基于PaddlePaddle2.0的蝴蝶图像识别分类——利用预训练残差网络ResNet101模型

作者：陆平

1. 蝴蝶识别分类任务概述

人工智能技术的应用领域日趋广泛，新的智能应用层出不穷。本项目将利用人工智能技术来对蝴蝶图像进行分类，需要能对蝴蝶的类别、属性进行细粒度的识别分类。相关研究工作者能够根据采集到的蝴蝶图片，快速识别图中蝴蝶的种类。期望能够有助于提升蝴蝶识别工作的效率和精度。

2. 创建项目和挂载数据

数据集都来源于网络公开数据（和鲸社区）。图片中所涉及的蝴蝶总共有9个属，20个物种，文件genus.txt中描述了9个属名，species.txt描述了20个物种名。

在创建项目时，可以为该项目挂载Butterfly20蝴蝶数据集，即便项目重启，该挂载的数据集也不会被自动清除。具体方法如下：首先采用notebook方式构建项目，项目创建框中的最下方有个数据集选项，选择“+添加数据集”。然后，弹出搜索框，在关键词栏目输入“bufferfly20”，便能够查询到该数据集。最后，选中该数据集，可以自动在项目中挂载该数据集了。

需要注意的是，每次重新打开该项目，data文件夹下除了挂载的数据集，其他文件都将被删除。

被挂载的数据集会自动出现在data目录之下，通常是压缩包的形式。在data/data63004目录，其中有两个压缩文件，分别是Butterfly20.zip和Butterfly20_test.zip。也可以利用下载功能把数据集下载到本地进行训练。

3. 初探蝴蝶数据集

我们看看蝴蝶图像数据长什么样子?

首先，解压缩数据。类以下几个步骤：

第一步，把当前路径转换到data目录，可以使用命令!cd data。在AI studio nootbook中可以使用Linux命令，需要在命令的最前面加上英文的感叹号(!)。用&&可以连接两个命令。用\号可以换行写代码。需要注意的是，每次重新打开该项目，data文件夹下除了挂载的数据集，其他文件都会被清空。因此，如果把数据保存在data目录中，每次重新启动项目时，都需要解压缩一下。如果想省事持久化保存，可以把数据保存在work目录下。

实际上，!加某命令的模式，等价于python中的get_ipython().system('某命令')模式。

第二步，利用unzip命令，把压缩包解压到当前路径。unzip的-q参数代表执行时不显示任何信息。unzip的-o参数代表不必先询问用户，unzip执行后覆盖原有的文件。两个参数合起来，可以写为-qo。

第三步，用rm命令可以把一些文件夹给删掉，比如，__MACOSX文件夹

```
!cd data &&\
unzip -qo data63004/Butterfly20_test.zip &&\
unzip -qo data63004/Butterfly20.zip &&\
rm -r __MACOSX
```

接着，我们分析一下数据集，发现Butterfly20文件夹中有很多子文件夹，每个子文件夹下又有很多图片，每个子文件夹的名字都是蝴蝶属种的名字。由此，可以推测每个文件夹下是样本，而样本的标签就是子文件夹的名字。

我们绘制data/Butterfly20/001.Atrophaneura_horishanus文件夹下的图片006.jpg。根据百度百科，Atrophaneura horishanus是凤蝶科、曙凤蝶属的一个物种。

我们再绘制data/Butterfly20/002.Atrophaneura_varuna文件夹下的图片006.jpg。根据百度百科，Atrophaneura varuna对应的中文名称是“瓦曙凤蝶”，它是凤蝶科、曙凤蝶属的另一个物种。

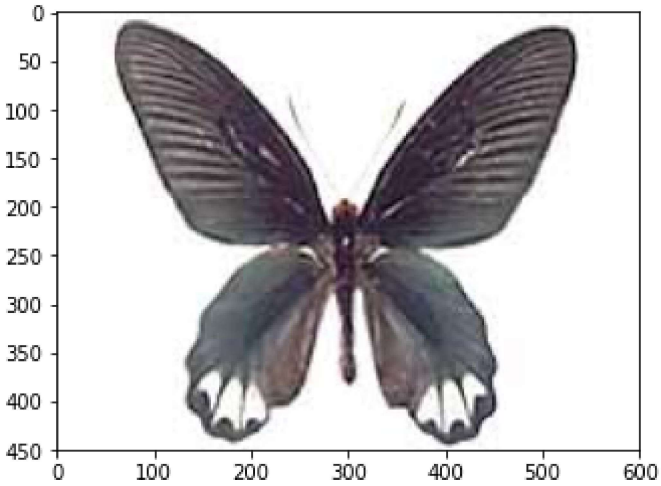
虽然乍一看蝴蝶都是相似的，但不同属种的蝴蝶在形状、颜色等细节方面还是存在很大的差别。

```
import matplotlib.pyplot as plt
import PIL.Image as Image

path='/home/aistudio/data/Butterfly20/001.Atrophaneura_horishanus/006.jpg'
img = Image.open(path)
plt.imshow(img)            #根据数组绘制图像
plt.show()                 #显示图像
```

```
<Figure size 640x480 with 1 Axes>
```

```
path='/home/aistudio/data/Butterfly20/002.Atrophaneura_varuna/006.jpg'
img = Image.open(path)
plt.imshow(img)            #根据数组绘制图像
plt.show()                 #显示图像
```

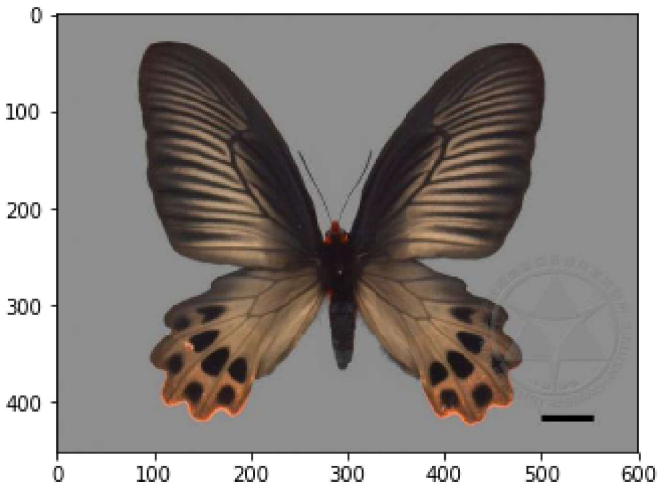


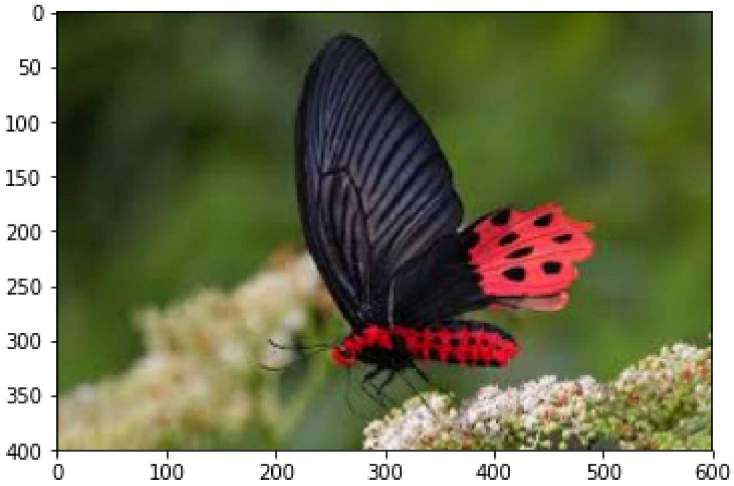
更具挑战的是，即便属于同一属种，不同的蝴蝶图片在角度、明暗、背景、姿态、颜色等方面均存在不小差别。甚至有的图片里面有多只蝴蝶。以下两张蝴蝶图片均出自同一个属种Atrophaneura horishanus。

```
path1='/home/aistudio/data/Butterfly20/001.Atrophaneura_horishanus/006.jpg'
path2='/home/aistudio/data/Butterfly20/001.Atrophaneura_horishanus/150.jpg'

img1 = Image.open(path1)
plt.imshow(img1)            #根据数组绘制图像
plt.show()

img2 = Image.open(path2)
plt.imshow(img2)            #根据数组绘制图像
plt.show()                 #显示图像
```





4. 准备数据

数据准备过程包括以下两个重点步骤：

一是建立样本数据读取路径与样本标签之间的关系。

二是构造读取器与数据预处理。可以写个自定义数据读取器，它继承于PaddlePaddle2.0的dataset类，在__getitem__方法中把自定义的预处理方法加载进去。

```
#以下代码用于建立样本数据读取路径与样本标签之间的关系
import os
import random

data_list = [] #用个列表保存每个样本的读取路径、标签

#由于属种名称本身是字符串，而输入模型的是数字。需要构造一个字典，把某个数字代表该属种名称。键是属种名称，值是整数。
label_list=[]
with open("/home/aistudio/data/species.txt") as f:
    for line in f:
        a,b = line.strip("\n").split(" ")
        label_list.append([b, int(a)-1])
label_dic = dict(label_list)

#获取Butterfly20目录下的所有子目录名称，保存进一个列表之中
class_list = os.listdir("/home/aistudio/data/Butterfly20")
class_list.remove('.DS_Store') #删掉列表中名为.DS_Store的元素，因为.DS_Store并没有样本。

for each in class_list:
    for f in os.listdir("/home/aistudio/data/Butterfly20/"+each):
        data_list.append(["/home/aistudio/data/Butterfly20/"+each+'/'+f,label_dic[each]])

#按文件顺序读取，可能造成很多属种图片存在序列相关，用random.shuffle方法把样本顺序彻底打乱。
random.shuffle(data_list)

#打印前十个，可以看出data_list列表中的每个元素是[样本读取路径, 样本标签]。
print(data_list[0:10])

#打印样本数量，一共有1866个样本。
print("样本数量是：{}".format(len(data_list)))
```

```
[['/home/aistudio/data/Butterfly20/006.Graphium_agamemnon/031.jpg', 5], ['/home/aistudio/data/Butterfly20/0
样本数量是：1866
```

```python
#以下代码用于构造读取器与数据预处理
#首先需要导入相关的模块
import paddle
from paddle.vision.transforms import Compose, ColorJitter, Resize,Transpose, Normalize
import cv2
import numpy as np
from PIL import Image
from paddle.io import Dataset

#自定义的数据预处理函数，输入原始图像，输出处理后的图像，可以借用paddle.vision.transforms的数据处理功能
def preprocess(img):
    transform = Compose([
        Resize(size=(224, 224)), #把数据长宽像素调成224*224
        Normalize(mean=[127.5, 127.5, 127.5], std=[127.5, 127.5, 127.5], data_format='HWC'), #标准化
        Transpose(), #原始数据形状维度是HWC格式，经过Transpose，转换为CHW格式
        ])
    img = transform(img).astype("float32")
    return img

#自定义数据读取器
class Reader(Dataset):
    def __init__(self, data, is_val=False):
        super().__init__()
        #在初始化阶段，把数据集划分训练集和测试集。由于在读取前样本已经被打乱顺序，取20%的样本作为测试集，80%的样本作为训练集
        self.samples = data[-int(len(data)*0.2):] if is_val else data[:-int(len(data)*0.2)]

    def __getitem__(self, idx):
        #处理图像
        img_path = self.samples[idx][0] #得到某样本的路径
        img = Image.open(img_path)
        if img.mode != 'RGB':
            img = img.convert('RGB')
        img = preprocess(img) #数据预处理--这里仅包括简单数据预处理，没有用到数据增强

        #处理标签
        label = self.samples[idx][1] #得到某样本的标签
        label = np.array([label], dtype="int64") #把标签数据类型转成int64
        return img, label

    def __len__(self):
        #返回每个Epoch中图片数量
        return len(self.samples)

#生成训练数据集实例
train_dataset = Reader(data_list, is_val=False)

#生成测试数据集实例
eval_dataset = Reader(data_list, is_val=True)

#打印一个训练样本
#print(train_dataset[1136][0])
print(train_dataset[1136][0].shape)
print(train_dataset[1136][1])
```

```
(3, 224, 224)
[18]
```

5. 建立模型

为了提升探索速度，建议首先选用比较成熟的基础模型，看看基础模型所能够达到的准确度。之后再试试模型融合，准确度是否有提升。最后可以试试自己独创模型。

为简便，这里直接采用101层的残差网络ResNet，并且采用预训练模式。为什么要采用预训练模型呢？因为通常模型参数采用随机初始化，而预训练模型参数初始值是一个比较确定的值。这个参数初始值是经历了大量任务训练而得来的，比如用CIFAR图像识别任务来训练模型，得到的参数。虽然蝴蝶识别任务和CIFAR图像识别任务是不同的，但可能存在某些机器视觉上的共性。用预训练模型可能能够较快地得到比较好的准确度。

在PaddlePaddle2.0中，使用预训练模型只需要设定模型参数pretained=True。值得注意的是，预训练模型得出的结果类别是1000维度，要用个线性变换，把类别转化为20维度。

```
#定义模型
class MyNet(paddle.nn.Layer):
    def __init__(self):
        super(MyNet,self).__init__()
        self.layer=paddle.vision.models.resnet50(pretrained=True)
        self.fc = paddle.nn.Linear(1000, 512)
        self.fc2=paddle.nn.Linear(512,20)
        self.flatten=paddle.nn.Flatten()
        self.relu=paddle.nn.ReLU()
        self.dropout=paddle.nn.Dropout(0.2)
    #网络的前向计算过程
    def forward(self,x):
        x=self.layer(x)
        x=self.flatten(x)
        x=self.fc(x)
        x=self.relu(x)
        x=self.dropout(x)
        x=self.fc2(x)
        return x
```

6. 应用高阶API训练模型

一是定义输入数据形状大小和数据类型。

二是实例化模型。如果要用高阶API，需要用Paddle.Model()对模型进行封装，如model = paddle.Model(model,inputs=input_define,labels=label_define)。

三是定义优化器。这个使用Adam优化器，学习率设置为0.0001，优化器中的学习率(learning_rate)参数很重要。要是训练过程中得到的准确率呈震荡状态，忽大忽小，可以试试进一步把学习率调低。

四是准备模型。这里用到高阶API，model.prepare()。

五是训练模型。这里用到高阶API，model.fit()。参数意义详见下述代码注释。

```
#定义输入
input_define = paddle.static.InputSpec(shape=[-1,3,224,224], dtype="float32", name="img")
label_define = paddle.static.InputSpec(shape=[-1,1], dtype="int64", name="label")

#实例化网络对象并定义优化器等训练逻辑
model = MyNet()
model = paddle.Model(model,inputs=input_define,labels=label_define) #用Paddle.Model()对模型进行封装
optimizer = paddle.optimizer.Adam(learning_rate=0.0001, parameters=model.parameters())
#上述优化器中的学习率(learning_rate)参数很重要。要是训练过程中得到的准确率呈震荡状态，忽大忽小，可以试试进一步把学习率调低。

model.prepare(optimizer=optimizer, #指定优化器
              loss=paddle.nn.CrossEntropyLoss(), #指定损失函数
              metrics=paddle.metric.Accuracy()) #指定评估方法
```

```
model.fit(train_data=train_dataset,      #训练数据集
          eval_data=eval_dataset,          #测试数据集
          batch_size=64,                   #一个批次的样本数量
          epochs=80,                       #迭代轮次
          save_dir="/home/aistudio/canshu", #把模型参数、优化器参数保存至自定义的文件夹
          save_freq=1,                     #设定每隔多少个epoch保存模型参数及优化器参数
          log_freq=100                      #打印日志的频率
)
```

```
The loss value printed in the log is the current step, and the metric is the average value of previous step
Epoch 1/80


/opt/conda/envs/python35-paddle120-env/lib/python3.7/site-packages/paddle/nn/layer/norm.py:636: UserWarning
  "When training, we now always track global mean and variance.")


step 24/24 - loss: 0.7714 - acc: 0.3972 - 507ms/step
save checkpoint at /home/aistudio/canshu/0
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1938 - acc: 0.6917 - 489ms/step
Eval samples: 373
Epoch 2/80
step 24/24 - loss: 0.3709 - acc: 0.9069 - 935ms/step
save checkpoint at /home/aistudio/canshu/1
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.0282 - acc: 0.7721 - 483ms/step
Eval samples: 373
Epoch 3/80
step 24/24 - loss: 0.0584 - acc: 0.9826 - 455ms/step
save checkpoint at /home/aistudio/canshu/2
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 0.8295 - acc: 0.8204 - 462ms/step
Eval samples: 373
Epoch 4/80
step 24/24 - loss: 0.2530 - acc: 0.9973 - 459ms/step
save checkpoint at /home/aistudio/canshu/3
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 0.8545 - acc: 0.8257 - 442ms/step
Eval samples: 373
Epoch 5/80
step 24/24 - loss: 0.0273 - acc: 0.9967 - 821ms/step
save checkpoint at /home/aistudio/canshu/4
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 0.9356 - acc: 0.8257 - 488ms/step

Eval samples: 373
Epoch 6/80
step 24/24 - loss: 0.0139 - acc: 0.9980 - 571ms/step
save checkpoint at /home/aistudio/canshu/5
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 0.9872 - acc: 0.8150 - 455ms/step
Eval samples: 373
Epoch 7/80
step 24/24 - loss: 0.0122 - acc: 0.9973 - 507ms/step
```

```
save checkpoint at /home/aistudio/canshu/6
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 0.8279 - acc: 0.8365 - 798ms/step
Eval samples: 373
Epoch 8/80
step 24/24 - loss: 0.0384 - acc: 0.9973 - 491ms/step
save checkpoint at /home/aistudio/canshu/7
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 0.8554 - acc: 0.8365 - 487ms/step
Eval samples: 373
Epoch 9/80
step 24/24 - loss: 0.0064 - acc: 0.9980 - 486ms/step
save checkpoint at /home/aistudio/canshu/8
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 0.9236 - acc: 0.8123 - 451ms/step
Eval samples: 373
Epoch 10/80
step 24/24 - loss: 0.0067 - acc: 0.9967 - 465ms/step
save checkpoint at /home/aistudio/canshu/9
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 0.8967 - acc: 0.8204 - 453ms/step
Eval samples: 373
Epoch 11/80
step 24/24 - loss: 0.0134 - acc: 0.9967 - 489ms/step
save checkpoint at /home/aistudio/canshu/10
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 0.9570 - acc: 0.8204 - 445ms/step
Eval samples: 373
Epoch 12/80
step 24/24 - loss: 0.0065 - acc: 0.9973 - 485ms/step
save checkpoint at /home/aistudio/canshu/11
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 0.9877 - acc: 0.8365 - 468ms/step
Eval samples: 373
Epoch 13/80
step 24/24 - loss: 0.0038 - acc: 0.9973 - 468ms/step
save checkpoint at /home/aistudio/canshu/12
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 0.9993 - acc: 0.8338 - 464ms/step
Eval samples: 373
Epoch 14/80
step 24/24 - loss: 0.0016 - acc: 0.9980 - 483ms/step
save checkpoint at /home/aistudio/canshu/13
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 0.9908 - acc: 0.8365 - 493ms/step
Eval samples: 373
Epoch 15/80
step 24/24 - loss: 8.7768e-04 - acc: 0.9973 - 482ms/step
save checkpoint at /home/aistudio/canshu/14
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 0.9416 - acc: 0.8311 - 491ms/step
Eval samples: 373
Epoch 16/80
```

```
step 24/24 - loss: 0.0803 - acc: 0.9967 - 486ms/step
save checkpoint at /home/aistudio/canshu/15
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 0.9738 - acc: 0.8391 - 488ms/step
Eval samples: 373
Epoch 17/80
step 24/24 - loss: 0.0012 - acc: 0.9980 - 467ms/step
save checkpoint at /home/aistudio/canshu/16
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 0.9511 - acc: 0.8445 - 482ms/step
Eval samples: 373
Epoch 18/80
step 24/24 - loss: 6.3367e-04 - acc: 0.9973 - 451ms/step
save checkpoint at /home/aistudio/canshu/17
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 0.9814 - acc: 0.8525 - 501ms/step
Eval samples: 373
Epoch 19/80
step 24/24 - loss: 4.6470e-04 - acc: 0.9973 - 492ms/step
save checkpoint at /home/aistudio/canshu/18
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 0.9448 - acc: 0.8445 - 493ms/step
Eval samples: 373
Epoch 20/80
step 24/24 - loss: 0.0087 - acc: 0.9973 - 449ms/step
save checkpoint at /home/aistudio/canshu/19
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 0.9668 - acc: 0.8445 - 539ms/step
Eval samples: 373
Epoch 21/80
step 24/24 - loss: 5.0827e-04 - acc: 0.9980 - 476ms/step
save checkpoint at /home/aistudio/canshu/20
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.0084 - acc: 0.8445 - 488ms/step
Eval samples: 373
Epoch 22/80
step 24/24 - loss: 3.6125e-04 - acc: 0.9973 - 457ms/step
save checkpoint at /home/aistudio/canshu/21
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.0282 - acc: 0.8472 - 461ms/step
Eval samples: 373
Epoch 23/80
step 24/24 - loss: 0.0084 - acc: 0.9967 - 795ms/step
save checkpoint at /home/aistudio/canshu/22
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1248 - acc: 0.8391 - 469ms/step
Eval samples: 373
Epoch 24/80
step 24/24 - loss: 0.0013 - acc: 0.9980 - 494ms/step
save checkpoint at /home/aistudio/canshu/23
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1500 - acc: 0.8338 - 456ms/step
Eval samples: 373
```

```
Epoch 25/80
step 24/24 - loss: 0.0072 - acc: 0.9980 - 469ms/step
save checkpoint at /home/aistudio/canshu/24
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1305 - acc: 0.8338 - 475ms/step
Eval samples: 373
Epoch 26/80
step 24/24 - loss: 0.0043 - acc: 0.9973 - 472ms/step
save checkpoint at /home/aistudio/canshu/25
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.0352 - acc: 0.8472 - 484ms/step
Eval samples: 373
Epoch 27/80
step 24/24 - loss: 0.0016 - acc: 0.9980 - 659ms/step
save checkpoint at /home/aistudio/canshu/26
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.0324 - acc: 0.8445 - 481ms/step
Eval samples: 373
Epoch 28/80
step 24/24 - loss: 0.0065 - acc: 0.9967 - 465ms/step
save checkpoint at /home/aistudio/canshu/27
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.0742 - acc: 0.8499 - 491ms/step
Eval samples: 373
Epoch 29/80
step 24/24 - loss: 0.0036 - acc: 0.9987 - 493ms/step
save checkpoint at /home/aistudio/canshu/28
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.0076 - acc: 0.8525 - 456ms/step
Eval samples: 373
Epoch 30/80
step 24/24 - loss: 5.1931e-04 - acc: 0.9980 - 463ms/step
save checkpoint at /home/aistudio/canshu/29
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1375 - acc: 0.8499 - 474ms/step
Eval samples: 373
Epoch 31/80
step 24/24 - loss: 0.0063 - acc: 0.9973 - 547ms/step
save checkpoint at /home/aistudio/canshu/30
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.0977 - acc: 0.8472 - 484ms/step
Eval samples: 373
Epoch 32/80
step 24/24 - loss: 3.0170e-04 - acc: 0.9973 - 520ms/step
save checkpoint at /home/aistudio/canshu/31
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.0065 - acc: 0.8338 - 479ms/step
Eval samples: 373
Epoch 33/80
step 24/24 - loss: 0.0015 - acc: 0.9980 - 446ms/step
save checkpoint at /home/aistudio/canshu/32
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.0495 - acc: 0.8338 - 489ms/step
```

```
Eval samples: 373
Epoch 34/80
step 24/24 - loss: 6.8332e-04 - acc: 0.9980 - 493ms/step
save checkpoint at /home/aistudio/canshu/33
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1248 - acc: 0.8284 - 493ms/step
Eval samples: 373
Epoch 35/80
step 24/24 - loss: 5.1613e-04 - acc: 0.9967 - 458ms/step
save checkpoint at /home/aistudio/canshu/34
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1603 - acc: 0.8418 - 460ms/step
Eval samples: 373
Epoch 36/80
step 24/24 - loss: 1.5649e-04 - acc: 0.9967 - 460ms/step
save checkpoint at /home/aistudio/canshu/35
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1336 - acc: 0.8445 - 495ms/step
Eval samples: 373
Epoch 37/80
step 24/24 - loss: 2.8873e-04 - acc: 0.9987 - 467ms/step
save checkpoint at /home/aistudio/canshu/36
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1163 - acc: 0.8231 - 447ms/step
Eval samples: 373
Epoch 38/80
step 24/24 - loss: 3.9291e-04 - acc: 0.9987 - 456ms/step
save checkpoint at /home/aistudio/canshu/37
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1905 - acc: 0.8365 - 445ms/step
Eval samples: 373
Epoch 39/80
step 24/24 - loss: 6.6256e-04 - acc: 0.9980 - 481ms/step
save checkpoint at /home/aistudio/canshu/38
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1891 - acc: 0.8365 - 463ms/step
Eval samples: 373
Epoch 40/80
step 24/24 - loss: 0.0068 - acc: 0.9967 - 457ms/step
save checkpoint at /home/aistudio/canshu/39
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1754 - acc: 0.8365 - 441ms/step
Eval samples: 373
Epoch 41/80
step 24/24 - loss: 0.0360 - acc: 0.9960 - 445ms/step
save checkpoint at /home/aistudio/canshu/40
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2446 - acc: 0.8311 - 537ms/step
Eval samples: 373
Epoch 42/80
step 24/24 - loss: 0.0010 - acc: 0.9973 - 592ms/step
save checkpoint at /home/aistudio/canshu/41
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
```

```
step 6/6 - loss: 1.2485 - acc: 0.8284 - 468ms/step
Eval samples: 373
Epoch 43/80
step 24/24 - loss: 0.0038 - acc: 0.9980 - 472ms/step
save checkpoint at /home/aistudio/canshu/42
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2157 - acc: 0.8365 - 454ms/step
Eval samples: 373
Epoch 44/80
step 24/24 - loss: 0.0028 - acc: 0.9960 - 458ms/step
save checkpoint at /home/aistudio/canshu/43
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1802 - acc: 0.8311 - 472ms/step
Eval samples: 373
Epoch 45/80
step 24/24 - loss: 0.0014 - acc: 0.9973 - 466ms/step
save checkpoint at /home/aistudio/canshu/44
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2510 - acc: 0.8391 - 499ms/step
Eval samples: 373
Epoch 46/80
step 24/24 - loss: 0.0015 - acc: 0.9967 - 506ms/step
save checkpoint at /home/aistudio/canshu/45
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2099 - acc: 0.8338 - 476ms/step
Eval samples: 373
Epoch 47/80
step 24/24 - loss: 0.0015 - acc: 0.9973 - 472ms/step
save checkpoint at /home/aistudio/canshu/46
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1837 - acc: 0.8365 - 554ms/step
Eval samples: 373
Epoch 48/80
step 24/24 - loss: 1.4137e-04 - acc: 0.9973 - 476ms/step
save checkpoint at /home/aistudio/canshu/47
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2703 - acc: 0.8311 - 510ms/step
Eval samples: 373
Epoch 49/80
step 24/24 - loss: 0.0076 - acc: 0.9973 - 461ms/step
save checkpoint at /home/aistudio/canshu/48
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1970 - acc: 0.8391 - 438ms/step
Eval samples: 373
Epoch 50/80
step 24/24 - loss: 0.0012 - acc: 0.9967 - 896ms/step
save checkpoint at /home/aistudio/canshu/49
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2333 - acc: 0.8231 - 479ms/step
Eval samples: 373
Epoch 51/80
step 24/24 - loss: 0.0017 - acc: 0.9960 - 466ms/step
save checkpoint at /home/aistudio/canshu/50
Eval begin...
```

```
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2720 - acc: 0.8391 - 479ms/step
Eval samples: 373
Epoch 52/80
step 24/24 - loss: 0.0041 - acc: 0.9980 - 460ms/step
save checkpoint at /home/aistudio/canshu/51
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.3451 - acc: 0.8445 - 446ms/step
Eval samples: 373
Epoch 53/80
step 24/24 - loss: 2.8336e-04 - acc: 0.9973 - 459ms/step
save checkpoint at /home/aistudio/canshu/52
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.3554 - acc: 0.8418 - 462ms/step
Eval samples: 373
Epoch 54/80
step 24/24 - loss: 2.1189e-04 - acc: 0.9987 - 441ms/step
save checkpoint at /home/aistudio/canshu/53
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2908 - acc: 0.8391 - 497ms/step
Eval samples: 373
Epoch 55/80
step 24/24 - loss: 2.5852e-04 - acc: 0.9967 - 453ms/step
save checkpoint at /home/aistudio/canshu/54
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.3168 - acc: 0.8418 - 443ms/step
Eval samples: 373
Epoch 56/80
step 24/24 - loss: 0.0362 - acc: 0.9973 - 469ms/step
save checkpoint at /home/aistudio/canshu/55
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.3405 - acc: 0.8365 - 961ms/step
Eval samples: 373
Epoch 57/80
step 24/24 - loss: 0.0026 - acc: 0.9980 - 464ms/step
save checkpoint at /home/aistudio/canshu/56
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2395 - acc: 0.8418 - 443ms/step
Eval samples: 373
Epoch 58/80
step 24/24 - loss: 1.0914e-04 - acc: 0.9980 - 464ms/step
save checkpoint at /home/aistudio/canshu/57
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.3219 - acc: 0.8338 - 487ms/step
Eval samples: 373
Epoch 59/80
step 24/24 - loss: 6.6677e-04 - acc: 0.9987 - 471ms/step
save checkpoint at /home/aistudio/canshu/58
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.3046 - acc: 0.8338 - 442ms/step
Eval samples: 373
Epoch 60/80
step 24/24 - loss: 0.0011 - acc: 0.9973 - 444ms/step
save checkpoint at /home/aistudio/canshu/59
```

```
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2470 - acc: 0.8365 - 443ms/step
Eval samples: 373
Epoch 61/80
step 24/24 - loss: 0.0016 - acc: 0.9973 - 512ms/step
save checkpoint at /home/aistudio/canshu/60
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2444 - acc: 0.8365 - 498ms/step
Eval samples: 373
Epoch 62/80
step 24/24 - loss: 0.0343 - acc: 0.9973 - 497ms/step
save checkpoint at /home/aistudio/canshu/61
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2627 - acc: 0.8365 - 498ms/step
Eval samples: 373
Epoch 63/80
step 24/24 - loss: 0.0060 - acc: 0.9973 - 461ms/step
save checkpoint at /home/aistudio/canshu/62
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2465 - acc: 0.8391 - 491ms/step
Eval samples: 373
Epoch 64/80
step 24/24 - loss: 0.0134 - acc: 0.9980 - 518ms/step
save checkpoint at /home/aistudio/canshu/63
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2220 - acc: 0.8365 - 463ms/step
Eval samples: 373
Epoch 65/80
step 24/24 - loss: 9.9867e-04 - acc: 0.9980 - 454ms/step
save checkpoint at /home/aistudio/canshu/64
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2250 - acc: 0.8365 - 451ms/step
Eval samples: 373
Epoch 66/80
step 24/24 - loss: 4.1172e-04 - acc: 0.9973 - 453ms/step
save checkpoint at /home/aistudio/canshu/65
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2338 - acc: 0.8311 - 459ms/step
Eval samples: 373
Epoch 67/80
step 24/24 - loss: 2.0886e-04 - acc: 0.9973 - 452ms/step
save checkpoint at /home/aistudio/canshu/66
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2522 - acc: 0.8418 - 501ms/step
Eval samples: 373
Epoch 68/80
step 24/24 - loss: 2.5482e-04 - acc: 0.9973 - 652ms/step
save checkpoint at /home/aistudio/canshu/67
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2666 - acc: 0.8338 - 509ms/step
Eval samples: 373
Epoch 69/80
step 24/24 - loss: 7.7463e-05 - acc: 0.9980 - 470ms/step
```

```
save checkpoint at /home/aistudio/canshu/68
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1849 - acc: 0.8418 - 473ms/step
Eval samples: 373
Epoch 70/80
step 24/24 - loss: 1.6319e-04 - acc: 0.9980 - 450ms/step
save checkpoint at /home/aistudio/canshu/69
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1997 - acc: 0.8391 - 475ms/step
Eval samples: 373
Epoch 71/80
step 24/24 - loss: 3.0103e-04 - acc: 0.9980 - 471ms/step
save checkpoint at /home/aistudio/canshu/70
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2446 - acc: 0.8365 - 472ms/step
Eval samples: 373
Epoch 72/80
step 24/24 - loss: 4.4041e-04 - acc: 0.9987 - 494ms/step
save checkpoint at /home/aistudio/canshu/71
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1863 - acc: 0.8391 - 495ms/step
Eval samples: 373
Epoch 73/80
step 24/24 - loss: 8.0260e-04 - acc: 0.9980 - 464ms/step
save checkpoint at /home/aistudio/canshu/72
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1768 - acc: 0.8418 - 474ms/step
Eval samples: 373
Epoch 74/80
step 24/24 - loss: 8.0764e-05 - acc: 0.9973 - 489ms/step
save checkpoint at /home/aistudio/canshu/73
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2066 - acc: 0.8391 - 508ms/step
Eval samples: 373
Epoch 75/80
step 24/24 - loss: 0.0015 - acc: 0.9987 - 459ms/step
save checkpoint at /home/aistudio/canshu/74
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.2847 - acc: 0.8445 - 472ms/step
Eval samples: 373
Epoch 76/80
step 24/24 - loss: 3.3070e-04 - acc: 0.9967 - 467ms/step
save checkpoint at /home/aistudio/canshu/75
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1844 - acc: 0.8418 - 492ms/step
Eval samples: 373
Epoch 77/80
step 24/24 - loss: 0.0016 - acc: 0.9993 - 444ms/step
save checkpoint at /home/aistudio/canshu/76
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1945 - acc: 0.8391 - 461ms/step
Eval samples: 373
Epoch 78/80
```

```
step 24/24 - loss: 7.0712e-05 - acc: 0.9967 - 477ms/step
save checkpoint at /home/aistudio/canshu/77
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1684 - acc: 0.8525 - 453ms/step
Eval samples: 373
Epoch 79/80
step 24/24 - loss: 0.0011 - acc: 0.9973 - 485ms/step
save checkpoint at /home/aistudio/canshu/78
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1363 - acc: 0.8445 - 467ms/step
Eval samples: 373
Epoch 80/80
step 24/24 - loss: 1.6439e-04 - acc: 0.9980 - 450ms/step
save checkpoint at /home/aistudio/canshu/79
Eval begin...
The loss value printed in the log is the current batch, and the metric is the average value of previous ste
step 6/6 - loss: 1.1791 - acc: 0.8472 - 443ms/step
Eval samples: 373
save checkpoint at /home/aistudio/canshu/final
```

7. 应用已经训练好的模型进行预测

如果是要参加建模比赛，通常赛事组织方会提供待预测的数据集，我们需要利用自己构建的模型，来对待预测数据集合中的数据标签进行预测。也就是说，我们其实并不知道到其真实标签是什么，只有比赛的组织方知道真实标签，我们的模型预测结果越接近真实结果，那么分数也就越高。

预测流程分为以下几个步骤：

一是构建数据读取器。因为预测数据集没有标签，该读取器写法和训练数据读取器不一样，建议重新写一个类，继承于Dataset基类。

二是实例化模型。如果要用高阶API，需要用Paddle.Model()对模型进行封装，如paddle.Model(MyNet(),inputs=input_define)，由于是预测模型，所以仅设定输入数据格式就好了。

三是读取刚刚训练好的参数。这个保存在/home/aistudio/work目录之下，如果指定的是final则是最后一轮训练后的结果。可以指定其他轮次的结果，比如model.load('/home/aistudio/work/30')，这里用到了高阶API，model.load()

四是准备模型。这里用到高阶API，model.prepare()。

五是读取待预测集合中的数据，利用已经训练好的模型进行预测。

六是结果保存。

```python
class InferDataset(Dataset):
    def __init__(self, img_path=None):
        """
        数据读取Reader(推理)
        :param img_path: 推理单张图片
        """
        super().__init__()
        if img_path:
            self.img_paths = [img_path]
        else:
            raise Exception("请指定需要预测对应图片路径")

    def __getitem__(self, index):
        # 获取图像路径
```

```
        img_path = self.img_paths[index]
        # 使用Pillow来读取图像数据并转成Numpy格式
        img = Image.open(img_path)
        if img.mode != 'RGB':
            img = img.convert('RGB')
        img = preprocess(img) #数据预处理--这里仅包括简单数据预处理，没有用到数据增强
        return img

    def __len__(self):
        return len(self.img_paths)

#实例化推理模型
model = paddle.Model(MyNet(),inputs=input_define)

#读取刚刚训练好的参数
model.load('/home/aistudio/canshu/final')

#准备模型
model.prepare()

#得到待预测数据集中每个图像的读取路径
infer_list=[]
with open("/home/aistudio/data/testpath.txt") as file_pred:
    for line in file_pred:
        infer_list.append("/home/aistudio/data/"+line.strip())

#模型预测结果通常是个数，需要获得其对应的文字标签。这里需要建立一个字典。
def get_label_dict2():
    label_list2=[]
    with open("/home/aistudio/data/species.txt") as filess:
        for line in filess:
            a,b = line.strip("\n").split(" ")
            label_list2.append([int(a)-1, b])
    label_dic2 = dict(label_list2)
    return label_dic2

label_dict2 = get_label_dict2()
#print(label_dict2)

#利用训练好的模型进行预测
results=[]
for infer_path in infer_list:
    infer_data = InferDataset(infer_path)
    result = model.predict(test_data=infer_data)[0] #关键代码，实现预测功能
    result = paddle.to_tensor(result)
    result = np.argmax(result.numpy()) #获得最大值所在的序号
    results.append("{}".format(label_dict2[result])) #查找该序号所对应的标签名字

#把结果保存起来
with open("work/result.txt", "w") as f:
    for r in results:
        f.write("{}\n".format(r))
```

```
Predict begin...
step 1/1 [==============================] - 49ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 51ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 56ms/step
Predict samples: 1
```

```
Predict begin...
step 1/1 [==============================] - 60ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 44ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 63ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 39ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 35ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 35ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
```

```
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 40ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 39ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 39ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 35ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 35ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
```

```
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 39ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 39ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 35ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 39ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 35ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
```

```
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 35ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 35ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 35ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 40ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 41ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
```

```
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 39ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 35ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 35ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 35ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 35ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 35ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 41ms/step
Predict samples: 1
Predict begin...
```

```
step 1/1 [==============================] - 45ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 42ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 45ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 39ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 39ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
```

```
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 36ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 39ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 39ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 49ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 50ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 49ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 43ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 53ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 46ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 72ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 62ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 56ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 72ms/step
```

```
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 83ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 54ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 106ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 161ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 81ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 111ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 132ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 81ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 97ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 138ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 75ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 97ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 139ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 76ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 76ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 65ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 63ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 64ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 62ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 64ms/step
Predict samples: 1
Predict begin...
```

```
step 1/1 [==============================] - 63ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 62ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 62ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 64ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 66ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 63ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 62ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 61ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 64ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 63ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 63ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 63ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 62ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 64ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 63ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 64ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 63ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 66ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 65ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 63ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 62ms/step
Predict samples: 1
```

```
Predict begin...
step 1/1 [==============================] - 63ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 63ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 63ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 62ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 61ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 64ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 63ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 68ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 57ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 38ms/step
Predict samples: 1
Predict begin...
step 1/1 [==============================] - 37ms/step
Predict samples: 1
```

该模型在测试集上的准确度约为85%左右。接下来，便可以探索使用数据增强、自定义模型等方式，看看能否取得更好的结果。