

# 浙江大学



## 本科实验报告

姓名： 严轶凡

学院： 控制科学与工程学院

系： 自动化

专业： 自动化（控制）

学号： 3200100917

指导教师： 周建光

年    月    日

# 浙江大学 实验报告

## 一、实验目的和要求（必填）

- 1、任务提供包括数据读取、基础模型、模型训练等基本代码
- 2、参赛选手需完成核心模型构建代码，并尽可能将模型调到最佳状态
- 3、模型单次推理时间不超过 10 秒

## 二、实验内容和原理（必填）

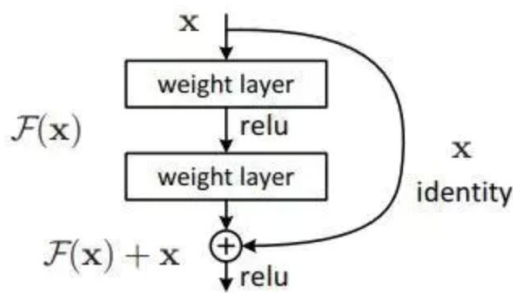
### 实验内容：

数据集包含了 1800 张各个明星的照片，对应的文件夹以他们的姓名命名。数据集中的照片来自 10 位明星，每个明星 180 张照片。需要开发机器学习模型，对明星照片进行识别。

### 实验原理：

本次实验中使用深度残差网络 Resnet 进行模型的训练和图像识别。

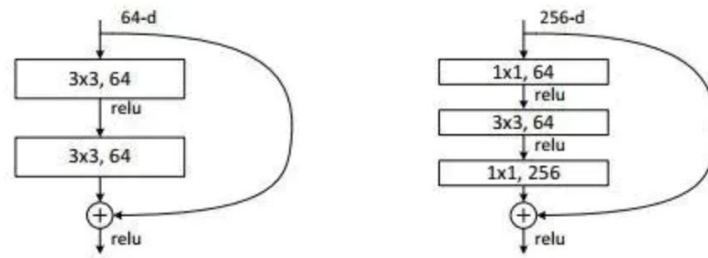
Resnet 一般用于图像分类问题，它是一种残差网络，其实是从 VGG 发展而来，可以解决传统网络深度加深而出现的损失增加问题。



深度残差网络中，残差网络多了一个 identity mapping 过程，即将这一层的输入不做任何处理直接加到这层网络的输出上。

残差神经网络由以下几个关键模块组成：

## 1、residual block/bottleneck 模块



左边就是传统的连接方式，通过两个卷积网络，得到最后的输出，然后再加上残差。当网络很深的时候，计算量就变大了，采用右侧的 bottleneck 结构可以有效减少参数量。bottleneck 包括两个  $1 \times 1$  的卷积核，这个卷积核的作用如下：

- (1) 升维和降维，比如 64 的最后成功映射到 256
- (2) 增加非线性，使网络拟合效果更好
- (3) 减少计算量

## 2、BN 模块

BN 是 Batch Normalization，在进行深度网络的构建的时候一般都会用到这一个模块，主要是在梯度下降算法过程中，需要花费大量时间去调参，比如学习率，dropout 比率等等。相比之下，BN 有以下优势：

- (1) 可以无需 dropout 和 L2 正则
- (2) 可以选择一个较大的学习率，使训练很快收敛

## 3、Relu 模块

常用的 Resnet 网络结构如下：

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

本次实验中，采用 Resnet50 神经网络进行图像分类的任务。

#### 四、操作方法与实验步骤

本次实验使用 pytorch 框架进行开发。

##### （一）数据集划分

使用 processing\_data 函数将数据集文件夹中的数据划分为训练集和验证集,同时将图片进行尺寸上的调整,以及数据归一化处理,以便于后续 pytorch 的模型训练。

```
train_data_loader, valid_data_loader = processing_data(  
    data_path=data_path,  
    height=train_height,  
    width=train_width,  
    batch_size=train_batch_size,  
    test_split=0.2  
)
```

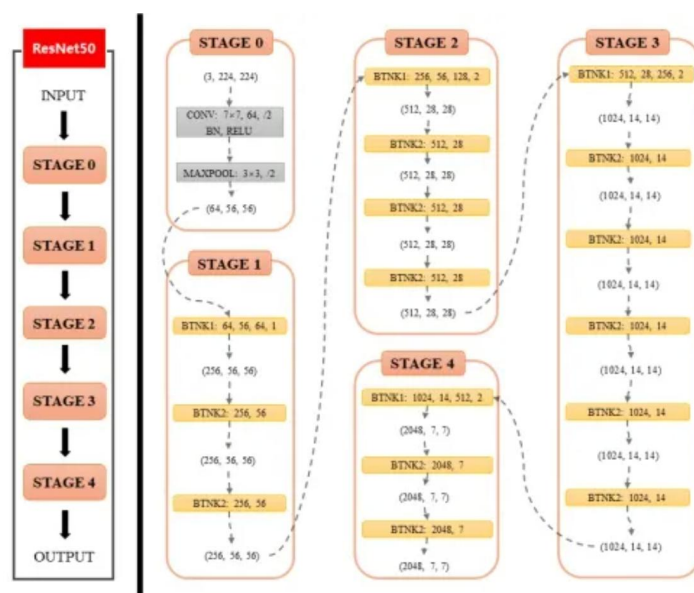
训练集和测试集大小比例按照 8: 2 进行划分。

并且对图像进行随机水平翻转、归一化处理。

## （二）模型结构定义

使用 pytorch 编写 resnet50 的网络结构。

Resnet 的整体结构和各个 Stage 具体结构如下图所示：



## （三）训练模型

训练轮数 epoch 设置为 100；

优化器 optimizer 使用 Adam 优化器；

损失函数 criterion 使用交叉熵函数 `CrossEntropyLoss()`；在模型训练过程中，分别记录每一轮训练过程中的训练误差 loss 以及模型在验证集上预测的正确率 correct，并且以正确率为评价指标，记录正确率最高时的模型结构参数。在模型训练结束后将模型权重参数以文件形式进行保存。

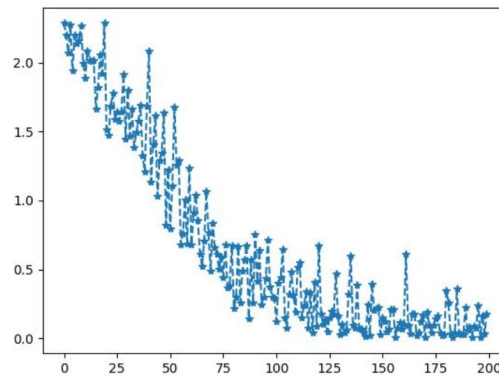
## （四）验证模型

分别向训练得到的模型中传入十个明星的照片，分别记录对每个明星识别的正确率，作为模型的评价指标。

## 五、实验数据结果与分析

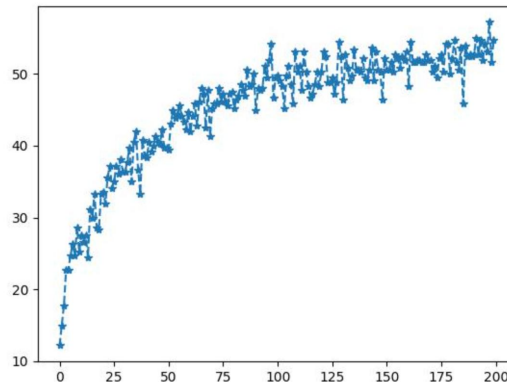
对模型进行连续两次 epoch = 100 轮训练，并且记录训练过程中的训练集交叉熵误差变化以及测试集上的明星脸识别正确率变化。

训练过程中，训练集交叉熵误差变化如图所示：



从曲线图中可以看到，随着训练轮数的增加，误差具有逐渐减小的趋势。

训练过程中，测试集上明星脸识别正确率变化如图所示：



从曲线图中可以看出，随着训练轮数的增加，在测试集上明星脸识别正确率逐渐升高，可见 resnet50 模型在验证集上也有较好的效果，并且较好的解决了过拟合的问题。最终选取训练过程中在测试集上准确率最高的模型权重参数进行保存。

在各个明星的照片测试中，测试结果如下：

CL 99.0 %  
FBB 99.0 %  
HG 98.0 %  
HJ 92.0 %  
LHR 99.0 %  
LSS 96.0 %  
LYF 98.0 %  
PYY 97.0 %  
TY 99.0 %  
YM 97.0 %

可见该模型经过训练，能够达到较高的识别准确率。

在测试系统中的 100 张照片上测试，可见该模型能够取得较高的测试评分。

用例测试

测试点	状态	时长	结果
在 100 张图片上测试模型	✓	44s	通过测试，识别正确率:79

## 六、讨论、心得

通过本次实验，我对机器学习训练深度模型的基本步骤有了更深的理解，同时掌握了使用 pytorch 框架进行深度学习框架搭建的流程和原理。

在本次明星人脸识别的实验过程中，首先考虑了使用 MobileNetV2 进行训练和图像分类任务，但是却出现了较严重的过拟合问题，导致在测试集上验证效果欠佳，后续更换采用了深度残差神经网络 Resnet，才取得了较为不错的图像分类效果。

ResNet 在深度学习历史上是里程碑式的模型，在 ResNet 之前，我们知道深度学习的模型一般都是在 20-30 层之间，但是在 ResNet 出现之后，把深度学习的模型层数提高到一百层以上，获得质的提升，其实 ResNet 可以实现一千层以上的网络模型。

残差网络较好的解决了梯度消失和退化问题，这也是在本次试验中残差神经网络能够取得较好效果的原因。