实验 4　PyTorch 与 CNN

姓名　学号　班级

日期

目录

1 PyTorch 入门

1.1 实验概览

1.1.1 实验内容

PyTorch 是基于 Torch 的 Python 机器学习库。本实验要求使用 PyTorch 对一个图片分类任务进行训练和测试，并尽可能优化测试准确率（test accuracy）。

1.1.2 实验原理

机器学习的本质是找到一个函数，使得对于给定的输入可以得到期望的输出。由于输入、输出一般都是很多维的，可以使用神经网络。

神经网络即为下图所示的网状结构，包含一个输入层（$\boldsymbol{x}$）、一个输出层（$\boldsymbol{y}$）与若干中间层（$\boldsymbol{h}$），每层包含若干神经元，层与层（神经元与神经元）之间有连接。

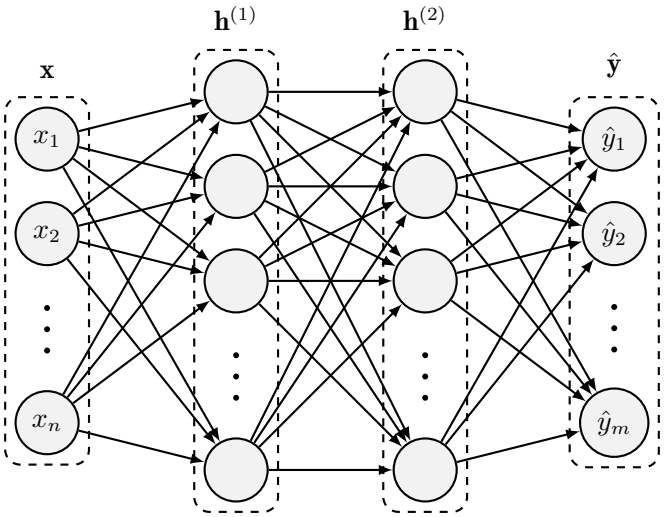

图 1: 神经网络示意图 [1]

在神经网络上有多种操作。图中每个箭头代表一个权重，通过计算若干个加权和可以得到下一层的值，这一操作称为前向传播，目的是计算输出。得到输出后，可以将其与预期值比较，得到损失（衡量结果好坏，越好值越小）。为了使损失尽量低，我们可以根据结果反过来改变前面的权重值，这被称为反向传播。

1.2 解决思路

实验要求我们补全计算测试准确率的部分，我们参照计算训练准确率的部分补全即可。

```
for batch_idx, (inputs, targets) in enumerate(testloader):
    inputs, targets = inputs, targets
    outputs = model(inputs)
    _, predicted = outputs.max(1)
    total += targets.size(0)
    correct += predicted.eq(targets).sum().item()
acc = 100. * correct / total
```

为了提高运行速度，我们将数据放在 GPU 上利用 CUDA 进行计算。

```
device = 'cpu'
if torch.cuda.is_available():
    device = 'cuda'
...
model = resnet20().to(device)
...
for batch_idx, (inputs, targets) in enumerate(trainloader):
    inputs, targets = inputs.to(device), targets.to(device)
```

实验要求我们使用各种办法提高最终的测试准确率。

我们将训练轮次提高到 20 轮。

```
start_epoch = 0
end_epoch = 19
```

同时，尝试使用不同的 lr 调整策略。这里我们使用：

```
lr.ReduceLROnPlateau(mode='max', factor=0.1, patience=1, min_lr=1e-3)
```

在完成一个训练轮次后，将测试准确率传递给调度器 `scheduler.step(metrics=acc)`。

这表示当测试准确率连续两轮不升反降时，我们将 lr 乘以 0.1，但最小不小于 0.001。这样做的好处是可以根据测试结果的好坏灵活地决定是否对 lr 进行调整，在准确率达到瓶颈时及时调低学习率，使测试准确率能尽快接近收敛。

此外，我们将 SGD 加上参数 `momentum=0.9`。我们可以将本例粗略地理解为一个凸优化问题，本例使用的优化器为 `optim.SGD`（Stochastic Gradient Descent），即在梯度下降（沿梯度方向更新参数）的基础上仅在一个 Batch 的数据上计算梯度，以提高运算效率。加上动量（momentum）参数相当于为 Gradient Descent 加上了惯性，每次更新参数时会取上一次的参数更新量，并加权加到本次的参数更新上，这可以减少震荡，使准确率曲线更加平滑。

在进行如上优化后，最终准确率可达 84.54%，比优化前的结果好约 10%。

1.3 运行结果

最终准确率：84.54%。

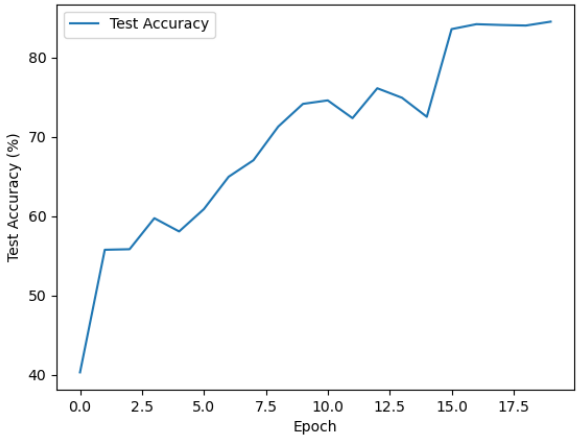

图 2: Test Accuracy 趋势图

最终模型请见 output/final.pth，逐 Epoch 快照请见 checkpoint/ckpt_*.pth。

1.4 结果分析与思考

1.4.1 结果分析

优化后的准确率相比优化前有较大提升，说明优化是有效的。

从图 2 可以直观看出，在 Epoch=13 和 Epoch=14 连续两轮准确率下降后，准确率突然提升，说明 lr 由 0.1 更新为 0.01，这证明我们优化后的 lr 调整策略是有效的。

1.4.2 思考

注：本节基于基础要求进行讨论，即原程序给定的无优化的 10 轮训练。

1. 训练准确率（train accuracy）和测试准确率（test accuracy）有什么关联和不同？

   从本质上来说，我们一开始将数据划分为训练集（50000 张）与测试集（10000 张）。Train accuracy 是在训练集上的运行结果，而 test accuracy 是在测试集上的运行结果。

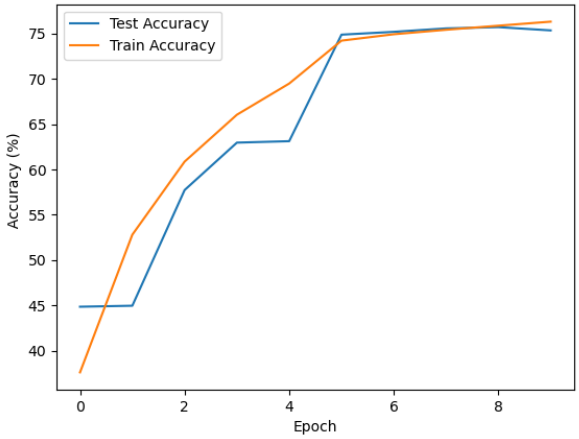

图 3: Test Accuracy 与 Train Accuracy 比较

由于神经网络的优化是在训练集上进行的，train accuracy 会呈现稳步提升的趋势，而 test accuracy 有可能因 overshoot 而出现波动。

就其相对大小而言，早期 test accuracy 可能高于 train accuracy，因为训练集在原始图片的基础上做了多重变换，如本例中的 `transforms.RandomCrop`（随机裁切）、`transforms.RandomHorizontalFlip`（随机水平翻转），而测试集没有做太多处理，所以训练集的难度实际上是高于测试集的。而后期 train accuracy 一般高于 test accuracy，因为神经网络是基于训练集进行训练的。

2. 在 lr 从 0.1 变到 0.01 后，准确率发生了什么变化？为什么？

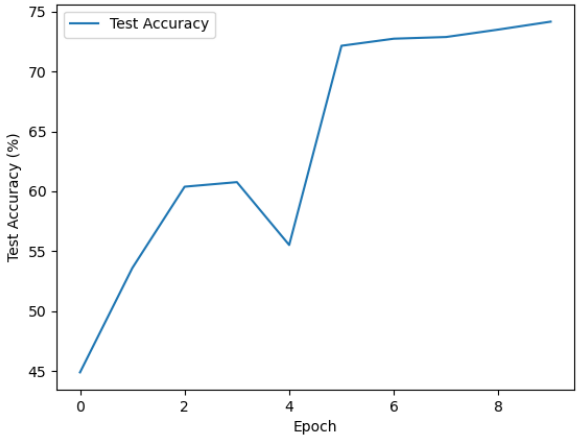

图 4: Test Accuracy 随 lr 的变化

如图所示，lr 由 0.1 变为 0.01 （由 Epoch=5 开始）后，准确率由原来的剧烈波动变为平稳上升。lr 变化之前，神经网络质量已经在 lr=0.1 的学习率下达到了较好水平，随后产生 overshoot，体现出 Epoch=4 处的剧烈波动。将 lr 变为 0.01 后，神经网络可以以更细的颗粒度继续优化，故体现为准确率平稳上升。

1.5 实验感想

通过本实验，我初步了解了机器学习的基本概念与神经网络的基本概念，补全了利用 PyTorch 进行图像分类的代码。同时，我对 test accuracy 等训练指标进行监测，提升对模型训练底层原理的理解。此外，还尝试对最终准确率进行优化。

2 CNN 图像检索

2.1 实验概览

2.1.1 实验内容

本实验要求构建一个图像库，使用各种卷积神经网络（CNN）提取图像的特征，并在库内检索外部图像的相似图像。

2.1.2 实验原理

以 ResNet50 为例，我们可以打印出其网络结构。

```
ResNet(
  (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3),
  ↪  bias=False)
  (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
  ↪  track_running_stats=True)
  (relu): ReLU(inplace=True)
  (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1,
  ↪  ceil_mode=False)
  (layer1): Sequential(
    ...
  )
  (layer2): Sequential(
    ...
  )
  (layer3): Sequential(
    ...
  )
  (layer4): Sequential(
    ...
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
  (fc): Linear(in_features=2048, out_features=1000, bias=True)
)
```

可以发现，最后一层为 FC 层（Fully Connected Layer），用于将 2048 维的图片向量转化为 1000 个类别的 probability。我们可以将 FC 层的前一层的输出提取出来，相当于提取了图片的特征。

```
x = model.conv1(x)
x = model.bn1(x)
x = model.relu(x)
x = model.maxpool(x)
x = model.layer1(x)
x = model.layer2(x)
x = model.layer3(x)
x = model.layer4(x)
x = model.avgpool(x)
```

我们将两个向量之间的夹角视作图像的相似度，进而实现相似图片检索。

2.2 解决思路

我们首先构建图像库，可以使用 ImageNet 数据集，因所需数据量较小，使用了 GitHub 上的一个子集 [2]。将 10 个类别的图像（每类别 5 张）分别放入 10 个文件夹，从网络上再单独下载每个类别的任意图像各一张，与文件夹同名，方便后期比较。

再看图像特征提取部分。除了 ResNet50，还可以使用其他神经网络来提取图像特征。以 AlexNet 为例，先打印其网络结构：

```
AlexNet(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(11, 11), stride=(4, 4), padding=(2, 2))
    (1): ReLU(inplace=True)
    (2): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1,
    ↪  ceil_mode=False)
    (3): Conv2d(64, 192, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (4): ReLU(inplace=True)
    (5): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1,
    ↪  ceil_mode=False)
    (6): Conv2d(192, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1,
    ↪  1))
    (7): ReLU(inplace=True)
    (8): Conv2d(384, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
    ↪  1))
    (9): ReLU(inplace=True)
    (10): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
    ↪  1))
    (11): ReLU(inplace=True)
    (12): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1,
    ↪  ceil_mode=False)
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(6, 6))
  (classifier): Sequential(
    (0): Dropout(p=0.5, inplace=False)
    (1): Linear(in_features=9216, out_features=4096, bias=True)
    (2): ReLU(inplace=True)
```

```
    (3): Dropout(p=0.5, inplace=False)
    (4): Linear(in_features=4096, out_features=4096, bias=True)
    (5): ReLU(inplace=True)
    (6): Linear(in_features=4096, out_features=1000, bias=True)
  )
)
```

类比上一章节的方法，不难发现只需运行 `features` 和 `avgpool` 两层即可。

```
x = model.features(x)
x = model.avgpool(x)
```

最后是图像检索，将向量归一化后计算内积，可得到两向量之间的夹角。

```
similarity = torch.dot(target_features[p1], image_features[p2])
```

对于每个类别的外部图像，我们检索图像库内与其最相似的 5 张图片，并将结果中属于其类别的占比作为准确率。此外，我们单独观察每个外部图像的最佳匹配是否与其为同类别，并单独计算 Top-1 准确率。

2.3 运行结果

ResNet50:

```
Matching ./target\0.png
Top 1 match (0.9957544207572937): ./dataset\0\ILSVRC2012_val_00001034.JPEG
Top 2 match (0.9955787658691406): ./dataset\0\ILSVRC2012_val_00006894.JPEG
Top 3 match (0.9955520629882812): ./dataset\1\ILSVRC2012_val_00002138.JPEG
Top 4 match (0.9954996109008789): ./dataset\7\ILSVRC2012_val_00001914.JPEG
Top 5 match (0.9954485893249512): ./dataset\0\ILSVRC2012_val_00007040.JPEG
Matching ./target\1.png
Top 1 match (0.9957874417304993): ./dataset\5\ILSVRC2012_val_00000657.JPEG
Top 2 match (0.9956573843955994): ./dataset\1\ILSVRC2012_val_00007197.JPEG
Top 3 match (0.9956352710723877): ./dataset\5\ILSVRC2012_val_00006012.JPEG
Top 4 match (0.9955881834030151): ./dataset\0\ILSVRC2012_val_00001034.JPEG
Top 5 match (0.9955872297286987): ./dataset\4\ILSVRC2012_val_00002214.JPEG
Matching ./target\2.png
Top 1 match (0.9956544637680054): ./dataset\2\ILSVRC2012_val_00000651.JPEG
Top 2 match (0.9953241348266602): ./dataset\5\ILSVRC2012_val_00006448.JPEG
```

```
Top 3 match (0.9952729344367981): ./dataset\2\ILSVRC2012_val_00002425.JPEG
Top 4 match (0.995227038860321): ./dataset\5\ILSVRC2012_val_00000657.JPEG
Top 5 match (0.9952166080474854): ./dataset\2\ILSVRC2012_val_00000747.JPEG
Matching ./target\3.png
Top 1 match (0.9957444667816162): ./dataset\3\ILSVRC2012_val_00000655.JPEG
Top 2 match (0.9956786632537842): ./dataset\4\ILSVRC2012_val_00002214.JPEG
Top 3 match (0.9956741333007812): ./dataset\8\ILSVRC2012_val_00004552.JPEG
Top 4 match (0.9956597089767456): ./dataset\3\ILSVRC2012_val_00002230.JPEG
Top 5 match (0.9956525564193726): ./dataset\5\ILSVRC2012_val_00000481.JPEG
Matching ./target\4.png
Top 1 match (0.9957857131958008): ./dataset\4\ILSVRC2012_val_00002214.JPEG
Top 2 match (0.9956408143043518): ./dataset\4\ILSVRC2012_val_00004748.JPEG
Top 3 match (0.9954824447631836): ./dataset\4\ILSVRC2012_val_00002683.JPEG
Top 4 match (0.9952818751335144): ./dataset\7\ILSVRC2012_val_00001276.JPEG
Top 5 match (0.995278000831604): ./dataset\5\ILSVRC2012_val_00000657.JPEG
Matching ./target\5.png
Top 1 match (0.9960558414459229): ./dataset\5\ILSVRC2012_val_00000481.JPEG
Top 2 match (0.9960392117500305): ./dataset\5\ILSVRC2012_val_00006448.JPEG
Top 3 match (0.9958397150039673): ./dataset\3\ILSVRC2012_val_00000655.JPEG
Top 4 match (0.9958265423774719): ./dataset\5\ILSVRC2012_val_00000657.JPEG
Top 5 match (0.9957547783851624): ./dataset\4\ILSVRC2012_val_00002214.JPEG
Matching ./target\6.png
Top 1 match (0.9958881735801697): ./dataset\6\ILSVRC2012_val_00000347.JPEG
Top 2 match (0.9957695007324219): ./dataset\7\ILSVRC2012_val_00001262.JPEG
Top 3 match (0.9957152605056763): ./dataset\8\ILSVRC2012_val_00001722.JPEG
Top 4 match (0.995651125907898): ./dataset\1\ILSVRC2012_val_00007197.JPEG
Top 5 match (0.9956128597259521): ./dataset\8\ILSVRC2012_val_00004552.JPEG
Matching ./target\7.png
Top 1 match (0.9954571723937988): ./dataset\8\ILSVRC2012_val_00004700.JPEG
Top 2 match (0.9954442977905273): ./dataset\7\ILSVRC2012_val_00001914.JPEG
Top 3 match (0.9954428672790527): ./dataset\7\ILSVRC2012_val_00001262.JPEG
Top 4 match (0.9954342842102051): ./dataset\8\ILSVRC2012_val_00004552.JPEG
Top 5 match (0.9954131245613098): ./dataset\9\ILSVRC2012_val_00004510.JPEG
Matching ./target\8.png
Top 1 match (0.9956743717193604): ./dataset\8\ILSVRC2012_val_00004700.JPEG
Top 2 match (0.9954821467399597): ./dataset\8\ILSVRC2012_val_00005862.JPEG
```

```
Top 3 match (0.9954208731651306): ./dataset\9\ILSVRC2012_val_00005732.JPEG
Top 4 match (0.995374321937561): ./dataset\9\ILSVRC2012_val_00004510.JPEG
Top 5 match (0.9953576922416687): ./dataset\8\ILSVRC2012_val_00004552.JPEG
Matching ./target\9.png
Top 1 match (0.996045708656311): ./dataset\7\ILSVRC2012_val_00001914.JPEG
Top 2 match (0.9960082173347473): ./dataset\9\ILSVRC2012_val_00002407.JPEG
Top 3 match (0.9958117604255676): ./dataset\8\ILSVRC2012_val_00004552.JPEG
Top 4 match (0.995793342590332): ./dataset\9\ILSVRC2012_val_00005732.JPEG
Top 5 match (0.9957758784294128): ./dataset\8\ILSVRC2012_val_00003107.JPEG
Accuracy (%): 46.0
Accuracy (top-1) (%): 70.0
```

AlexNet:

```
Matching ./target\0.png
Top 1 match (0.37292996048927307): ./dataset\0\ILSVRC2012_val_00003632.JPEG
Top 2 match (0.37170466780662537): ./dataset\0\ILSVRC2012_val_00001034.JPEG
Top 3 match (0.37105539441108704): ./dataset\1\ILSVRC2012_val_00000293.JPEG
Top 4 match (0.35198140144348145): ./dataset\1\ILSVRC2012_val_00007197.JPEG
Top 5 match (0.34770098328590393): ./dataset\0\ILSVRC2012_val_00003332.JPEG
Matching ./target\1.png
Top 1 match (0.2908517122268677): ./dataset\1\ILSVRC2012_val_00007197.JPEG
Top 2 match (0.28494158387184143): ./dataset\2\ILSVRC2012_val_00002425.JPEG
Top 3 match (0.28235486149787903): ./dataset\5\ILSVRC2012_val_00006448.JPEG
Top 4 match (0.28041037917137146): ./dataset\2\ILSVRC2012_val_00000747.JPEG
Top 5 match (0.27608662843704224): ./dataset\5\ILSVRC2012_val_00000481.JPEG
Matching ./target\2.png
Top 1 match (0.45887482166290283): ./dataset\2\ILSVRC2012_val_00000651.JPEG
Top 2 match (0.36213722825050354): ./dataset\2\ILSVRC2012_val_00002425.JPEG
Top 3 match (0.320125013589859): ./dataset\2\ILSVRC2012_val_00000747.JPEG
Top 4 match (0.2989462614059448): ./dataset\2\ILSVRC2012_val_00002191.JPEG
Top 5 match (0.2850078940391540 5): ./dataset\5\ILSVRC2012_val_00005669.JPEG
Matching ./target\3.png
Top 1 match (0.376450777053833): ./dataset\5\ILSVRC2012_val_00005669.JPEG
Top 2 match (0.32415249943733215): ./dataset\3\ILSVRC2012_val_00000655.JPEG
Top 3 match (0.28864389657974243): ./dataset\5\ILSVRC2012_val_00000481.JPEG
Top 4 match (0.28496599197387695): ./dataset\3\ILSVRC2012_val_00002403.JPEG
```

```
Top 5 match (0.28393781185150146): ./dataset\5\ILSVRC2012_val_00006448.JPEG
Matching ./target\4.png
Top 1 match (0.3656787872314453): ./dataset\4\ILSVRC2012_val_00004748.JPEG
Top 2 match (0.35376036167144775): ./dataset\5\ILSVRC2012_val_00006448.JPEG
Top 3 match (0.3535393178462982): ./dataset\4\ILSVRC2012_val_00002214.JPEG
Top 4 match (0.32445549964904785): ./dataset\5\ILSVRC2012_val_00000481.JPEG
Top 5 match (0.2715451717376709): ./dataset\4\ILSVRC2012_val_00000457.JPEG
Matching ./target\5.png
Top 1 match (0.4693306088447571): ./dataset\5\ILSVRC2012_val_00006448.JPEG
Top 2 match (0.3697713315486908): ./dataset\5\ILSVRC2012_val_00005669.JPEG
Top 3 match (0.33630791306495667): ./dataset\5\ILSVRC2012_val_00000481.JPEG
Top 4 match (0.331800639629364): ./dataset\4\ILSVRC2012_val_00004748.JPEG
Top 5 match (0.2891748547554016): ./dataset\5\ILSVRC2012_val_00000657.JPEG
Matching ./target\6.png
Top 1 match (0.3453251123428345): ./dataset\6\ILSVRC2012_val_00000347.JPEG
Top 2 match (0.33746588230133057): ./dataset\6\ILSVRC2012_val_00002732.JPEG
Top 3 match (0.3245896100997925): ./dataset\6\ILSVRC2012_val_00001326.JPEG
Top 4 match (0.318187952041626): ./dataset\6\ILSVRC2012_val_00001677.JPEG
Top 5 match (0.29854699969291687): ./dataset\6\ILSVRC2012_val_00004909.JPEG
Matching ./target\7.png
Top 1 match (0.3035036325454712): ./dataset\7\ILSVRC2012_val_00001914.JPEG
Top 2 match (0.2652604877948761): ./dataset\7\ILSVRC2012_val_00001276.JPEG
Top 3 match (0.24388737976551056): ./dataset\5\ILSVRC2012_val_00000657.JPEG
Top 4 match (0.24312818050384521): ./dataset\1\ILSVRC2012_val_00002138.JPEG
Top 5 match (0.2367219179868698): ./dataset\8\ILSVRC2012_val_00001722.JPEG
Matching ./target\8.png
Top 1 match (0.3927282691001892): ./dataset\8\ILSVRC2012_val_00005862.JPEG
Top 2 match (0.37346163392066956): ./dataset\8\ILSVRC2012_val_00001722.JPEG
Top 3 match (0.359977126121521): ./dataset\8\ILSVRC2012_val_00004700.JPEG
Top 4 match (0.3546079695224762): ./dataset\8\ILSVRC2012_val_00003107.JPEG
Top 5 match (0.27949976921081543): ./dataset\1\ILSVRC2012_val_00002138.JPEG
Matching ./target\9.png
Top 1 match (0.460374653339386): ./dataset\9\ILSVRC2012_val_00002407.JPEG
Top 2 match (0.3974464535713196): ./dataset\9\ILSVRC2012_val_00004510.JPEG
Top 3 match (0.39023250341415405): ./dataset\9\ILSVRC2012_val_00005732.JPEG
Top 4 match (0.3345493674278259): ./dataset\9\ILSVRC2012_val_00005721.JPEG
```

```
Top 5 match (0.31766676902770996): ./dataset\8\ILSVRC2012_val_00003107.JPEG
Accuracy (%): 64.0
Accuracy (top-1) (%): 90.0
```

2.4 结果分析与思考

ResNet50 准确率 46%，AlexNet 准确率 64%，属于可接受水平，说明我们提取图像特征的方式是正确的。

由于每个类别的图像只有 5 张，进行 Top-5 检索反而有可能降低准确率。我们单独观察 Top-1 准确率：ResNet50 70%，AlexNet 90%，均较高。

2.5 实验感想

通过本实验，我实践了通过神经网络提取图像特征的方法，对神经网络各部分的作用、神经网络相比传统方法的优势有了更深理解。

A 参考资料

[1] Fraser Love. Nntikz: Tikz diagrams for deep learning and neural networks, 2024.

[2] Na Dongbin ndb796. Ndb796/small-imagenet-validation-dataset-1000-classes: This is a subset of the imagenet validation dataset. this dataset has 5 images per class., 2021.