



# 模式识别及应用

## 课程设计实验报告

- 题 目： 基于 ResNeXt 网络结构的表情包分类器
- 指 导 老 师： 罗 达
- 班 级： 2016 级机器人 4 班
- 姓 名： 吴 俊 鸿
- 学 号： 201641404313
- 提 交 时 间： 2019 年 6 月 25 日
- 课程设计项目地址： <https://github.com/TonyNgcn/meme-classification>



## 目录

|     |                                |    |
|-----|--------------------------------|----|
| 一、  | 问题背景与描述.....                   | 3  |
| 二、  | 解决方法 .....                     | 3  |
| 1.  | 采用的技术.....                     | 3  |
| (1) | 模型引入 .....                     | 3  |
| (2) | ResNeXt 网络结构介绍.....            | 4  |
| 2.  | 预处理.....                       | 6  |
| (1) | 数据集来源.....                     | 6  |
| (2) | 数据预处理（本地） .....                | 7  |
| (3) | 数据预处理（服务器端） .....              | 7  |
| (4) | 模型训练过程 .....                   | 8  |
| (5) | 模型预测 .....                     | 8  |
| 三、  | 实验对比 .....                     | 10 |
| 1.  | ResNeXt 在原版 Cifar-10 数据集上..... | 10 |
| 2.  | ResNeXt 在表情包数据集上 .....         | 11 |
| 3.  | ResNeXt 网络在表情包数据集上的实验结果对比..... | 13 |
| 四、  | 总结 .....                       | 17 |



## 一、 问题背景与描述

当前，我们在网上聊天的时候会使用各种各样的表情包。表情包可以用来简短精悍有趣地表达自己的内心想法，它不仅新潮幽默，更是能激发人的智慧和人的各种语言天赋。弥补了只有文字不能表达的情景，可以让我们的聊天更加有趣。当见识到各种各样的表情包时，我们产生了一个想法，能不能让电脑自动的分类各种各样的表情包呢？这也是我今天课程设计的主题。

## 二、 解决方法

### 1. 采用的技术

#### (1) 模型引入

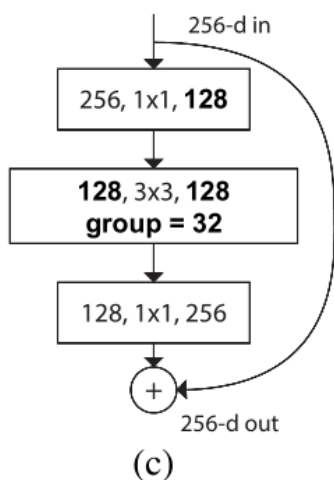
对于表情包分类问题，实际上就是所谓的图片分类问题。

对于图片分类的问题，有很多著名的数据集，比如说 ImageNet、Cifar-10 等。由于 Cifar-10 数据集只有 10 个类别, 60000 张 10 个类别的  $32 \times 32$  的图片, 即每个类别有 6000 张图片。这个 Cifar-10 与我今天要搭建的表情包分类器所用的数据集类型比较接近。因此我选用了可以分类 Cifar-10 的 ResNeXt 网络结构，这个是近年来提出的解决图片分类问题

的深度学习网络模型，在处理 Cifar-10 等数据集上的图片分类问题时都能取得不错的效果。

## (2) ResNeXt 网络结构介绍

ResNeXt 网络结构是由《Aggregated Residual Transformations for Deep Neural Networks》这篇学术论文提出的。在论文中，我们可以知道该网络是通过重复构建块来构建的，该构建块聚合了具有相同拓扑的一组转换。我们的简单设计产生了一个同构的多分支架构，只需要设置一些超参数。下面是一个基本块的模型。



基本块的实现代码如下：

```
1. def residual_layer(self, input_x, out_dim, layer_num, res_block=blo  
   cks):  
2.     for i in range(res_block):  
3.         input_dim = int(np.shape(input_x)[-1])  
4.         if input_dim * 2 == out_dim:  
5.             flag = True  
6.             stride = 2  
7.             channel = input_dim // 2
```



```
8.         else:
9.             flag = False
10.            stride = 1
11.            x = self.split_layer(input_x, stride=stride, layer_name='split_layer_'+layer_num+'_'+str(i))
12.            x = self.transition_layer(x, out_dim=out_dim, scope='transition_layer_'+layer_num+'_'+str(i))
13.            x = self.squeeze_excitation_layer(x, out_dim=out_dim, ratio=reduction_ratio, layer_name='squeeze_layer_'+layer_num+'_'+str(i))

14.            if flag is True :
15.                pad_input_x = Average_pooling(input_x)
16.                pad_input_x = tf.pad(pad_input_x, [[0, 0], [0, 0], [0, 0], [0, 0], [channel, channel]]) # [?, height, width, channel]
17.            else :
18.                pad_input_x = input_x
19.            input_x = Relu(x + pad_input_x)
20.        return input_x
```

而这个网络的构建就是由像上图的基本块构成，原版针对Cifar-10 的 ResNeXt 网络，由 4 个模块构成，输入的卷积层 1，用于处理输入，剩下的 3 个都是基本块，将第一个卷积层的输出用于输入，每一层基本块的输出作为下一个基本块的输入。最后接池化层和全连接层，用于输出最终的类别，相关的实现代码见下面：

```
1. def Build_SEnet(self, input_x):
2.     # only cifar10 architecture
3.     input_x = self.first_layer(input_x, scope='first_layer')
4.     x = self.residual_layer(input_x, out_dim=64, layer_num='1')
5.     x = self.residual_layer(x, out_dim=128, layer_num='2')
6.     x = self.residual_layer(x, out_dim=256, layer_num='3')
7.     x = Global_Average_Pooling(x)
8.     x = flatten(x)
9.     x = Fully_connected(x, layer_name='final_fully_connected')
10.    return x
```



以下的图片选自原始的论文中，介绍了 ResNeXt 网络结构的组成（右边），与之前的陈述是一致的：

| stage     | output  | ResNet-50   | ResNeXt-50 (32×4d)  |
|-----------|---------|---|---|
| conv1     | 112×112 | 7×7, 64, stride 2   | 7×7, 64, stride 2   |
| conv2     | 56×56   | 3×3 max pool, stride 2  | 3×3 max pool, stride 2  |
|           |         | $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$    | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128, C=32 \\ 1 \times 1, 256 \end{bmatrix} \times 3$    |
| conv3     | 28×28   | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$  | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256, C=32 \\ 1 \times 1, 512 \end{bmatrix} \times 4$    |
| conv4     | 14×14   | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512, C=32 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$   |
| conv5     | 7×7     | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 1024 \\ 3 \times 3, 1024, C=32 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$ |
|           | 1×1     | global average pool<br>1000-d fc, softmax   | global average pool<br>1000-d fc, softmax   |
| # params. |         | <b>25.5</b> ×10 <sup>6</sup>  | <b>25.0</b> ×10 <sup>6</sup>  |
| FLOPs     |         | <b>4.1</b> ×10 <sup>9</sup>   | <b>4.2</b> ×10 <sup>9</sup>   |

Table 1. **(Left)** ResNet-50. **(Right)** ResNeXt-50 with a 32×4d template (using the reformulation in Fig. 3(c)). Inside the brackets are the shape of a residual block, and outside the brackets is the number of stacked blocks on a stage. “C=32” suggests grouped convolutions [24] with 32 groups. *The numbers of parameters and FLOPs are similar between these two models.*

## 2. 预处理

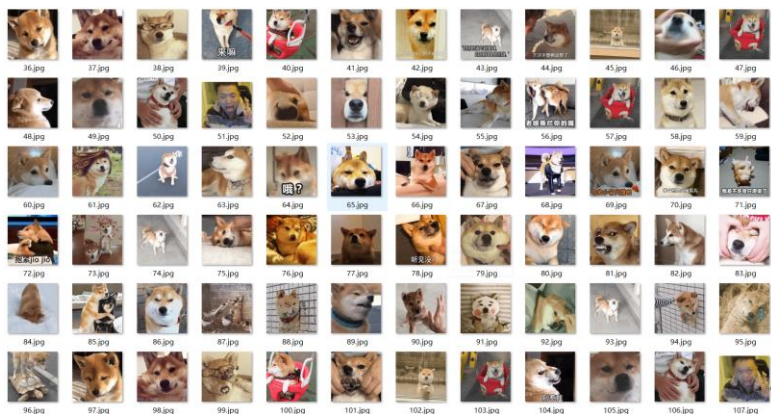
### (1) 数据集来源

表情包的数据集是从搜狗输入法的工具箱搜索关键词进行下载并保存到相关的文件夹内，由于数据量不大而且需要



保证分类的质量，全部的表情包图片都需要经过人工的分类和整理。

如下图是狗的表情包。



## (2) 数据预处理（本地）

数据需要仿照 Cifar-10 数据集的格式进行处理。

对于所有已获得的表情包图片，如果其中有 gif 格式的文件，提取当中的几帧直接保存成 jpg 格式的文件，而对于其他格式的图片，都转换成 jpg 格式的图片。然后把图片大小修改为  $32 \times 32$ 。由于数据量不足，这里把同一张图片进行了旋转操作，一张图片变成了 4 张。重命名图片为“类别\_图片编号”并把全部图片放在同一个文件夹中。压缩文件夹，并上传到服务器，进行后续的训练操作。

## (3) 数据预处理（服务器端）

在模型训练之前，需要进行以下预处理操作：解压刚才上传的文件。对每个图片，用 opencv 读取，变成  $32 \times 32 \times 3$  的



矩阵，并从文件名获得 label。用 sklearn 工具分出训练集和测试集（8:2）。把 label 转换成 onehot 编码。随机打乱顺序。

#### (4) 模型训练过程

整个模型都是在 Colab 进行训练的，由于该模型需要的显存大于 14G，我的笔记本电脑不足以完成训练。训练的方法是先利用上面预处理号的数据集，然后修改从 [https://github.com/taki0112/SENet-Tensorflow/blob/master/SE\\_ResNeXt.py](https://github.com/taki0112/SENet-Tensorflow/blob/master/SE_ResNeXt.py) 下载的 ResNeXt 模型代码。该代码是针对 Cifar-10 数据集制作的，因此需要修改部分代码以适应该表情包数据集。模型的每次训练都需要运行 100 个迭代，其详细的训练过程可以看下面的网址：

[https://colab.research.google.com/drive/1YfkNHLj9KDEy-\\_cUMDQtn\\_4NCkeDb2tQ](https://colab.research.google.com/drive/1YfkNHLj9KDEy-_cUMDQtn_4NCkeDb2tQ)

#### (5) 模型预测

训练好得到的模型，可以修改部分代码以用于预测单张图片。得到的结果是一个 prediction，它可以给出需要预测的图片经过模型之后的得分。取最高分的类别作为最终的类别输出就可以了。





如下图，这个表情包分类的结果是 1 类别分数最高，所以这个表情包就是“蘑菇头”的类别。



- 00-小猪佩奇-130
- 01-蘑菇头-146
- 02-可爱柴犬-152
- 03-猥琐猫-136
- 04-猫-134
- 05-笑-121
- 06-熊本熊-151
- 07-熊猫头-175
- 08-文字-120
- 09-长草颜-168

| 0         | 1          | 2         | 3          | 4         | 5         | 6           | 7          | 8         | 9         |
|-----------|------------|-----------|------------|-----------|-----------|-------------|------------|-----------|-----------|
| -4.292079 | 12.9024725 | -1.966367 | 0.36090595 | 1.1618128 | 4.3680816 | -0.45070356 | 0.45959228 | -8.160566 | -8.158982 |

但是，这个也会有预测错误的情况出现，比如说下面的这种情况，应该输出的是“熊本熊”类别，却错误地分类成“蘑菇头”。

```
saver.restore(sess, ckpt.model_checkpoint_path)
print("Loading success")
else:
    print("No checkpoint")
prediction = sess.run(logits, feed_dict=x_feed_dict)
max_index = np.argmax(prediction)
result = prediction_dict[max_index]
print(result)
img = cv.imread(predict_imgPath)
fig = plt.figure()
plt.imshow(img)
fig.show()
with tf.Session(config=tf.ConfigProto(device_list=[tf.ConfigProto.GPU])) as sess:
    predict
2019-06-25 20:28:48.816128: I c:\users\user\source\repos\tensorflow\tensorflow\core\common_runtime\gpu\gpu_device.cc:1404] Adding visible gpu devices: 0
2019-06-25 20:28:48.816128: I c:\users\user\source\repos\tensorflow\tensorflow\core\common_runtime\gpu\gpu_device.cc:965] Device interconnect StreamExecutor
2019-06-25 20:28:48.816435: I c:\users\user\source\repos\tensorflow\tensorflow\core\common_runtime\gpu\gpu_device.cc:971] 0
2019-06-25 20:28:48.816648: I c:\users\user\source\repos\tensorflow\tensorflow\core\common_runtime\gpu\gpu_device.cc:984] 0: N
2019-06-25 20:28:48.816990: I c:\users\user\source\repos\tensorflow\tensorflow\core\common_runtime\gpu\gpu_device.cc:1097] Created TensorFlow device (/job:loc
Loading success
蘑菇头
```





### 三、 实验对比

#### 1. ResNeXt 在原版 Cifar-10 数据集上

对于这个 ResNeXt 网络结构，实验主要是做了对比。首先是对原版 ResNeXt 网络在 Cifar-10 下的训练结果进行测试。

经过 100 次迭代，ResNeXt 网络结构在 cifar-10 训练集上的准确率为 100%，测试集为 94.57%（如下图）

```
epoch: 96/100, train_loss: 0.0060, train_acc: 0.9999, test_loss: 0.1792, test_acc: 0.9459
epoch: 97/100, train_loss: 0.0060, train_acc: 0.9999, test_loss: 0.1797, test_acc: 0.9457
epoch: 98/100, train_loss: 0.0061, train_acc: 1.0000, test_loss: 0.1795, test_acc: 0.9455
epoch: 99/100, train_loss: 0.0060, train_acc: 1.0000, test_loss: 0.1792, test_acc: 0.9455
epoch: 100/100, train_loss: 0.0059, train_acc: 1.0000, test_loss: 0.1790, test_acc: 0.9457
```



## 2. ResNeXt 在表情包数据集上

接下来是我自己的表情包分类器，对于网络部分的修改，都是以上述介绍部分的 ResNeXt 网络结构的基本块为单位。

### (1) 未经修改的 ResNeXt (v1.3)

第 1.3 版本，网络结构与 ResNeXt 一样，只是修改部分代码适应该表情包数据集。

```
def Build_SEnet(self, input_x):  
    # only cifar10 architecture  
  
    input_x = self.first_layer(input_x, scope='first_layer')  
  
    x = self.residual_layer(input_x, out_dim=64, layer_num='1')  
    x = self.residual_layer(x, out_dim=128, layer_num='2')  
    x = self.residual_layer(x, out_dim=256, layer_num='3')  
  
    x = Global_Average_Pooling(x)  
    x = flatten(x)  
  
    x = Fully_connected(x, layer_name='final_fully_connected')  
    return x
```

经过 100 次迭代的训练后，得出了训练集上的准确率是 100%，测试集上的准确率是 97.08%（如下图）

```
epoch: 96/100, train_loss: 0.0032, train_acc: 1.0000, test_loss: 0.0759, test_acc: 0.9825  
epoch: 97/100, train_loss: 0.0033, train_acc: 1.0000, test_loss: 0.0764, test_acc: 0.9833  
epoch: 98/100, train_loss: 0.0033, train_acc: 1.0000, test_loss: 0.0756, test_acc: 0.9825  
epoch: 99/100, train_loss: 0.0034, train_acc: 1.0000, test_loss: 0.0763, test_acc: 0.9825  
epoch: 100/100, train_loss: 0.0034, train_acc: 1.0000, test_loss: 0.0760, test_acc: 0.9798
```



## (2) ResNeXt 增加一个基本块 (v1.4)

第 1.4 版本，网络结构在 ResNeXt 的基础上 (1.3 版本)，增加了一个 residual\_layer (基本块)。

```
def Build_SEnet(self, input_x):  
    # only cifar10 architecture  
  
    input_x = self.first_layer(input_x, scope='first_layer')  
  
    x = self.residual_layer(input_x, out_dim=64, layer_num='1')  
    x = self.residual_layer(x, out_dim=128, layer_num='2')  
    x = self.residual_layer(x, out_dim=256, layer_num='3')  
    x = self.residual_layer(x, out_dim=512, layer_num='4')  
  
    x = Global_Average_Pooling(x)  
    x = flatten(x)  
  
    x = Fully_connected(x, layer_name='final_fully_connected')  
    return x
```

经过 100 次迭代的训练后，得出了训练集上的准确率是 100%，测试集上的准确率是 98.77% (如下图)

```
epoch: 96/100, train_loss: 0.0023, train_acc: 1.0000, test_loss: 0.0681, test_acc: 0.9877  
epoch: 97/100, train_loss: 0.0025, train_acc: 1.0000, test_loss: 0.0673, test_acc: 0.9877  
epoch: 98/100, train_loss: 0.0025, train_acc: 1.0000, test_loss: 0.0681, test_acc: 0.9877  
epoch: 99/100, train_loss: 0.0026, train_acc: 1.0000, test_loss: 0.0682, test_acc: 0.9877  
epoch: 100/100, train_loss: 0.0024, train_acc: 1.0000, test_loss: 0.0659, test_acc: 0.9877
```



### (3) ResNeXt 减少一个基本块 (v1.5)

第 1.5 版本，网络结构在 ResNeXt 的基础上 (1.3 版本)，减少了一个 residual\_layer (基本块)。

```
def Build_SEnet(self, input_x):  
    # only cifar10 architecture  
  
    input_x = self.first_layer(input_x, scope='first_layer')  
  
    x = self.residual_layer(input_x, out_dim=64, layer_num='1')  
    x = self.residual_layer(x, out_dim=128, layer_num='2')  
  
    x = Global_Average_Pooling(x)  
    x = flatten(x)  
  
    x = Fully_connected(x, layer_name='final_fully_connected')  
    return x
```

经过 100 次迭代的训练后，得出了训练集上的准确率是 100%，测试集上的准确率是 98.07% (如下图)

```
epoch: 96/100, train_loss: 0.0048, train_acc: 1.0000, test_loss: 0.0600, test_acc: 0.9807  
epoch: 97/100, train_loss: 0.0046, train_acc: 1.0000, test_loss: 0.0602, test_acc: 0.9807  
epoch: 98/100, train_loss: 0.0043, train_acc: 1.0000, test_loss: 0.0607, test_acc: 0.9807  
epoch: 99/100, train_loss: 0.0048, train_acc: 1.0000, test_loss: 0.0601, test_acc: 0.9807  
epoch: 100/100, train_loss: 0.0046, train_acc: 1.0000, test_loss: 0.0611, test_acc: 0.9807
```

### 3. ResNeXt 网络在表情包数据集上的实验结果对比

把上述训练过程的模型损失和模型准确率进行整理后得到下列表格：

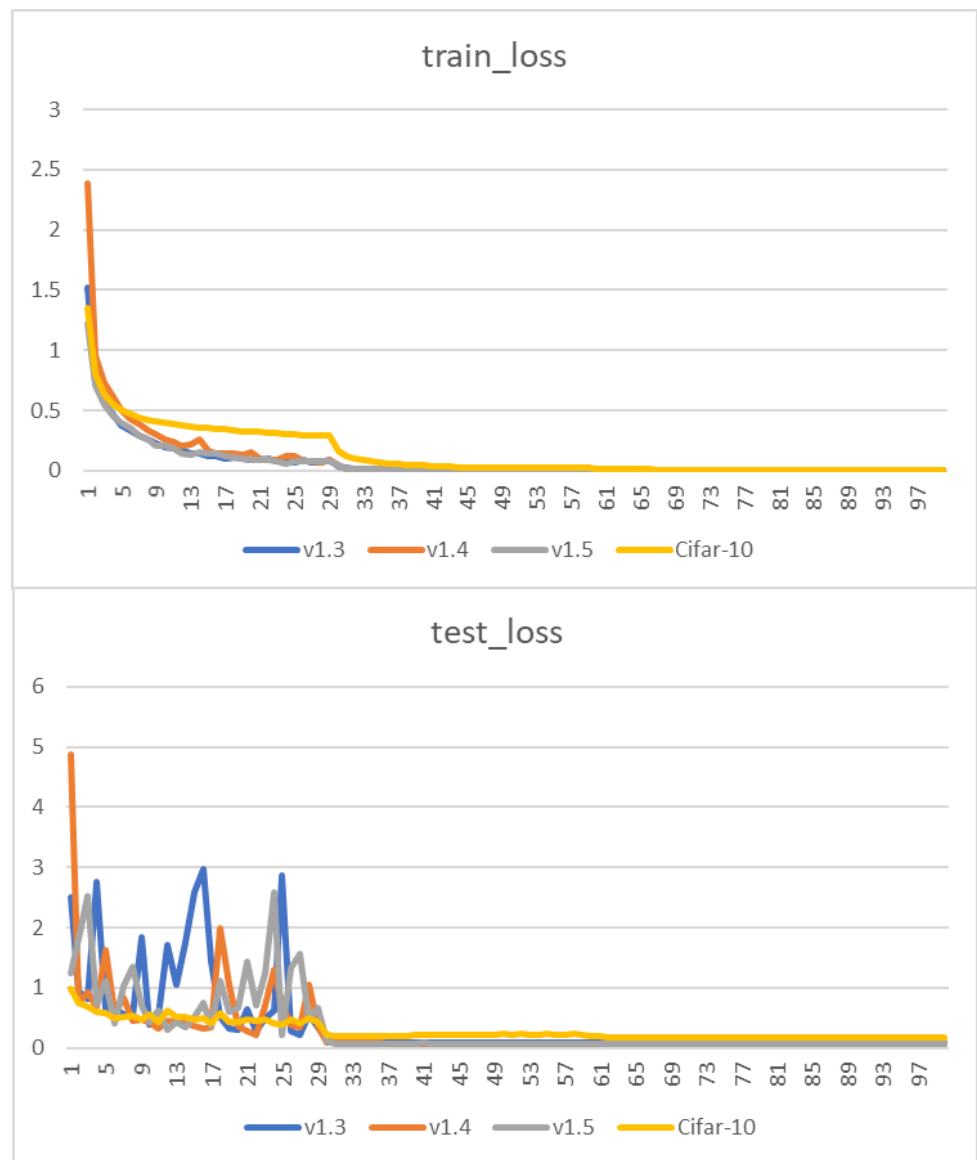


| epoch | v1.3<br>train_loss | v1.3<br>train_acc | v1.3<br>test_loss | v1.3<br>test_acc | v1.4<br>train_loss | v1.4<br>train_acc | v1.4<br>test_loss | v1.4<br>test_acc | v1.5<br>train_loss | v1.5<br>train_acc | v1.5<br>test_loss | v1.5<br>test_acc |
|-------|--------------------|-------------------|-------------------|------------------|--------------------|-------------------|-------------------|------------------|--------------------|-------------------|-------------------|------------------|
| 1     | 1.5237             | 0.5992            | 2.5079            | 0.4474           | 2.3858             | 0.5124            | 4.8741            | 0.5281           | 1.2263             | 0.6386            | 1.2319            | 0.5851           |
| 2     | 0.7301             | 0.7969            | 0.9444            | 0.6298           | 0.954              | 0.7246            | 0.7517            | 0.7246           | 0.713              | 0.795             | 1.8093            | 0.5675           |
| 3     | 0.5545             | 0.856             | 0.8066            | 0.7211           | 0.726              | 0.7929            | 0.9107            | 0.7386           | 0.5484             | 0.8552            | 2.5288            | 0.5474           |
| 4     | 0.4585             | 0.889             | 2.7603            | 0.6351           | 0.6091             | 0.8394            | 0.6927            | 0.807            | 0.4564             | 0.8838            | 0.7317            | 0.7491           |
| 5     | 0.3814             | 0.9119            | 0.6498            | 0.7798           | 0.5034             | 0.8749            | 1.6319            | 0.7982           | 0.3989             | 0.9044            | 1.1154            | 0.6649           |
| 6     | 0.3392             | 0.933             | 0.6694            | 0.764            | 0.4252             | 0.9072            | 0.568             | 0.7868           | 0.358              | 0.9209            | 0.4168            | 0.8553           |
| 7     | 0.2879             | 0.9489            | 0.5575            | 0.8123           | 0.3835             | 0.9188            | 0.8279            | 0.736            | 0.2965             | 0.9367            | 1.02              | 0.7228           |
| 8     | 0.256              | 0.957             | 0.526             | 0.8114           | 0.3374             | 0.9347            | 0.4425            | 0.8623           | 0.2547             | 0.9517            | 1.3517            | 0.7167           |
| 9     | 0.2325             | 0.9624            | 1.8418            | 0.6605           | 0.2989             | 0.949             | 0.4798            | 0.8351           | 0.2112             | 0.9674            | 0.704             | 0.7807           |
| 10    | 0.1993             | 0.9723            | 0.3894            | 0.8684           | 0.2607             | 0.9656            | 0.425             | 0.843            | 0.2036             | 0.9744            | 0.4269            | 0.8711           |
| 11    | 0.1841             | 0.9831            | 0.5343            | 0.8333           | 0.238              | 0.9691            | 0.3194            | 0.8912           | 0.1874             | 0.9794            | 0.619             | 0.8211           |
| 12    | 0.1731             | 0.9796            | 1.7052            | 0.7386           | 0.2052             | 0.9758            | 0.4427            | 0.8491           | 0.1421             | 0.9868            | 0.2917            | 0.8921           |
| 13    | 0.1456             | 0.9882            | 1.0554            | 0.75             | 0.2116             | 0.9775            | 0.4185            | 0.8605           | 0.1365             | 0.9907            | 0.4357            | 0.8491           |
| 14    | 0.1394             | 0.9887            | 1.7567            | 0.7719           | 0.2594             | 0.9703            | 0.3965            | 0.8579           | 0.1528             | 0.9891            | 0.3345            | 0.8728           |
| 15    | 0.1197             | 0.9922            | 2.593             | 0.6912           | 0.1676             | 0.9868            | 0.3627            | 0.8675           | 0.1423             | 0.9898            | 0.5242            | 0.8158           |
| 16    | 0.1217             | 0.9932            | 2.9808            | 0.686            | 0.1421             | 0.9896            | 0.3264            | 0.8939           | 0.1423             | 0.9918            | 0.757             | 0.7737           |
| 17    | 0.1022             | 0.9941            | 1.4257            | 0.7632           | 0.1449             | 0.9902            | 0.335             | 0.8921           | 0.1246             | 0.9909            | 0.3734            | 0.8825           |
| 18    | 0.114              | 0.9937            | 0.5178            | 0.8289           | 0.1456             | 0.9931            | 1.9846            | 0.6711           | 0.1142             | 0.9944            | 1.1182            | 0.7553           |
| 19    | 0.0943             | 0.9973            | 0.3266            | 0.9044           | 0.1281             | 0.9919            | 1.0586            | 0.7711           | 0.0972             | 0.9957            | 0.5999            | 0.8456           |
| 20    | 0.0913             | 0.9976            | 0.3009            | 0.8965           | 0.1522             | 0.9918            | 0.3717            | 0.8737           | 0.0849             | 0.9967            | 0.6587            | 0.8316           |
| 21    | 0.0897             | 0.9972            | 0.6519            | 0.8289           | 0.1013             | 0.9963            | 0.2734            | 0.9018           | 0.0886             | 0.9978            | 1.4372            | 0.7474           |
| 22    | 0.0944             | 0.9954            | 0.2792            | 0.9132           | 0.0859             | 0.9976            | 0.2188            | 0.9307           | 0.0906             | 0.9974            | 0.7048            | 0.7991           |
| 23    | 0.0757             | 0.9976            | 0.488             | 0.8649           | 0.0868             | 0.9976            | 0.6966            | 0.8605           | 0.0814             | 0.998             | 1.2382            | 0.7737           |
| 24    | 0.0911             | 0.9965            | 0.6155            | 0.8237           | 0.1207             | 0.9947            | 1.3025            | 0.8026           | 0.0588             | 0.9991            | 2.5782            | 0.6965           |
| 25    | 0.0724             | 0.9993            | 2.8552            | 0.7044           | 0.1235             | 0.9935            | 0.7369            | 0.7982           | 0.0732             | 0.9985            | 0.2202            | 0.9325           |
| 26    | 0.0849             | 0.9976            | 0.287             | 0.9132           | 0.083              | 0.997             | 0.3892            | 0.8877           | 0.0829             | 0.997             | 1.3355            | 0.7632           |
| 27    | 0.063              | 0.9993            | 0.2192            | 0.9412           | 0.0811             | 0.9974            | 0.3403            | 0.8904           | 0.0808             | 0.9983            | 1.5686            | 0.8053           |
| 28    | 0.0699             | 0.9983            | 0.5575            | 0.843            | 0.0689             | 0.9987            | 1.0416            | 0.8351           | 0.0728             | 0.9983            | 0.4676            | 0.8711           |
| 29    | 0.0872             | 0.9967            | 0.3426            | 0.893            | 0.0832             | 0.9976            | 0.3667            | 0.9035           | 0.0737             | 0.9976            | 0.6669            | 0.8298           |
| 30    | 0.0382             | 0.9928            | 0.1254            | 0.9623           | 0.0461             | 0.9883            | 0.0959            | 0.9675           | 0.0494             | 0.9905            | 0.1076            | 0.9649           |
| 31    | 0.0208             | 0.9974            | 0.108             | 0.9711           | 0.0178             | 0.9978            | 0.0862            | 0.9737           | 0.0185             | 0.9976            | 0.0688            | 0.9816           |
| 32    | 0.0152             | 0.998             | 0.1001            | 0.9728           | 0.0143             | 0.9985            | 0.0831            | 0.9781           | 0.0143             | 0.9996            | 0.0707            | 0.9825           |
| 33    | 0.0108             | 0.9996            | 0.0985            | 0.9737           | 0.0109             | 0.9996            | 0.0811            | 0.9816           | 0.0116             | 0.9998            | 0.0658            | 0.9798           |
| 34    | 0.0102             | 0.9996            | 0.0937            | 0.9728           | 0.0103             | 0.9991            | 0.0763            | 0.9798           | 0.0102             | 0.9998            | 0.0691            | 0.9798           |
| 35    | 0.0089             | 0.9996            | 0.09              | 0.9737           | 0.0092             | 0.9993            | 0.0756            | 0.9825           | 0.01               | 0.9993            | 0.0619            | 0.9825           |
| 36    | 0.0081             | 0.9998            | 0.0906            | 0.9772           | 0.0079             | 0.9996            | 0.0757            | 0.9789           | 0.0091             | 1                 | 0.0723            | 0.9798           |
| 37    | 0.007              | 1                 | 0.0896            | 0.9772           | 0.0083             | 0.9998            | 0.0737            | 0.9807           | 0.0089             | 0.9993            | 0.0694            | 0.9807           |
| 38    | 0.0072             | 0.9996            | 0.0909            | 0.9789           | 0.0066             | 0.9998            | 0.0733            | 0.9833           | 0.0084             | 0.9998            | 0.0683            | 0.9816           |
| 39    | 0.0061             | 0.9996            | 0.089             | 0.9763           | 0.0064             | 0.9998            | 0.0715            | 0.9833           | 0.0079             | 1                 | 0.0709            | 0.9816           |
| 40    | 0.0061             | 1                 | 0.0867            | 0.9763           | 0.0059             | 0.9998            | 0.0731            | 0.9851           | 0.007              | 1                 | 0.0646            | 0.9816           |
| 41    | 0.0065             | 0.9996            | 0.0868            | 0.9781           | 0.0054             | 1                 | 0.0732            | 0.9851           | 0.0075             | 1                 | 0.0788            | 0.9798           |
| 42    | 0.0058             | 0.9998            | 0.0851            | 0.9754           | 0.0048             | 1                 | 0.0726            | 0.9851           | 0.0072             | 1                 | 0.0624            | 0.9816           |
| 43    | 0.0058             | 0.9996            | 0.0833            | 0.9816           | 0.0046             | 0.9998            | 0.0727            | 0.986            | 0.0068             | 1                 | 0.071             | 0.9789           |
| 44    | 0.0049             | 1                 | 0.0839            | 0.9781           | 0.0045             | 1                 | 0.072             | 0.986            | 0.0062             | 1                 | 0.0706            | 0.9798           |
| 45    | 0.0047             | 1                 | 0.0838            | 0.9789           | 0.0041             | 1                 | 0.0707            | 0.986            | 0.0064             | 0.9998            | 0.0709            | 0.9798           |
| 46    | 0.0049             | 1                 | 0.0829            | 0.9789           | 0.0043             | 0.9998            | 0.0676            | 0.9868           | 0.0068             | 0.9998            | 0.0681            | 0.9807           |
| 47    | 0.0043             | 1                 | 0.0854            | 0.9789           | 0.0039             | 1                 | 0.0699            | 0.9851           | 0.0065             | 0.9998            | 0.0679            | 0.9807           |
| 48    | 0.0045             | 1                 | 0.0836            | 0.9816           | 0.0036             | 1                 | 0.0699            | 0.9868           | 0.0058             | 1                 | 0.0681            | 0.9816           |
| 49    | 0.0043             | 1                 | 0.083             | 0.9807           | 0.0035             | 1                 | 0.0658            | 0.9868           | 0.0061             | 0.9998            | 0.0744            | 0.9789           |
| 50    | 0.0042             | 1                 | 0.0809            | 0.9816           | 0.0039             | 0.9996            | 0.0727            | 0.9851           | 0.0058             | 1                 | 0.0658            | 0.9798           |
| 51    | 0.0042             | 1                 | 0.0794            | 0.9789           | 0.0035             | 1                 | 0.0692            | 0.9868           | 0.0061             | 1                 | 0.0746            | 0.9789           |
| 52    | 0.0035             | 1                 | 0.0806            | 0.9807           | 0.0036             | 1                 | 0.0678            | 0.9868           | 0.0053             | 1                 | 0.0624            | 0.9825           |
| 53    | 0.004              | 1                 | 0.0799            | 0.9798           | 0.0028             | 1                 | 0.0702            | 0.9868           | 0.0058             | 1                 | 0.0638            | 0.9807           |
| 54    | 0.004              | 1                 | 0.0808            | 0.9789           | 0.0035             | 1                 | 0.07              | 0.9833           | 0.0049             | 1                 | 0.0624            | 0.9807           |
| 55    | 0.0039             | 1                 | 0.0784            | 0.9798           | 0.0032             | 1                 | 0.0707            | 0.9886           | 0.0054             | 1                 | 0.0693            | 0.9825           |
| 56    | 0.0039             | 1                 | 0.0786            | 0.9816           | 0.0031             | 1                 | 0.0693            | 0.9868           | 0.005              | 1                 | 0.0672            | 0.9807           |
| 57    | 0.004              | 1                 | 0.0786            | 0.9807           | 0.0032             | 1                 | 0.0712            | 0.9877           | 0.0051             | 1                 | 0.0726            | 0.9789           |
| 58    | 0.0034             | 1                 | 0.0781            | 0.9798           | 0.0029             | 1                 | 0.069             | 0.9868           | 0.0056             | 1                 | 0.0674            | 0.9798           |
| 59    | 0.0037             | 1                 | 0.0757            | 0.9781           | 0.0032             | 1                 | 0.0701            | 0.9877           | 0.0051             | 1                 | 0.0701            | 0.9789           |
| 60    | 0.0036             | 1                 | 0.0769            | 0.9781           | 0.0025             | 1                 | 0.0674            | 0.9868           | 0.005              | 1                 | 0.0602            | 0.9807           |
| 61    | 0.0036             | 1                 | 0.0758            | 0.9789           | 0.0032             | 0.9996            | 0.0693            | 0.9877           | 0.005              | 1                 | 0.0621            | 0.9807           |
| 62    | 0.0034             | 1                 | 0.0773            | 0.9781           | 0.0026             | 1                 | 0.0688            | 0.9877           | 0.0047             | 1                 | 0.0612            | 0.9798           |
| 63    | 0.0035             | 1                 | 0.077             | 0.9798           | 0.0027             | 1                 | 0.0689            | 0.9877           | 0.0048             | 1                 | 0.0609            | 0.9807           |
| 64    | 0.0035             | 0.9998            | 0.077             | 0.9772           | 0.0026             | 1                 | 0.0672            | 0.9877           | 0.0051             | 0.9998            | 0.0622            | 0.9807           |
| 65    | 0.0032             | 1                 | 0.0766            | 0.9798           | 0.0026             | 1                 | 0.0674            | 0.9877           | 0.0048             | 1                 | 0.063             | 0.9807           |
| 66    | 0.0035             | 1                 | 0.0757            | 0.9798           | 0.0026             | 1                 | 0.0656            | 0.9877           | 0.0048             | 1                 | 0.0628            | 0.9807           |
| 67    | 0.0038             | 1                 | 0.076             | 0.9798           | 0.0032             | 1                 | 0.0669            | 0.9877           | 0.0047             | 1                 | 0.0614            | 0.9807           |
| 68    | 0.0033             | 1                 | 0.0765            | 0.9798           | 0.0026             | 1                 | 0.0672            | 0.9877           | 0.005              | 1                 | 0.062             | 0.9807           |
| 69    | 0.0037             | 0.9998            | 0.0765            | 0.9789           | 0.0026             | 1                 | 0.0669            | 0.9877           | 0.0045             | 1                 | 0.0631            | 0.9807           |
| 70    | 0.0035             | 1                 | 0.0758            | 0.9798           | 0.0024             | 1                 | 0.0678            | 0.9877           | 0.005              | 1                 | 0.0637            | 0.9807           |
| 71    | 0.0033             | 1                 | 0.0773            | 0.9807           | 0.0023             | 1                 | 0.0665            | 0.9877           | 0.0049             | 0.9998            | 0.0614            | 0.9807           |
| 72    | 0.0037             | 1                 | 0.076             | 0.9798           | 0.0027             | 0.9998            | 0.0677            | 0.9877           | 0.005              | 1                 | 0.0626            | 0.9807           |
| 73    | 0.0038             | 0.9998            | 0.0766            | 0.9816           | 0.0025             | 1                 | 0.0668            | 0.9877           | 0.005              | 1                 | 0.061             | 0.9807           |
| 74    | 0.0033             | 1                 | 0.0762            | 0.9789           | 0.0029             | 0.9998            | 0.0666            | 0.9877           | 0.0052             | 1                 | 0.059             | 0.9807           |
| 75    | 0.0037             | 1                 | 0.0762            | 0.9807           | 0.0026             | 1                 | 0.067             | 0.9877           | 0.0045             | 1                 | 0.0612            | 0.9807           |
| 76    | 0.0035             | 1                 | 0.0769            | 0.9816           | 0.0029             | 1                 | 0.0665            | 0.9877           | 0.0048             | 1                 | 0.0607            | 0.9807           |
| 77    | 0.0035             | 1                 | 0.076             | 0.9807           | 0.0023             | 1                 | 0.066             | 0.9877           | 0.0049             | 1                 | 0.0593            | 0.9807           |
| 78    | 0.0033             | 0.9998            | 0.0755            | 0.9807           | 0.0023             | 1                 | 0.0679            | 0.9877           | 0.0045             | 1                 | 0.0617            | 0.9807           |
| 79    | 0.0037             | 1                 | 0.0759            | 0.9798           | 0.0025             | 1                 | 0.0668            | 0.9877           | 0.0048             | 1                 | 0.0615            | 0.9807           |
| 80    | 0.0035             | 1                 | 0.077             | 0.9807           | 0.0023             | 1                 | 0.0672            | 0.9877           | 0.0044             | 1                 | 0.0607            | 0.9807           |
| 81    | 0.0033             | 1                 | 0.0757            | 0.9816           | 0.0027             | 1                 | 0.0676            | 0.9877           | 0.0046             | 1                 | 0.0597            | 0.9807           |
| 82    | 0.0037             | 1                 | 0.0761            | 0.9816           | 0.0023             | 1                 | 0.0676            | 0.9877           | 0.0048             | 1                 | 0.0599            | 0.9807           |
| 83    | 0.0034             | 1                 | 0.0763            | 0.9816           | 0.0028             | 1                 | 0.0672            | 0.9877           | 0.0047             | 1                 | 0.062             | 0.9816           |
| 84    | 0.0033             | 1                 | 0.0759            | 0.9798           | 0.0023             | 1                 | 0.0675            | 0.9877           | 0.0048             | 1                 | 0.0613            | 0.9807           |
| 85    | 0.0034             | 1                 | 0.0757            | 0.9807           | 0.0024             | 1                 | 0.0663            | 0.9877           | 0.0045             | 1                 | 0.0606            | 0.9807           |
| 86    | 0.0034             | 1                 | 0.0766            | 0.9798           | 0.0025             | 1                 | 0.0673            | 0.9877           | 0.0045             | 1                 | 0.0601            | 0.9807           |
| 87    | 0.0034             | 1                 | 0.0763            | 0.9833           | 0.0023             | 1                 | 0.0666            | 0.9877           | 0.0048             | 1                 | 0.0617            | 0.9807           |
| 88    | 0.0032             | 1                 | 0.0757            | 0.9807           | 0.0035             | 0.9996            | 0.0676            | 0.9877           | 0.0046             | 1                 | 0.0599            | 0.9807           |
| 89    | 0.0033             | 1                 | 0.0762            | 0.9825           | 0.0024             | 1                 | 0.067             | 0.9877           | 0.0047             | 1                 | 0.062             | 0.9807           |
| 90    | 0.0034             | 1                 | 0.076             | 0.9816           | 0.0023             | 1                 | 0.068             | 0.9877           | 0.0043             | 1                 | 0.0606            | 0.9807           |
| 91    | 0.0032             |                   |                   |                  |                    |                   |                   |                  |                    |                   |                   |                  |



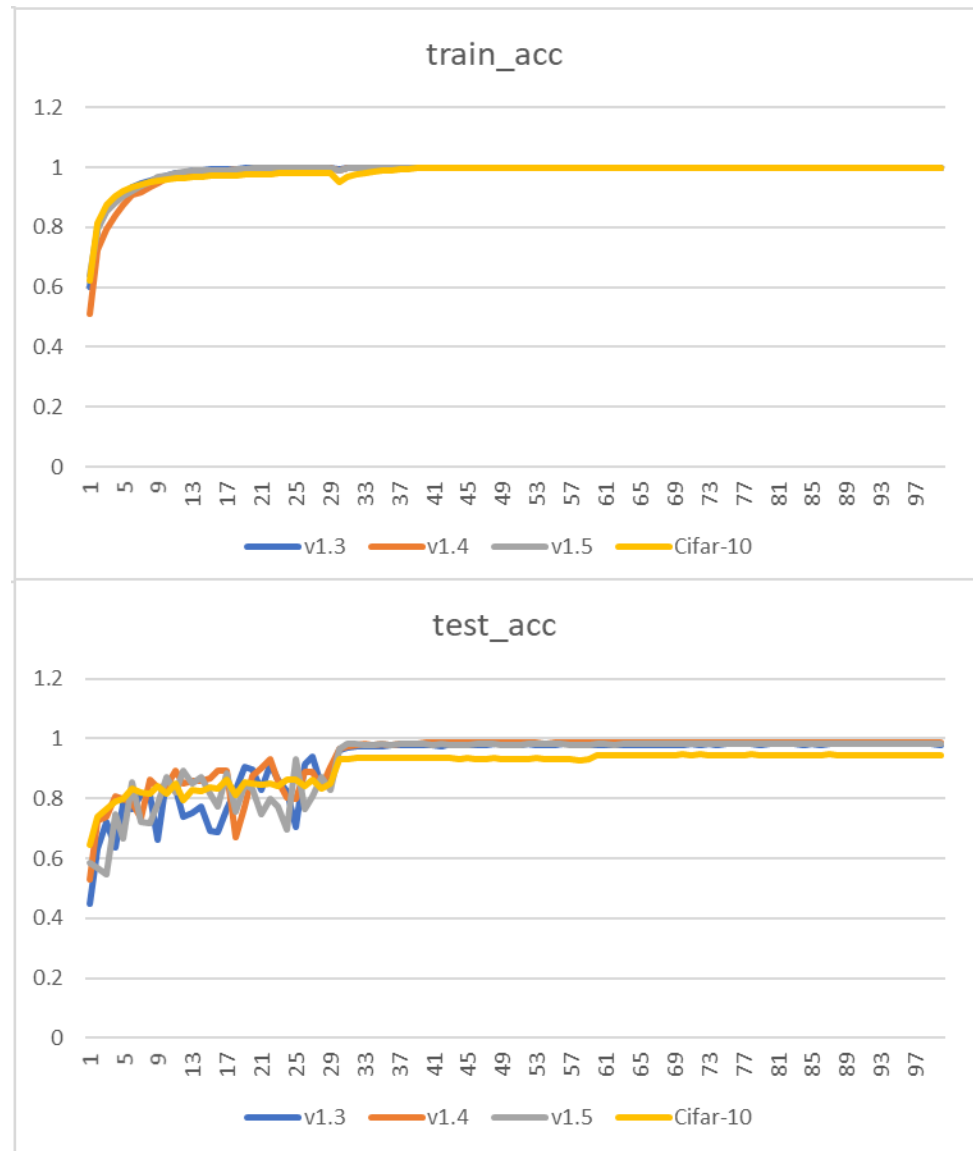
对于这 4 个模型，上述数据经整理后绘图如下（上述数据没有关于 Cifar-10 的数据，但是整理后的折线图有这个数据）：

以下是模型在训练集和测试集上面的损失，横坐标是 epoch 的次数，纵坐标是损失值：





以下是模型在训练集和测试集上面的准确率，横坐标是 epoch 的次数，纵坐标是准确率：







## 四、 总结

基于 ResNeXt 网络结构的表情包分类器在本次课程设计中取得了非常好的效果，分类器在训练集上的准确率达到 100%，而在测试集上的准确率也达到了 97%，且经过对比，发现增加或减少一个 ResNeXt 网络的基本块，对模型的准确率的影响并不大。虽然在刚开始的时候，准确率的波动较大，而且准确率比原版的 ResNeXt 网络有较大的差异，但由于在运行 100 次迭代后影响较小，因此，我们可以认为该网络结构在这个表情包数据集上的效果很好。但是，不可否认的是，由于数据集的规模较少（总共只有 4585（训练集）+1147（测试集）=5732 张图片，而且这个已经是经过旋转操作增加数据集后的数据），以及进行旋转操作增加数据量的方法的约束，这样去制作数据集的方式可能不太规范，并会造成较大的误差。而在使用单张图片验证模型正确率的过程中，我们认为这个模型（v1.3）在预测部分表情包图片时仍有错误的情况出现。在下一阶段，可能仍需要使用更多的数据对模型进行训练和测试。