

```
%pip install torch torchvision --upgrade
import torch
import torch.nn as nn
import torch.optim as optim
from torch.optim.lr_scheduler import OneCycleLR
from torchvision import models, transforms, datasets
from torch.utils.data import DataLoader
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.utils.class_weight import compute_class_weight
import numpy as np

Requirement already satisfied: torch in c:\users\ekaansh\appdata\local\programs\python\python311\lib\site-packages (2.5.1+cu121)
Collecting torch
  Downloading torch-2.9.0-cp311-cp311-win_amd64.whl.metadata (30 kB)
Requirement already satisfied: torchvision in c:\users\ekaansh\appdata\local\programs\python\python311\lib\site-packages (0.20.1+cu121)
Collecting torchvision
  Downloading torchvision-0.24.0-cp311-cp311-win_amd64.whl.metadata (5.9 kB)
Requirement already satisfied: filelock in c:\users\ekaansh\appdata\local\programs\python\python311\lib\site-packages (from torch) (3.13.1)
Requirement already satisfied: typing-extensions>=4.10.0 in c:\users\ekaansh\appdata\local\programs\python\python311\lib\site-packages (from torch) (4.15.0)
Collecting sympy>=1.13.3 (from torch)
  Using cached sympy-1.14.0-py3-none-any.whl.metadata (12 kB)
Requirement already satisfied: networkx>=2.5.1 in c:\users\ekaansh\appdata\local\programs\python\python311\lib\site-packages (from torch) (3.3)
Requirement already satisfied: jinja2 in c:\users\ekaansh\appdata\local\programs\python\python311\lib\site-packages (from torch) (3.1.4)
Requirement already satisfied: fsspec>=0.8.5 in c:\users\ekaansh\appdata\local\programs\python\python311\lib\site-packages (from torch) (2024.6.1)
Requirement already satisfied: numpy in c:\users\ekaansh\appdata\local\programs\python\python311\lib\site-packages (from torchvision) (1.26.4)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in c:\users\ekaansh\appdata\local\programs\python\python311\lib\site-packages (from torchvision) (11.0.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in c:\users\ekaansh\appdata\local\programs\python\python311\lib\site-packages (from sympy>=1.13.3->torch) (1.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\ekaansh\appdata\local\programs\python\python311\lib\site-packages (from jinja2->torch) (2.1.5)
Downloading torch-2.9.0-cp311-cp311-win_amd64.whl (109.3 MB)
```

```
----- 0.0/109.3 MB ? eta -----  
----- 1.0/109.3 MB 7.1 MB/s eta  
0:00:16 ----- 4.7/109.3 MB 13.6 MB/s eta  
0:00:08 ----- 6.6/109.3 MB 11.8 MB/s eta  
0:00:09 ----- 7.3/109.3 MB 11.3 MB/s eta  
0:00:10 ----- 11.5/109.3 MB 11.8 MB/s  
eta 0:00:09 ----- 15.7/109.3 MB 12.1 MB/s  
eta 0:00:08 ----- 18.6/109.3 MB 12.2 MB/s  
eta 0:00:08 ----- 20.7/109.3 MB 12.0 MB/s  
eta 0:00:08 ----- 23.3/109.3 MB 12.1 MB/s  
eta 0:00:08 ----- 24.4/109.3 MB 11.7 MB/s  
eta 0:00:08 ----- 24.4/109.3 MB 11.7 MB/s  
eta 0:00:08 ----- 24.4/109.3 MB 11.7 MB/s  
eta 0:00:08 ----- 25.7/109.3 MB 9.2 MB/s eta  
0:00:10 ----- 27.0/109.3 MB 9.1 MB/s eta  
0:00:10 ----- 27.5/109.3 MB 8.6 MB/s eta  
0:00:10 ----- 29.9/109.3 MB 7.4 MB/s eta  
0:00:11 ----- 30.7/109.3 MB 7.2 MB/s eta  
0:00:11 ----- 32.2/109.3 MB 7.2 MB/s eta  
0:00:11 ----- 34.1/109.3 MB 7.3 MB/s eta  
0:00:11 ----- 35.9/109.3 MB 7.3 MB/s eta  
0:00:11 ----- 36.7/109.3 MB 7.2 MB/s eta  
0:00:11
```

```
----- 37.5/109.3 MB 7.0 MB/s eta  
0:00:11 ----- 37.7/109.3 MB 7.0 MB/s eta  
0:00:11 ----- 38.3/109.3 MB 6.6 MB/s eta  
0:00:11 ----- 38.5/109.3 MB 6.5 MB/s eta  
0:00:11 ----- 39.1/109.3 MB 6.3 MB/s eta  
0:00:12 ----- 40.4/109.3 MB 6.3 MB/s eta  
0:00:11 ----- 41.9/109.3 MB 6.4 MB/s eta  
0:00:11 ----- 44.8/109.3 MB 6.6 MB/s eta  
0:00:10 ----- 46.1/109.3 MB 6.6 MB/s eta  
0:00:10 ----- 47.4/109.3 MB 6.6 MB/s eta  
0:00:10 ----- 48.5/109.3 MB 6.5 MB/s eta  
0:00:10 ----- 50.6/109.3 MB 6.6 MB/s eta  
0:00:09 ----- 52.4/109.3 MB 6.7 MB/s eta  
0:00:09 ----- 54.3/109.3 MB 6.7 MB/s eta  
0:00:09 ----- 56.1/109.3 MB 6.8 MB/s eta  
0:00:08 ----- 58.2/109.3 MB 6.8 MB/s eta  
0:00:08 ----- 60.0/109.3 MB 6.9 MB/s eta  
0:00:08 ----- 61.9/109.3 MB 6.9 MB/s eta  
0:00:07 ----- 64.0/109.3 MB 7.0 MB/s eta  
0:00:07 ----- 65.5/109.3 MB 7.0 MB/s eta  
0:00:07 ----- 66.6/109.3 MB 7.0 MB/s eta  
0:00:07 ----- 69.2/109.3 MB 7.1 MB/s eta  
0:00:06 ----- 71.8/109.3 MB 7.2 MB/s eta  
0:00:06 ----- 73.4/109.3 MB 7.2 MB/s eta  
0:00:05 ----- 75.5/109.3 MB 7.3 MB/s eta
```

```
0:00:05                               -.- 78.1/109.3 MB 7.4 MB/s eta
0:00:05                               -.- 78.4/109.3 MB 7.4 MB/s eta
0:00:05                               -.- 78.4/109.3 MB 7.4 MB/s eta
0:00:05                               -.- 78.4/109.3 MB 7.4 MB/s eta
0:00:05                               -.- 80.7/109.3 MB 7.1 MB/s eta
0:00:05                               -.- 81.8/109.3 MB 7.0 MB/s eta
0:00:04                               -.- 83.4/109.3 MB 7.1 MB/s eta
0:00:04                               -.- 84.7/109.3 MB 7.0 MB/s eta
0:00:04                               -.- 87.0/109.3 MB 7.1 MB/s eta
0:00:04                               -.- 89.7/109.3 MB 7.2 MB/s eta
0:00:03                               -.- 91.2/109.3 MB 7.2 MB/s eta
0:00:03                               -.- 91.8/109.3 MB 7.1 MB/s eta
0:00:03                               -.- 93.1/109.3 MB 7.1 MB/s eta
0:00:03                               -.- 94.6/109.3 MB 7.1 MB/s eta
0:00:03                               -.- 96.2/109.3 MB 7.1 MB/s eta
0:00:02                               -.- 98.0/109.3 MB 7.2 MB/s eta
0:00:02                               -.- 100.1/109.3 MB 7.2 MB/s
eta 0:00:02                           -.- 102.2/109.3 MB 7.2 MB/s
eta 0:00:01                           -.- 103.8/109.3 MB 7.2 MB/s
eta 0:00:01                           -.- 104.6/109.3 MB 7.2 MB/s
eta 0:00:01                           -.- 107.5/109.3 MB 7.3 MB/s
eta 0:00:01                           -.- 109.1/109.3 MB 7.3 MB/s
eta 0:00:01                           -.- 109.1/109.3 MB 7.3 MB/s
eta 0:00:01                           -.- 109.1/109.3 MB 7.3 MB/s
```


Successfully uninstalled sympy-1.13.1


```
----- 2/3 [torchvision]
```

Successfully installed sympy-1.14.0 torch-2.9.0 torchvision-0.24.0
Note: you may need to restart the kernel to use updated packages.

```
WARNING: Ignoring invalid distribution ~ensorflow (c:\Users\Ekaansh\AppData\Local\Programs\Python\Python311\Lib\site-packages)
WARNING: Ignoring invalid distribution ~hap (c:\Users\Ekaansh\AppData\Local\Programs\Python\Python311\Lib\site-packages)
WARNING: Ignoring invalid distribution ~ensorflow (c:\Users\Ekaansh\AppData\Local\Programs\Python\Python311\Lib\site-packages)
WARNING: Ignoring invalid distribution ~ensorflow (c:\Users\Ekaansh\AppData\Local\Programs\Python\Python311\Lib\site-packages)
WARNING: Ignoring invalid distribution ~hap (c:\Users\Ekaansh\AppData\Local\Programs\Python\Python311\Lib\site-packages)
WARNING: Failed to remove contents in a temporary directory 'C:\Users\Ekaansh\AppData\Local\Programs\Python\Python311\Lib\site-packages\~orch'.
You can safely remove it manually.
WARNING: Failed to remove contents in a temporary directory 'C:\Users\Ekaansh\AppData\Local\Programs\Python\Python311\Lib\site-packages\~orchvision'.
You can safely remove it manually.
WARNING: Ignoring invalid distribution ~ensorflow (c:\Users\Ekaansh\AppData\Local\Programs\Python\Python311\Lib\site-packages)
WARNING: Ignoring invalid distribution ~hap (c:\Users\Ekaansh\AppData\Local\Programs\Python\Python311\Lib\site-packages)
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
torchaudio 2.5.1+cu121 requires torch==2.5.1+cu121, but you have torch 2.9.0 which is incompatible.

[notice] A new release of pip is available: 25.2 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
-----  
-----
```

```
ImportError                                     Traceback (most recent call
last)
Cell In[4], line 2
      1 get_ipython().run_line_magic('pip', 'install torch torchvision
--upgrade')
----> 2 import torch
      3 import torch.nn as nn
      4 import torch.optim as optim

File c:\Users\Ekaansh\AppData\Local\Programs\Python\Python311\Lib\
site-packages\torch\__init__.py:50
    42     return False
    45 from torch._utils import (
    46     _functionalize_sync as _sync,
    47     _import_dotted_name,
    48     classproperty,
    49 )
---> 50 from torch._utils_internal import (
    51     get_file_path,
    52     prepare_multiprocessing_environment,
    53     profiler_allow_cudagraph_cupti_lazy_reinit_cuda12,
    54     USE_GLOBAL_DEPS,
    55     USE_RTLD_GLOBAL_WITH_LIBTORCH,
    56 )
    57 from torch.torch_version import __version__ as __version__
    60 if TYPE_CHECKING:

ImportError: cannot import name
'profiler_allow_cudagraph_cupti_lazy_reinit_cuda12' from
'torch._utils_internal' (c:\Users\Ekaansh\AppData\Local\Programs\
Python\Python311\Lib\site-packages\torch\_utils_internal.py)

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print("Device:", device)

Device: cuda

train_transform = transforms.Compose([
    transforms.RandomResizedCrop(224, scale=(0.8, 1.0)),
    transforms.RandomHorizontalFlip(),
    transforms.RandomRotation(15),
    transforms.ColorJitter(brightness=0.2, contrast=0.2),
    transforms.ToTensor(),
    transforms.Normalize([0.485], [0.229])
])

val_test_transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
```

```

        transforms.Normalize([0.485], [0.229])
    ])

train_data = datasets.ImageFolder(r"D:\datasets\chest_xray\chest_xray\train",
                                 transform=train_transform)
val_data   = datasets.ImageFolder(r"D:\datasets\chest_xray\chest_xray\val",
                                 transform=val_test_transform)
test_data  = datasets.ImageFolder(r"D:\datasets\chest_xray\chest_xray\test",
                                 transform=val_test_transform)

train_loader = DataLoader(train_data, batch_size=32, shuffle=True,
                           num_workers=2)
val_loader   = DataLoader(val_data, batch_size=32, shuffle=False,
                           num_workers=2)
test_loader  = DataLoader(test_data, batch_size=32, shuffle=False,
                           num_workers=2)

y_train = [label for _, label in train_data.samples]
class_weights = compute_class_weight(class_weight='balanced',
                                      classes=np.unique(y_train),
                                      y=y_train)
class_weights = torch.tensor(class_weights,
                             dtype=torch.float).to(device)
print("Class Weights:", class_weights)

Class Weights: tensor([1.9448, 0.6730], device='cuda:0')

model = models.densenet121(weights=models.DenseNet121_Weights.DEFAULT)

num_ftrs = model.classifier.in_features
model.classifier = nn.Sequential(
    nn.Linear(num_ftrs, 256),
    nn.ReLU(),
    nn.Dropout(0.4),
    nn.Linear(256, 2)
)

for name, param in model.named_parameters():
    param.requires_grad = False
    if "denseblock3" in name or "denseblock4" in name or "classifier" in name:
        param.requires_grad = True

model = model.to(device)

criterion = nn.CrossEntropyLoss(weight=class_weights)

optimizer = optim.AdamW(model.parameters(), lr=1e-4, weight_decay=1e-4)

```

```

scheduler = OneCycleLR(optimizer, max_lr=1e-3,
steps_per_epoch=len(train_loader), epochs=10)

def train_model(model, train_loader, val_loader, epochs=10):
    best_acc = 0
    for epoch in range(epochs):
        model.train()
        running_loss, correct, total = 0.0, 0, 0

        for inputs, labels in train_loader:
            inputs, labels = inputs.to(device), labels.to(device)

            optimizer.zero_grad()
            outputs = model(inputs)
            loss = criterion(outputs, labels)
            loss.backward()
            optimizer.step()
            scheduler.step()

            running_loss += loss.item()
            _, preds = torch.max(outputs, 1)
            correct += (preds == labels).sum().item()
            total += labels.size(0)

        train_acc = correct / total * 100
        val_acc = evaluate_model(model, val_loader)

        print(f"Epoch {epoch+1}/{epochs}, Loss:
{running_loss/len(train_loader):.4f}, "
              f"Train Acc: {train_acc:.2f}%, Val Acc: {val_acc:.2f}%")

        if val_acc > best_acc:
            best_acc = val_acc
            torch.save(model.state_dict(),
"best_pneumonia_densenet121.pt")

def evaluate_model(model, loader):
    model.eval()
    correct, total = 0, 0
    with torch.no_grad():
        for inputs, labels in loader:
            inputs, labels = inputs.to(device), labels.to(device)
            outputs = model(inputs)
            _, preds = torch.max(outputs, 1)
            correct += (preds == labels).sum().item()
            total += labels.size(0)
    return correct / total * 100

train_model(model, train_loader, val_loader, epochs=10)

```

```
Epoch 1/10, Loss: 0.2653, Train Acc: 91.55%, Val Acc: 87.50%
Epoch 2/10, Loss: 0.1324, Train Acc: 95.28%, Val Acc: 93.75%
Epoch 3/10, Loss: 0.1325, Train Acc: 94.94%, Val Acc: 100.00%
Epoch 4/10, Loss: 0.1026, Train Acc: 96.20%, Val Acc: 93.75%
Epoch 5/10, Loss: 0.0858, Train Acc: 97.24%, Val Acc: 100.00%
Epoch 6/10, Loss: 0.0721, Train Acc: 97.47%, Val Acc: 100.00%
Epoch 7/10, Loss: 0.0663, Train Acc: 97.60%, Val Acc: 81.25%
Epoch 8/10, Loss: 0.0496, Train Acc: 97.85%, Val Acc: 100.00%
Epoch 9/10, Loss: 0.0447, Train Acc: 98.43%, Val Acc: 93.75%
Epoch 10/10, Loss: 0.0399, Train Acc: 98.60%, Val Acc: 87.50%

model.load_state_dict(torch.load("best_pneumonia_densenet121.pt"))
model.eval()

C:\Users\Ekaansh\AppData\Local\Temp\ipykernel_18684\1610110734.py:1:
FutureWarning: You are using `torch.load` with `weights_only=False`  

(the current default value), which uses the default pickle module  

implicitly. It is possible to construct malicious pickle data which  

will execute arbitrary code during unpickling (See  

https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for more details). In a future release, the default value for  

`weights_only` will be flipped to `True`. This limits the functions  

that could be executed during unpickling. Arbitrary objects will no  

longer be allowed to be loaded via this mode unless they are  

explicitly allowlisted by the user via  

`torch.serialization.add_safe_globals`. We recommend you start setting  

`weights_only=True` for any use case where you don't have full control  

of the loaded file. Please open an issue on GitHub for any issues  

related to this experimental feature.

model.load_state_dict(torch.load("best_pneumonia_densenet121.pt"))

DenseNet(
    features): Sequential(
        (conv0): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2),
padding=(3, 3), bias=False)
        (norm0): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu0): ReLU(inplace=True)
        (pool0): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1,
ceil_mode=False)
        (denseblock1): _DenseBlock(
            (denselayer1): _DenseLayer(
                (norm1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
                (relu1): ReLU(inplace=True)
                (conv1): Conv2d(64, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
                (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
                (relu2): ReLU(inplace=True)
```

```
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer2): _DenseLayer(
        (norm1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(96, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer3): _DenseLayer(
        (norm1): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer4): _DenseLayer(
        (norm1): BatchNorm2d(160, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(160, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer5): _DenseLayer(
        (norm1): BatchNorm2d(192, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(192, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
```

```
padding=(1, 1), bias=False)
    )
    (denselayer6): _DenseLayer(
        (norm1): BatchNorm2d(224, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(224, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
)
(transition1): _Transition(
    (norm): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (pool): AvgPool2d(kernel_size=2, stride=2, padding=0)
)
(denseblock2): _DenseBlock(
    (denselayer1): _DenseLayer(
        (norm1): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer2): _DenseLayer(
        (norm1): BatchNorm2d(160, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(160, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer3): _DenseLayer(
```

```
        (norm1): BatchNorm2d(192, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(192, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer4): _DenseLayer(
        (norm1): BatchNorm2d(224, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(224, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer5): _DenseLayer(
        (norm1): BatchNorm2d(256, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer6): _DenseLayer(
        (norm1): BatchNorm2d(288, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(288, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer7): _DenseLayer(
        (norm1): BatchNorm2d(320, eps=1e-05, momentum=0.1,
```

```
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(320, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer8): _DenseLayer(
        (norm1): BatchNorm2d(352, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(352, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer9): _DenseLayer(
        (norm1): BatchNorm2d(384, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(384, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer10): _DenseLayer(
        (norm1): BatchNorm2d(416, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(416, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer11): _DenseLayer(
        (norm1): BatchNorm2d(448, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
```

```
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(448, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer12): _DenseLayer(
        (norm1): BatchNorm2d(480, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(480, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
)
(transition2): _Transition(
    (norm): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (pool): AvgPool2d(kernel_size=2, stride=2, padding=0)
)
(denseblock3): _DenseBlock(
    (denselayer1): _DenseLayer(
        (norm1): BatchNorm2d(256, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer2): _DenseLayer(
        (norm1): BatchNorm2d(288, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(288, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
```

```
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer3): _DenseLayer(
        (norm1): BatchNorm2d(320, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(320, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer4): _DenseLayer(
        (norm1): BatchNorm2d(352, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(352, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer5): _DenseLayer(
        (norm1): BatchNorm2d(384, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(384, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer6): _DenseLayer(
        (norm1): BatchNorm2d(416, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(416, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
```

```
affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
(denselayer7): _DenseLayer(
    (norm1): BatchNorm2d(448, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(448, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
(denselayer8): _DenseLayer(
    (norm1): BatchNorm2d(480, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(480, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
(denselayer9): _DenseLayer(
    (norm1): BatchNorm2d(512, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
(denselayer10): _DenseLayer(
    (norm1): BatchNorm2d(544, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(544, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
```

```
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer11): _DenseLayer(
        (norm1): BatchNorm2d(576, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(576, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer12): _DenseLayer(
        (norm1): BatchNorm2d(608, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(608, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer13): _DenseLayer(
        (norm1): BatchNorm2d(640, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(640, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer14): _DenseLayer(
        (norm1): BatchNorm2d(672, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(672, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
```

```
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer15): _DenseLayer(
        (norm1): BatchNorm2d(704, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(704, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer16): _DenseLayer(
        (norm1): BatchNorm2d(736, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(736, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer17): _DenseLayer(
        (norm1): BatchNorm2d(768, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(768, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer18): _DenseLayer(
        (norm1): BatchNorm2d(800, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(800, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
```

```
padding=(1, 1), bias=False)
)
(denselayer19): _DenseLayer(
    (norm1): BatchNorm2d(832, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(832, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
(denselayer20): _DenseLayer(
    (norm1): BatchNorm2d(864, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(864, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
(denselayer21): _DenseLayer(
    (norm1): BatchNorm2d(896, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(896, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
(denselayer22): _DenseLayer(
    (norm1): BatchNorm2d(928, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(928, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
```

```
(denselayer23): _DenseLayer(
    (norm1): BatchNorm2d(960, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(960, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
(denselayer24): _DenseLayer(
    (norm1): BatchNorm2d(992, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(992, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
)
(transition3): _Transition(
    (norm): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (pool): AvgPool2d(kernel_size=2, stride=2, padding=0)
)
(denseblock4): _DenseBlock(
    (denselayer1): _DenseLayer(
        (norm1): BatchNorm2d(512, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer2): _DenseLayer(
        (norm1): BatchNorm2d(544, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
```

```
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(544, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer3): _DenseLayer(
        (norm1): BatchNorm2d(576, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(576, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer4): _DenseLayer(
        (norm1): BatchNorm2d(608, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(608, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer5): _DenseLayer(
        (norm1): BatchNorm2d(640, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(640, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer6): _DenseLayer(
        (norm1): BatchNorm2d(672, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
```

```
(conv1): Conv2d(672, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
(denselayer7): _DenseLayer(
    (norm1): BatchNorm2d(704, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
    (conv1): Conv2d(704, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
(denselayer8): _DenseLayer(
    (norm1): BatchNorm2d(736, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
    (conv1): Conv2d(736, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
(denselayer9): _DenseLayer(
    (norm1): BatchNorm2d(768, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
    (conv1): Conv2d(768, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
(denselayer10): _DenseLayer(
    (norm1): BatchNorm2d(800, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
    (conv1): Conv2d(800, 128, kernel_size=(1, 1), stride=(1, 1),
```

```
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer11): _DenseLayer(
        (norm1): BatchNorm2d(832, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(832, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer12): _DenseLayer(
        (norm1): BatchNorm2d(864, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(864, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer13): _DenseLayer(
        (norm1): BatchNorm2d(896, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(896, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer14): _DenseLayer(
        (norm1): BatchNorm2d(928, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(928, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
```

```

        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer15): _DenseLayer(
        (norm1): BatchNorm2d(960, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(960, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer16): _DenseLayer(
        (norm1): BatchNorm2d(992, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(992, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1,
affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
)
(norm5): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
)
(classifier): Sequential(
    (0): Linear(in_features=1024, out_features=256, bias=True)
    (1): ReLU()
    (2): Dropout(p=0.4, inplace=False)
    (3): Linear(in_features=256, out_features=2, bias=True)
)
)

y_true, y_pred = [], []
with torch.no_grad():
    for inputs, labels in test_loader:
        inputs, labels = inputs.to(device), labels.to(device)
        outputs = model(inputs)
        _, preds = torch.max(outputs, 1)
        y_true.extend(labels.cpu().numpy())
        y_pred.extend(preds.cpu().numpy())

```

```
print("\nClassification Report:\n", classification_report(y_true,
y_pred, target_names=["PNEUMONIA", "NORMAL"]))
print("\nConfusion Matrix:\n", confusion_matrix(y_true, y_pred))
```

Classification Report:

	precision	recall	f1-score	support
PNEUMONIA	0.98	0.86	0.91	234
NORMAL	0.92	0.99	0.95	390
accuracy			0.94	624
macro avg	0.95	0.92	0.93	624
weighted avg	0.94	0.94	0.94	624

Confusion Matrix:

```
[[201  33]
 [ 5 385]]
```