下载预训练权重文件

In [2]:
```
!pip install download
```

```
Looking in indexes: http://repo.myhuaweicloud.com/repository/pypi/simple
Collecting download
  Downloading http://repo.myhuaweicloud.com/repository/pypi/packages/37/45/01e7455a9659528e77a414b222326d4c525796e4f571bbabcb2e
0ff3d1f4/download-0.3.5-py3-none-any.whl (8.8 kB)
Requirement already satisfied: tqdm in /home/ma-user/anaconda3/envs/MindSpore/lib/python3.9/site-packages (from download) (4.6
6.4)
Requirement already satisfied: requests in /home/ma-user/anaconda3/envs/MindSpore/lib/python3.9/site-packages (from download)
(2.23.0)
Requirement already satisfied: six in /home/ma-user/anaconda3/envs/MindSpore/lib/python3.9/site-packages (from download) (1.16.
0)
Collecting urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
  Downloading http://repo.myhuaweicloud.com/repository/pypi/packages/56/aa/4ef5aa67a9a62505db124a5cb5262332d1d4153462eb8fd89c9f
a41e5d92/urllib3-1.25.11-py2.py3-none-any.whl (127 kB)
     |████████████████████████████████| 127 kB 63.5 MB/s eta 0:00:01
Requirement already satisfied: idna<3,>=2.5 in /home/ma-user/anaconda3/envs/MindSpore/lib/python3.9/site-packages (from request
s->download) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /home/ma-user/anaconda3/envs/MindSpore/lib/python3.9/site-packages (from re
quests->download) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /home/ma-user/anaconda3/envs/MindSpore/lib/python3.9/site-packages (from r
equests->download) (2024.2.2)
Installing collected packages: urllib3, download
  Attempting uninstall: urllib3
    Found existing installation: urllib3 1.26.7
    Uninstalling urllib3-1.26.7:
      Successfully uninstalled urllib3-1.26.7
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is th
e source of the following dependency conflicts.
modelarts 1.4.28 requires lxml==5.1.0, but you have lxml 4.9.3 which is incompatible.
modelarts 1.4.28 requires matplotlib==3.5.2, but you have matplotlib 3.5.1 which is incompatible.
modelarts 1.4.28 requires prettytable<=3.7.0, but you have prettytable 3.10.0 which is incompatible.
modelarts 1.4.28 requires requests==2.31.0, but you have requests 2.23.0 which is incompatible.
modelarts 1.4.28 requires tqdm<=4.66.1, but you have tqdm 4.66.4 which is incompatible.
modelarts 1.4.28 requires typing-extensions==4.7.1, but you have typing-extensions 4.11.0 which is incompatible.
modelarts 1.4.28 requires urllib3==1.26.18, but you have urllib3 1.25.11 which is incompatible.
Successfully installed download-0.3.5 urllib3-1.25.11
WARNING: You are using pip version 21.0.1; however, version 24.3.1 is available.
You should consider upgrading via the '/home/ma-user/anaconda3/envs/MindSpore/bin/python3.9 -m pip install --upgrade pip' comma
nd.
```

In [3]:
```
from download import download
url = "https://ascend-professional-construction-dataset.obs.cn-north-4.myhuaweicloud.com:443/ComputerVision/mobilenetV2-200_10
```

```
path = download(url, "./", kind="zip", replace=True)
# from download import download
```

```
Downloading data from https://ascend-professional-construction-dataset.obs.cn-north-4.myhuaweicloud.com:443/ComputerVision/mobi
lenetV2-200_1067.zip (25.5 MB)

file_sizes: 100%|████████████████████████| 26.7M/26.7M [00:00<00:00, 108MB/s]
Extracting zip file...
Successfully downloaded / unzipped to ./
```

In [ ]:
```
import zipfile
import os

# Define the paths to the zip files
zip_files = "mobilenetV2-200_1067.zip"

# Extract each zip file to the current directory
with zipfile.ZipFile(zip_file, 'r') as zip_ref:
    # Extract all the contents into the current directory
    zip_ref.extractall(os.getcwd())
    print(f"Extracted {zip_file} to {os.getcwd()}")
```

导入库

In [1]:
```
import math
import numpy as np
import os
import random

from matplotlib import pyplot as plt
from easydict import EasyDict
from PIL import Image
import numpy as np
import mindspore.nn as nn
from mindspore import ops as P
from mindspore.ops import add
from mindspore import Tensor
import mindspore.common.dtype as mstype
import mindspore.dataset as de
```

```
import mindspore.dataset.vision as C
import mindspore.dataset.transforms as C2
import mindspore as ms
from mindspore import set_context, nn, Tensor, load_checkpoint, save_checkpoint, export
from mindspore.train import Model
from mindspore.train import Callback, LossMonitor, ModelCheckpoint, CheckpointConfig

# os.environ[&#39;GLOG_v&#39;] = &#39;3&#39; # Log level includes 3(ERROR), 2(WARNING), 1(INFO), 0(DEBUG).
# set_context(mode=ms.GRAPH_MODE, device_target="Ascend", device_id=0) # 设置采用图模式执行，设备为Ascend#

# Log level includes 3(ERROR), 2(WARNING), 1(INFO), 0(DEBUG).
os.environ['GLOG_v'] = '3'  # Set logging level
set_context(mode=ms.GRAPH_MODE, device_target="Ascend", device_id=0)  # 设置采用图模式执行，设备为Ascend
```

In [2]:

```
# 垃圾分类数据集标签，以及用于标签映射的字典。
# 垃圾分类数据集标签，以汉字和标签映射的字典
garbage_classes = {
    '第一部分': ['NS', 'S'],
    '第二部分': ['Normal', 'Mild', 'Moderate', 'Severe']
}

class_cn = ['NS', 'S']
class_en = ['NS', 'S']
index_en = {'NS': 0, 'S': 1}

# 配置参数
config = EasyDict({
    "num_classes": 2,
    "image_height": 224,
    "image_width": 224,
    "data_split": (0.9, 0.1),
    "backbone_out_channels": 1280,
    "batch_size": 32,
    "eval_batch_size": 8,
    "epochs": 20,
    "lr_max": 0.0005,
    "momentum": 0.9,
    "weight_decay": 1e-4,
    "save_ckpt_epochs": 1,
    "save_ckpt_path": "./ckpt2", #./表示当前目录下
```

```
    # "dataset_path": "./data_en",
    "dataset_path": "./model_generator",  # Updated to new dataset path
    "class_index": index_en,
    "pretrained_ckpt": "./mobilenetV2-200_1067.ckpt"
})
```

数据集定义

In [3]:

```
def create_dataset(dataset_path, config, training=True, buffer_size=1000):
    """
    create a train or eval dataset

    Args:
        dataset_path (string): the path of dataset.
        config (struct): the config of train and eval in different platform.

    Returns:
        train_dataset, val_dataset
    """
    data_path = os.path.join(dataset_path, 'train' if training else 'test')
    ds = de.ImageFolderDataset(data_path, num_parallel_workers=4, class_indexing=config.class_index)
    resize_height = config.image_height
    resize_width = config.image_width

    normalize_op = C.Normalize(mean=[0.485*255, 0.456*255, 0.406*255], std=[0.229*255, 0.224*255, 0.225*255])
    change_swap_op = C.HWC2CHW()
    type_cast_op = C2.TypeCast(mstype.int32)

    if training:
        crop_decode_resize = C.RandomCropDecodeResize(resize_height, scale=(0.08, 1.0), ratio=(0.75, 1.333))
        horizontal_flip_op = C.RandomHorizontalFlip(prob=0.5)
        #color_adjust = C.RandomColorAdjust(brightness=0.4, contrast=0.4, saturation=0.4)

        #train_trans = [crop_decode_resize, horizontal_flip_op, color_adjust, normalize_op, change_swap_op]
        train_trans = [crop_decode_resize, horizontal_flip_op, normalize_op, change_swap_op]
        train_ds = ds.map(input_columns="image", operations=train_trans, num_parallel_workers=4)
        train_ds = train_ds.map(input_columns="label", operations=type_cast_op, num_parallel_workers=4)

        train_ds = train_ds.shuffle(buffer_size=buffer_size)
```

```
        ds = train_ds.batch(config.batch_size, drop_remainder=True)

    else:
        decode_op = C.Decode()
        resize_op = C.Resize((int(resize_width * 0.875), int(resize_height * 0.875)))
        center_crop = C.CenterCrop(resize_width)

        eval_trans = [decode_op, resize_op, center_crop, normalize_op, change_swap_op]
        eval_ds = ds.map(input_columns="image", operations=eval_trans, num_parallel_workers=4)
        eval_ds = eval_ds.map(input_columns="label", operations=type_cast_op, num_parallel_workers=4)
        ds = eval_ds.batch(config.eval_batch_size, drop_remainder=True)

    return ds
```

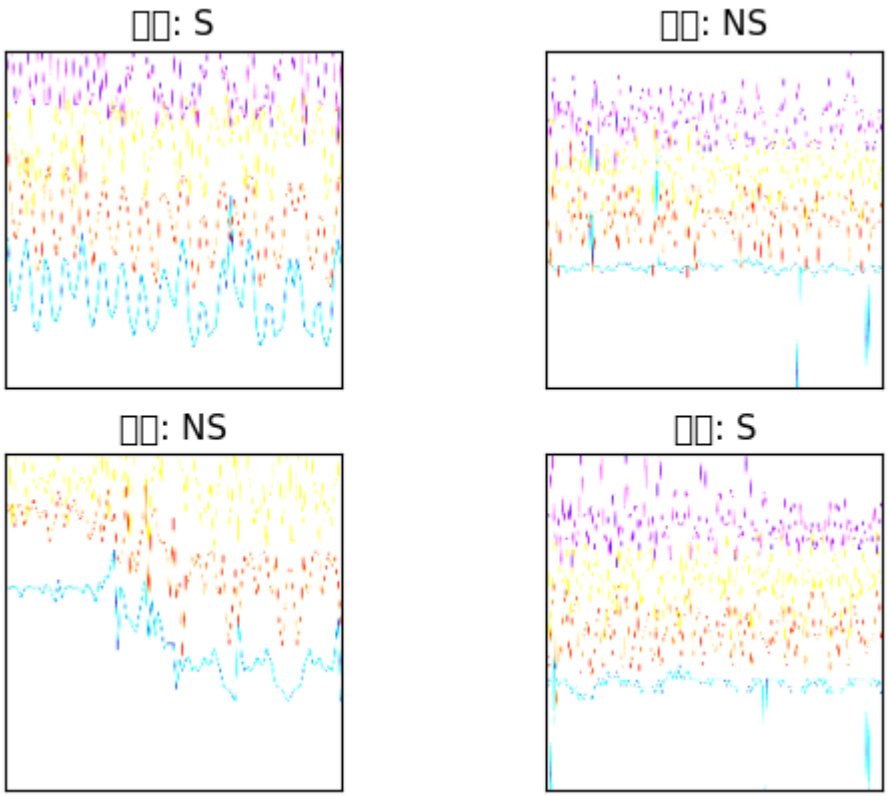
In [4]:

```
# 显示处理过的前5张图片
ds = create_dataset(dataset_path=config.dataset_path, config=config, training=True)
print(ds.get_dataset_size())
data = ds.create_dict_iterator(output_numpy=True)._get_next()
images = data['image']
labels = data['label']

for i in range(1, 5):
    plt.subplot(2, 2, i)
    plt.imshow(np.transpose(images[i], (1, 2, 0)))
    plt.title(f'标签: {class_en[labels[i]]}')
    plt.xticks([])
    plt.yticks([])

plt.show()
```

```
612
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



MobileNetV2模型搭建

In [4]:
```
import math
from mindspore import nn, Tensor, ops as P
import numpy as np

__all__ = ['MobileNetV2', 'MobileNetV2Backbone', 'MobileNetV2Head', 'mobilenet_v2']

def _make_divisible(v, divisor, min_value=None):
    if min_value is None:
        min_value = divisor
    new_v = max(min_value, int(v + divisor / 2) // divisor * divisor)
    if new_v < 0.9 * v:
        new_v += divisor
```

```
    return new_v

class GlobalAvgPooling(nn.Cell):
    def __init__(self):
        super(GlobalAvgPooling, self).__init__()

    def construct(self, x):
        x = P.ReduceMean()(x, (2, 3))
        return x

class ConvBNReLU(nn.Cell):
    def __init__(self, in_planes, out_planes, kernel_size=3, stride=1, groups=1):
        super(ConvBNReLU, self).__init__()
        padding = (kernel_size - 1) // 2
        in_channels = in_planes
        out_channels = out_planes
        if groups == 1:
            conv = nn.Conv2d(in_channels, out_channels, kernel_size, stride, pad_mode='pad', padding=padding)
        else:
            out_channels = in_planes
            conv = nn.Conv2d(in_channels, out_channels, kernel_size, stride, pad_mode='pad',
                             padding=padding, group=in_channels)

        layers = [conv, nn.BatchNorm2d(out_planes), nn.ReLU6()]
        self.features = nn.SequentialCell(layers)

    def construct(self, x):
        output = self.features(x)
        return output

class InvertedResidual(nn.Cell):
    def __init__(self, inp, oup, stride, expand_ratio):
        super(InvertedResidual, self).__init__()
        assert stride in [1, 2]

        hidden_dim = int(round(inp * expand_ratio))
        self.use_res_connect = stride == 1 and inp == oup

        layers = []
        if expand_ratio != 1:
            layers.append(ConvBNReLU(inp, hidden_dim, kernel_size=1))
```

```
        layers.extend([
            ConvBNReLU(hidden_dim, hidden_dim, stride=stride, groups=hidden_dim),
            nn.Conv2d(hidden_dim, oup, kernel_size=1, stride=1, has_bias=False),
            nn.BatchNorm2d(oup),
        ])
        self.conv = nn.SequentialCell(layers)
        self.cast = P.Cast()

    def construct(self, x):
        identity = x
        x = self.conv(x)
        if self.use_res_connect:
            return P.Add()(identity, x)
        return x

class MobileNetV2Backbone(nn.Cell):
    def __init__(self, width_mult=1., inverted_residual_setting=None, round_nearest=8,
                 input_channel=32, last_channel=1280):
        super(MobileNetV2Backbone, self).__init__()
        block = InvertedResidual
        self.cfgs = inverted_residual_setting or [
            [1, 16, 1, 1],
            [6, 24, 2, 2],
            [6, 32, 3, 2],
            [6, 64, 4, 2],
            [6, 96, 3, 1],
            [6, 160, 3, 2],
            [6, 320, 1, 1],
        ]

        input_channel = _make_divisible(input_channel * width_mult, round_nearest)
        self.out_channels = _make_divisible(last_channel * max(1.0, width_mult), round_nearest)
        features = [ConvBNReLU(3, input_channel, stride=2)]

        for t, c, n, s in self.cfgs:
            output_channel = _make_divisible(c * width_mult, round_nearest)
            for i in range(n):
                stride = s if i == 0 else 1
                features.append(block(input_channel, output_channel, stride, expand_ratio=t))
                input_channel = output_channel
        features.append(ConvBNReLU(input_channel, self.out_channels, kernel_size=1))
```

```
        self.features = nn.SequentialCell(features)
        self._initialize_weights()

    def construct(self, x):
        x = self.features(x)
        return x

    def _initialize_weights(self):
        self.init_parameters_data()
        for _, m in self.cells_and_names():
            if isinstance(m, nn.Conv2d):
                n = m.kernel_size[0] * m.kernel_size[1] * m.out_channels
                m.weight.set_data(Tensor(np.random.normal(0, np.sqrt(2. / n), m.weight.data.shape).astype("float32")))
                if m.bias is not None:
                    m.bias.set_data(Tensor(np.zeros(m.bias.data.shape, dtype="float32")))
            elif isinstance(m, nn.BatchNorm2d):
                m.gamma.set_data(Tensor(np.ones(m.gamma.data.shape, dtype="float32")))
                m.beta.set_data(Tensor(np.zeros(m.beta.data.shape, dtype="float32")))

    @property
    def get_features(self):
        return self.features

class MobileNetV2Head(nn.Cell):
    def __init__(self, input_channel=1280, num_classes=1000, has_dropout=False, activation="None"):
        super(MobileNetV2Head, self).__init__()
        head = ([GlobalAvgPooling(), nn.Dense(input_channel, num_classes, has_bias=True)] if not has_dropout else
                [GlobalAvgPooling(), nn.Dropout(0.2), nn.Dense(input_channel, num_classes, has_bias=True)])
        self.head = nn.SequentialCell(head)
        self.need_activation = True
        if activation == "Sigmoid":
            self.activation = nn.Sigmoid()
        elif activation == "Softmax":
            self.activation = nn.Softmax()
        else:
            self.need_activation = False
        self._initialize_weights()

    def construct(self, x):
        x = self.head(x)
        if self.need_activation:
```

```
            x = self.activation(x)
        return x

    def _initialize_weights(self):
        self.init_parameters_data()
        for _, m in self.cells_and_names():
            if isinstance(m, nn.Dense):
                m.weight.set_data(Tensor(np.random.normal(0, 0.01, m.weight.data.shape).astype("float32")))
                if m.bias is not None:
                    m.bias.set_data(Tensor(np.zeros(m.bias.data.shape, dtype="float32")))

    @property
    def get_head(self):
        return self.head

class MobileNetV2(nn.Cell):
    def __init__(self, num_classes=1000, width_mult=1., has_dropout=False, inverted_residual_setting=None,
                 round_nearest=8, input_channel=32, last_channel=1280):
        super(MobileNetV2, self).__init__()
        self.backbone = MobileNetV2Backbone(width_mult=width_mult, inverted_residual_setting=inverted_residual_setting,
                                            round_nearest=round_nearest, input_channel=input_channel,
                                            last_channel=last_channel).get_features
        self.head = MobileNetV2Head(input_channel=self.backbone.out_channels, num_classes=num_classes,
                                    has_dropout=has_dropout).get_head

    def construct(self, x):
        x = self.backbone(x)
        x = self.head(x)
        return x

class MobileNetV2Combine(nn.Cell):
    def __init__(self, backbone, head):
        super(MobileNetV2Combine, self).__init__(auto_prefix=False)
        self.backbone = backbone
        self.head = head

    def construct(self, x):
        x = self.backbone(x)
        x = self.head(x)
        return x
```

```
def mobilenet_v2(backbone, head):
    return MobileNetV2Combine(backbone, head)

def cosine_lr_schedule(total_steps, lr_init, lr_end, lr_max, warmup_steps):
    """
    Generate learning rate array with a cosine decay and linear warmup.

    Args:
       total_steps (int): Total training steps.
       lr_init (float): Initial learning rate.
       lr_end (float): Final learning rate.
       lr_max (float): Maximum learning rate.
       warmup_steps (int): Number of warmup steps.

    Returns:
       list: Learning rate array.
    """
    lr_init, lr_end, lr_max = float(lr_init), float(lr_end), float(lr_max)
    decay_steps = total_steps - warmup_steps
    lr_all_steps = []
    inc_per_step = (lr_max - lr_init) / warmup_steps if warmup_steps else 0
    for i in range(total_steps):
        if i < warmup_steps:
            lr = lr_init + inc_per_step * (i + 1)
        else:
            cosine_decay = 0.5 * (1 + math.cos(math.pi * (i - warmup_steps) / decay_steps))
            lr = (lr_max - lr_end) * cosine_decay + lr_end
        lr_all_steps.append(lr)

    return lr_all_steps
```

添加检查点Checkpoint

In [5]:

```
def switch_precision(net, data_type):
    if ms.get_context('device_target') == "Ascend":
        net.to_float(data_type)
        for _, cell in net.cells_and_names():
            if isinstance(cell, nn.Dense):
                cell.to_float(ms.float32)
```

模型训练与测试

In [7]:
```
from mindspore.amp import FixedLossScaleManager
from mindspore import save_checkpoint
import os
import mindspore as ms

# 设置绝对路径以避免路径解析问题
CKPT_PATH = os.path.abspath("./ckpt")
os.makedirs(CKPT_PATH, exist_ok=True)

LOSS_SCALE = 1024

# 加载数据集
train_dataset = create_dataset(dataset_path=config.dataset_path, config=config)
eval_dataset = create_dataset(dataset_path=config.dataset_path, config=config)
step_size = train_dataset.get_dataset_size()

# 设置模型
backbone = MobileNetV2Backbone()

# 冻结backbone的参数（如果不需要训练）
for param in backbone.get_parameters():
    param.requires_grad = False

# 从预训练模型中加载参数
load_checkpoint(config.pretrained_ckpt, backbone)

head = MobileNetV2Head(input_channel=backbone.out_channels, num_classes=config.num_classes)
network = mobilenet_v2(backbone, head)

# 定义损失函数、优化器和学习率调度
loss = nn.SoftmaxCrossEntropyWithLogits(sparse=True, reduction='mean')
loss_scale = FixedLossScaleManager(LOSS_SCALE, drop_overflow_update=False)
lrs = cosine_lr_schedule(config.epochs * step_size, lr_init=0.0, lr_end=1e-5, lr_max=config.lr_max, warmup_steps=5)
opt = nn.Momentum(network.trainable_params(), learning_rate=lrs, momentum=config.momentum, weight_decay=config.weight_decay, l

# 训练循环函数
```

```python
def train_loop(model, dataset, loss_fn, optimizer):
    def forward_fn(data, label):
        logits = model(data)
        loss = loss_fn(logits, label)
        return loss

    grad_fn = ms.value_and_grad(forward_fn, None, optimizer.parameters)

    def train_step(data, label):
        loss, grads = grad_fn(data, label)
        optimizer(grads)
        return loss

    size = dataset.get_dataset_size()
    model.set_train()
    for batch, (data, label) in enumerate(dataset.create_tuple_iterator()):
        loss = train_step(data, label)

        if batch % 10 == 0:
            loss_val, current = loss.asnumpy(), batch
            print(f"loss: {loss_val:>7f}  [{current:>3d}/{size:>3d}]")

# 测试循环函数
def test_loop(model, dataset, loss_fn):
    num_batches = dataset.get_dataset_size()
    model.set_train(False)
    total, test_loss, correct = 0, 0, 0
    for data, label in dataset.create_tuple_iterator():
        pred = model(data)
        total += data.shape[0]
        test_loss += loss_fn(pred, label).asnumpy()
        correct += (pred.argmax(axis=1) == label).asnumpy().sum()
    test_loss /= num_batches
    correct /= total
    print(f"Test: \n Accuracy: {(100 * correct):>0.1f}%, Avg loss: {test_loss:>8f} \n")

# 训练与评估
print("============== Starting Training ==============")
epochs = config.epochs
for t in range(epochs):
    print(f"Epoch {t + 1}\n-------------------------------")
```

```
    train_loop(network, train_dataset, loss, opt)
    save_checkpoint(network, os.path.join(CKPT_PATH, "save_mobilenetV2_model.ckpt"))
    test_loop(network, eval_dataset, loss)
print("Done!")
```

```
============== Starting Training ==============
Epoch 1
-------------------------------
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:30:12.977.236 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:30:12.977.343 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:30:12.977.384 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.708438  [  0/495]
loss: 0.712111  [ 10/495]
loss: 0.707566  [ 20/495]
loss: 0.689527  [ 30/495]
loss: 0.694866  [ 40/495]
loss: 0.710247  [ 50/495]
loss: 0.695616  [ 60/495]
loss: 0.705321  [ 70/495]
loss: 0.708085  [ 80/495]
loss: 0.693837  [ 90/495]
loss: 0.686858  [100/495]
loss: 0.669556  [110/495]
loss: 0.708219  [120/495]
loss: 0.688764  [130/495]
loss: 0.680666  [140/495]
loss: 0.696999  [150/495]
loss: 0.706277  [160/495]
loss: 0.656959  [170/495]
loss: 0.708687  [180/495]
loss: 0.687958  [190/495]
loss: 0.681713  [200/495]
loss: 0.701048  [210/495]
loss: 0.696189  [220/495]
loss: 0.674965  [230/495]
loss: 0.702232  [240/495]
loss: 0.710959  [250/495]
loss: 0.684295  [260/495]
loss: 0.680727  [270/495]
loss: 0.684963  [280/495]
loss: 0.729168  [290/495]
loss: 0.708849  [300/495]
loss: 0.670948  [310/495]
loss: 0.670604  [320/495]
loss: 0.702844  [330/495]
loss: 0.687347  [340/495]
loss: 0.718339  [350/495]
loss: 0.717262  [360/495]
loss: 0.692823  [370/495]
loss: 0.719563  [380/495]
loss: 0.705458  [390/495]
loss: 0.688337  [400/495]
```

```
loss: 0.732877  [410/495]
loss: 0.698940  [420/495]
loss: 0.722252  [430/495]
loss: 0.703091  [440/495]
loss: 0.692376  [450/495]
loss: 0.708687  [460/495]
loss: 0.700499  [470/495]
loss: 0.670104  [480/495]
loss: 0.675386  [490/495]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:32:11.167.660 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/302252503.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:32:11.167.733 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/302252503.py]
Test:
 Accuracy: 51.0%, Avg loss: 0.694756

Epoch 2
-------------------------------
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:33:13.291.473 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:33:13.291.549 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:33:13.291.632 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.705563  [  0/495]
loss: 0.704740  [ 10/495]
loss: 0.675834  [ 20/495]
loss: 0.689108  [ 30/495]
loss: 0.660324  [ 40/495]
loss: 0.691196  [ 50/495]
loss: 0.682279  [ 60/495]
loss: 0.694589  [ 70/495]
loss: 0.695640  [ 80/495]
loss: 0.687188  [ 90/495]
loss: 0.727910  [100/495]
loss: 0.668724  [110/495]
loss: 0.664111  [120/495]
loss: 0.705636  [130/495]
loss: 0.728979  [140/495]
loss: 0.671073  [150/495]
loss: 0.691831  [160/495]
loss: 0.689627  [170/495]
loss: 0.719485  [180/495]
loss: 0.714889  [190/495]
loss: 0.674531  [200/495]
loss: 0.705724  [210/495]
loss: 0.684852  [220/495]
loss: 0.704719  [230/495]
loss: 0.700953  [240/495]
loss: 0.684280  [250/495]
loss: 0.695738  [260/495]
loss: 0.681850  [270/495]
loss: 0.697805  [280/495]
loss: 0.679014  [290/495]
loss: 0.710482  [300/495]
loss: 0.704378  [310/495]
loss: 0.705209  [320/495]
loss: 0.702618  [330/495]
loss: 0.689934  [340/495]
loss: 0.678443  [350/495]
loss: 0.677363  [360/495]
loss: 0.675260  [370/495]
loss: 0.674855  [380/495]
loss: 0.685595  [390/495]
loss: 0.694823  [400/495]
```

```
loss: 0.686931  [410/495]
loss: 0.705822  [420/495]
loss: 0.689454  [430/495]
loss: 0.699612  [440/495]
loss: 0.686824  [450/495]
loss: 0.702218  [460/495]
loss: 0.687165  [470/495]
loss: 0.692366  [480/495]
loss: 0.692219  [490/495]
Test:
 Accuracy: 53.0%, Avg loss: 0.690684

Epoch 3
-------------------------------
```

```
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:34:57.087.605 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:34:57.087.682 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:34:57.087.787 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.680452  [  0/495]
loss: 0.680854  [ 10/495]
loss: 0.689828  [ 20/495]
loss: 0.724346  [ 30/495]
loss: 0.678035  [ 40/495]
loss: 0.668018  [ 50/495]
loss: 0.707147  [ 60/495]
loss: 0.664843  [ 70/495]
loss: 0.684689  [ 80/495]
loss: 0.674111  [ 90/495]
loss: 0.695017  [100/495]
loss: 0.715011  [110/495]
loss: 0.685498  [120/495]
loss: 0.696532  [130/495]
loss: 0.692658  [140/495]
loss: 0.692809  [150/495]
loss: 0.693304  [160/495]
loss: 0.696907  [170/495]
loss: 0.690144  [180/495]
loss: 0.684091  [190/495]
loss: 0.681318  [200/495]
loss: 0.729686  [210/495]
loss: 0.666857  [220/495]
loss: 0.681778  [230/495]
loss: 0.681765  [240/495]
loss: 0.657266  [250/495]
loss: 0.664264  [260/495]
loss: 0.678228  [270/495]
loss: 0.686544  [280/495]
loss: 0.663975  [290/495]
loss: 0.668080  [300/495]
loss: 0.662382  [310/495]
loss: 0.705142  [320/495]
loss: 0.652370  [330/495]
loss: 0.681370  [340/495]
loss: 0.686598  [350/495]
loss: 0.688156  [360/495]
loss: 0.707548  [370/495]
loss: 0.670923  [380/495]
loss: 0.692052  [390/495]
loss: 0.688887  [400/495]
```

```
loss: 0.670217  [410/495]
loss: 0.709376  [420/495]
loss: 0.708610  [430/495]
loss: 0.719994  [440/495]
loss: 0.660425  [450/495]
loss: 0.708782  [460/495]
loss: 0.708978  [470/495]
loss: 0.721061  [480/495]
loss: 0.722906  [490/495]
Test:
 Accuracy: 54.8%, Avg loss: 0.687002

Epoch 4
-------------------------------
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:36:41.867.879 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:36:41.867.947 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:36:41.868.066 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.688385  [  0/495]
loss: 0.679003  [ 10/495]
loss: 0.685462  [ 20/495]
loss: 0.703293  [ 30/495]
loss: 0.665142  [ 40/495]
loss: 0.664140  [ 50/495]
loss: 0.668516  [ 60/495]
loss: 0.696744  [ 70/495]
loss: 0.691300  [ 80/495]
loss: 0.679093  [ 90/495]
loss: 0.673931  [100/495]
loss: 0.686270  [110/495]
loss: 0.701359  [120/495]
loss: 0.695866  [130/495]
loss: 0.700507  [140/495]
loss: 0.685303  [150/495]
loss: 0.699739  [160/495]
loss: 0.665772  [170/495]
loss: 0.723025  [180/495]
loss: 0.732311  [190/495]
loss: 0.713856  [200/495]
loss: 0.686138  [210/495]
loss: 0.717526  [220/495]
loss: 0.673204  [230/495]
loss: 0.685328  [240/495]
loss: 0.698663  [250/495]
loss: 0.673475  [260/495]
loss: 0.693568  [270/495]
loss: 0.701914  [280/495]
loss: 0.679799  [290/495]
loss: 0.644043  [300/495]
loss: 0.665810  [310/495]
loss: 0.679134  [320/495]
loss: 0.708141  [330/495]
loss: 0.681037  [340/495]
loss: 0.656583  [350/495]
loss: 0.693775  [360/495]
loss: 0.664412  [370/495]
loss: 0.673209  [380/495]
loss: 0.655644  [390/495]
loss: 0.694372  [400/495]
```

```
loss: 0.695234  [410/495]
loss: 0.680019  [420/495]
loss: 0.718102  [430/495]
loss: 0.686874  [440/495]
loss: 0.687347  [450/495]
loss: 0.672181  [460/495]
loss: 0.688841  [470/495]
loss: 0.691237  [480/495]
loss: 0.658227  [490/495]
Test:
 Accuracy: 55.8%, Avg loss: 0.684726

Epoch 5
-------------------------------
```

```
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:38:25.780.308 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:38:25.780.392 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:38:25.780.524 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.688623  [  0/495]
loss: 0.687048  [ 10/495]
loss: 0.661670  [ 20/495]
loss: 0.677068  [ 30/495]
loss: 0.693388  [ 40/495]
loss: 0.690032  [ 50/495]
loss: 0.708001  [ 60/495]
loss: 0.667129  [ 70/495]
loss: 0.680315  [ 80/495]
loss: 0.687232  [ 90/495]
loss: 0.682410  [100/495]
loss: 0.667521  [110/495]
loss: 0.647496  [120/495]
loss: 0.683156  [130/495]
loss: 0.693579  [140/495]
loss: 0.694724  [150/495]
loss: 0.697037  [160/495]
loss: 0.669873  [170/495]
loss: 0.694216  [180/495]
loss: 0.676769  [190/495]
loss: 0.718207  [200/495]
loss: 0.698159  [210/495]
loss: 0.674195  [220/495]
loss: 0.674150  [230/495]
loss: 0.710601  [240/495]
loss: 0.706090  [250/495]
loss: 0.674888  [260/495]
loss: 0.672854  [270/495]
loss: 0.660534  [280/495]
loss: 0.669528  [290/495]
loss: 0.691075  [300/495]
loss: 0.671895  [310/495]
loss: 0.694464  [320/495]
loss: 0.694098  [330/495]
loss: 0.702529  [340/495]
loss: 0.701181  [350/495]
loss: 0.662881  [360/495]
loss: 0.664848  [370/495]
loss: 0.671366  [380/495]
loss: 0.690690  [390/495]
loss: 0.694995  [400/495]
```

```
loss: 0.689377  [410/495]
loss: 0.700985  [420/495]
loss: 0.712077  [430/495]
loss: 0.697753  [440/495]
loss: 0.680582  [450/495]
loss: 0.662814  [460/495]
loss: 0.697502  [470/495]
loss: 0.672252  [480/495]
loss: 0.669202  [490/495]
Test:
 Accuracy: 57.3%, Avg loss: 0.682395

Epoch 6
-------------------------------
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:40:10.945.782 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:40:10.945.862 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:40:10.945.908 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.667535  [  0/495]
loss: 0.689177  [ 10/495]
loss: 0.649587  [ 20/495]
loss: 0.695463  [ 30/495]
loss: 0.693370  [ 40/495]
loss: 0.698308  [ 50/495]
loss: 0.693743  [ 60/495]
loss: 0.701106  [ 70/495]
loss: 0.666986  [ 80/495]
loss: 0.703462  [ 90/495]
loss: 0.693117  [100/495]
loss: 0.674267  [110/495]
loss: 0.662406  [120/495]
loss: 0.700176  [130/495]
loss: 0.691225  [140/495]
loss: 0.644173  [150/495]
loss: 0.661340  [160/495]
loss: 0.677600  [170/495]
loss: 0.657831  [180/495]
loss: 0.700420  [190/495]
loss: 0.670274  [200/495]
loss: 0.676093  [210/495]
loss: 0.684055  [220/495]
loss: 0.668735  [230/495]
loss: 0.623678  [240/495]
loss: 0.686573  [250/495]
loss: 0.711051  [260/495]
loss: 0.706364  [270/495]
loss: 0.664111  [280/495]
loss: 0.712410  [290/495]
loss: 0.705943  [300/495]
loss: 0.671443  [310/495]
loss: 0.697774  [320/495]
loss: 0.728221  [330/495]
loss: 0.683130  [340/495]
loss: 0.679379  [350/495]
loss: 0.650525  [360/495]
loss: 0.667704  [370/495]
loss: 0.675837  [380/495]
loss: 0.681754  [390/495]
loss: 0.696423  [400/495]
```

```
loss: 0.723219  [410/495]
loss: 0.716996  [420/495]
loss: 0.696176  [430/495]
loss: 0.644541  [440/495]
loss: 0.686183  [450/495]
loss: 0.681344  [460/495]
loss: 0.687882  [470/495]
loss: 0.675057  [480/495]
loss: 0.703956  [490/495]
Test:
 Accuracy: 58.1%, Avg loss: 0.680977

Epoch 7
-------------------------------
```

```
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:41:56.707.881 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:41:56.707.952 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:41:56.708.085 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.682417  [  0/495]
loss: 0.693110  [ 10/495]
loss: 0.665414  [ 20/495]
loss: 0.707264  [ 30/495]
loss: 0.643610  [ 40/495]
loss: 0.683703  [ 50/495]
loss: 0.667978  [ 60/495]
loss: 0.683555  [ 70/495]
loss: 0.678403  [ 80/495]
loss: 0.665124  [ 90/495]
loss: 0.747333  [100/495]
loss: 0.664621  [110/495]
loss: 0.660675  [120/495]
loss: 0.694821  [130/495]
loss: 0.651308  [140/495]
loss: 0.713522  [150/495]
loss: 0.655356  [160/495]
loss: 0.676953  [170/495]
loss: 0.657784  [180/495]
loss: 0.690500  [190/495]
loss: 0.656986  [200/495]
loss: 0.700514  [210/495]
loss: 0.673057  [220/495]
loss: 0.650725  [230/495]
loss: 0.663508  [240/495]
loss: 0.679937  [250/495]
loss: 0.692644  [260/495]
loss: 0.716396  [270/495]
loss: 0.652104  [280/495]
loss: 0.696009  [290/495]
loss: 0.645524  [300/495]
loss: 0.665024  [310/495]
loss: 0.699644  [320/495]
loss: 0.714808  [330/495]
loss: 0.732764  [340/495]
loss: 0.694429  [350/495]
loss: 0.698200  [360/495]
loss: 0.647999  [370/495]
loss: 0.673915  [380/495]
loss: 0.710462  [390/495]
loss: 0.649759  [400/495]
```

```
loss: 0.694471  [410/495]
loss: 0.650931  [420/495]
loss: 0.681225  [430/495]
loss: 0.663454  [440/495]
loss: 0.668919  [450/495]
loss: 0.687042  [460/495]
loss: 0.641240  [470/495]
loss: 0.683084  [480/495]
loss: 0.664545  [490/495]
Test:
 Accuracy: 58.6%, Avg loss: 0.679063

Epoch 8
-------------------------------
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:43:39.509.445 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:43:39.509.539 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:43:39.509.806 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.697282  [  0/495]
loss: 0.675097  [ 10/495]
loss: 0.674818  [ 20/495]
loss: 0.667060  [ 30/495]
loss: 0.699057  [ 40/495]
loss: 0.648606  [ 50/495]
loss: 0.689548  [ 60/495]
loss: 0.667749  [ 70/495]
loss: 0.698532  [ 80/495]
loss: 0.639052  [ 90/495]
loss: 0.673864  [100/495]
loss: 0.693592  [110/495]
loss: 0.671800  [120/495]
loss: 0.689286  [130/495]
loss: 0.673849  [140/495]
loss: 0.698219  [150/495]
loss: 0.676354  [160/495]
loss: 0.653185  [170/495]
loss: 0.672869  [180/495]
loss: 0.708606  [190/495]
loss: 0.643522  [200/495]
loss: 0.652836  [210/495]
loss: 0.703245  [220/495]
loss: 0.717940  [230/495]
loss: 0.621427  [240/495]
loss: 0.675758  [250/495]
loss: 0.645306  [260/495]
loss: 0.730090  [270/495]
loss: 0.714624  [280/495]
loss: 0.646769  [290/495]
loss: 0.658203  [300/495]
loss: 0.673871  [310/495]
loss: 0.706457  [320/495]
loss: 0.675587  [330/495]
loss: 0.673996  [340/495]
loss: 0.701080  [350/495]
loss: 0.709313  [360/495]
loss: 0.638150  [370/495]
loss: 0.652686  [380/495]
loss: 0.672724  [390/495]
loss: 0.652931  [400/495]
```

```
loss: 0.688289  [410/495]
loss: 0.685152  [420/495]
loss: 0.688403  [430/495]
loss: 0.648880  [440/495]
loss: 0.679826  [450/495]
loss: 0.685273  [460/495]
loss: 0.689849  [470/495]
loss: 0.657736  [480/495]
loss: 0.670813  [490/495]
Test:
 Accuracy: 59.1%, Avg loss: 0.678099

Epoch 9
-------------------------------
```

```
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:45:21.835.643 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:45:21.835.720 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:45:21.835.835 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.686960  [  0/495]
loss: 0.657303  [ 10/495]
loss: 0.661675  [ 20/495]
loss: 0.701003  [ 30/495]
loss: 0.691226  [ 40/495]
loss: 0.720456  [ 50/495]
loss: 0.652308  [ 60/495]
loss: 0.631646  [ 70/495]
loss: 0.640417  [ 80/495]
loss: 0.669887  [ 90/495]
loss: 0.669153  [100/495]
loss: 0.653155  [110/495]
loss: 0.667522  [120/495]
loss: 0.674027  [130/495]
loss: 0.703844  [140/495]
loss: 0.703319  [150/495]
loss: 0.744115  [160/495]
loss: 0.694870  [170/495]
loss: 0.661938  [180/495]
loss: 0.655252  [190/495]
loss: 0.673201  [200/495]
loss: 0.684793  [210/495]
loss: 0.628158  [220/495]
loss: 0.716454  [230/495]
loss: 0.657150  [240/495]
loss: 0.632986  [250/495]
loss: 0.659744  [260/495]
loss: 0.684909  [270/495]
loss: 0.672735  [280/495]
loss: 0.716808  [290/495]
loss: 0.665454  [300/495]
loss: 0.700988  [310/495]
loss: 0.677596  [320/495]
loss: 0.672778  [330/495]
loss: 0.623502  [340/495]
loss: 0.730659  [350/495]
loss: 0.659904  [360/495]
loss: 0.662331  [370/495]
loss: 0.674646  [380/495]
loss: 0.646035  [390/495]
loss: 0.699163  [400/495]
```

```
loss: 0.657047  [410/495]
loss: 0.746921  [420/495]
loss: 0.702165  [430/495]
loss: 0.661595  [440/495]
loss: 0.694413  [450/495]
loss: 0.687162  [460/495]
loss: 0.700773  [470/495]
loss: 0.680239  [480/495]
loss: 0.659909  [490/495]
Test:
 Accuracy: 59.7%, Avg loss: 0.677361

Epoch 10
-------------------------------
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:47:09.421.850 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:47:09.421.926 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:47:09.422.006 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.686937  [  0/495]
loss: 0.694121  [ 10/495]
loss: 0.707875  [ 20/495]
loss: 0.675484  [ 30/495]
loss: 0.719265  [ 40/495]
loss: 0.693903  [ 50/495]
loss: 0.683529  [ 60/495]
loss: 0.723091  [ 70/495]
loss: 0.662026  [ 80/495]
loss: 0.668603  [ 90/495]
loss: 0.672087  [100/495]
loss: 0.693380  [110/495]
loss: 0.680898  [120/495]
loss: 0.641615  [130/495]
loss: 0.644090  [140/495]
loss: 0.693192  [150/495]
loss: 0.677089  [160/495]
loss: 0.710637  [170/495]
loss: 0.654869  [180/495]
loss: 0.692416  [190/495]
loss: 0.682411  [200/495]
loss: 0.687341  [210/495]
loss: 0.667452  [220/495]
loss: 0.666441  [230/495]
loss: 0.673383  [240/495]
loss: 0.675456  [250/495]
loss: 0.695585  [260/495]
loss: 0.694587  [270/495]
loss: 0.674368  [280/495]
loss: 0.641072  [290/495]
loss: 0.637264  [300/495]
loss: 0.700594  [310/495]
loss: 0.671600  [320/495]
loss: 0.669436  [330/495]
loss: 0.709558  [340/495]
loss: 0.658335  [350/495]
loss: 0.692960  [360/495]
loss: 0.660738  [370/495]
loss: 0.703070  [380/495]
loss: 0.624511  [390/495]
loss: 0.645180  [400/495]
```

```
loss: 0.682053  [410/495]
loss: 0.682081  [420/495]
loss: 0.672512  [430/495]
loss: 0.648227  [440/495]
loss: 0.649063  [450/495]
loss: 0.663310  [460/495]
loss: 0.671826  [470/495]
loss: 0.655811  [480/495]
loss: 0.643432  [490/495]
Test:
 Accuracy: 59.8%, Avg loss: 0.675966

Epoch 11
-------------------------------
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:48:53.415.225 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:48:53.415.320 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:48:53.415.361 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.699676  [  0/495]
loss: 0.703663  [ 10/495]
loss: 0.681936  [ 20/495]
loss: 0.642875  [ 30/495]
loss: 0.657298  [ 40/495]
loss: 0.653012  [ 50/495]
loss: 0.726263  [ 60/495]
loss: 0.699932  [ 70/495]
loss: 0.700809  [ 80/495]
loss: 0.680409  [ 90/495]
loss: 0.648270  [100/495]
loss: 0.660931  [110/495]
loss: 0.710545  [120/495]
loss: 0.696739  [130/495]
loss: 0.632524  [140/495]
loss: 0.695593  [150/495]
loss: 0.658619  [160/495]
loss: 0.654523  [170/495]
loss: 0.666098  [180/495]
loss: 0.666835  [190/495]
loss: 0.705666  [200/495]
loss: 0.687586  [210/495]
loss: 0.674695  [220/495]
loss: 0.719120  [230/495]
loss: 0.641737  [240/495]
loss: 0.707493  [250/495]
loss: 0.663688  [260/495]
loss: 0.706115  [270/495]
loss: 0.731240  [280/495]
loss: 0.638721  [290/495]
loss: 0.648331  [300/495]
loss: 0.675699  [310/495]
loss: 0.702934  [320/495]
loss: 0.665835  [330/495]
loss: 0.679848  [340/495]
loss: 0.652872  [350/495]
loss: 0.690994  [360/495]
loss: 0.681753  [370/495]
loss: 0.661887  [380/495]
loss: 0.719156  [390/495]
loss: 0.660808  [400/495]
```

```
loss: 0.691879  [410/495]
loss: 0.687525  [420/495]
loss: 0.648511  [430/495]
loss: 0.621726  [440/495]
loss: 0.665913  [450/495]
loss: 0.674207  [460/495]
loss: 0.667921  [470/495]
loss: 0.696494  [480/495]
loss: 0.675142  [490/495]
Test:
 Accuracy: 59.9%, Avg loss: 0.675130

Epoch 12
-------------------------------
```

```
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:50:39.589.870 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:50:39.589.944 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:50:39.590.004 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.713649  [  0/495]
loss: 0.698666  [ 10/495]
loss: 0.622515  [ 20/495]
loss: 0.659979  [ 30/495]
loss: 0.683871  [ 40/495]
loss: 0.667016  [ 50/495]
loss: 0.674441  [ 60/495]
loss: 0.637845  [ 70/495]
loss: 0.669959  [ 80/495]
loss: 0.686625  [ 90/495]
loss: 0.662342  [100/495]
loss: 0.704227  [110/495]
loss: 0.690382  [120/495]
loss: 0.666780  [130/495]
loss: 0.630953  [140/495]
loss: 0.680582  [150/495]
loss: 0.616917  [160/495]
loss: 0.703317  [170/495]
loss: 0.665074  [180/495]
loss: 0.633476  [190/495]
loss: 0.705048  [200/495]
loss: 0.627096  [210/495]
loss: 0.703748  [220/495]
loss: 0.643042  [230/495]
loss: 0.672828  [240/495]
loss: 0.658019  [250/495]
loss: 0.687536  [260/495]
loss: 0.648899  [270/495]
loss: 0.674002  [280/495]
loss: 0.686301  [290/495]
loss: 0.617749  [300/495]
loss: 0.689693  [310/495]
loss: 0.680017  [320/495]
loss: 0.706132  [330/495]
loss: 0.691778  [340/495]
loss: 0.664366  [350/495]
loss: 0.647969  [360/495]
loss: 0.619394  [370/495]
loss: 0.695875  [380/495]
loss: 0.723654  [390/495]
loss: 0.687858  [400/495]
```

```
loss: 0.655135  [410/495]
loss: 0.679142  [420/495]
loss: 0.648359  [430/495]
loss: 0.674179  [440/495]
loss: 0.673752  [450/495]
loss: 0.673078  [460/495]
loss: 0.695894  [470/495]
loss: 0.664042  [480/495]
loss: 0.624295  [490/495]
Test:
 Accuracy: 59.6%, Avg loss: 0.674242

Epoch 13
-------------------------------
```

```
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:52:23.671.352 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:52:23.671.428 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:52:23.671.539 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.668477  [  0/495]
loss: 0.643898  [ 10/495]
loss: 0.662685  [ 20/495]
loss: 0.689101  [ 30/495]
loss: 0.658688  [ 40/495]
loss: 0.694770  [ 50/495]
loss: 0.663791  [ 60/495]
loss: 0.684760  [ 70/495]
loss: 0.680727  [ 80/495]
loss: 0.671094  [ 90/495]
loss: 0.634360  [100/495]
loss: 0.675732  [110/495]
loss: 0.680843  [120/495]
loss: 0.703233  [130/495]
loss: 0.709203  [140/495]
loss: 0.625741  [150/495]
loss: 0.708254  [160/495]
loss: 0.647787  [170/495]
loss: 0.719607  [180/495]
loss: 0.689178  [190/495]
loss: 0.664021  [200/495]
loss: 0.636087  [210/495]
loss: 0.689339  [220/495]
loss: 0.686469  [230/495]
loss: 0.611519  [240/495]
loss: 0.700570  [250/495]
loss: 0.676587  [260/495]
loss: 0.697418  [270/495]
loss: 0.683072  [280/495]
loss: 0.689726  [290/495]
loss: 0.672312  [300/495]
loss: 0.647084  [310/495]
loss: 0.664178  [320/495]
loss: 0.692339  [330/495]
loss: 0.650285  [340/495]
loss: 0.708591  [350/495]
loss: 0.687693  [360/495]
loss: 0.703632  [370/495]
loss: 0.691052  [380/495]
loss: 0.697424  [390/495]
loss: 0.646282  [400/495]
```

```
loss: 0.718168  [410/495]
loss: 0.660069  [420/495]
loss: 0.648595  [430/495]
loss: 0.643197  [440/495]
loss: 0.697821  [450/495]
loss: 0.671517  [460/495]
loss: 0.718383  [470/495]
loss: 0.748984  [480/495]
loss: 0.682253  [490/495]
Test:
 Accuracy: 59.4%, Avg loss: 0.676174

Epoch 14
-------------------------------
```

```
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:54:08.525.651 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:54:08.525.718 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:54:08.525.797 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.642056  [  0/495]
loss: 0.692277  [ 10/495]
loss: 0.669877  [ 20/495]
loss: 0.697376  [ 30/495]
loss: 0.641600  [ 40/495]
loss: 0.689329  [ 50/495]
loss: 0.683192  [ 60/495]
loss: 0.682822  [ 70/495]
loss: 0.677405  [ 80/495]
loss: 0.658143  [ 90/495]
loss: 0.706174  [100/495]
loss: 0.676680  [110/495]
loss: 0.640454  [120/495]
loss: 0.686792  [130/495]
loss: 0.694433  [140/495]
loss: 0.662413  [150/495]
loss: 0.681742  [160/495]
loss: 0.657704  [170/495]
loss: 0.694603  [180/495]
loss: 0.766552  [190/495]
loss: 0.707543  [200/495]
loss: 0.667700  [210/495]
loss: 0.678157  [220/495]
loss: 0.703809  [230/495]
loss: 0.700111  [240/495]
loss: 0.657082  [250/495]
loss: 0.664611  [260/495]
loss: 0.668370  [270/495]
loss: 0.678301  [280/495]
loss: 0.680694  [290/495]
loss: 0.688896  [300/495]
loss: 0.699138  [310/495]
loss: 0.654747  [320/495]
loss: 0.624307  [330/495]
loss: 0.674233  [340/495]
loss: 0.644704  [350/495]
loss: 0.695565  [360/495]
loss: 0.681630  [370/495]
loss: 0.650114  [380/495]
loss: 0.674159  [390/495]
loss: 0.634178  [400/495]
```

```
loss: 0.649409  [410/495]
loss: 0.641939  [420/495]
loss: 0.687863  [430/495]
loss: 0.657458  [440/495]
loss: 0.658842  [450/495]
loss: 0.669055  [460/495]
loss: 0.655078  [470/495]
loss: 0.687294  [480/495]
loss: 0.639737  [490/495]
Test:
 Accuracy: 59.9%, Avg loss: 0.675860

Epoch 15
-------------------------------
```

```
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:55:52.401.028 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:55:52.401.101 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:55:52.401.195 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.648179  [  0/495]
loss: 0.661022  [ 10/495]
loss: 0.659933  [ 20/495]
loss: 0.642191  [ 30/495]
loss: 0.661867  [ 40/495]
loss: 0.626731  [ 50/495]
loss: 0.624409  [ 60/495]
loss: 0.704832  [ 70/495]
loss: 0.665281  [ 80/495]
loss: 0.639081  [ 90/495]
loss: 0.671420  [100/495]
loss: 0.649703  [110/495]
loss: 0.654428  [120/495]
loss: 0.653876  [130/495]
loss: 0.666523  [140/495]
loss: 0.698708  [150/495]
loss: 0.703012  [160/495]
loss: 0.668180  [170/495]
loss: 0.662343  [180/495]
loss: 0.682789  [190/495]
loss: 0.671787  [200/495]
loss: 0.684753  [210/495]
loss: 0.650161  [220/495]
loss: 0.673264  [230/495]
loss: 0.647228  [240/495]
loss: 0.670792  [250/495]
loss: 0.648842  [260/495]
loss: 0.679928  [270/495]
loss: 0.710290  [280/495]
loss: 0.654888  [290/495]
loss: 0.656178  [300/495]
loss: 0.691057  [310/495]
loss: 0.675444  [320/495]
loss: 0.673584  [330/495]
loss: 0.632668  [340/495]
loss: 0.630785  [350/495]
loss: 0.673049  [360/495]
loss: 0.642970  [370/495]
loss: 0.700496  [380/495]
loss: 0.651049  [390/495]
loss: 0.707506  [400/495]
```

```
loss: 0.690132  [410/495]
loss: 0.653523  [420/495]
loss: 0.661964  [430/495]
loss: 0.654356  [440/495]
loss: 0.645735  [450/495]
loss: 0.690190  [460/495]
loss: 0.678048  [470/495]
loss: 0.715882  [480/495]
loss: 0.674296  [490/495]
Test:
 Accuracy: 60.4%, Avg loss: 0.674502

Epoch 16
-------------------------------
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:57:41.143.207 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:57:41.143.281 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:57:41.143.319 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.613005  [  0/495]
loss: 0.702214  [ 10/495]
loss: 0.694013  [ 20/495]
loss: 0.650505  [ 30/495]
loss: 0.717314  [ 40/495]
loss: 0.622305  [ 50/495]
loss: 0.709715  [ 60/495]
loss: 0.714006  [ 70/495]
loss: 0.710696  [ 80/495]
loss: 0.651167  [ 90/495]
loss: 0.645249  [100/495]
loss: 0.712925  [110/495]
loss: 0.694602  [120/495]
loss: 0.649873  [130/495]
loss: 0.681969  [140/495]
loss: 0.642990  [150/495]
loss: 0.636840  [160/495]
loss: 0.663405  [170/495]
loss: 0.695074  [180/495]
loss: 0.690066  [190/495]
loss: 0.647917  [200/495]
loss: 0.665279  [210/495]
loss: 0.655318  [220/495]
loss: 0.687812  [230/495]
loss: 0.653362  [240/495]
loss: 0.679188  [250/495]
loss: 0.689177  [260/495]
loss: 0.662277  [270/495]
loss: 0.635366  [280/495]
loss: 0.645763  [290/495]
loss: 0.663760  [300/495]
loss: 0.668766  [310/495]
loss: 0.699764  [320/495]
loss: 0.647191  [330/495]
loss: 0.728582  [340/495]
loss: 0.656270  [350/495]
loss: 0.712765  [360/495]
loss: 0.614631  [370/495]
loss: 0.660345  [380/495]
loss: 0.703687  [390/495]
loss: 0.721857  [400/495]
```

```
loss: 0.619397  [410/495]
loss: 0.663801  [420/495]
loss: 0.688458  [430/495]
loss: 0.607334  [440/495]
loss: 0.633237  [450/495]
loss: 0.710772  [460/495]
loss: 0.666202  [470/495]
loss: 0.635552  [480/495]
loss: 0.691998  [490/495]
Test:
 Accuracy: 60.2%, Avg loss: 0.674355

Epoch 17
-------------------------------
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:59:26.519.061 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:59:26.519.132 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-14:59:26.519.169 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.671068  [  0/495]
loss: 0.662775  [ 10/495]
loss: 0.665251  [ 20/495]
loss: 0.702628  [ 30/495]
loss: 0.705152  [ 40/495]
loss: 0.681432  [ 50/495]
loss: 0.659729  [ 60/495]
loss: 0.634799  [ 70/495]
loss: 0.725746  [ 80/495]
loss: 0.636869  [ 90/495]
loss: 0.653738  [100/495]
loss: 0.645076  [110/495]
loss: 0.648463  [120/495]
loss: 0.652043  [130/495]
loss: 0.650284  [140/495]
loss: 0.670410  [150/495]
loss: 0.659281  [160/495]
loss: 0.697459  [170/495]
loss: 0.671641  [180/495]
loss: 0.675265  [190/495]
loss: 0.647406  [200/495]
loss: 0.642791  [210/495]
loss: 0.635116  [220/495]
loss: 0.704724  [230/495]
loss: 0.645437  [240/495]
loss: 0.620732  [250/495]
loss: 0.656222  [260/495]
loss: 0.636708  [270/495]
loss: 0.686038  [280/495]
loss: 0.641271  [290/495]
loss: 0.667795  [300/495]
loss: 0.701276  [310/495]
loss: 0.603800  [320/495]
loss: 0.693950  [330/495]
loss: 0.642683  [340/495]
loss: 0.709779  [350/495]
loss: 0.652673  [360/495]
loss: 0.641148  [370/495]
loss: 0.701555  [380/495]
loss: 0.660629  [390/495]
loss: 0.721309  [400/495]
```

```
loss: 0.634826  [410/495]
loss: 0.662603  [420/495]
loss: 0.642023  [430/495]
loss: 0.658222  [440/495]
loss: 0.680208  [450/495]
loss: 0.629760  [460/495]
loss: 0.711736  [470/495]
loss: 0.653848  [480/495]
loss: 0.703633  [490/495]
Test:
 Accuracy: 60.1%, Avg loss: 0.674627

Epoch 18
-------------------------------
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-15:01:10.943.395 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-15:01:10.943.486 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-15:01:10.943.524 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.678383  [  0/495]
loss: 0.655536  [ 10/495]
loss: 0.676603  [ 20/495]
loss: 0.666071  [ 30/495]
loss: 0.658737  [ 40/495]
loss: 0.659571  [ 50/495]
loss: 0.704526  [ 60/495]
loss: 0.632266  [ 70/495]
loss: 0.636142  [ 80/495]
loss: 0.690914  [ 90/495]
loss: 0.664995  [100/495]
loss: 0.681604  [110/495]
loss: 0.642340  [120/495]
loss: 0.691696  [130/495]
loss: 0.648372  [140/495]
loss: 0.650117  [150/495]
loss: 0.596206  [160/495]
loss: 0.692997  [170/495]
loss: 0.697134  [180/495]
loss: 0.715608  [190/495]
loss: 0.634346  [200/495]
loss: 0.727418  [210/495]
loss: 0.709479  [220/495]
loss: 0.650124  [230/495]
loss: 0.669561  [240/495]
loss: 0.665203  [250/495]
loss: 0.627023  [260/495]
loss: 0.675313  [270/495]
loss: 0.667809  [280/495]
loss: 0.611382  [290/495]
loss: 0.673644  [300/495]
loss: 0.686907  [310/495]
loss: 0.689375  [320/495]
loss: 0.666225  [330/495]
loss: 0.679596  [340/495]
loss: 0.702459  [350/495]
loss: 0.700853  [360/495]
loss: 0.648450  [370/495]
loss: 0.724869  [380/495]
loss: 0.696375  [390/495]
loss: 0.668852  [400/495]
```

```
loss: 0.712734  [410/495]
loss: 0.677177  [420/495]
loss: 0.680369  [430/495]
loss: 0.711844  [440/495]
loss: 0.694734  [450/495]
loss: 0.709023  [460/495]
loss: 0.725131  [470/495]
loss: 0.683235  [480/495]
loss: 0.695780  [490/495]
Test:
 Accuracy: 60.3%, Avg loss: 0.674880

Epoch 19
-------------------------------
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-15:02:56.182.726 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-15:02:56.182.809 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-15:02:56.182.848 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.672690  [  0/495]
loss: 0.666390  [ 10/495]
loss: 0.725459  [ 20/495]
loss: 0.653435  [ 30/495]
loss: 0.680044  [ 40/495]
loss: 0.721230  [ 50/495]
loss: 0.687053  [ 60/495]
loss: 0.630240  [ 70/495]
loss: 0.662340  [ 80/495]
loss: 0.660859  [ 90/495]
loss: 0.670233  [100/495]
loss: 0.690092  [110/495]
loss: 0.644161  [120/495]
loss: 0.632652  [130/495]
loss: 0.622733  [140/495]
loss: 0.655521  [150/495]
loss: 0.671452  [160/495]
loss: 0.666414  [170/495]
loss: 0.692645  [180/495]
loss: 0.677100  [190/495]
loss: 0.653624  [200/495]
loss: 0.672473  [210/495]
loss: 0.653340  [220/495]
loss: 0.658364  [230/495]
loss: 0.669613  [240/495]
loss: 0.708762  [250/495]
loss: 0.658293  [260/495]
loss: 0.678534  [270/495]
loss: 0.663960  [280/495]
loss: 0.648322  [290/495]
loss: 0.687938  [300/495]
loss: 0.649023  [310/495]
loss: 0.712743  [320/495]
loss: 0.683605  [330/495]
loss: 0.659957  [340/495]
loss: 0.738909  [350/495]
loss: 0.639957  [360/495]
loss: 0.684686  [370/495]
loss: 0.706819  [380/495]
loss: 0.672066  [390/495]
loss: 0.666173  [400/495]
```

```
loss: 0.667474  [410/495]
loss: 0.664981  [420/495]
loss: 0.623092  [430/495]
loss: 0.636469  [440/495]
loss: 0.667737  [450/495]
loss: 0.666966  [460/495]
loss: 0.646230  [470/495]
loss: 0.680831  [480/495]
loss: 0.692888  [490/495]
Test:
 Accuracy: 60.3%, Avg loss: 0.673614

Epoch 20
-------------------------------
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-15:04:48.375.221 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-15:04:48.375.311 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
[ERROR] CORE(2105,ffff86ddc0b0,python):2024-12-03-15:04:48.375.443 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get re
alpath failed, path[/tmp/ipykernel_2105/2466609572.py]
```

```
loss: 0.628379  [  0/495]
loss: 0.704851  [ 10/495]
loss: 0.659700  [ 20/495]
loss: 0.672256  [ 30/495]
loss: 0.638861  [ 40/495]
loss: 0.662239  [ 50/495]
loss: 0.656230  [ 60/495]
loss: 0.653852  [ 70/495]
loss: 0.688318  [ 80/495]
loss: 0.673717  [ 90/495]
loss: 0.639234  [100/495]
loss: 0.673172  [110/495]
loss: 0.691210  [120/495]
loss: 0.653283  [130/495]
loss: 0.674223  [140/495]
loss: 0.680240  [150/495]
loss: 0.698765  [160/495]
loss: 0.677761  [170/495]
loss: 0.705806  [180/495]
loss: 0.687323  [190/495]
loss: 0.709016  [200/495]
loss: 0.677341  [210/495]
loss: 0.644262  [220/495]
loss: 0.667501  [230/495]
loss: 0.694551  [240/495]
loss: 0.648156  [250/495]
loss: 0.674360  [260/495]
loss: 0.680038  [270/495]
loss: 0.686017  [280/495]
loss: 0.712901  [290/495]
loss: 0.681569  [300/495]
loss: 0.711304  [310/495]
loss: 0.655036  [320/495]
loss: 0.699419  [330/495]
loss: 0.644613  [340/495]
loss: 0.636392  [350/495]
loss: 0.628719  [360/495]
loss: 0.673630  [370/495]
loss: 0.662899  [380/495]
loss: 0.697854  [390/495]
loss: 0.719078  [400/495]
```

```
loss: 0.694758  [410/495]
loss: 0.653718  [420/495]
loss: 0.644590  [430/495]
loss: 0.673009  [440/495]
loss: 0.646291  [450/495]
loss: 0.688894  [460/495]
loss: 0.646847  [470/495]
loss: 0.725700  [480/495]
loss: 0.651198  [490/495]
Test:
 Accuracy: 60.4%, Avg loss: 0.674094

Done!
```

模型预测

In [ ]:

```python
import os
import random

def mk_file(file_path: str):
    if os.path.exists(file_path):
        # 如果文件夹存在，则先删除原文件夹在重新创建
        rmtree(file_path)
    os.makedirs(file_path)

file_name = 'x16'
folder_path = './For_test/'+file_name  # 文件夹的路径
files = os.listdir(folder_path)  # 获取文件夹中的文件列表


total_files = len(files)  # 文件总数
split_ratio = 0.5  # 拆分比例，这里假设拆分为两个文件夹，比例为0.5
split_count = int(total_files * split_ratio)  # 计算拆分的数量
random_split_count = random.randint(0, split_count)  # 生成随机拆分的数量

random_files = random.sample(files, random_split_count)  # 从文件列表中随机选择指定数量的文件

import shutil
```

```
folder1_path = './For_test2/' +file_name + '/A'  # 第一个文件夹的路径
folder2_path = './For_test2/' +file_name + '/N'  # 第二个文件夹的路径
mk_file(folder1_path)
mk_file(folder2_path)

for file in files:
    file_path = os.path.join(folder_path, file)  # 文件的完整路径
    if file in random_files:  # 如果文件在拆分的文件列表中
        shutil.move(file_path, folder1_path)  # 将文件移动到第一个文件夹
    else:
        shutil.move(file_path, folder2_path)  # 将文件移动到第二个文件夹

import numpy as np
import os
from PIL import Image
from mindspore import Tensor, load_checkpoint
from sklearn.metrics import confusion_matrix, roc_curve, auc
import matplotlib.pyplot as plt

CKPT = "save_mobilenetV2_model.ckpt"

def image_process(image):
    """Process one image at a time.

    Args:
        image: shape (H, W, C)
    """
    mean = [0.485 * 255, 0.456 * 255, 0.406 * 255]
    std = [0.229 * 255, 0.224 * 255, 0.225 * 255]
    image = (np.array(image) - mean) / std
    image = image.transpose((2, 0, 1))  # Change to (C, H, W) format
    img_tensor = Tensor(np.array([image], np.float32))  # Add batch dimension
    return img_tensor

def infer_one(network, image_path):
    """Infer a single image and return predicted label."""
    image = Image.open(image_path).resize((config.image_height, config.image_width))
    logits = network(image_process(image))
    pred = np.argmax(logits.asnumpy(), axis=1)[0]
    return pred
```

```python
def infer_folder(network, folder_path, label_map):
    """Infer all images in a folder."""
    true_labels = []
    pred_labels = []

    for class_name, label in label_map.items():
        class_folder = os.path.join(folder_path, class_name)
        if not os.path.exists(class_folder):
            continue

        for image_name in os.listdir(class_folder):
            image_path = os.path.join(class_folder, image_name)
            true_labels.append(label)
            pred_labels.append(infer_one(network, image_path))

    return true_labels, pred_labels

def plot_metrics(true_labels, pred_labels, label_map):
    """Plot confusion matrix and ROC curve."""
    # Confusion Matrix
    cm = confusion_matrix(true_labels, pred_labels)
    print("Confusion Matrix:\n", cm)

    plt.figure(figsize=(10, 5))
    plt.subplot(1, 2, 1)
    plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
    plt.title("Confusion Matrix")
    plt.colorbar()
    plt.xticks(ticks=np.arange(len(label_map)), labels=label_map.keys(), rotation=45)
    plt.yticks(ticks=np.arange(len(label_map)), labels=label_map.keys())
    plt.ylabel("True Label")
    plt.xlabel("Predicted Label")

    # ROC Curve
    n_classes = len(label_map)
    y_true = np.eye(n_classes)[true_labels]  # One-hot encode true labels
    y_pred_proba = np.eye(n_classes)[pred_labels]  # Simulated probabilities for simplicity

    plt.subplot(1, 2, 2)
    for i, label in enumerate(label_map.keys()):
        fpr, tpr, _ = roc_curve(y_true[:, i], y_pred_proba[:, i])
```

```
        roc_auc = auc(fpr, tpr)
        plt.plot(fpr, tpr, label=f"Class {label} (AUC = {roc_auc:.2f})")

    plt.plot([0, 1], [0, 1], "k--")
    plt.title("ROC Curve")
    plt.xlabel("False Positive Rate")
    plt.ylabel("True Positive Rate")
    plt.legend(loc="lower right")
    plt.tight_layout()
    plt.show()

def infer():
    # Load network
    backbone = MobileNetV2Backbone(last_channel=config.backbone_out_channels)
    head = MobileNetV2Head(input_channel=backbone.out_channels, num_classes=config.num_classes)
    network = mobilenet_v2(backbone, head)
    load_checkpoint(os.path.join(config.save_ckpt_path, CKPT), network)

    # Define label mapping (adjust according to your dataset)
    label_map = {"A": 0, "N": 1}  # Example class mapping

    # Folder containing subfolders for each class
    folder_path = "./For_test2/" +file_name

    # Perform inference and compute metrics
    true_labels, pred_labels = infer_folder(network, folder_path, label_map)
    A_num = len(pred_labels)-sum(pred_labels)
    AHI = A_num*60/sum(pred_labels)
    print("AHI: %.2f" %(AHI))
    plot_metrics(true_labels, pred_labels, label_map)

# Run inference
infer()
```

In [6]:

```
import numpy as np
import os
from PIL import Image
from mindspore import Tensor, load_checkpoint
from sklearn.metrics import confusion_matrix, roc_curve, auc
import matplotlib.pyplot as plt
```

```python
def image_process(image):
    """Process one image at a time.

    Args:
        image: shape (H, W, C)
    """
    mean = [0.485 * 255, 0.456 * 255, 0.406 * 255]
    std = [0.229 * 255, 0.224 * 255, 0.225 * 255]
    image = (np.array(image) - mean) / std
    image = image.transpose((2, 0, 1))  # Change to (C, H, W) format
    img_tensor = Tensor(np.array([image], np.float32))  # Add batch dimension
    return img_tensor

def infer_one(network, image_path):
    """Infer a single image and return predicted label."""
    image = Image.open(image_path).resize((config.image_height, config.image_width))
    logits = network(image_process(image))
    pred = np.argmax(logits.asnumpy(), axis=1)[0]
    return pred

def infer_folder(network, folder_path, label_map):
    """Infer all images in a folder."""
    true_labels = []
    pred_labels = []

    for class_name, label in label_map.items():
        class_folder = os.path.join(folder_path, class_name)
        if not os.path.exists(class_folder):
            continue

        for image_name in os.listdir(class_folder):
            image_path = os.path.join(class_folder, image_name)
            true_labels.append(label)
            pred_labels.append(infer_one(network, image_path))

    return true_labels, pred_labels

def plot_metrics(true_labels, pred_labels, label_map):
    """Plot confusion matrix and ROC curve."""
```

```
    # Confusion Matrix
    cm = confusion_matrix(true_labels, pred_labels)
    print("Confusion Matrix:\n", cm)

    plt.figure(figsize=(10, 5))
    plt.subplot(1, 2, 1)
    plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
    plt.title("Confusion Matrix")
    plt.colorbar()
    plt.xticks(ticks=np.arange(len(label_map)), labels=label_map.keys(), rotation=45)
    plt.yticks(ticks=np.arange(len(label_map)), labels=label_map.keys())
    plt.ylabel("True Label")
    plt.xlabel("Predicted Label")

    # ROC Curve
    n_classes = len(label_map)
    y_true = np.eye(n_classes)[true_labels]  # One-hot encode true labels
    y_pred_proba = np.eye(n_classes)[pred_labels]  # Simulated probabilities for simplicity

    plt.subplot(1, 2, 2)
    for i, label in enumerate(label_map.keys()):
        fpr, tpr, _ = roc_curve(y_true[:, i], y_pred_proba[:, i])
        roc_auc = auc(fpr, tpr)
        plt.plot(fpr, tpr, label=f"Class {label} (AUC = {roc_auc:.2f})")

    plt.plot([0, 1], [0, 1], "k--")
    plt.title("ROC Curve")
    plt.xlabel("False Positive Rate")
    plt.ylabel("True Positive Rate")
    plt.legend(loc="lower right")
    plt.tight_layout()
    plt.show()

def infer_sleep(folder_path):
    # Load network
    backbone = MobileNetV2Backbone(last_channel=config.backbone_out_channels)
    head = MobileNetV2Head(input_channel=backbone.out_channels, num_classes=config.num_classes)
    network = mobilenet_v2(backbone, head)
    load_checkpoint(os.path.join(config.save_ckpt_path, CKPT), network)

    # Define label mapping (adjust according to your dataset)
```

```
    label_map = {"NS": 0, "S": 1}  # Example class mapping

    # Folder containing subfolders for each class
    #folder_path = "./For_test2/" +file_name

    # Perform inference and compute metrics
    true_labels, pred_labels = infer_folder(network, folder_path, label_map)
    S_num = sum(pred_labels)
    print("Sleep_time: %.2f min" %(S_num/2))
    return S_num/2
    #plot_metrics(true_labels, pred_labels, label_map)

def infer_sas(folder_path):
    # Load network
    backbone = MobileNetV2Backbone(last_channel=config.backbone_out_channels)
    head = MobileNetV2Head(input_channel=backbone.out_channels, num_classes=config.num_classes)
    network = mobilenet_v2(backbone, head)
    load_checkpoint(os.path.join(config.save_ckpt_path, CKPT), network)

    # Define label mapping (adjust according to your dataset)
    label_map = {"N": 0, "A": 1}  # Example class mapping

    # Perform inference and compute metrics
    true_labels, pred_labels = infer_folder(network, folder_path, label_map)
    A_num = sum(pred_labels)
    print("A_num: %d" %(A_num))
    return A_num

# Run inference
mother_name = './0-AP'
folder_path = mother_name + "/For_test_s"
config.save_ckpt_path = "./ckpt2"
CKPT = "save_mobilenetV2_SLEEP_model.ckpt"

# 获取当前路径下的文件名，返回List
fileNames = os.listdir(folder_path)
for file in fileNames:
    # 将文件命加入到当前文件路径后面
    newDir = os.path.join(folder_path,file)
    time_60s = infer_sleep(newDir)
    print("%s 睡眠时长= %.2f min" %(file, time_60s))
```

```
# folder_path = mother_name + "./For_test/" +file_name
# config.save_ckpt_path = "./ckpt"
# CKPT = "save_mobilenetV2_model.ckpt"
# sas_num = infer_sas(folder_path)

# print("AHI = %.2f" %(sas_num*120/time_30s))
```

```
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:35:11.681.061 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:35:11.681.180 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 433.00 min
c09 睡眠时长= 433.00 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:35:58.283.057 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:35:58.283.145 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 448.00 min
c04 睡眠时长= 448.00 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:36:47.793.442 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:36:47.793.521 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 362.50 min
b04 睡眠时长= 362.50 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:37:33.058.461 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:37:33.058.558 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 441.00 min
a18 睡眠时长= 441.00 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:38:25.046.846 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:38:25.046.933 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
```

```
Sleep_time: 401.50 min
c07 睡眠时长= 401.50 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:39:15.291.932 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:39:15.292.007 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 457.50 min
b01 睡眠时长= 457.50 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:40:07.124.613 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:40:07.124.694 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 483.00 min
a13 睡眠时长= 483.00 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:40:58.785.803 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:40:58.785.881 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 513.50 min
b02 睡眠时长= 513.50 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:41:54.127.799 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:41:54.127.917 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 438.00 min
a17 睡眠时长= 438.00 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:42:42.393.432 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:42:42.393.514 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 499.50 min
a02 睡眠时长= 499.50 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:43:38.567.067 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:43:38.567.152 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 484.50 min
a01 睡眠时长= 484.50 min
```

```
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:44:28.912.286 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:44:28.912.377 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 506.00 min
a07 睡眠时长= 506.00 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:45:22.384.640 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:45:22.384.736 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 465.00 min
c01 睡眠时长= 465.00 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:46:15.558.005 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:46:15.558.100 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 485.00 min
a19 睡眠时长= 485.00 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:47:05.163.054 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:47:05.163.157 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 516.00 min
a14 睡眠时长= 516.00 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:47:56.439.987 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:47:56.440.084 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 441.50 min
c08 睡眠时长= 441.50 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:48:42.761.254 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:48:42.761.335 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 392.00 min
a04 睡眠时长= 392.00 min
```

```
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:49:32.619.144 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:49:32.619.230 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 492.50 min
a20 睡眠时长= 492.50 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:50:22.712.531 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:50:22.712.622 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 470.50 min
c02 睡眠时长= 470.50 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:51:12.826.156 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:51:12.826.243 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 431.50 min
c06 睡眠时长= 431.50 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:51:59.524.631 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:51:59.524.718 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 485.50 min
a15 睡眠时长= 485.50 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:52:56.761.474 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:52:56.761.562 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 408.00 min
c10 睡眠时长= 408.00 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:53:42.752.503 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:53:42.752.594 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 512.50 min
a03 睡眠时长= 512.50 min
```

```
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:54:34.675.341 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:54:34.675.423 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 522.00 min
a12 睡眠时长= 522.00 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:55:27.377.001 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:55:27.377.087 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 393.50 min
b03 睡眠时长= 393.50 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:56:12.364.502 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:56:12.364.588 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 460.00 min
a09 睡眠时长= 460.00 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:57:08.738.761 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:57:08.738.871 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 431.50 min
c03 睡眠时长= 431.50 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:57:56.000.708 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:57:56.000.792 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 431.00 min
c05 睡眠时长= 431.00 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:58:42.577.812 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:58:42.577.904 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 456.50 min
a11 睡眠时长= 456.50 min
```

```
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:59:30.014.346 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-21:59:30.014.439 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 505.50 min
a10 睡眠时长= 505.50 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-22:00:22.984.696 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-22:00:22.984.787 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 447.00 min
a05 睡眠时长= 447.00 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-22:01:12.664.118 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-22:01:12.664.228 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 268.50 min
b05 睡眠时长= 268.50 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-22:01:51.461.491 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-22:01:51.461.569 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 452.50 min
a08 睡眠时长= 452.50 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-22:02:53.307.107 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-22:02:53.307.184 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 416.00 min
a16 睡眠时长= 416.00 min
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-22:03:41.592.270 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
[ERROR] CORE(23748,ffff9038e0b0,python):2024-12-03-22:03:41.592.363 [mindspore/core/utils/file_utils.cc:253] GetRealPath] Get r
ealpath failed, path[/tmp/ipykernel_23748/302252503.py]
Sleep_time: 501.00 min
a06 睡眠时长= 501.00 min
```

In [ ]:

```python
import numpy as np
import os
```

```python
from PIL import Image
from mindspore import Tensor, load_checkpoint
from sklearn.metrics import confusion_matrix, roc_curve, auc
import matplotlib.pyplot as plt


def image_process(image):
    """Process one image at a time.

    Args:
        image: shape (H, W, C)
    """
    mean = [0.485 * 255, 0.456 * 255, 0.406 * 255]
    std = [0.229 * 255, 0.224 * 255, 0.225 * 255]
    image = (np.array(image) - mean) / std
    image = image.transpose((2, 0, 1))  # Change to (C, H, W) format
    img_tensor = Tensor(np.array([image], np.float32))  # Add batch dimension
    return img_tensor

def infer_one(network, image_path):
    """Infer a single image and return predicted label."""
    image = Image.open(image_path).resize((config.image_height, config.image_width))
    logits = network(image_process(image))
    pred = np.argmax(logits.asnumpy(), axis=1)[0]
    return pred

def infer_folder(network, folder_path, label_map):
    """Infer all images in a folder."""
    true_labels = []
    pred_labels = []

    for class_name, label in label_map.items():
        class_folder = os.path.join(folder_path, class_name)
        if not os.path.exists(class_folder):
            continue

        for image_name in os.listdir(class_folder):
            image_path = os.path.join(class_folder, image_name)
            true_labels.append(label)
            pred_labels.append(infer_one(network, image_path))
```

```
    return true_labels, pred_labels

def plot_metrics(true_labels, pred_labels, label_map):
    """Plot confusion matrix and ROC curve."""
    # Confusion Matrix
    cm = confusion_matrix(true_labels, pred_labels)
    print("Confusion Matrix:\n", cm)

    plt.figure(figsize=(10, 5))
    plt.subplot(1, 2, 1)
    plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
    plt.title("Confusion Matrix")
    plt.colorbar()
    plt.xticks(ticks=np.arange(len(label_map)), labels=label_map.keys(), rotation=45)
    plt.yticks(ticks=np.arange(len(label_map)), labels=label_map.keys())
    plt.ylabel("True Label")
    plt.xlabel("Predicted Label")

    # ROC Curve
    n_classes = len(label_map)
    y_true = np.eye(n_classes)[true_labels]  # One-hot encode true labels
    y_pred_proba = np.eye(n_classes)[pred_labels]  # Simulated probabilities for simplicity

    plt.subplot(1, 2, 2)
    for i, label in enumerate(label_map.keys()):
        fpr, tpr, _ = roc_curve(y_true[:, i], y_pred_proba[:, i])
        roc_auc = auc(fpr, tpr)
        plt.plot(fpr, tpr, label=f"Class {label} (AUC = {roc_auc:.2f})")

    plt.plot([0, 1], [0, 1], "k--")
    plt.title("ROC Curve")
    plt.xlabel("False Positive Rate")
    plt.ylabel("True Positive Rate")
    plt.legend(loc="lower right")
    plt.tight_layout()
    plt.show()

def infer_sleep(folder_path):
    # Load network
    backbone = MobileNetV2Backbone(last_channel=config.backbone_out_channels)
    head = MobileNetV2Head(input_channel=backbone.out_channels, num_classes=config.num_classes)
```

```
    network = mobilenet_v2(backbone, head)
    load_checkpoint(os.path.join(config.save_ckpt_path, CKPT), network)

    # Define label mapping (adjust according to your dataset)
    label_map = {"NS": 0, "S": 1}  # Example class mapping

    # Folder containing subfolders for each class
    #folder_path = "./For_test2/" +file_name

    # Perform inference and compute metrics
    true_labels, pred_labels = infer_folder(network, folder_path, label_map)
    S_num = sum(pred_labels)
    print("Sleep_time: %.2f min" %(S_num/2))
    return S_num/2
    #plot_metrics(true_labels, pred_labels, label_map)

def infer_sas(folder_path):
    # Load network
    backbone = MobileNetV2Backbone(last_channel=config.backbone_out_channels)
    head = MobileNetV2Head(input_channel=backbone.out_channels, num_classes=config.num_classes)
    network = mobilenet_v2(backbone, head)
    load_checkpoint(os.path.join(config.save_ckpt_path, CKPT), network)

    # Define label mapping (adjust according to your dataset)
    label_map = {"N": 0, "A": 1}  # Example class mapping

    # Perform inference and compute metrics
    true_labels, pred_labels = infer_folder(network, folder_path, label_map)
    A_num = sum(pred_labels)
    print("A_num: %d" %(A_num))
    return A_num

# Run inference
mother_name = './0-AP'
folder_path = mother_name + "/For_test1"
config.save_ckpt_path = "./ckpt"
CKPT = "save_mobilenetV2_model.ckpt"

# 获取当前路径下的文件名，返回List
fileNames = os.listdir(folder_path)
for file in fileNames:
```

```
    # 将文件命加入到当前文件路径后面
    newDir = os.path.join(folder_path,file)
    time_60s = infer_sas(newDir)
    print("%s 呼吸障碍次数= %d min" %(file, time_60s))




# folder_path = mother_name + "./For_test/" +file_name
# config.save_ckpt_path = "./ckpt"
# CKPT = "save_mobilenetV2_model.ckpt"
# sas_num = infer_sas(folder_path)

# print("AHI = %.2f" %(sas_num*120/time_30s))
```