机器学习 (EE369) Project2

基于深度学习的生活垃圾图片分类

Classification of Garbage Picture Based on Deep Learing

姓名: 蒋逸伟

学号:517161910005

目录

1 引言

1.1 背景介绍

随着《上海市生活垃圾管理条例》[?] 在 2019 年 7 月 1 日正式开始执行, 全国范围内的垃圾分类的热潮正式掀开。垃圾分类对于人类的可持续发展以及对生态环境的保护具有重要的意义，它可以帮助上游企业更好地实现资源的可回收利用以及垃圾填埋等。对于垃圾分类，传统上只能依靠人进行识别手工拣拾，耗时耗力。近年来随着深度学习技术的发展，利用搭建好的神经网络对图片进行特征提取然后进行分类识别技术已经成熟，AlexNet[?], ResNet[?] 等神经网络均在传统的 CNN 上进行了改进从而在 CIFAR1000,ImageNet 等数据集上实现了较好的分类效果，他们同样在垃圾图片分类这个任务上具有巨大的潜力。

1.2 问题目标

这个问题的目标在于给出一个垃圾的图片 x_i 能够将它正确的归并在 {” 纸板”,” 玻璃”,” 金属”,” 纸张”,” 塑料”,” 其他垃圾”} 中的一类。

1.3 数据概况

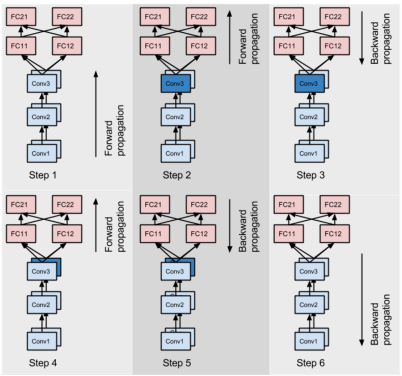
用于解决本个问题的数据集包括六类: 分别包含 cardboard(393), glass (491), metal (400), paper (584), plastic (472) and trash (127). 划分后的数据集：训练集：1768，验证集：328，测试集：431。

2 方法与技术

本文在上述的数据集上分别应用了 AlexNet,MobileNet,DenseNet,ResNet,VGG,EfficientNet, 多种神经网络进行训练。

2.1 AlexNet

本深度学习网络是 Alex 和 Hinton 参加 ILSVRC2012 比赛的卷积网络论文，本网络结构也是开启 ImageNet 数据集更大，更深 CNN 的开山之作，本文对 CNN 的一些改进成为以后 CNN 网络通用的结构；在一些报告中被称为 Alex-net，之后在 Imagenet 上取得更好结果的 ZF-net，SPP-net，VGG 等网络，都是在其基础上修改得到。[?]



2.2 VGG

AlexNet 在 LeNet 的基础上增加了 3 个卷积层。但 AlexNet 作者对它们的卷积窗口、输出通道数和构造顺序均做了大量的调整。虽然 AlexNet 指明了深度卷积神经网络可以取得出色的结果，但并没有提供简单的规则以指导后来的研究者如何设计新的网络。我们将在本章的后续几节里介绍几种不同的深度网络设计思路。它的名字来源于论文作者所在的实验室 Visual Geometry Group 。[?]VGG 提出了可以通过重复使用简单的基础块来构建深度模型的思路。

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	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	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 **conv1-256**	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 **conv1-512**	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 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

2.3 ResNet

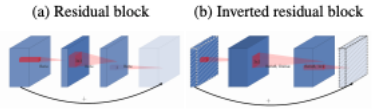
而 Resnet[?] 网络作者则想到了常规计算机视觉领域常用的 residual representation 的概念，并进一步将它应用在了 CNN 模型的构建当中，于是就有了基本的 residual learning 的 block。它通过使用多个有参层来学习输入输出之间的残差表示，而非像一般 CNN 网络（如 Alexnet/VGG 等）那样使用有参层来直接尝试学习输入、输出之间的映射。实验表明使用一般意义上的有参层来直接学习残差比直接学习输入、输出间映射要容易得多（收敛速度更快），也有效得多（可通过使用更多的层来达到更高的分类精度）。

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				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

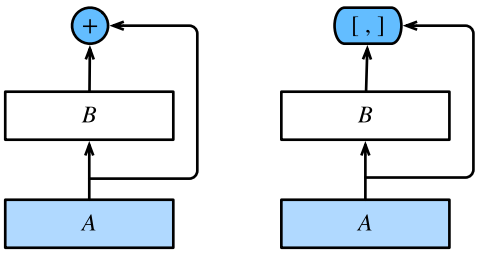
ures for ImageNet. Building blocks are shown in brackets (see also Fig. 5), with the numbers of block

2.4 MobileNet

MobileNet[?] 的基本单元是深度级可分离卷积（depthwise separable convolution），其实这种结构之前已经被使用在 Inception 模型中。深度级可分离卷积其实是一种可分解卷积操作（factorized convolutions），其可以分解为两个更小的操作：depthwise convolution 和 pointwise convolution。Depthwise convolution 和标准卷积不同，对于标准卷积其卷积核是用在所有的输入通道上（input channels），而 depthwise convolution 针对每个输入通道采用不同的卷积核，就是说一个卷积核对应一个输入通道，所以说 depthwise convolution 是 depth 级别的操作。而 pointwise convolution 其实就是普通的卷积，只不过其采用 1x1 的卷积核。对于 depthwise separable convolution，其首先是采用 depthwise convolution 对不同输入通道分别进行卷积，然后采用 pointwise convolution 将上面的输出再进行结合，这样其实整体效果和一个标准卷积是差不多的，但是会大大减少计算量和模型参数量。



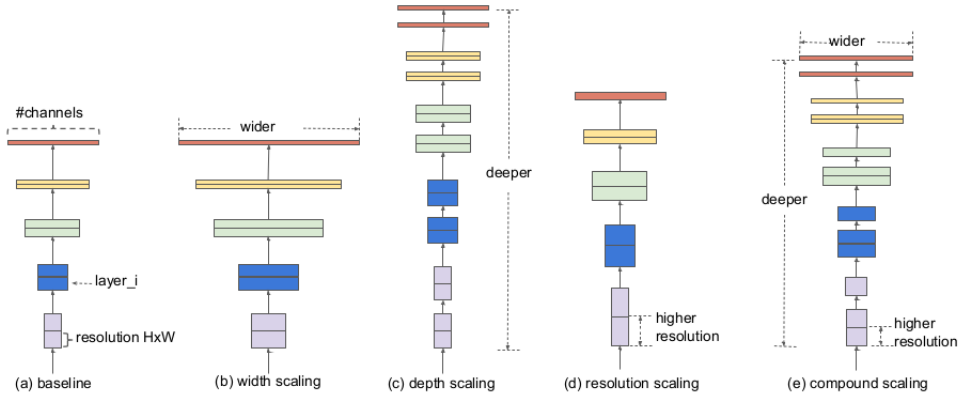
2.5 DenseNet



与 ResNet 的主要区别在于，DenseNet 里模块 B 的输出不是像 ResNet 那样和模块 A 的输出相加，而是在通道维上连结。这样模块 A 的输出可以直接传入模块 B 后面的层。在这个设计里，模块 A 直接跟模块 B 后面的所有层连接在了一起。这也是它被称为“稠密连接”的原因。

2.6 EfficientNet

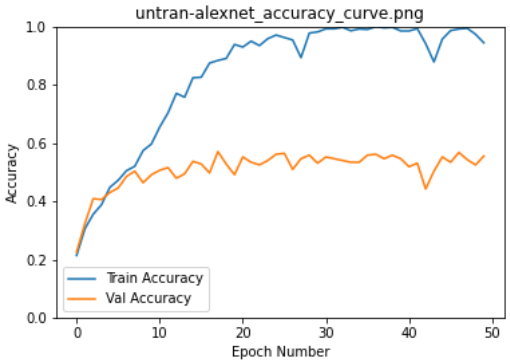
[?] 作者希望找到一个可以同时兼顾速度与精度的模型放缩方法，为此，作者重新审视了前人提出的模型放缩的几个维度：网络深度、网络宽度、图像分辨率，前人的文章多是放大其中的一个维度以达到更高的准确率，比如 ResNet-18 到 ResNet-152 是通过增加网络深度的方法来提高准确率。

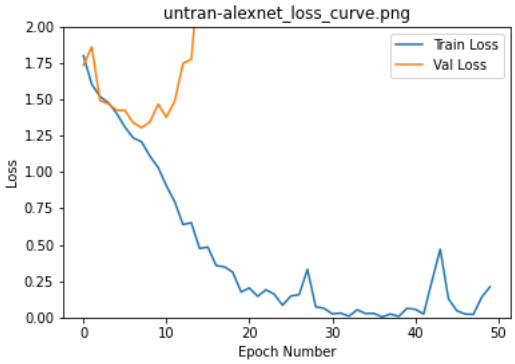


3 实验结果

3.1 未进行迁移学习（未加载与训练参数）的神经网络训练结果

3.1.1 AlexNet

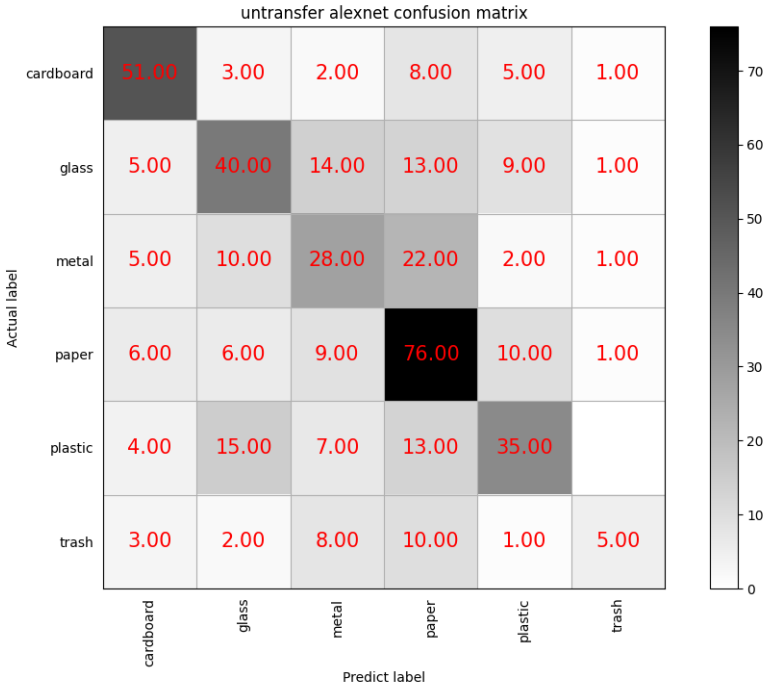




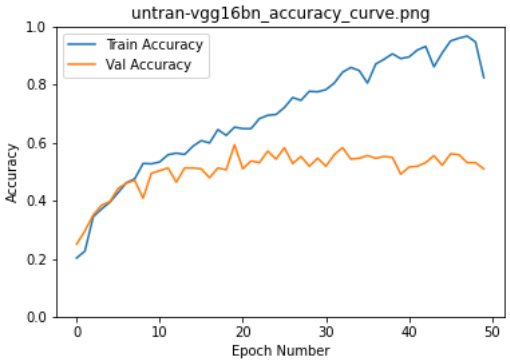
在测试集上的训练结果

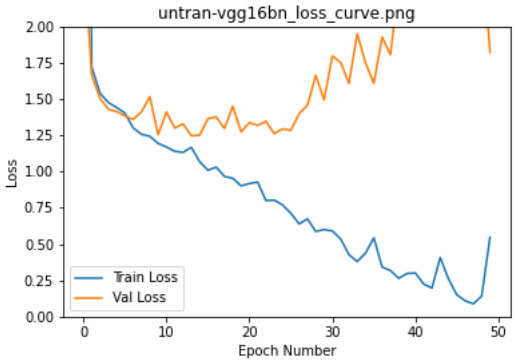
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.9025522	0.81902552	0.81438515	0.77262181	0.84686775	0.9350348
Recall	0.72857143	0.48780488	0.41176471	0.7037037	0.47297297	0.17241379
Precision	0.68918919	0.52631579	0.41176471	0.53521127	0.56451613	0.55555556
F1 Score	0.70833333	0.50632911	0.41176471	0.608	0.51470588	0.26315789

混淆矩阵：



3.1.2 VGG16_bn

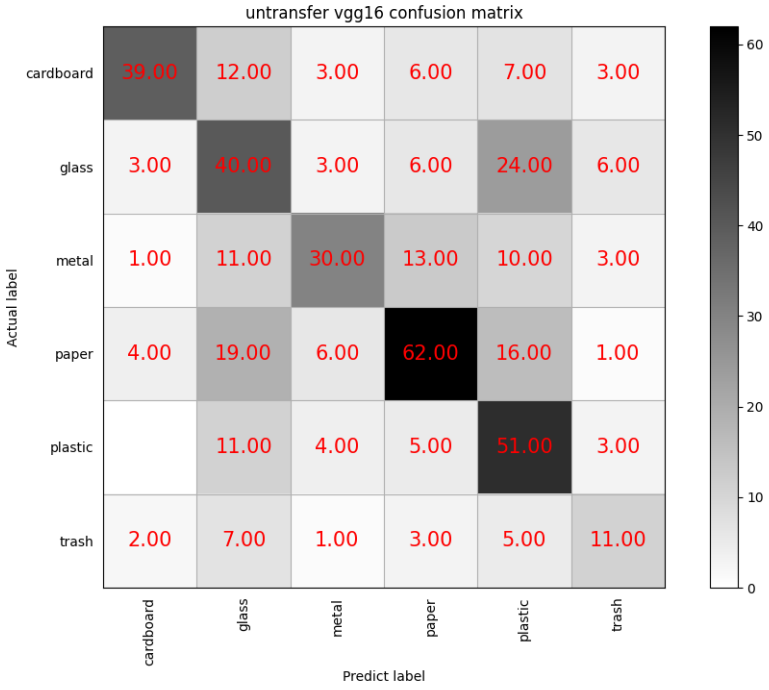




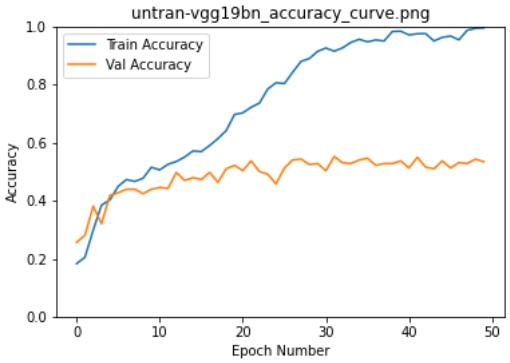
测试集上的结果：

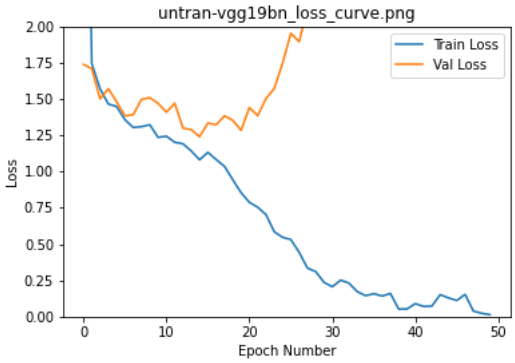
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.90487239	0.76334107	0.87238979	0.81670534	0.80278422	0.92111369
Recall	0.55714286	0.48780488	0.44117647	0.57407407	0.68918919	0.37931034
Precision	0.79591837	0.4	0.63829787	0.65263158	0.45132743	0.40740741
F1 Score	0.65546218	0.43956044	0.52173913	0.61083744	0.54545455	0.39285714

混淆矩阵：



3.1.3 VGG19_bn

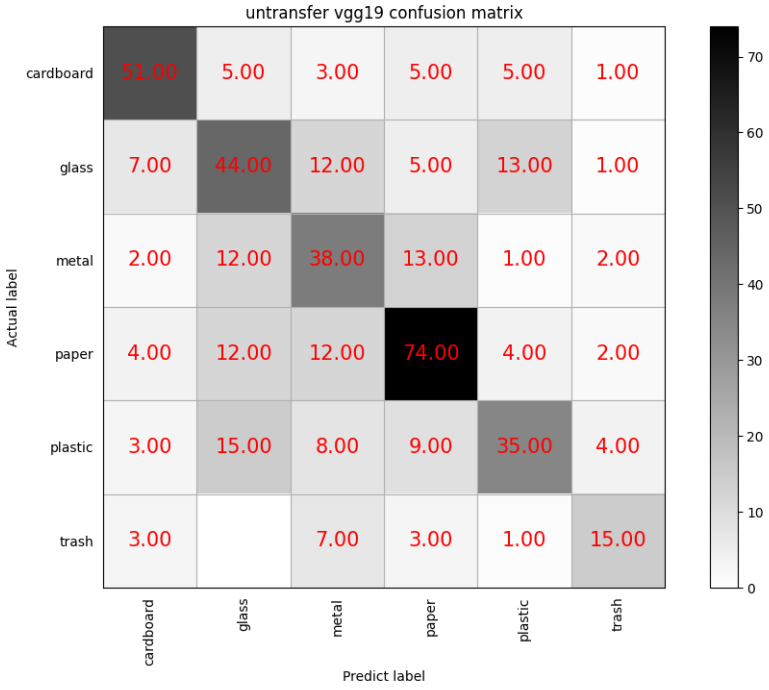




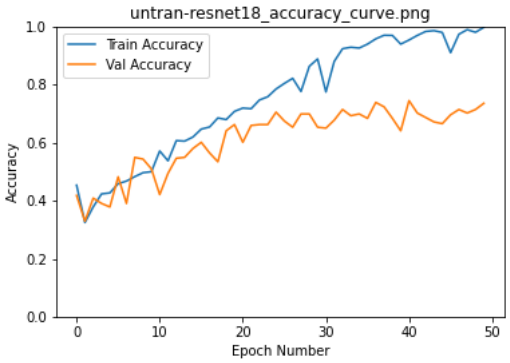
测试集上的训练结果

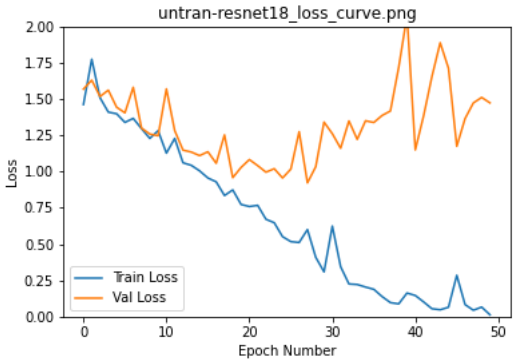
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.91183295	0.80974478	0.83294664	0.83990719	0.85382831	0.94431555
Recall	0.72857143	0.53658537	0.55882353	0.68518519	0.47297297	0.51724138
Precision	0.72857143	0.5	0.475	0.67889908	0.59322034	0.6
F1 Score	0.72857143	0.51764706	0.51351351	0.68202765	0.52631579	0.55555556

混淆矩阵：



3.1.4 ResNet18

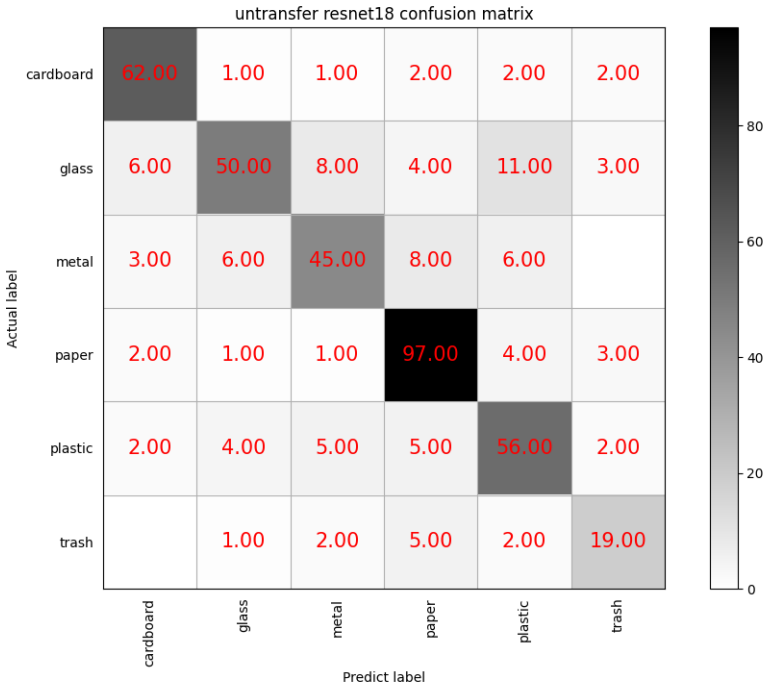




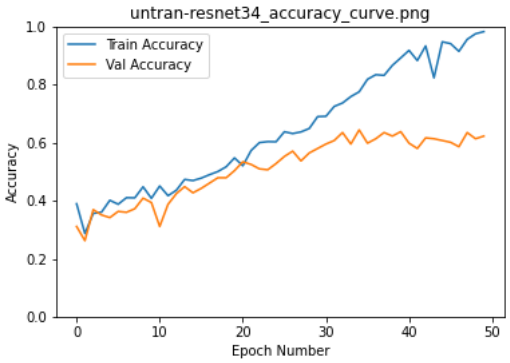
测试集上的结果：

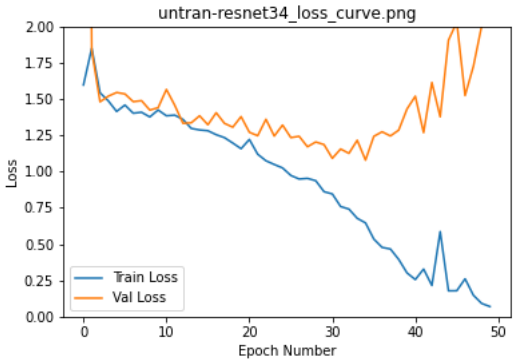
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.9512761	0.89559165	0.90719258	0.9187935	0.90023202	0.95359629
Recall	0.88571429	0.6097561	0.66176471	0.89814815	0.75675676	0.65517241
Precision	0.82666667	0.79365079	0.72580645	0.80165289	0.69135802	0.65517241
F1 Score	0.85517241	0.68965517	0.69230769	0.84716157	0.72258065	0.65517241

混淆矩阵：



3.1.5 ResNet34

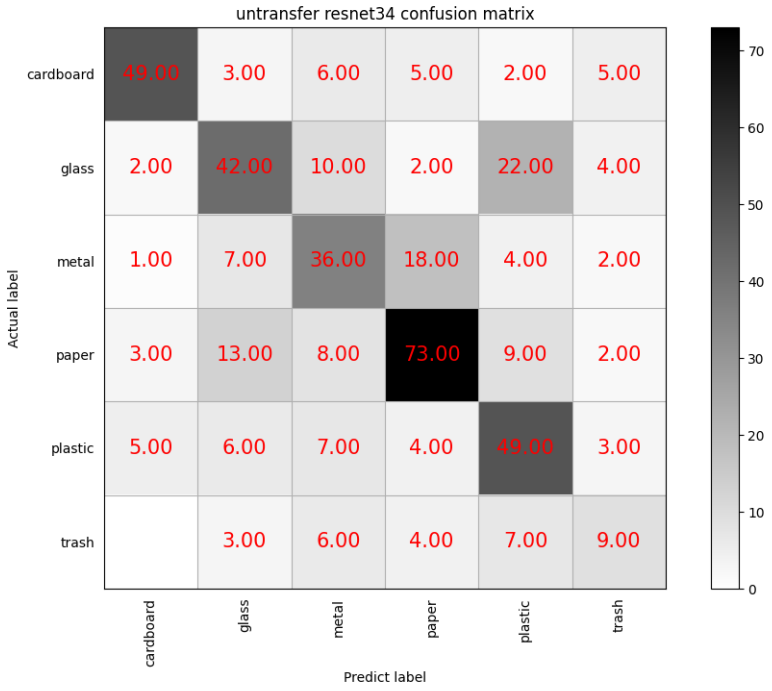




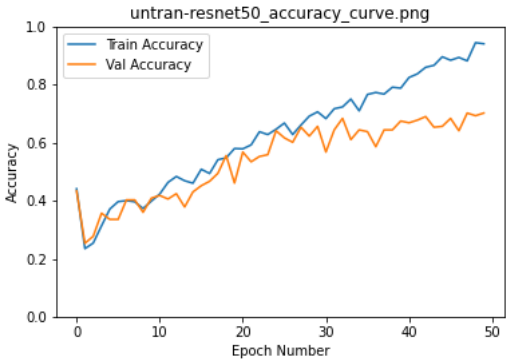
测试集上的结果：

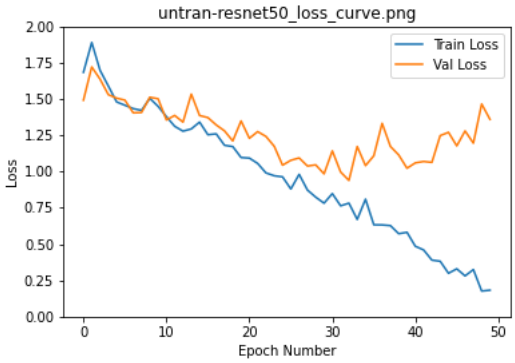
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.92575406	0.83294664	0.83990719	0.84222738	0.83990719	0.91647332
Recall	0.7	0.51219512	0.52941176	0.67592593	0.66216216	0.31034483
Precision	0.81666667	0.56756757	0.49315068	0.68867925	0.52688172	0.36
F1 Score	0.75384615	0.53846154	0.5106383	0.68224299	0.58682635	0.33333333

混淆矩阵：



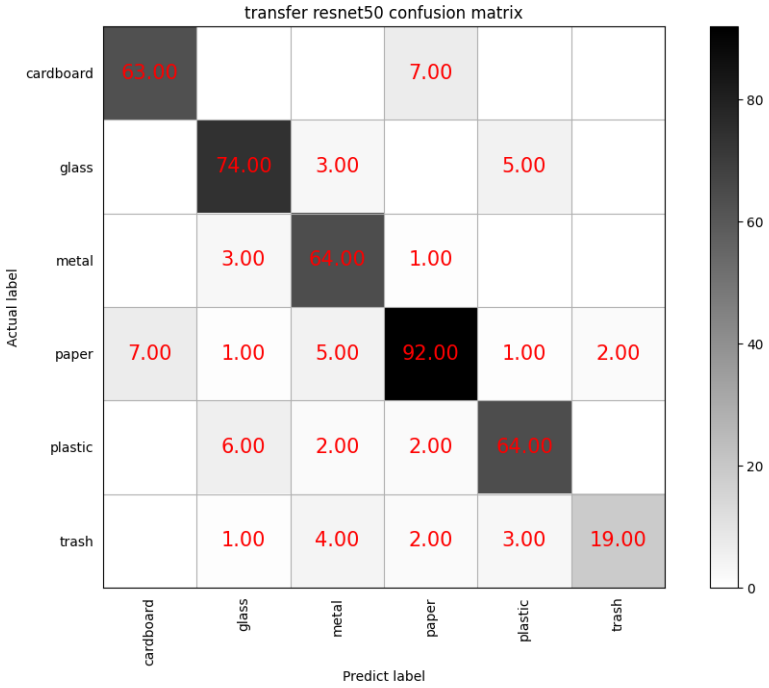
3.1.6 ResNet50





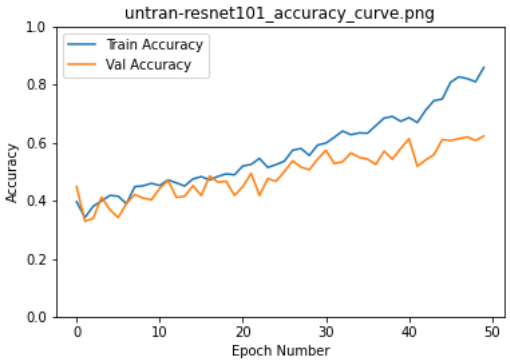
测试集上的结果：

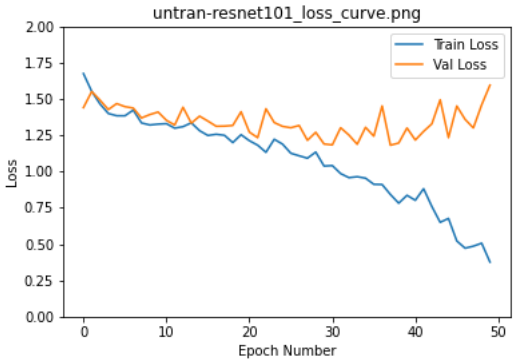
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.9675174	0.95591647	0.95823666	0.9350348	0.95591647	0.97215777
Recall	0.9	0.90243902	0.94117647	0.85185185	0.86486486	0.65517241
Precision	0.9	0.87058824	0.82051282	0.88461538	0.87671233	0.9047619
F1 Score	0.9	0.88622754	0.87671233	0.86792453	0.8707483	0.76



混淆矩阵：

3.1.7 ResNet101

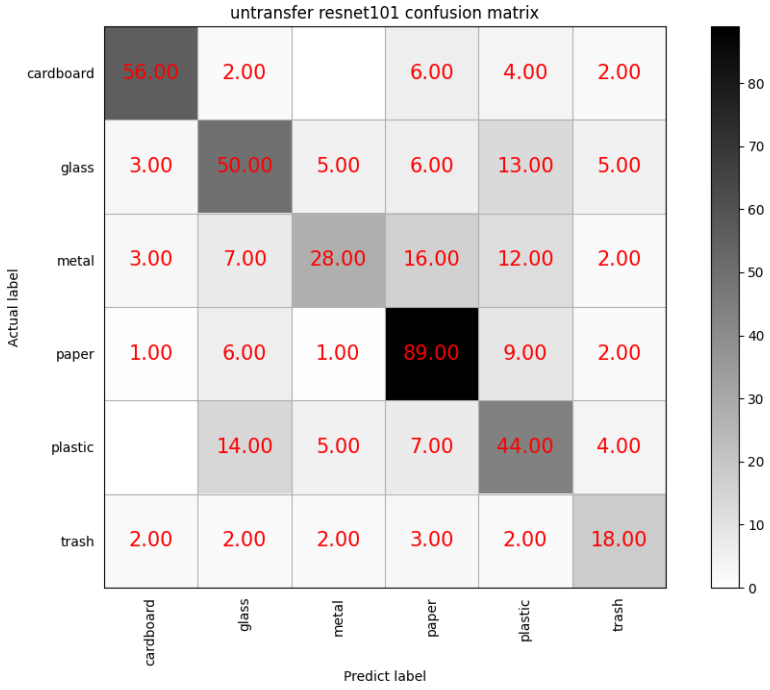




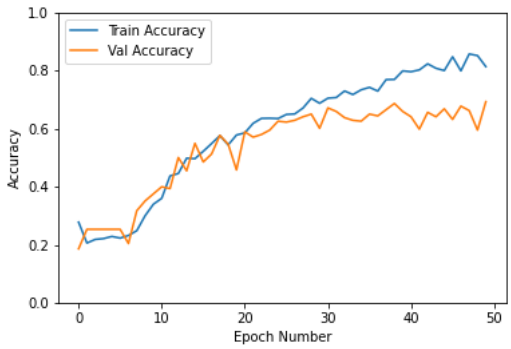
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.94431555	0.87935035	0.87935035	0.90951276	0.86542923	0.92807425
Recall	0.74285714	0.6097561	0.67647059	0.80555556	0.72972973	0.48275862
Precision	0.89655172	0.71428571	0.60526316	0.82857143	0.58695652	0.46666667
F1 Score	0.8125	0.65789474	0.63888889	0.81690141	0.65060241	0.47457627

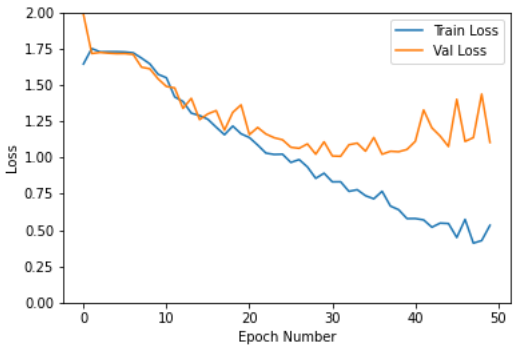
测试集上的结果：

混淆矩阵：



3.1.8 MobileNet_V2

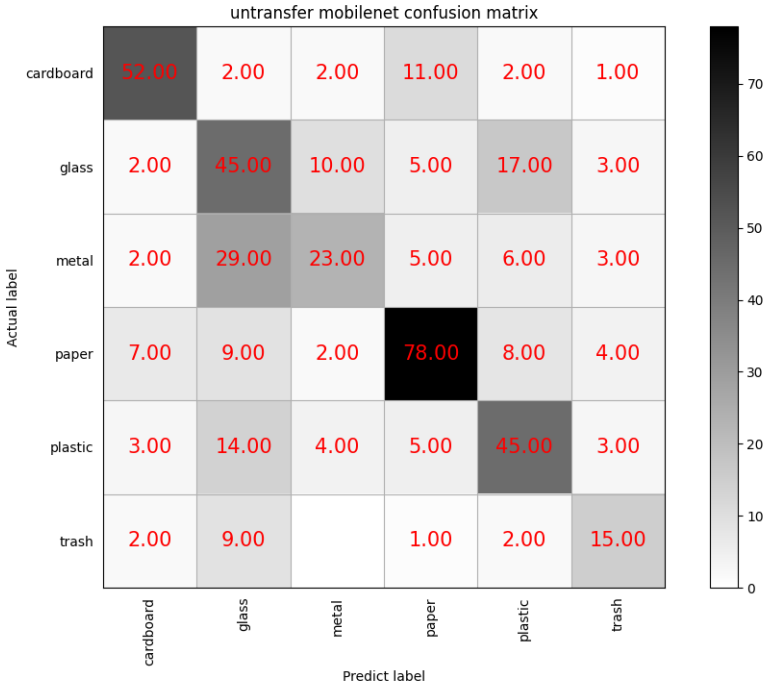




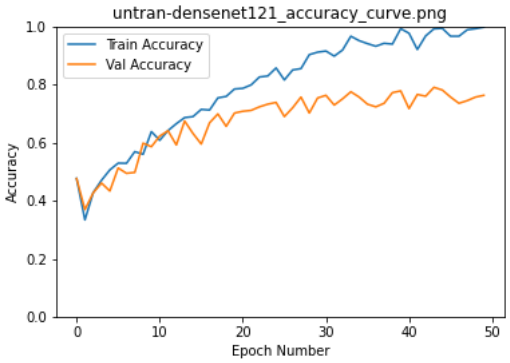
测试集上的结果：

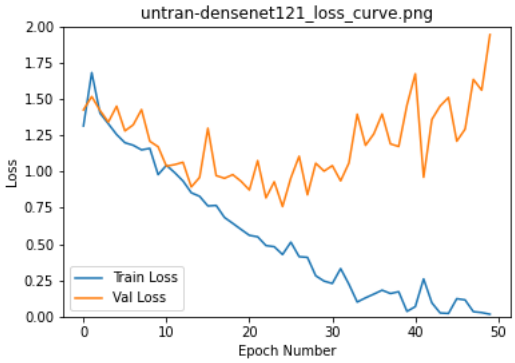
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.92111369	0.76798144	0.85382831	0.86774942	0.85150812	0.9350348
Recall	00.74285714	0.54878049	0.33823529	0.72222222	0.60810811	0.51724138
Precision	0.76470588	0.41666667	0.56097561	0.74285714	0.5625	0.51724138
F1 Score	0.75362319	0.47368421	0.42201835	0.73239437	0.58441558	0.51724138

混淆矩阵：



3.1.9 DenseNet121

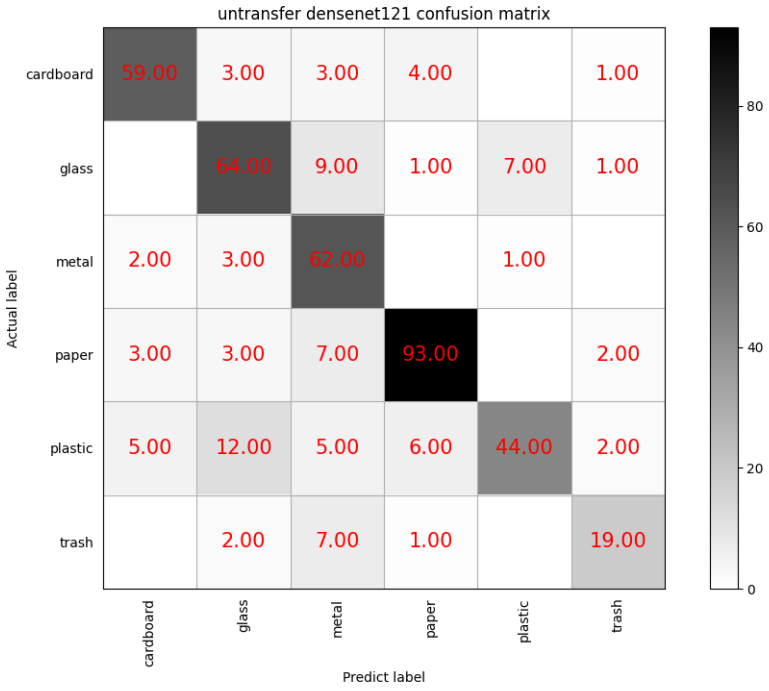




测试集上的结果：

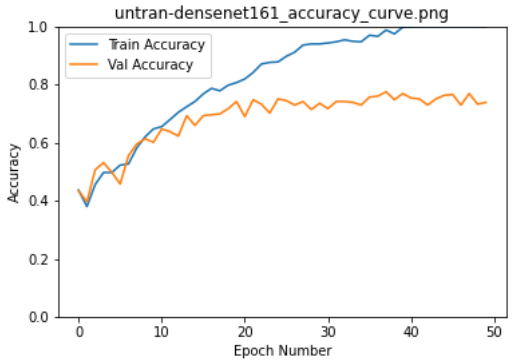
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.9512761	0.90487239	0.91415313	0.93735499	0.91183295	0.96287703
Recall	0.84285714	0.7804878	0.91176471	0.86111111	0.59459459	0.65517241
Precision	0.85507246	0.73563218	0.66666667	0.88571429	0.84615385	0.76
F1 Score	0.84892086	0.75739645	0.77018634	0.87323944	0.6984127	0.7037037

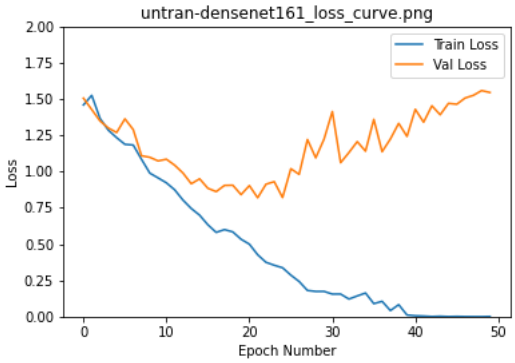
混淆矩阵：



3.1.10 DenseNet161

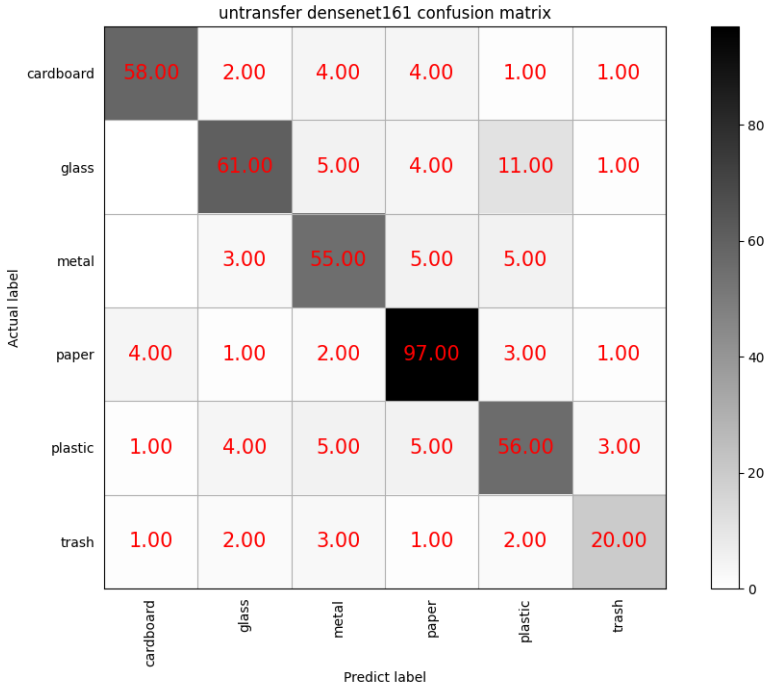
测试集上的结果：





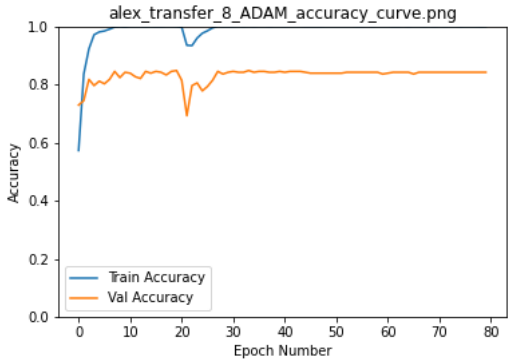
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.95823666	0.92343387	0.92575406	0.93039443	0.90719258	0.96519722
Recall	0.82857143	0.74390244	0.80882353	0.89814815	0.75675676	0.68965517
Precision	0.90625	0.83561644	0.74324324	0.8362069	0.71794872	0.76923077
F1 Score	0.86567164	0.78709677	0.77464789	0.86607143	0.73684211	0.72727273

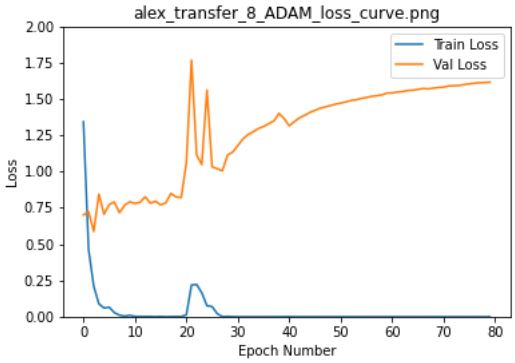
混淆矩阵:



3.2 进行迁移学习（加载了云训练参数）的神经网络训练结果

3.2.1 AlexNet

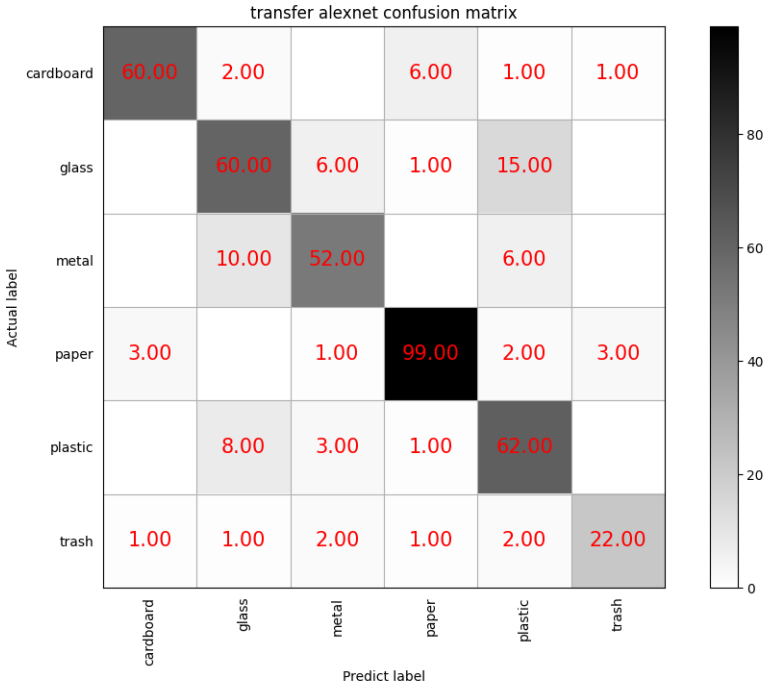




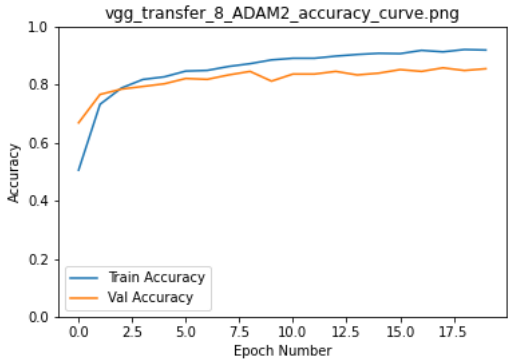
在测试集上的训练结果

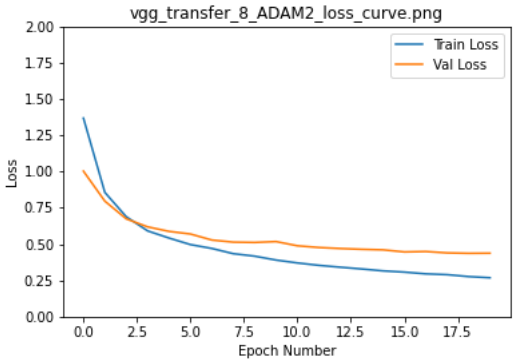
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.9675174	0.90023202	0.9350348	0.95823666	0.91183295	0.97447796
Recall	0.85714286	0.73170732	0.76470588	0.91666667	0.83783784	0.75862069
Precision	0.9375	0.74074074	0.8125	0.91666667	0.70454545	0.84615385
F1 Score	0.89552239	0.73619632	0.78787879	0.91666667	0.7654321	0.8

混淆矩阵：



3.2.2 VGG16_bn

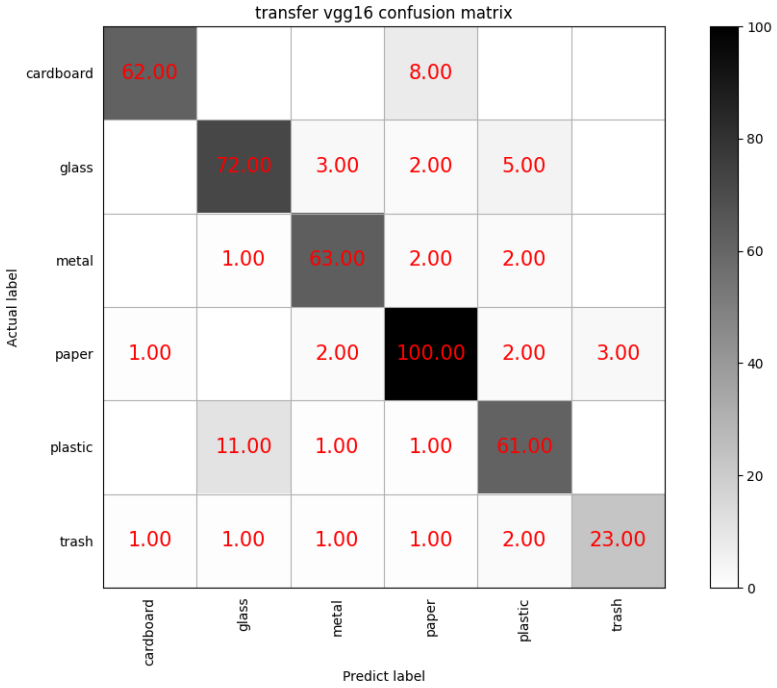




测试集上的结果：

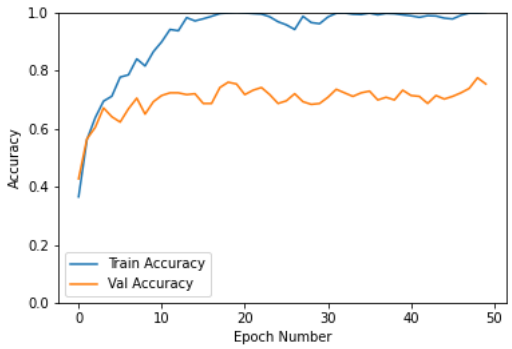
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.97679814	0.94663573	0.97215777	0.94895592	0.94431555	0.97911833
Recall	0.88571429	0.87804878	0.92647059	0.92592593	0.82432432	0.79310345
Precision	00.96875	0.84705882	0.9	0.87719298	0.84722222	0.88461538
F1 Score	0.92537313	0.86227545	0.91304348	0.9009009	0.83561644	0.83636364

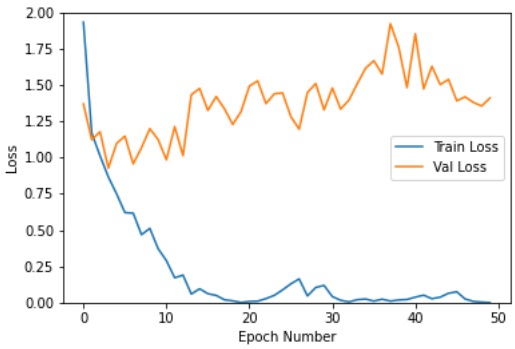
混淆矩阵：



/

3.2.3 VGG19_bn

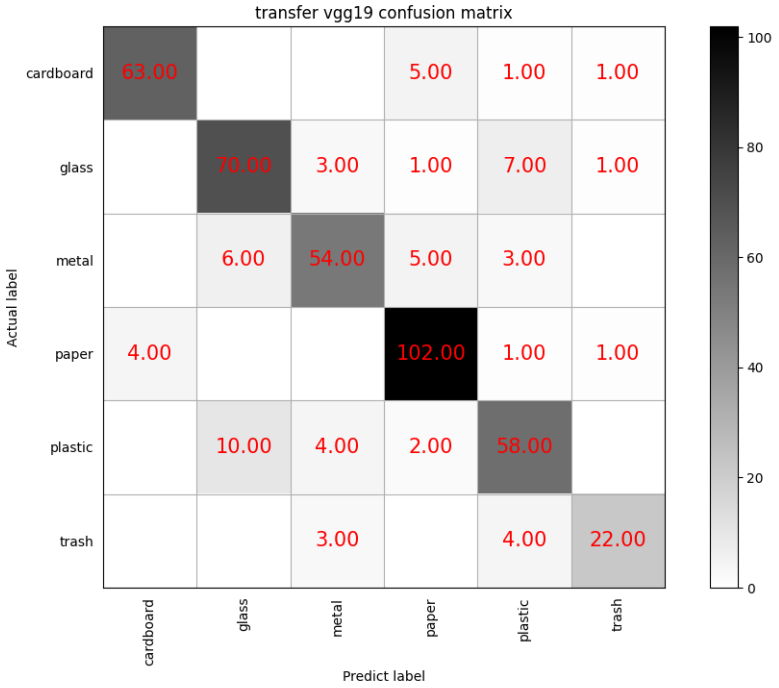




测试集上的训练结果

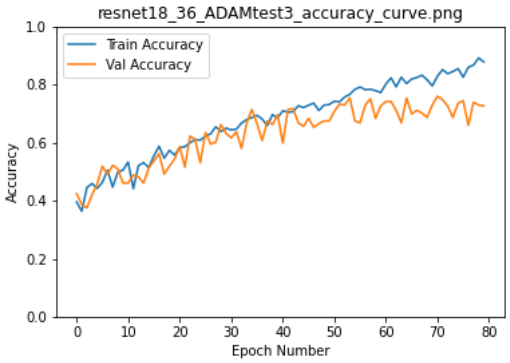
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.97447796	0.9350348	0.94431555	0.95591647	0.92575406	0.97679814
Recall	0.9	0.85365854	0.79411765	0.94444444	0.78378378	0.75862069
Precision	0.94029851	0.81395349	0.84375	0.88695652	0.78378378	0.88
F1 Score	0.91970803	0.83333333	0.81818182	0.91479821	0.78378378	0.81481481

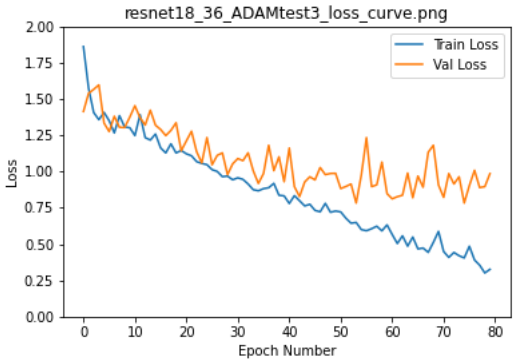
混淆矩阵：



/

3.2.4 ResNet18

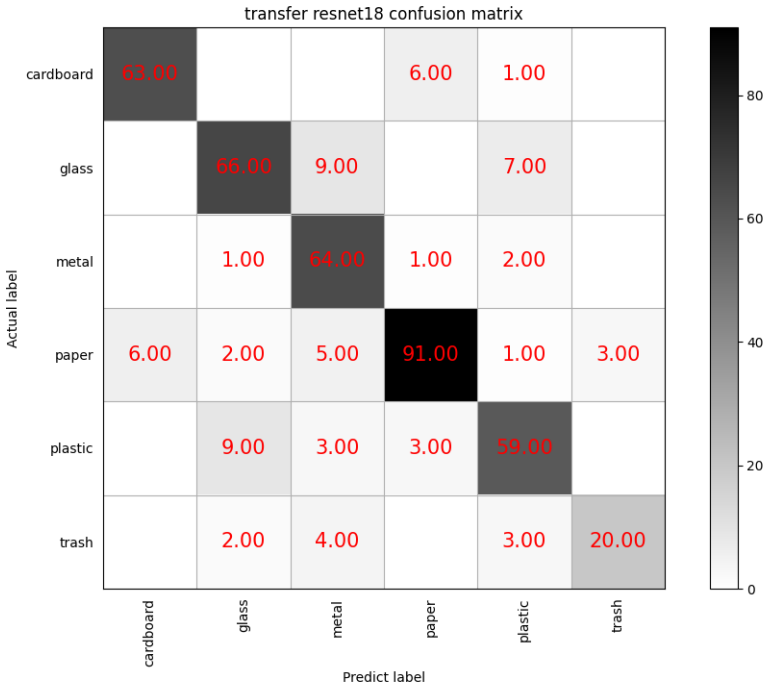




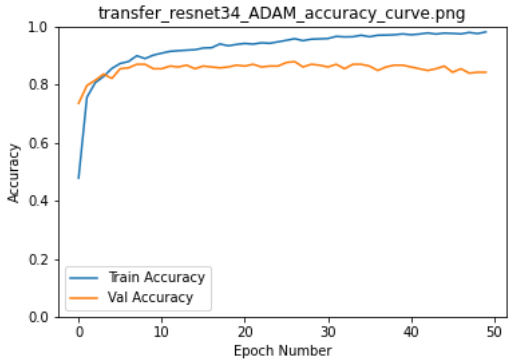
测试集上的结果：

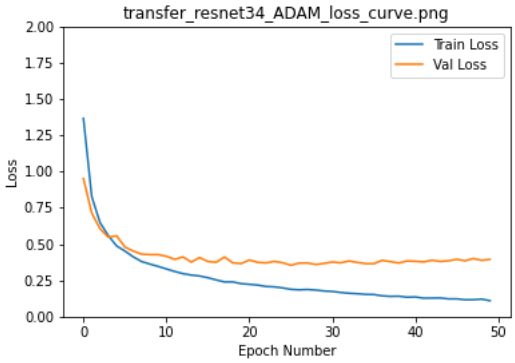
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.96983759	0.93039443	0.94199536	0.93735499	0.93271462	0.97215777
Recall	0.9	0.80487805	0.94117647	0.84259259	0.7972973	0.68965517
Precision	0.91304348	0.825	0.75294118	0.9009901	0.80821918	0.86956522
F1 Score	0.90647482	0.81481481	0.83660131	0.8708134	0.80272109	0.76923077

混淆矩阵：



3.2.5 ResNet34

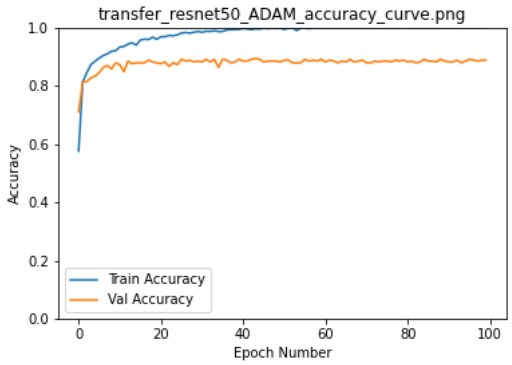


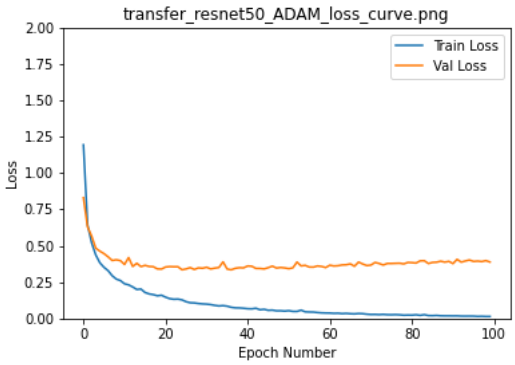


测试集上的结果：

类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.95823666	0.92343387	0.92575406	0.93039443	0.90719258	0.96519722
Recall	0.82857143	0.74390244	0.80882353	0.89814815	0.75675676	0.68965517
Precision	0.90625	0.83561644	0.74324324	0.8362069	0.71794872	0.76923077
F1 Score	0.86567164	0.78709677	0.77464789	0.86607143	0.73684211	0.72727273

3.2.6 ResNet50

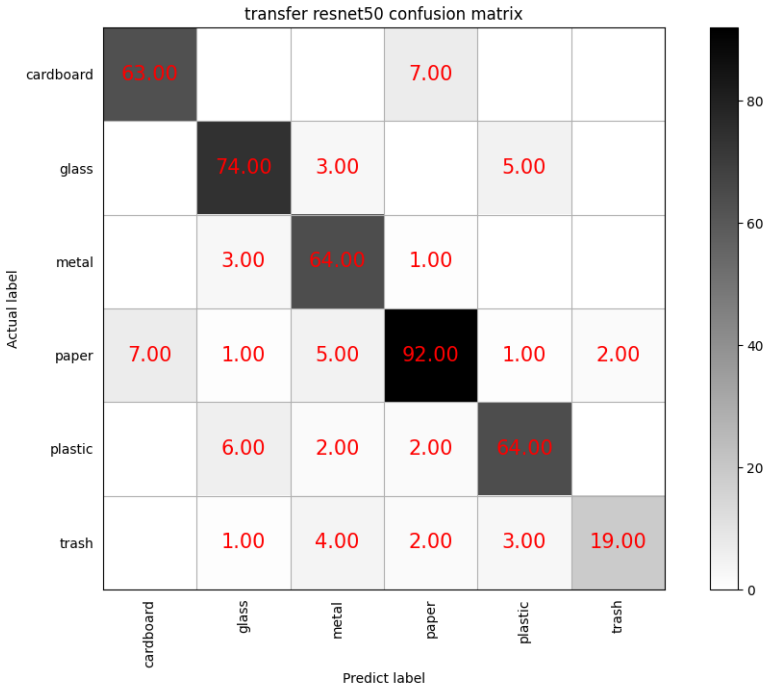




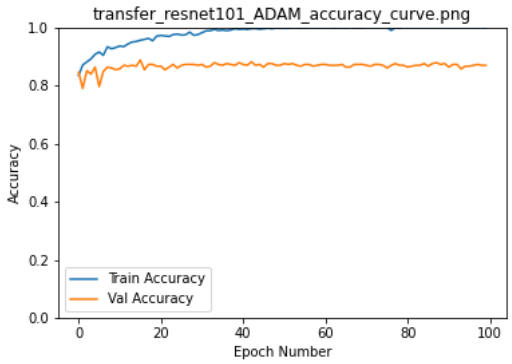
测试集上的结果：

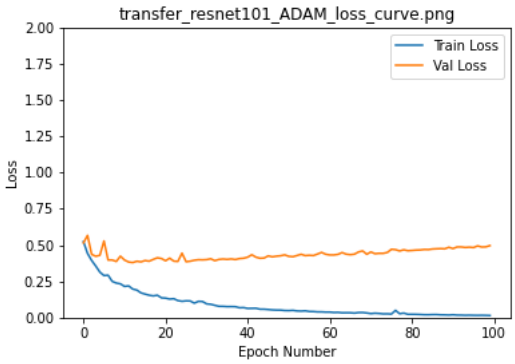
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.97911833	0.95359629	0.96983759	0.95359629	0.92343387	0.96519722
Recall	0.91428571	0.84146341	0.91176471	0.90740741	0.87837838	0.62068966
Precision	0.95522388	0.90789474	0.89855072	0.90740741	0.73033708	0.81818182
F1 Score	0.93430657	0.87341772	0.905109497	0.90740741	0.79754601	0.70588235

混淆矩阵：



3.2.7 ResNet101

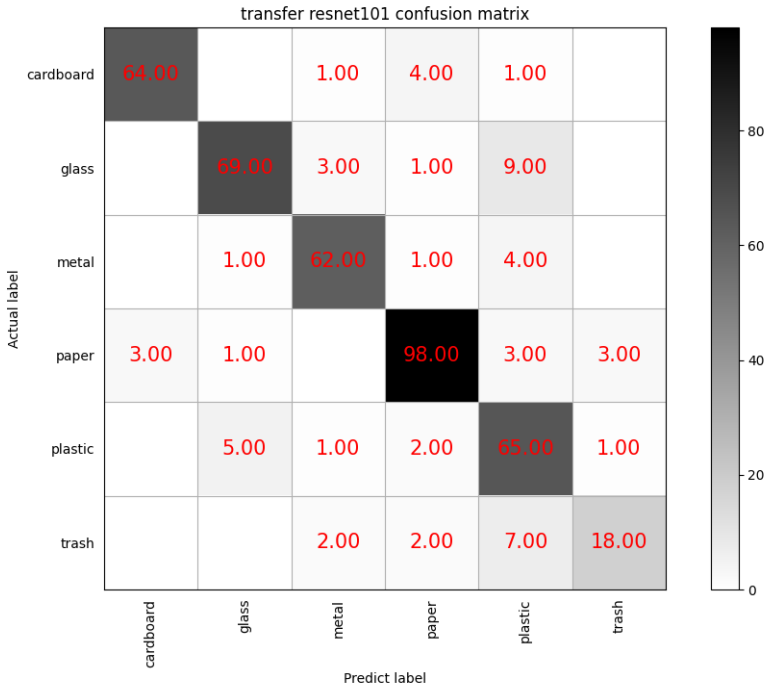




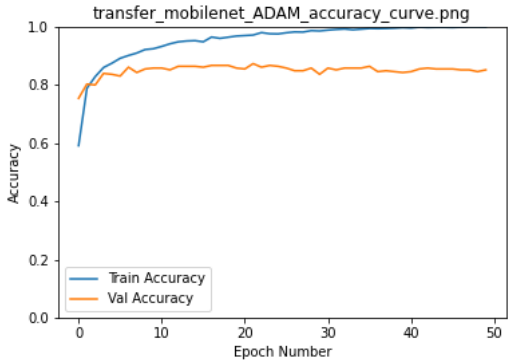
测试集上的结果：

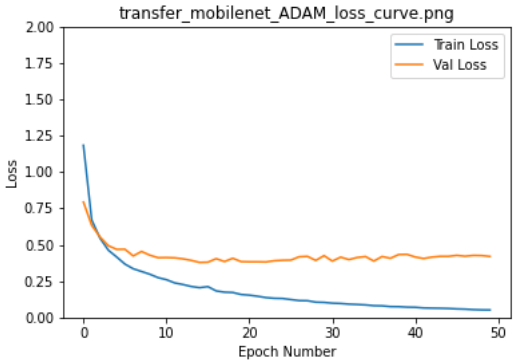
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.94663573	0.85382831	0.87703016	0.86774942	0.83758701	0.93967517
Recall	0.8	0.6097561	0.41176471	0.82407407	0.59459459	0.62068966
Precision	0.86153846	0.61728395	0.68292683	0.7007874	0.52380952	0.54545455
F1 Score	0.82962963	0.61349693	0.51376147	0.75744681	0.55696203	0.58064516

混淆矩阵：



3.2.8 MobileNet_V2

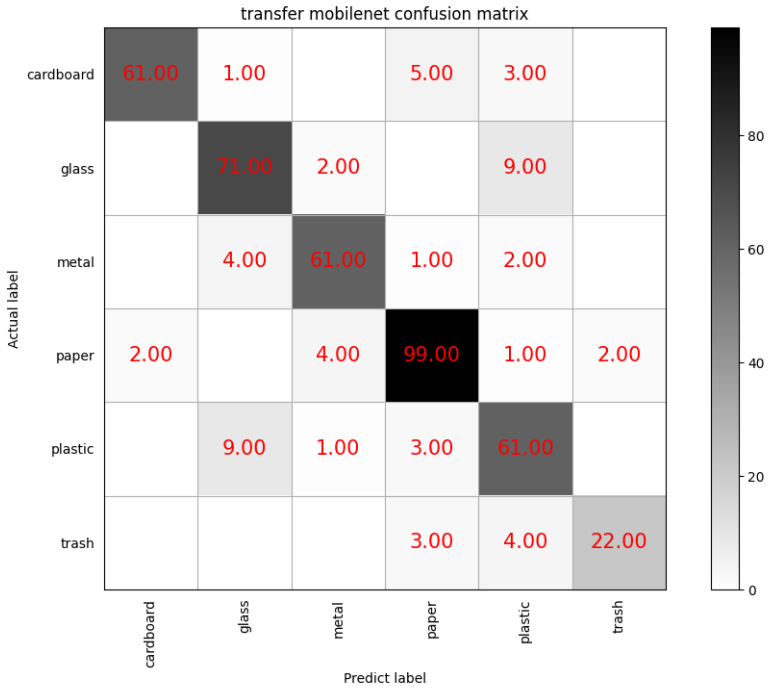




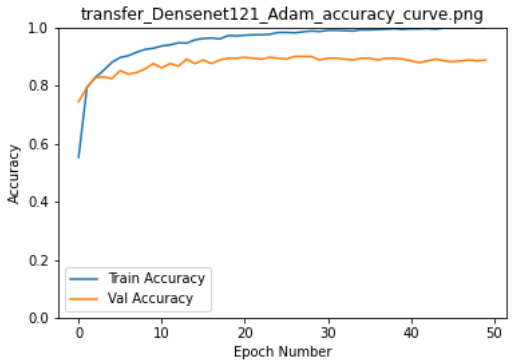
测试集上的结果：

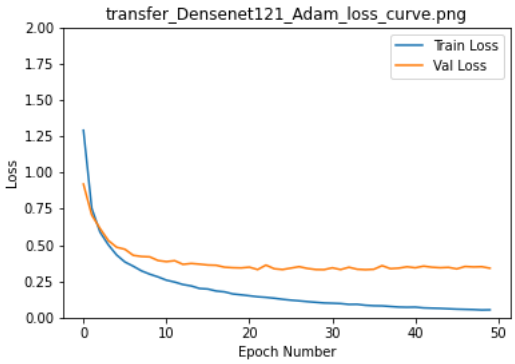
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.97447796	0.94199536	0.9675174	0.9512761	0.92575406	0.97911833
Recall	0.87142857	0.86585366	0.89705882	0.91666667	0.82432432	0.75862069
Precision	0.96825397	0.83529412	0.89705882	0.89189189	0.7625	0.91666667
F1 Score	0.91729323	0.8502994	0.89705882	0.90410959	0.79220779	0.83018868

混淆矩阵：



3.2.9 DenseNet121

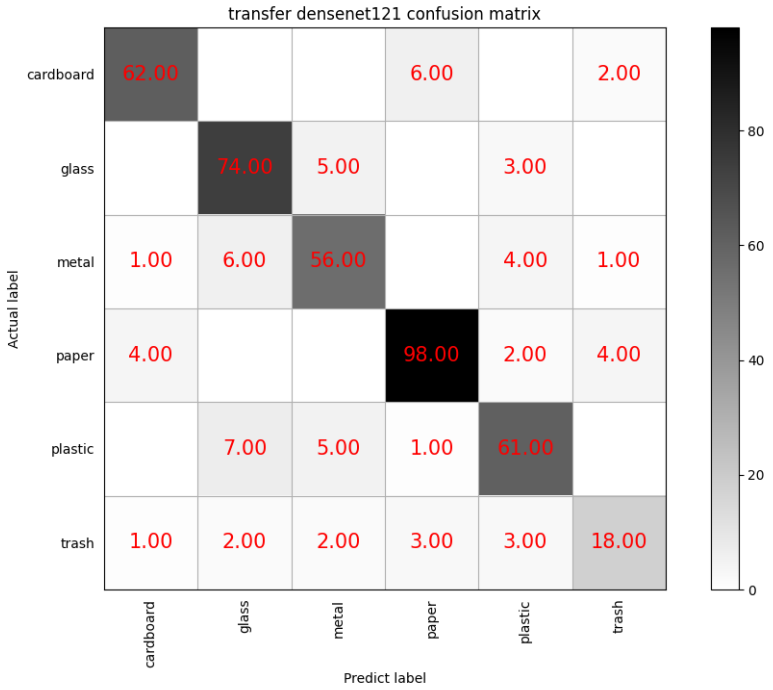




测试集上的结果：

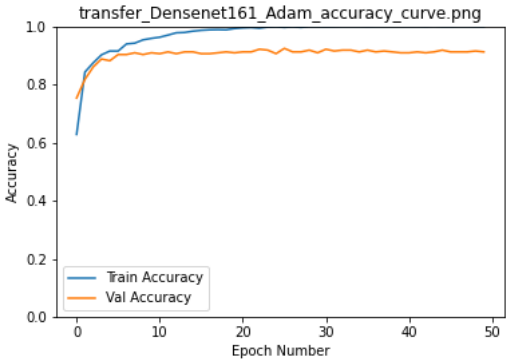
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.9675174	0.94663573	0.94431555	0.95359629	0.94199536	0.95823666
Recall	0.88571429	0.90243902	0.82352941	0.90740741	0.82432432	0.62068966
Precision	0.91176471	0.83146067	0.82352941	0.90740741	0.83561644	0.72
F1 Score	0.89855072	0.86549708	0.82352941	0.90740741	0.82993197	0.66666667

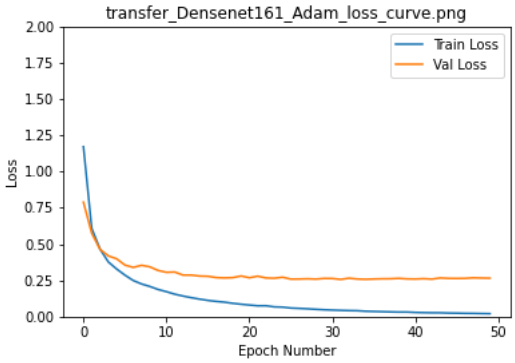
混淆矩阵：



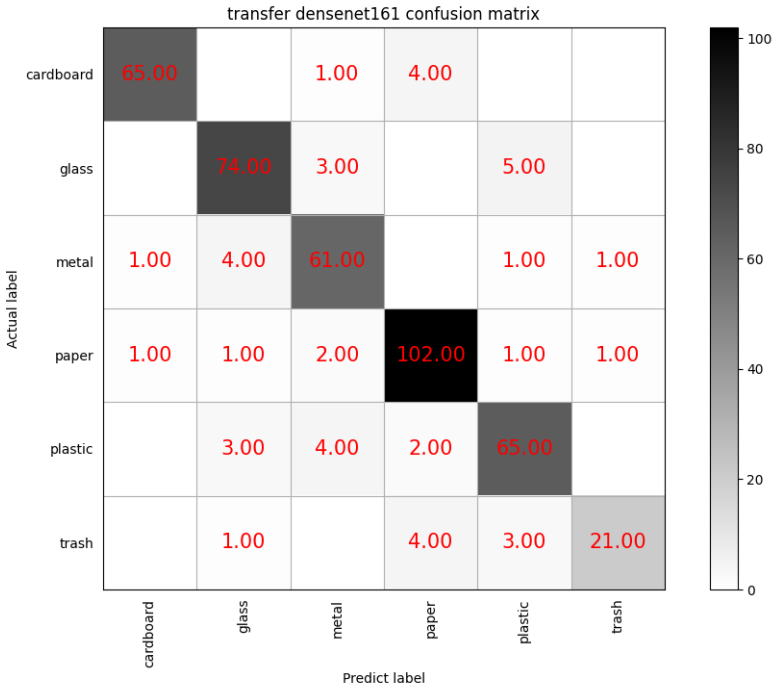
3.2.10 DenseNet161

测试集上的结果:





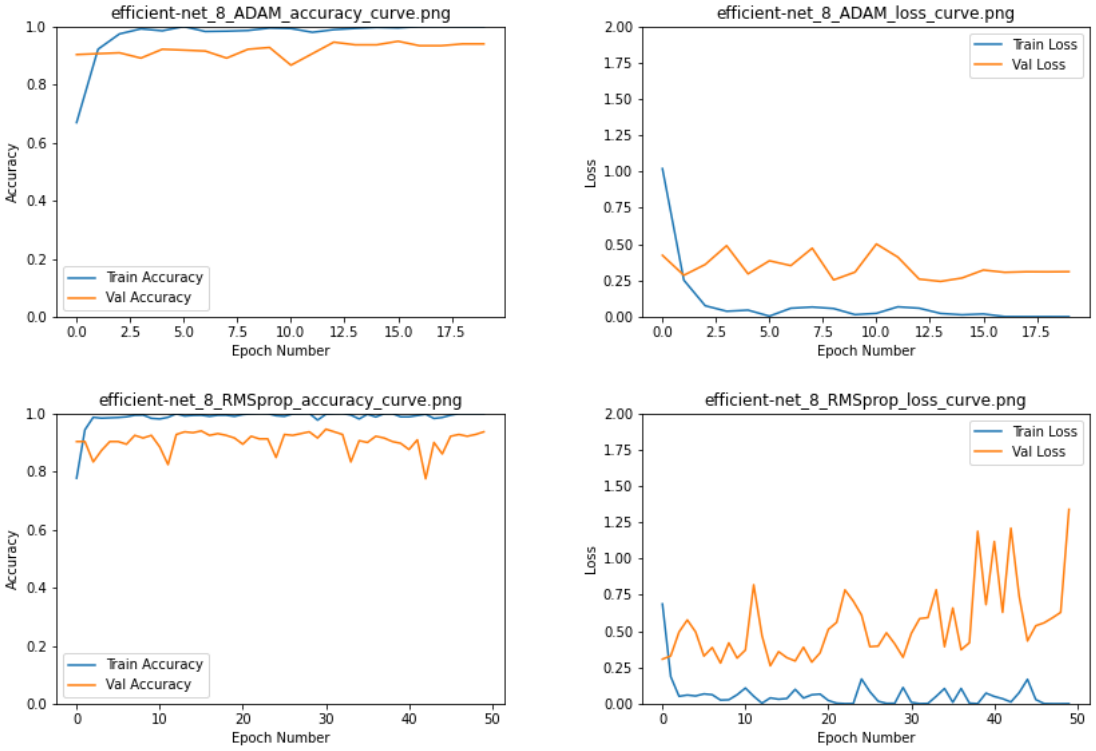
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.9837587	0.96055684	0.96055684	0.96287703	0.95591647	0.97679814
Recall	0.92857143	0.90243902	0.89705882	0.94444444	0.87837838	0.72413793
Precision	0.97014925	0.89156627	0.85915493	0.91071429	0.86666667	0.91304348
F1 Score	0.94890511	0.8969697	0.87769784	0.92727273	0.87248322	0.80769231

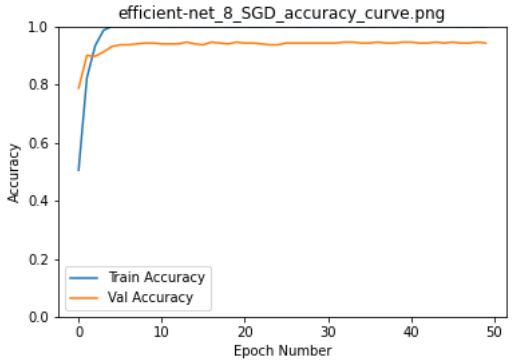


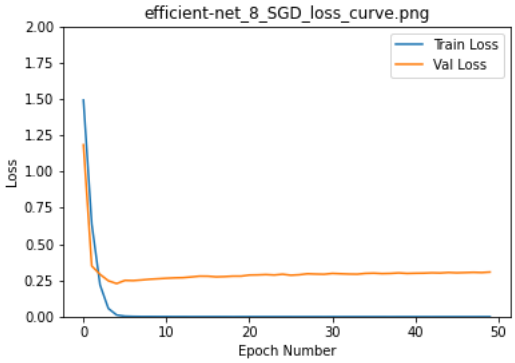
port/cm/

3.3 不同优化方法下的对比

3.3.1 Adam VS SGD VS RMSprop







Adam 数据

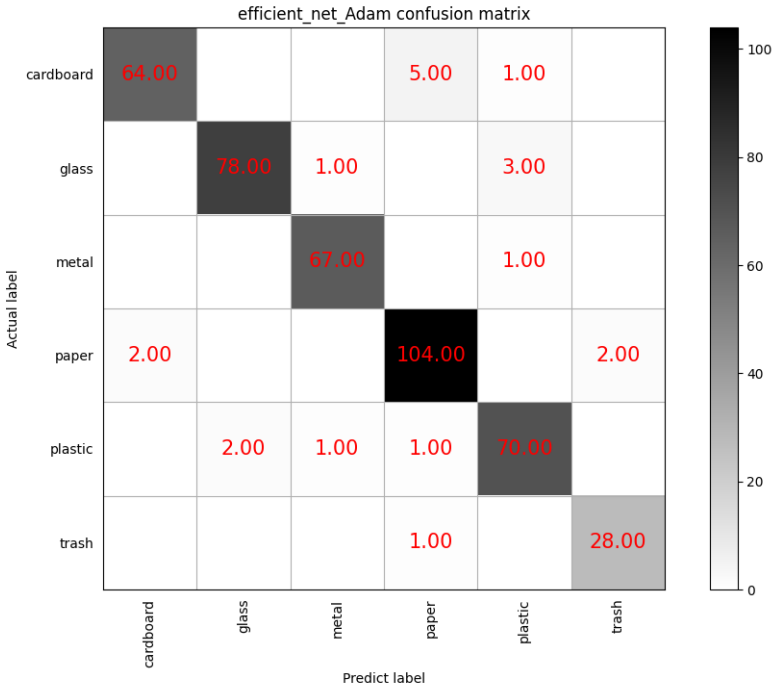
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.98	0.99	0.99	0.97	0.98	0.99
Recall	0.91	0.95	0.99	0.96	0.95	0.97
Precision	0.97	0.97	0.97	0.94	0.93	0.93
F1 Score	0.94	0.96	0.98	0.95	0.94	0.95

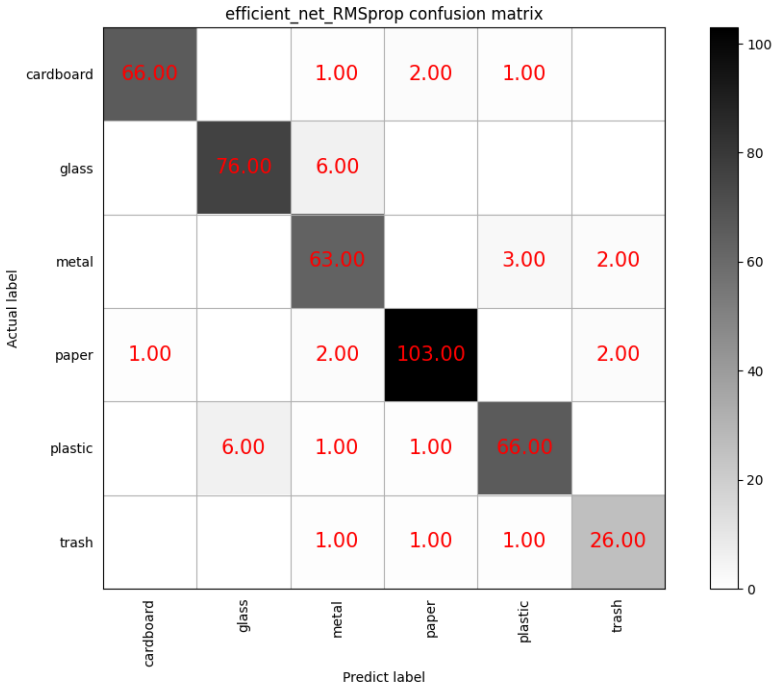
RMSprop 数据

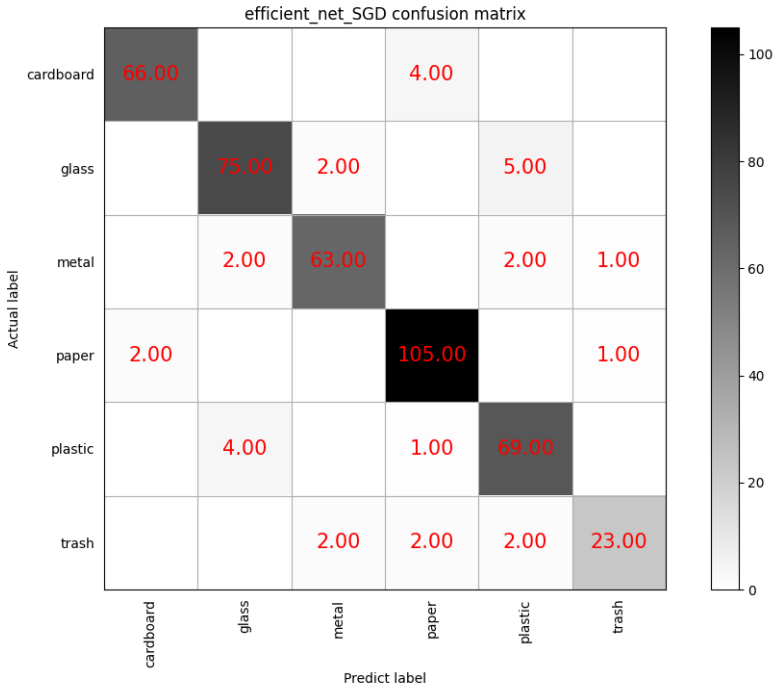
类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.99	0.97	0.96	0.98	0.97	0.98
Recall	0.94	0.93	0.93	0.95	0.89	0.9
Precision	0.99	0.93	0.85	0.96	0.93	0.87
F1 Score	0.96	0.93	0.89	0.96	0.91	0.88

SGD 数据

类别	Cardboard	Glass	Metal	Paper	Plastic	Trash
Accuracy	0.99	0.97	0.98	0.98	0.97	0.98
Recall	0.94	0.91	0.93	0.97	0.93	0.79
Precision	0.97	0.93	0.94	0.94	0.88	0.92
F1 Score	0.96	0.92	0.93	0.95	0.91	0.85


efficient_net_Adam confusion matrix
cardboard
glass
metal
paper
plastic
trash
Actual label
64.00 5.00 1.00
78.00 1.00 3.00
67.00 1.00
2.00 104.00 2.00
2.00 1.00 1.00 70.00
1.00 28.00
cardboard
glass
metal
paper
plastic
trash
Predict label
100
80
60
40
20
0


efficient_net_RMSprop confusion matrix
cardboard
glass
metal
paper
plastic
trash
Actual label
66.00 1.00 2.00 1.00
76.00 6.00
63.00 3.00 2.00
1.00 2.00 103.00 2.00
6.00 1.00 1.00 66.00
1.00 1.00 1.00 26.00
cardboard
glass
metal
paper
plastic
trash
Predict label
100
80
60
40
20
0



3.4 模型性能对比

训练硬件：RTX2080TI Batchsize:36 EfficientNet batchsize 是 12

模型种类	模型文件大小（MB）	训练时间（每个 epoch）
AlexNet	165	9.40
VGG16	537	10.984
VGG19	558	12.376
MobileNet	9.12	8.6232
ResNet18	44.8	8.1402
ResNet34	85.3	• 9.7024
ResNet50	94.3	10.497
ResNet101	171	18.81
DenseNet121	28.3	8.47
DenseNet161	107	11.747
EfficientNet_B7(RMSProp)	257	195.79
EfficientNet_B7(Adam)	257	165.29
EfficientNet_B7(SGD)	257	170.13

4 总结

4.1 迁移学习与普通监督学习的对比

- 迁移学习模型更容易收敛模型没有发生过拟合的现象

- 直接加载模型容易发生过拟合的现象
- 迁移模型在验证集上的表现更好，在测试集上也展现的较高的准确度

4.2 几种模型的对比

- Efficient Net 得到了最好的训练结果在各个分类上的准确率能够达到 92 以上, 召回率 F1 的 score 等表现都非常好
- MobileNet 展现了非常好的便携性和训练速度它在精度和性能上面做到了最好的平衡

4.3 模型层数与 BN 的使用

- ResNet 分别测试了 18、34、50、101 四种模型发现精度并不是随着模型的深度的增加而增加，50 与 101 的表现区别不大而对于 DenseNet 161 的表现比较出色但是训练时间较长
- 实验中有测试 vgg16 与 vgg16_bn 两种模型带了 BatchNormalization 的模型收敛的更快，同时最后的准确度较好。

4.4 几种训练方式的对比

- 在训练 EfficientNet 的时候发现 SGD 最后得到的结果非常好就是时间耗费有点多
- RMSprop 中间发生了非常多的动荡最后发生了过拟合
- Adam 算法它比较兼顾时间和性能

5 讨论与展望

未来可以在模型结构上进行修改从而达到更加好的结果。
可以压缩模型参数改善训练速度和储存空间的问题
对于过拟合的现象找到较合适的解决方案。