



同濟大學
TONGJI UNIVERSITY

同济大学《深度学习》 课程实验报告

基于 fashion-mnist 数据集的时尚服
装识别实验

任课老师：范睿

班级：10069801

2023 年 11 月

摘要

本实验旨在使用 fashion-mnist 数据集，通过手动构建神经网络进行图像分类，将不同类型的时尚商品准确地归类到对应的类别中。同时本实验的训练过程进行了可视化，记录了几种网络训练过程的 acc 和 loss；还在测试集上对于几种模型进行了评估 Accuracy、Recall、Precision、F1-score、AUROC，并且绘制了混淆矩阵。

关键词：深度学习，神经网络，CNN, 可视化

目 录

摘要.....	I
第一章 实验内容简述	1
1.1 问题描述	1
1.2 问题背景	1
1.3 数据集介绍	1
1.4 实验环境	2
第二章 核心算法	3
2.1 数据集导入与预处理	3
2.2 神经网络的搭建	3
2.2.1 LeNet 网络	3
2.2.2 Alexnet 网络	3
2.2.3 VGG16 网络	4
2.2.4 MyCNN	5
2.3 优化器	5
2.3.1 SGD 优化器	5
2.3.2 Adam 优化器	5
2.4 损失函数	6
2.5 超参数的设置	6
第三章 实验结果	7
3.1 训练过程	7
3.2 评估过程	8
3.2.1 Accuracy、Recall、Precision、F1-score	8
3.2.2 AUROC	8
3.2.3 混淆矩阵	9
3.2.4 总结	9
第四章 实验总结	11
4.1 实验心得	11
4.2 后续展望	11
参考文献	12

第一章 实验内容简述

1.1 问题描述

在计算机视觉领域，图像分类是一个重要的问题，它在许多应用中都具有关键作用。本实验旨在使用 fashion-mnist 数据集，通过手动构建经典的神经网络以及自己搭建的神经网络进行图像分类，将不同类型的时尚商品准确地归类到对应的类别中。该问题的关键目标是训练一个具有良好泛化能力的模型，使其能够对未见过的时尚图像进行准确分类。

1.2 问题背景

fashion-mnist 数据集是一个包含 10 个时尚类别的灰度图像数据集，每个类别包含 6000 个 28x28 像素的图像。该数据集是 MNIST 的扩展，旨在提供一个更具挑战性的图像分类任务。时尚商品的分类在电子商务和时尚行业等领域有着广泛的应用，因此研究如何有效地进行时尚图像分类具有实际意义。

1.3 数据集介绍

“Fashion-MNIST”是一个用于机器学习和深度学习的图像分类数据集，由 Zalando Research 创建。这个数据集扩展了传统的 MNIST，旨在为研究人员提供一个更具挑战性的任务。它包含 10 个类别的灰度图像，代表着各种时尚物品，如 T 恤、裤子、套头衫等。每个图像的大小为 28x28 像素，与 MNIST 相同，使其成为一个轻量级的数据集，下面是数据集的分类与对应的标签：

标签	类别
0	T 恤 (T-shirt/top)
1	裤子 (Trousers)
2	套头衫 (Pullover)
3	连衣裙 (Dress)
4	外套 (Coat)
5	凉鞋 (Sandal)
6	衬衫 (Shirt)
7	运动鞋 (Sneaker)
8	包 (Bag)
9	靴子 (Ankle boot)

训练集和测试集分别包含 60,000 和 10,000 张图像，为模型的训练和评估提供了丰富的数据。相对于传统的 MNIST 数据集，Fashion-MNIST 提供

了更具挑战性的分类任务，因为时尚物品的外观变化更为复杂，需要模型更深入地理解图像的特征。

1.4 实验环境

本次实验在本地进行代码的编写与调试，在 featurize 网站中租用 GPU 进行训练，以下为本实验的训练环境：

项目	内容
CPU	Intel 5 核 E5-2680 v4
GPU	Nvidia RTX3060
主存	26GB
OS	Linux
CUDA	11.6
编程语言	Python 3.7
框架	pytorch 1.10

第二章 核心算法

2.1 数据集导入与预处理

fashion-mnist 数据集的获取方式比较多,既可以从开源地址或者是 kaggle 等网站获取,也可以从 pytorch 的内置函数中获取。本实验在训练前对数据集进行了预处理,采用的方法是标准化。标准化是一种常见的技术,通过调整输入数据的尺度,使其均值为零、方差为一。标准化有助于提高模型的收敛速度、稳定梯度传播、增强泛化能力,以及使神经网络优化算法更有效。在实际代码中,通过 `torchvision.transforms.Normalize` 对图像数据进行标准化处理,进一步优化了深度学习模型在 FashionMNIST 数据集上的训练效果。

2.2 神经网络的搭建

这部分主要是介绍了本次实验所用到的一些网络,其中包括 LeNet 网络, Alexnet 网络, VGG16 网络以及自己搭建的一个比较简单的 CNN 网络。

2.2.1 LeNet 网络

LeNet (LeNet-5) 是由 Yann LeCun 等人^[1]于 1998 年提出的卷积神经网络 (CNN) 架构。LeNet 的结构相对简单,由卷积层、池化层和全连接层组成。LeNet 的核心思想是通过交替使用卷积和池化层,逐渐减小特征图的空间尺寸,最终连接全连接层进行分类。下面是 LeNet 网络的网络结构图:

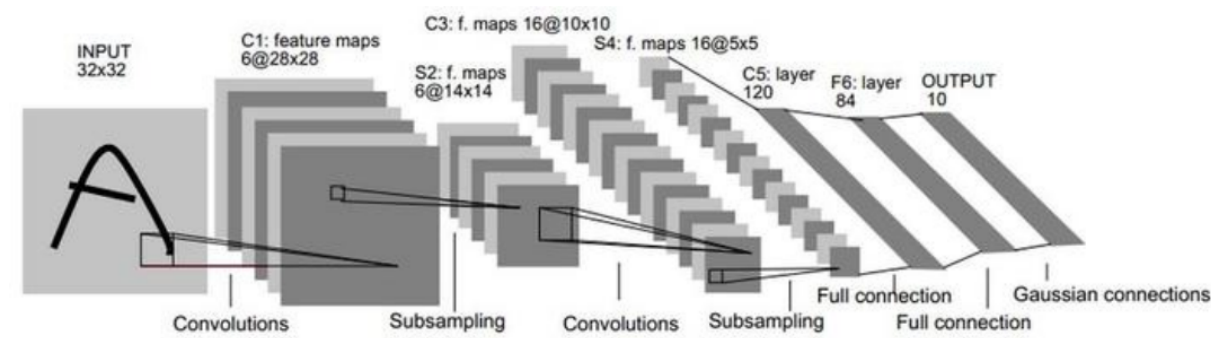


图 2.1: Structure of the LeNet network

2.2.2 Alexnet 网络

在该网络的基础上,为满足本实验数据集的需求,需要将输入大小调整为 28×28 。AlexNet 是在 2012 年由 Alex Krizhevsky 等人设计的深度卷积神经网络^[2],该网络的特点在于其相对较深的架构,包括五个卷积层、池化

层和三个全连接层。采用 ReLU 作为激活函数，引入 Dropout 层以缓解过拟合问题，同时使用局部响应归一化（LRN）层增强模型对不同特征的响应能力。下面是 AlexNet 网络的基本结构：

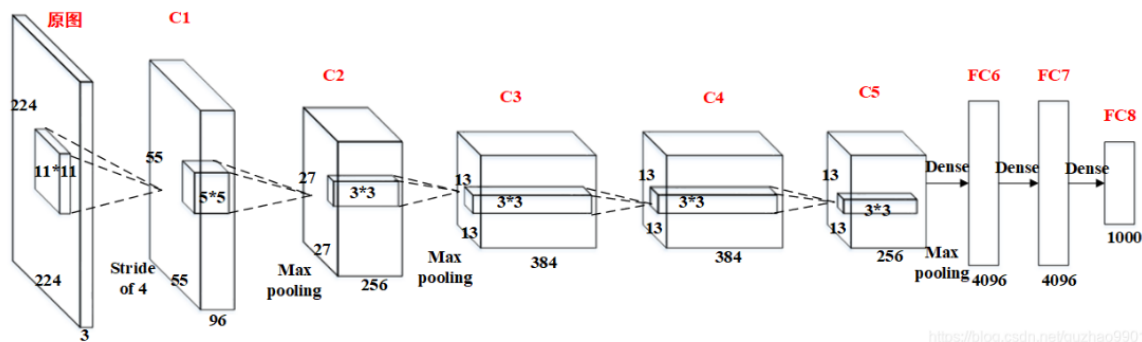


图 2.2: Structure of the AlexNet network

在该网络的基础上，为满足本实验数据集的需求，需要将输入大小调整为 28×28 ，并且在输出层后另外加一层全连接层以实现 10 分类的效果（该网络原本为 1000 分类）

2.2.3 VGG16 网络

VGG16 是一种深度卷积神经网络，由 Oxford 的视觉几何组（Visual Geometry Group）提出^[3]。VGG16 的网络架构主要由卷积层和全连接层组成。卷积层使用小尺寸的 3×3 卷积核，采用 ReLU 激活函数，有助于引入非线性特征。在卷积层之间，VGG16 使用 2×2 的最大池化层进行下采样，以逐渐减小空间维度。全连接层负责生成最终的类别预测。下面是 VGG16 网络的基本结构：

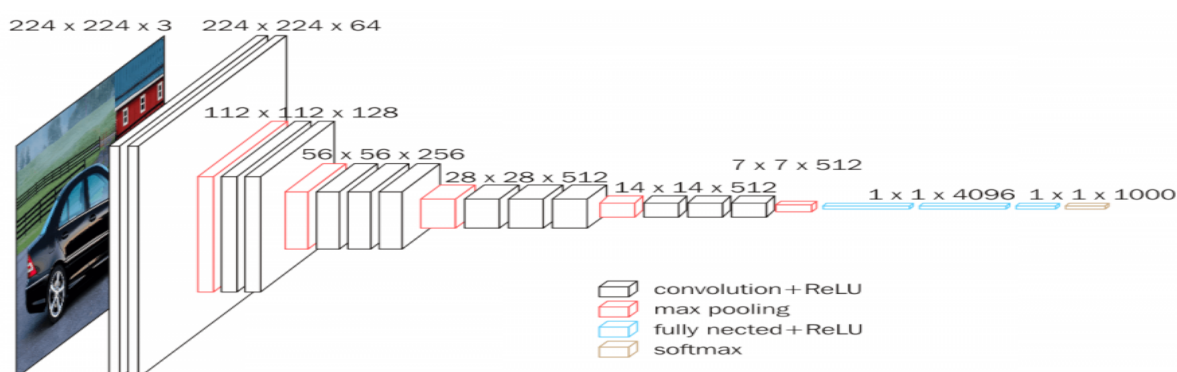


图 2.3: Structure of the VGG16 network

同理，同样需要调整该网络的输入部分与输出部分以适应该实验的要求。同时在实际实验中发现，如果按照原有的 VGG16 网络进行该数据集上的实验，发现会报以下的错误：“RuntimeError: Given input size: (512x1x1). Calculated output size: (512x0x0). Output size is too small” 经过自己的分析后发现，该问题是由于 VGG16 网络中具有非常多的最大池化层，而该数据集

的图片大小非常小，仅仅为 28×28 ，经过多次池化操作后得到的图片非常小，无法进入下一层的网络。本人尝试的方法是改变池化层的步数或者是删除掉一层池化层。但是在进行上述处理之后进行训练，发现训练过程中 acc 非常低，大概只有 10%，经过分析，认为还是 VGG16 网络的复杂性太高，对于本实验的数据集来说并不适合，VGG 网络的特点在于其应用多层的最大池化层进行下采样，然而对于本来图片大小就非常小的数据集来说，经过多次池化后很难提取到准确的特征。

经过删除一些池化层和卷积层，并调整对应层的大小之后进行训练，发现 ACC 有所提高。由于需要对模型本身进行较多的改动，所以后续的报告中没有对该模型进行评估与分析。

2.2.4 MyCNN

经过学习动手搭建上述的三种经典的主要用于神经网络，对基本的分类网络有了一定的认知，所以自己手动构建了一个简单的 CNN 网络用于实现本实验。网络的模型非常简单，只包含了两个卷积层（并进行了最大池化层与激活函数）和两个全连接层，同时还设置有一个 dropout 层，并且最后返回 softmax 函数的结果。

2.3 优化器

优化器的主要任务是通过根据损失函数的梯度信息调整模型参数，实现最小化损失函数的目标。本次实验采用了 SGD 和 Adam 两种优化器进行模型的训练。

2.3.1 SGD 优化器

SGD (Stochastic Gradient Descent) 是深度学习中基础的优化算法之一。它通过随机选择样本来计算梯度并更新模型参数，逐步优化目标函数。这种简单直接的方法使得 SGD 在训练神经网络时非常高效。通过不断迭代样本和更新参数，SGD 帮助模型逐步学习并提高性能。

2.3.2 Adam 优化器

Adam 优化器是一种在深度学习中广泛使用的优化算法，通过综合梯度的一阶矩估计（均值）和二阶矩估计（非中心化方差），实现自适应学习率的调整。其核心特点包括对梯度信息的动态调整，采用指数滑动平均计算梯度的一阶和二阶矩估计，并通过偏差修正纠正初始时的偏差。Adam 算法的更新步骤中，引入了学习率调整项，使每个参数都有独立的学习率，从而适应性更强。这些特性使得 Adam 在训练神经网络时能够更快地收敛，同时对不同尺度的参数和稀疏梯度表现出较好的性能。

2.4 损失函数

当处理多分类任务时，我们需要一个损失函数来衡量模型的预测输出与实际标签之间的差异。这就是多分类交叉熵损失函数的作用。它基于信息论中的交叉熵概念，通过衡量两个概率分布之间的相似度来评估模型的性能。

在多分类任务中，模型的输出是一个概率分布，表示每个类别的预测概率。与之对应的实际标签通常采用 one-hot 编码，其中只有正确类别对应的位置为 1，其他位置为 0。

交叉熵是一种衡量两个概率分布之间差异的指标。对于两个概率分布 (P) 和 (Q)，它们之间的交叉熵定义为各个事件概率的乘积和的负对数。将其应用于多分类问题中，实际标签的概率分布可以看作是 (P)，而模型的预测概率分布可以看作是 (Q)。

多分类交叉熵损失函数的计算公式可以表示为对每个样本的实际标签和模型预测之间的交叉熵的求和，并取平均。这样得到的损失函数能够量化模型的性能，并在训练过程中被最小化。通过优化算法和反向传播算法，模型的权重和偏置会被调整，以最小化多分类交叉熵损失函数。这使得模型能够更好地适应训练数据，并在测试数据上取得更好的性能。损失函数的公式如下：

$$[L(P, Q) = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C P(i) \cdot \log(Q(i))]$$

2.5 超参数的设置

在本实验中，涉及到的超参数包括训练轮数 epoch，训练集和数据集的 batchsize，优化器的学习率 learning rate,SGD 优化器中用到的动量 momentum，随机数种子 random seed 等。经过多次实验尝试后，选定训练轮数为 20 轮，learning rate 初始设置为 0.01 并且在第十轮后变为 0.001

第三章 实验结果

3.1 训练过程

该部分展示了在训练过程中三种网络(LeNet 网络, AlexNet 网络, MyNet 网络) 的 Acc 变化情况和 loss 的变化情况。下面是可视化的图片。

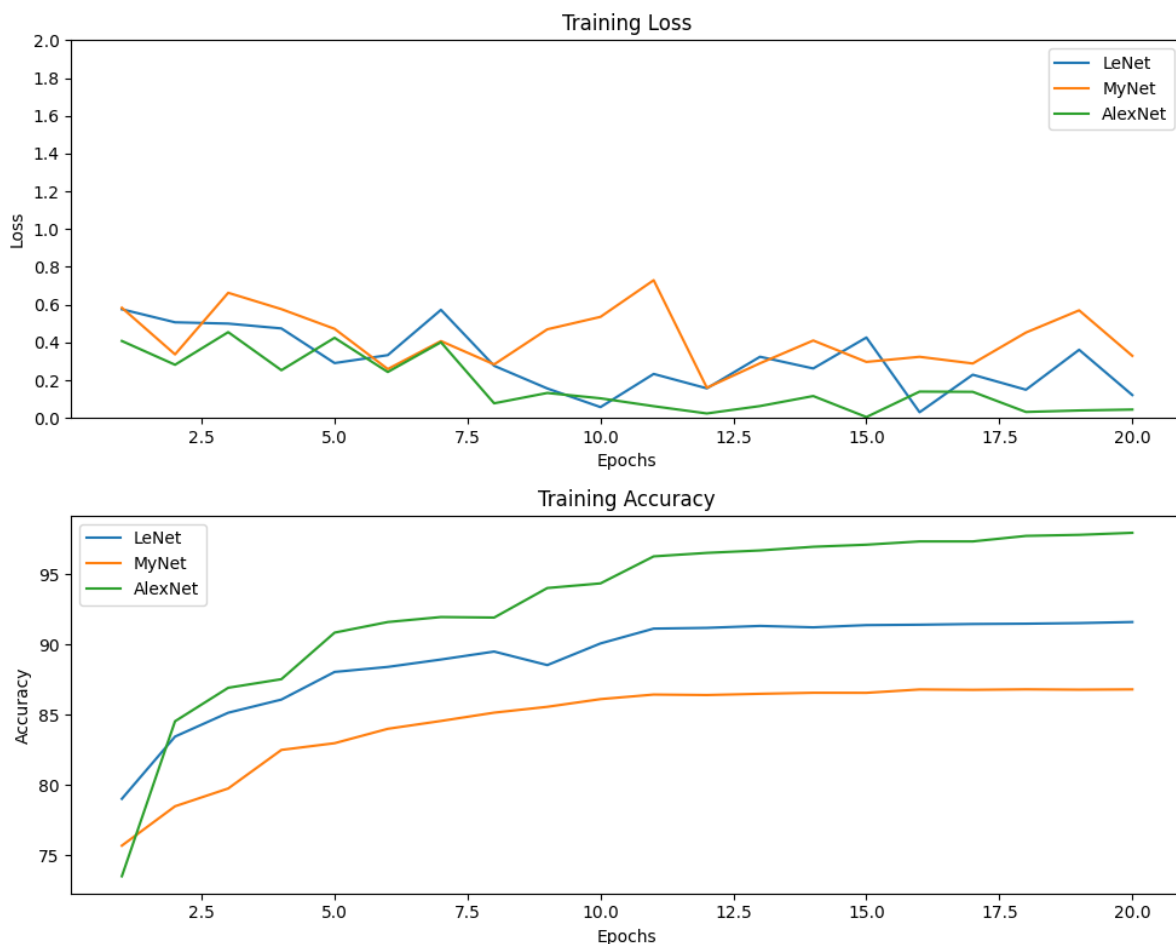


图 3.1: training loss and training accuracy

从图中可以明显看出,当训练轮次(epoch)为20的时候,三种网络均表现出了比较好的效果,损失基本上呈现下降趋势,而在训练集上的准确率也比较高,其中效果最好的是 AlexNet 网络,它的损失基本趋近于 0,而准确率达到了百分之九十七以上;效果最差的是由我自己搭建的神经网络,其损失波动比较明显,但整体上还是呈现下降的趋势,而准确率刚刚超过百分之 85。LeNet 的效果介于二者之间。

3.2 评估过程

该部分展示了对三种网络在测试集上进行评估，通过 Accuracy、Recall、Precision、F1-score、AUROC 这五种指标进行测评以及绘制混淆矩阵进行可视化。这里是通过调用 `sklearn.metrics` 中内置的函数来计算这几个指标。

3.2.1 Accuracy、Recall、Precision、F1-score

首先是前四个指标：

网络 \ 指标	Accuracy	Recall	Precision	F1-score
AlexNet	0.9162	0.9162	0.9159	0.9157
LeNet	0.8939	0.8939	0.8939	0.8939
MyNet	0.8588	0.8588	0.8589	0.8527

这里发现在四个指标中 Accuracy 和 Recall 都是一样的。而对于 LeNet，甚至出现了四个指标全部相同的情况。经过查阅相关参考资料后发现是在求这几个指标是 = 时 average 用的方法不同会导致结果不同，如果使用 micro-average 方法来求，就会出现这样的结果，有 $\text{micro-precision} = \text{micro-recall} = \text{micro-F1-score} = \text{accuracy}$ 。^[4]

解决方法的话就是换一种平均的方法 $\text{average} = \text{'macro'}$ 或者“weighted”。这里由于时间的缘故没有进行进一步的测试。

3.2.2 AUROC

下面是三种网络分别在测试集上的 AUROC：

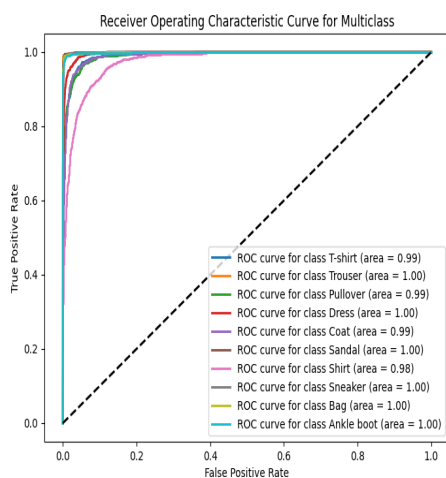


图 3.2: AlexNet AUROC

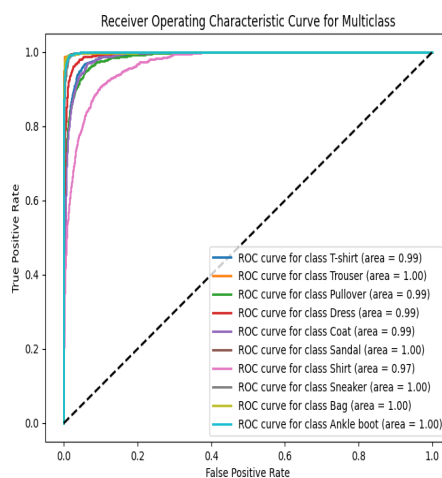


图 3.3: LeNet AUROC

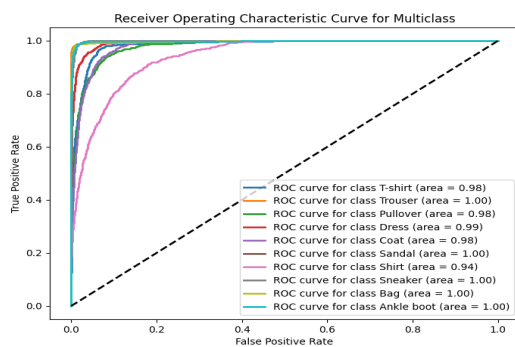


图 3.4: MyNet AUROC

3.2.3 混淆矩阵

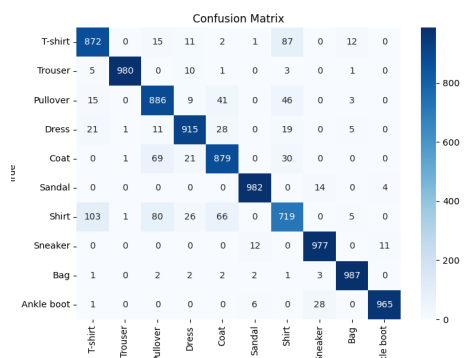


图 3.5: AlexNet matrix

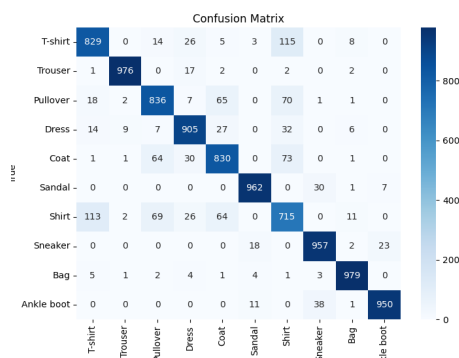


图 3.6: LeNet matrix

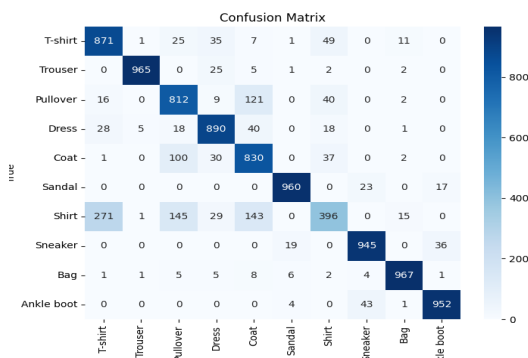


图 3.7: MyNet matrix

3.2.4 总结

在测试集上，三种模型同样取得了比较好的效果，其中 AlexNet 的各项指标依旧是最优的，基本都超过了百分之 91，而 LeNet 也有百分之 89 左右的结果，MyNet 表现最差，有百分之 85 左右。

但是和三者测试集上的效果进行对比发现，AlexNet 和 LeNet 均在一定程度上出现效果的下滑，而 MyNet 基本保持相同的效果，这就说明前两种网络出现了过拟合的可能性，而 MyNet 由于结构非常简单，没有出现过拟合现象。出现过拟合可能是训练轮数过多导致的，也可能是网络本身的结构过于复杂。

第四章 实验总结

4.1 实验心得

在这个实验中，我尝试了手动搭建了几个经典的卷积神经网络模型，比如 LeNet、VGG16、AlexNet 以及自己搭建了简单的 CNN 网络等。在搭建这些模型的过程中，我学会了如何设计不同层级的卷积、池化、批量归一化和全连接层，以及如何在这些层之间合理地连接和堆叠，以达到更好的分类性能。

除了学会了搭建模型外，我也对其他的一些概念有了更加深刻的掌握，比如了解了深度学习常见的几项评价指标，并且由于实验过程中产生的一些问题，对这几项指标的计算方法有了更加清晰的认识；同时也学会了如何更好地写深度学习的代码，

4.2 后续展望

尽管在本次实验中通过自行搭建神经网络进行图像分类，取得了较为满意的分类结果，但是，仍然存在一些潜在的改进方向和进一步研究的可能。可以采取一系列策略来进一步提升图像分类模型。首先，可以尝试使用不同的卷积核大小和通道数，或者增加网络深度，以更好地捕捉时尚图像中的抽象特征。同时还可以考虑调整模型的超参数，如学习率、批量大小和训练轮数，调整这些超参数可能会对模型的性能产生显著影响。

另一个关键方向是考虑模型的泛化能力。本次实验的数据集的图片的大小仅为 28×28 ，而目前来看一般我们通过各种途径得到的图片大小远远超过这个，如果直接将拍摄得到的图片大小转为 28×28 ，识别效果会比测试集上的效果差很多。通过引入更多的变换，如旋转、缩放和平移，对图像进行处理，同时我们还可以扩展训练数据集，提高模型对各种变化的适应能力。这有助于提高模型的泛化性能，特别是在面对真实世界中多样化的时尚图像时候。

此外，可以考虑模型的可解释性。探索可解释性工具和方法，例如用 Grad-CAM 等方法可以帮助我们理解模型在决策时关注的特定区域，提高模型的可信度和可理解性。

通过不断地实验和改进，期望能够建立一个更强大、高效的时尚图像分类模型，使其在实际应用中表现出色。这些进一步的工作将有助于深化我们对深度学习在时尚领域中的应用理解，并为未来的研究提供更多的启示。

参考文献

- [1] Y LeCun, L Bottou, Y Bengio, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278–2324.
- [2] A Krizhevsky, I Sutskever, G E Hinton. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25.
- [3] K Simonyan, A Zisserman. Very deep convolutional networks for large-scale image recognition[J]. ArXiv preprint arXiv:1409.1556, 2014.
- [4] 江南马杀鸡. Skmetrics 输出 acc、precision、recall、f1 值相同的问题. <https://blog.csdn.net/fujikoo/article/details/119926390>.