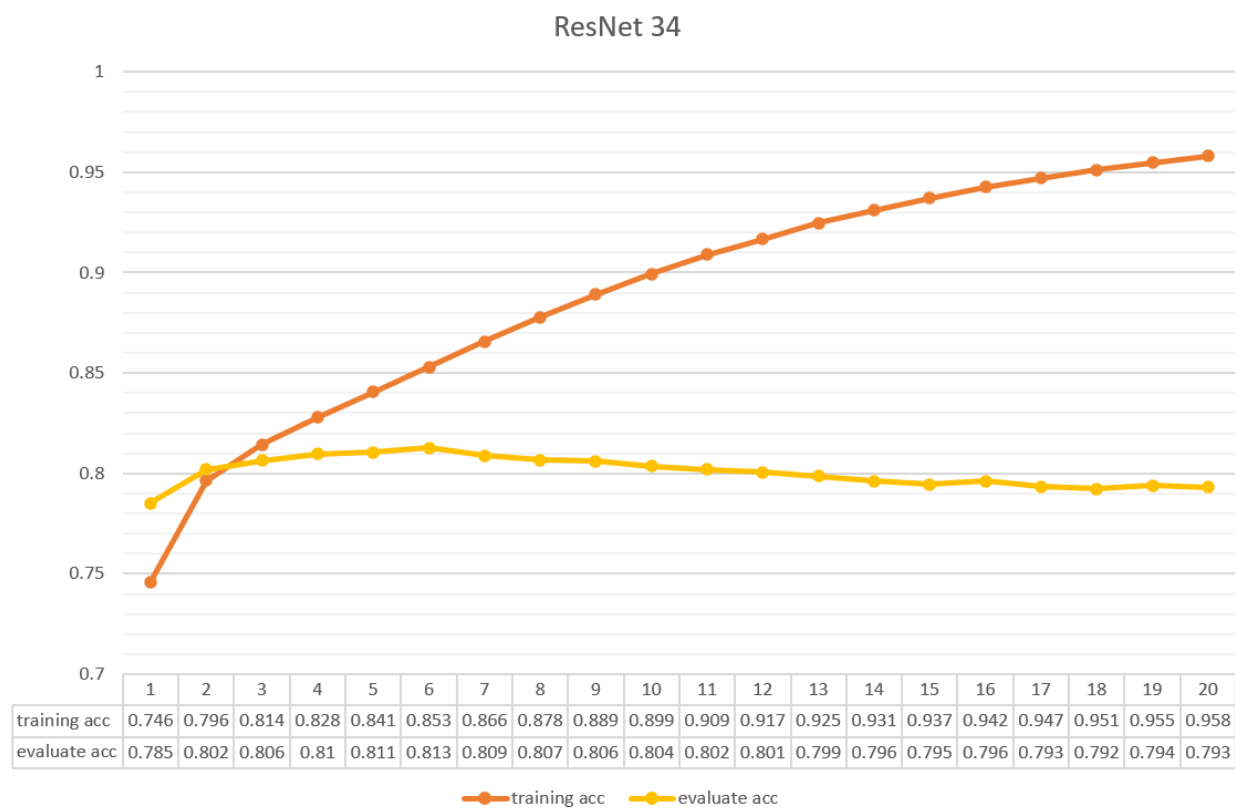


图5-9 ResNet34 Training Accuracy

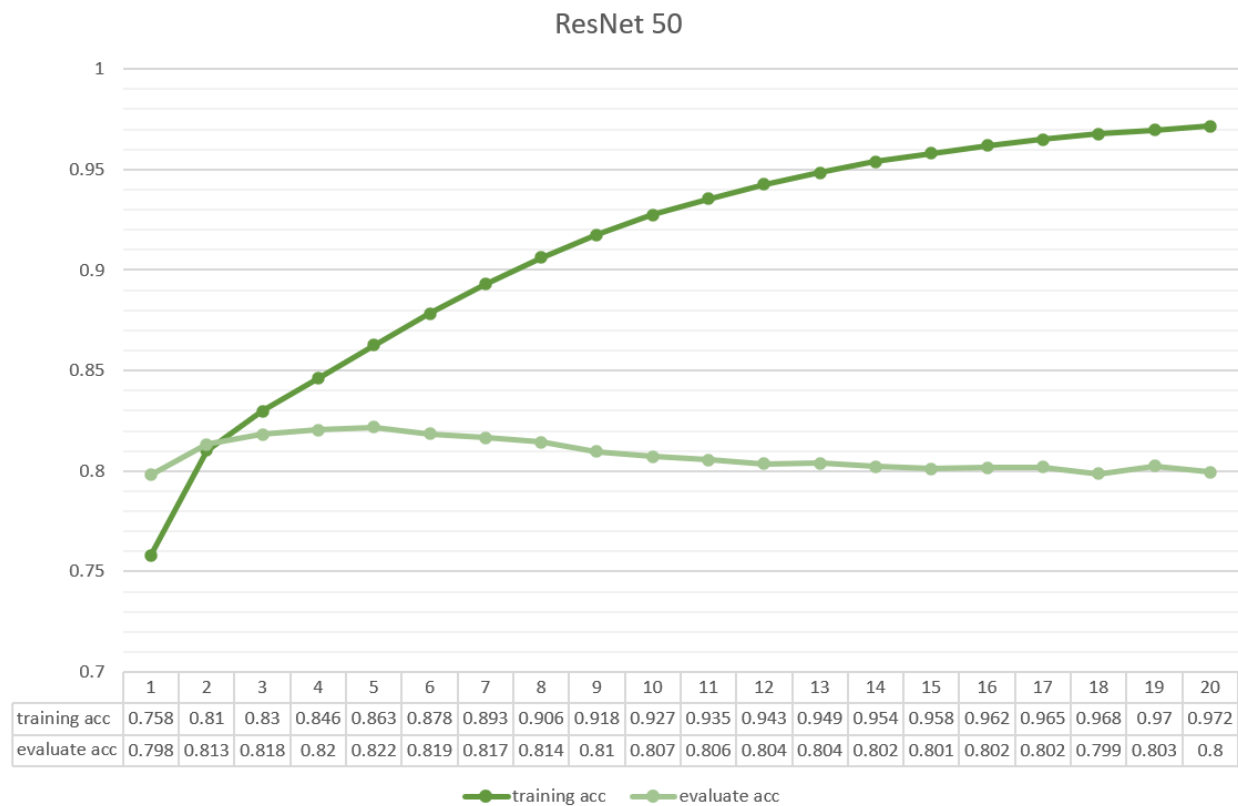


图5-10 ResNet50 Training Accuracy

从图中可以看出，随着训练轮数的增加，上述三个网络在训练集上的分类准确率均会稳定增加，但在第5、6轮后，上述三个网络在评估集上的分类准确率都发生了一定程度的下降，并且没有回升的征兆；在50轮训练的过程中模型分类准确率存在类似的趋势，此处不另外绘图。从图片下方的表格中可以看出，在有限轮数的训练下，层数越深的ResNet网络可以获得更高的训练准确率和评估准确率，但是在评估集上分类的准确率差异比在训练集上要小许多。

我们认为出现上述状况的最主要的原因是图片较小，特征较少，在训练轮数过多后模型在训练集上产生了过拟合。考虑到评估集的参考价值，我们猜测能给出更好的评估集准确率的模型能够更好地对测试集数据进行分类。此外，考虑到学习率被我们固定在一个常用值0.0001上，我们推测分类器在评估集上的准确率下降也和步长较大跳出了局部最优解，结果模型反而收敛困难，loss不降反升有关。我们引入了 [StepLR](#) 来动态调整学习率，先以较大步长快速降低loss，提升收敛速度，再以较低学习率对模型进行微调，以期获得更好的分类性能。

基于上述猜测，我们增加了以下两种分类器训练的方式：

1. 保持学习率为0.0001，保存在评估数据集上获得最高准确率的模型参数；
2. 将初始学习率给定为0.01，每经过5轮的训练将其下调为原学习率的0.1，保存在评估数据集上获得最高准确率的模型参数。

对于如上所述总计12个分类器，我们分别测试了其在测试数据集上的分类准确率，结果如下所示：

	ResNet18	ResNet34	ResNet50
20 Epoches	79.02%	79.42%	79.86%
50 Epoches	78.45%	78.22%	78.35%
Best Eval Acc	78.91%	79.05%	79.03%
Best Eval Acc with StepLR	79.92%	80.19%	79.77%

从表中可以看出，ResNet网络在手绘草图数据集上训练出的分类器在其测试集上的最佳准确率为80.19%。总的来说，采用了学习率衰减手段的分类器具有最好的分类效果，其次是在学习率固定为0.0001的情况下，进行20轮训练的分类器。训练50轮得到的分类器相对于训练20轮得到的分类器准确率肉眼可见地下降了，而直接使用在评估集上获得最高准确率的模型参数来进行分类的效果也不如我们的预想。

对比前文图中给出的数据，需要注意的是，我们在测试集上获得的最终测试准确率其实相对于分类器在第1轮训练时给出的在评估集上分类的准确率并无明显提升。我们怀疑这是因为手绘草图的可解释性相对于真实图片来说较差，即便不同类别的手绘图片之间有着较大的分布差距，使其一开始就能够被较为准确地分类，但实际上图片在训练过程中提供的有用信息极其有限，分类器习得的特征不充分，没有办法大幅度地提升分类性能。此外，ResNet网络在进行第一轮卷积计算时卷积核的大小为7*7，对比我们图片的大小28*28，它下采样的程度偏严重，特征提取效率显而易见是比较糟糕的。这让我们的分类器训练结果带上了一些随机性，很难确定分类效果是否受到了一些偶然因素的影响。

即便如此，考虑到评估集在第5、6轮后的下降趋势，我们认为出现训练轮数过多以至于分类效果变差的情况也绝非纯粹的偶然。在训练集上的过拟合可以很好地解释这种状况，而层数越深的网络发生过拟合的速度更快、更严重，ResNet模型引以为傲的深度优势反而成为了劣势。此外，在评估集上获得最高准确率的模型参数通常是第5轮前后的训练产物，此时虽然模型在评估集上有着较好的表现，但实际上可能并没能很好地收敛，在训练集上的分类效果欠佳也是合理的结果。我们猜测以评估集分类准确率大幅度下跌为节点来存储模型参数能够获得比直接取最优评估集分类准确率获得更好的效果，但是时间有限，我们也没有一个很好的定义“大幅度下跌”的标准，故而没有实行以该方案进行的模型训练和测试。

最后我们可以看出，一个合适的学习率对模型的影响是比较明显的。很显然，以0.01为初始并且在每5轮训练后以0.1的幅度衰减的学习率对于ResNet34模型在手绘草图数据集上的训练是最为合适的，但在ResNet18和ResNet50上并没有突出的效能。相对于训练20轮的分类性能，ResNet18在学习率衰减的场景下略有提升，但ResNet50甚至下降了。如果后续有机会，多尝试几组不同学习率、不同衰减方式和不同模型的组合，我们或许可以给出更优的ResNet网络在手绘草图数据集上的训练参数。但是总的来说，ResNet网络用于该数据集的分类性能上限有限，不够适宜。

5.2.2 VGGNet

在本次实验中，我们使用PyTorch深度学习框架搭建了VGG16和VGG19网络，并在网络基础上进行了一些修改，增加了BN层，并且选择性的去掉了一些池化层。在搭建好网络类后，用 `model = VGG19(n_classes=25)` 即可创建模型。

为了保证实验的可重复性，参数需要固定而不是选择随机初始化。对于实验的设置方面，我们使用Adam作为优化器，mini batch的大小设置为64，初始学习率设置为0.001，并且使用ReduceLROnPlateau scheduler来动态调整学习率，即对平均loss进行监测，若若干次loss均保持较小差异，则降低学习率。

考虑到不同种类的数据分布比较均匀，不存在数据失衡的问题，我们采用准确率(Accuracy)来对模型进行评估。

我们分别对VGG16和VGG19两种不同深度的网络进行了实验，我们记录了10轮训练的模型在验证集上的准确率，实验结果如下：

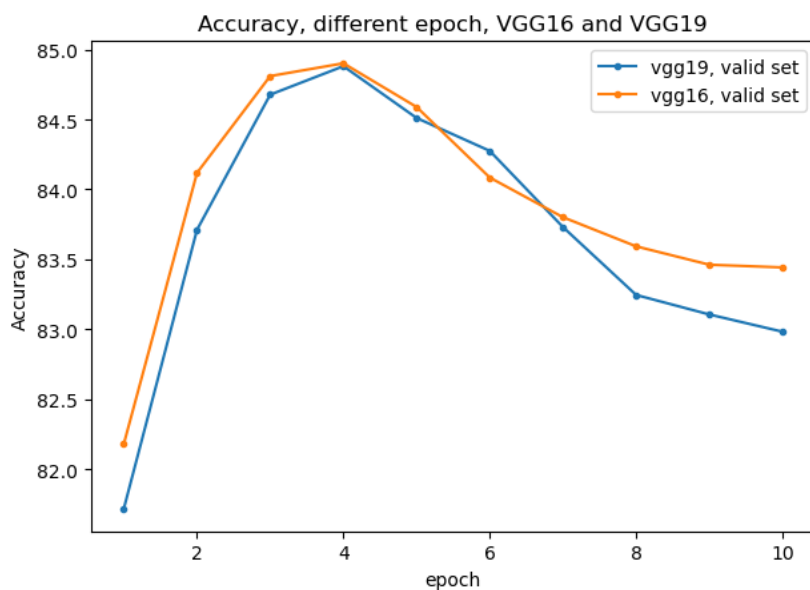


图5-11 VGG16 and VGG19 Training

从图中可以看出，两个不同的网络在验证集上的分类准确率均先上升后下降最后稳定，而训练轮数等于4时，模型具有最好的性能，两个模型的最优性能差距很小。随着epoch的增大，模型显然出现了过拟合的现象。对比两个网络，VGG19具有更大的深度，出现过拟合后在验证集上的准确率也相对VGG16更低，即网络越复杂，越容易过拟合。

在实验过程中，由于只能保留两个池化层，我们也对保留不同池化层的不同模型均做了训练和测试，epoch=10时在验证集上的准确率均在82%左右，差异很小，故我们选择了去掉第1、3和第5个最大池化层的模型。

可以得出，VGG16可能在该草图分类问题上具有更优秀的性能。不同的网络在测试集上的结果如下表：

	VGG16	VGG19
epoch=4	84.90 %	85.06 %
epoch=10	83.53 %	83.13 %

表中可以看出，VGG网络在手绘草图数据集上训练后，模型的分类准确率最高为85.06 %。我们发现训练轮数更大的模型性能相对训练轮数较小的模型有着明显下降，而实验中的loss是一直下降的，这说明网络出现了过拟合。而VGG16和VGG19的最高准确率分别是84.90%和85.06%，差异不大，可以认为具有相同的性能，这也从侧面说明了使用VGG网络深度为16和19时基本处于最佳的表现水平，不必再对VGG13、VGG11进行更多的实验。

总的来说，对草图数据进行适配后的VGG16和VGG19模型在本次项目的草图分类任务中均有良好的表现，其中修改后的VGG19网络模型对草图的25类分类准确率有85.06 %的表现水平。此外，我们尝试将VGG网络的浅层特征下采样后与深层特征进行特征融合，希望将图片数据的位置结构信息和纹理信息综合考量，但实现过程中遇到的问题太多，由于时间原因未能得到较好结果。

6 总结与展望

本项目中涉及的所有模型的最佳结果如下表所示：

模型	训练集大小	准确率
SVM	5000*25	59.74%
PCA-SVM	5000*25	26.51%
KNN	10000*25	45.48%
MLP	10000*25	29.02%
ResNet18	70000*25	79.92%
ResNet34	70000*25	80.19%
ResNet50	70000*25	79.86%
VGG16	70000*25	84.90%
VGG19	70000*25	85.06%

从表中可以看出，传统的分类模型的表现并不理想，分类效果最好的SVM模型需要大量的内存以及长时间的计算，而运行更快的其他模型的准确率都低于50%。分析原因主要在于这些模型的输入是展平的特征，他们无法有效利用像素之间的空间信息进行分类，而对于手绘草图这样图像和纹理信息稀少的图片来说这些模型能够利用的信息就更加稀缺了。从TSNE可视化的结果也可以看出，虽然理论上手绘草图中存在大量留白空间，但PCA降维之后的样本聚集并没有实质性的加强，也能从侧面说明信息的稀缺而不是冗余。因此，我们认为这些传统模型在本项目场景中并不适用。

此外，虽然相对于传统的分类模型有质的飞跃，但ResNet网络的表现也存在难以逾越的天花板。分类效果最好的ResNet34模型对于学习率进行了一定程度上的微调，但相较于预训练模型对于评估集的分类准确率（约为78%），80.19%这个结果并没有显著的提升。我们认为最主要的原因是图片缺乏精确的细节且尺寸过小，ResNet的结构并没有办法精确地提取和学习图片的特征。在这样的前提下，ResNet网络为加深层数而设计的残差单元反而成为了负累，试图提取的特征越为复杂精细，模型在训练集上越容易出现过拟合。因此，我们认为ResNet网络在本项目场景中并不是最优选择。

相比较ResNet，VGG网络的表现有一些提升。由于原图尺寸很小，仅为 28×28 的单通道灰度图像，ResNet的首层卷积核大小为7，显得有些过大了，从而导致丢失掉了很多信息，而VGGNet的 3×3 小卷积核更为合理一些，可以提取到更多结构上的信息，而不是深层网络得到的纹理信息，所以表现相对ResNet更好一些。

总体而言，本项目探索了以以像素图为输入的手绘草图分类任务的多种实现方式，并分析了各个模型的优点和不足。我们认为，想要提高模型的性能，一个可行的探索方向就是结合像素图信息和笔画顺序信息，或者对原始数据进行增强，但由于时间仓促，我们没能进一步探索更多改进的可能性。

我们的成员分工及贡献如下表所示：

姓名	工作内容	贡献率
林宇欣： 220030910004	生成png格式数据；完成ResNet网络在数据集上的训练和测试；撰写报告	1/3
胡屹垚： 220030910016	生成npz格式数据；完成VGG网络在数据集上的训练和测试；撰写报告	1/3
王祎雯： 220030910002	相关文献调研；生成npz格式数据；完成传统方法分类器在数据集上的训练和测试；撰写报告	1/3

我们的源代码可以通过此链接克隆至本地：<https://github.com/linyx1998/CS420-Machine-Learning-Project.git>

7 参考文献

1. Yu Q , Yang Y , Liu F , et al. Sketch-a-Net: A Deep Neural Network that Beats Humans[J]. International Journal of Computer Vision, 2017, 122(3):411-425.
2. Li L , Zou C , Zheng Y , et al. Sketch-R2CNN: An RNN-Rasterization-CNN Architecture for Vector Sketch Recognition[J]. IEEE Transactions on Visualization and Computer Graphics, 2020, PP(99):1-1.
3. Ha D , Eck D . A Neural Representation of Sketch Drawings[C], 2017.
4. He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
5. Simonyan, K., Zisserman, A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
6. Eitz M,Hays J,Alexa M. How do humans sketch objects?[J].ACM Transactions on Graphics,2012,31(4): 44:1-44:10