WQF7009 Explainable AI: Assignment 3 - XAI on Chest X-Ray Pneumonia Classification

1. Setup & Import

In [1]:
```
!uv build ../
!uv pip install ../dist/wqf7009_a3-0.1.0-py3-none-any.whl
```

```
Building source distribution (uv build backend)...
Building wheel from source distribution (uv build backend)...
Successfully built d:\jherng\Workspace\university\masters\courses\year2526_sem1\wqf7009_explainable_artificial_intelligence\assignment3\wqf7009-a3\dist\wqf7009_a3-0.1.0.tar.gz
Successfully built d:\jherng\Workspace\university\masters\courses\year2526_sem1\wqf7009_explainable_artificial_intelligence\assignment3\wqf7009-a3\dist\wqf7009_a3-0.1.0-py3-none-any.whl
Using Python 3.12.12 environment at: D:\jherng\Workspace\university\masters\courses\year2526_sem1\wqf7009_explainable_artificial_intelligence\assignment3\wqf7009-a3\.venv
Resolved 47 packages in 1.13s
Prepared 1 package in 14ms
Uninstalled 1 package in 11ms
Installed 1 package in 10ms
 ~ wqf7009-a3==0.1.0 (from file:///D:/jherng/Workspace/university/masters/courses/year2526_sem1/wqf7009_explainable_artificial_intelligence/assignment3/wqf7009-a3/dist/wqf7009_a3-0.1.0-py3-none-any.whl)
```

In [2]:
```
%load_ext autoreload
%autoreload 2

import os
from pathlib import Path

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import torch
import torchsummary
from pytorch_grad_cam import GradCAM
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score

from wqf7009_a3.dataset import (
    IMAGENET_MEAN,
    IMAGENET_STD,
    IMG_SIZE,
    get_dataloader,
    transform,
)
from wqf7009_a3.infer import predict
from wqf7009_a3.loss import get_weighted_bce_loss
from wqf7009_a3.models import get_model
from wqf7009_a3.trainer import evaluate
from wqf7009_a3.utils import set_seed
from wqf7009_a3.xai import RISE, visualize_gradcam, visualize_rise
```

In [3]:
```
PROJECT_ROOT = Path(os.getcwd()).parent
DATA_DIR = PROJECT_ROOT / "data" / "chest_xray"
MODELS_DIR = PROJECT_ROOT / "models"  # Dir for model dumps
FIG_DIR = PROJECT_ROOT / "docs" / "figures"  # Dir for figures
LOG_DIR = PROJECT_ROOT / "runs"  # Dir for TensorBoard logs

TRAIN_DIR = DATA_DIR / "train"
VAL_DIR = DATA_DIR / "val"
TEST_DIR = DATA_DIR / "test"

CLASSES = ["NORMAL", "PNEUMONIA"]
BATCH_SIZE = 32  # Increase until out-of-memory errors occur
NUM_WORKERS = 4  # Number of subprocesses for data loading
DEVICE = torch.device("cuda" if torch.cuda.is_available() else "cpu")

set_seed(42)
LOG_DIR.mkdir(parents=True, exist_ok=True)
MODELS_DIR.mkdir(parents=True, exist_ok=True)
FIG_DIR.mkdir(parents=True, exist_ok=True)
```

In [4]:
```
print(f"Project root directory: {PROJECT_ROOT.as_posix()}")
print(f"Training data directory: {TRAIN_DIR.relative_to(PROJECT_ROOT).as_posix()}")
print(f"Validation data directory: {VAL_DIR.relative_to(PROJECT_ROOT).as_posix()}")
print(f"Test data directory: {TEST_DIR.relative_to(PROJECT_ROOT).as_posix()}")
print(f"Using device: {DEVICE}")
```

```
Project root directory: d:/jherng/Workspace/university/masters/courses/year2526_sem1/wqf7009_explainable_artificial_intelligence/assignment3/wqf7009-a3
Training data directory: data/chest_xray/train
Validation data directory: data/chest_xray/val
Test data directory: data/chest_xray/test
Using device: cuda
```

2. `Task 1` - Data Preparation & Pre-processing (5%)

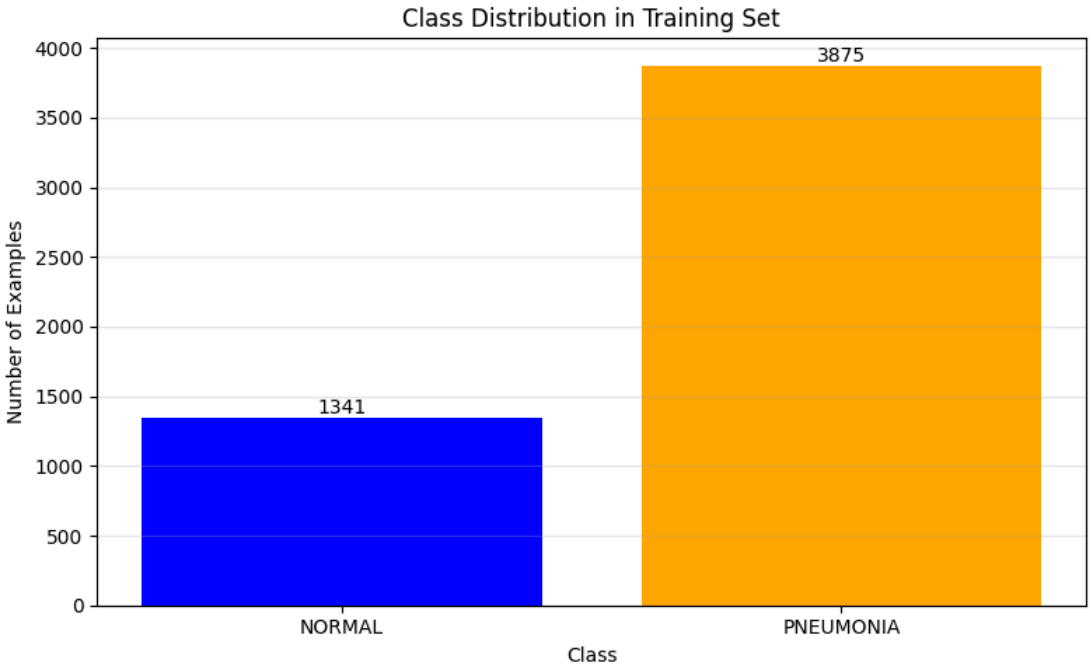
2.1. Explore the Dataset

Based on the plotted class distribution, we can observe that the dataset is imbalanced, with a higher number of 'NORMAL' cases (1341) compared to 'PNEUMONIA' cases (3875). This imbalance may affect the performance of machine learning models, as they might be biased towards the majority class.

To address this, we'll use ***class weighting*** during model training to give more importance to the minority class ('NORMAL').

In [5]:
```python
# Helper function to count images in each class
def count_images_by_class(data_dir, class_name):
    """Count number of images in a class directory."""
    return len(list((data_dir / class_name).glob("*.jpeg")))

# List number of examples in each class
print("Dataset Statistics:")
print("-" * 50)
for class_dir in CLASSES:
    train_count = count_images_by_class(TRAIN_DIR, class_dir)
    val_count = count_images_by_class(VAL_DIR, class_dir)
    test_count = count_images_by_class(TEST_DIR, class_dir)
    print(f"{class_dir:12s}: Train={train_count:4d}, Val={val_count:3d}, Test={test_count:3d}")

# Plot the class distribution
train_counts = [count_images_by_class(TRAIN_DIR, cls) for cls in CLASSES]
plt.figure(figsize=(8, 5))
plt.bar(CLASSES, train_counts, color=["blue", "orange"])
plt.title("Class Distribution in Training Set")
plt.xlabel("Class")
plt.ylabel("Number of Examples")
plt.grid(axis="y", alpha=0.3)
for i, count in enumerate(train_counts):
    plt.text(i, count, str(count), ha="center", va="bottom")
plt.tight_layout()
plt.savefig(FIG_DIR / "class_distribution.png", dpi=150)
plt.show()
```

```
Dataset Statistics:
--------------------------------------------------
NORMAL      : Train=1341, Val=  8, Test=234
PNEUMONIA   : Train=3875, Val=  8, Test=390
```



2.2. Data Pre-processing

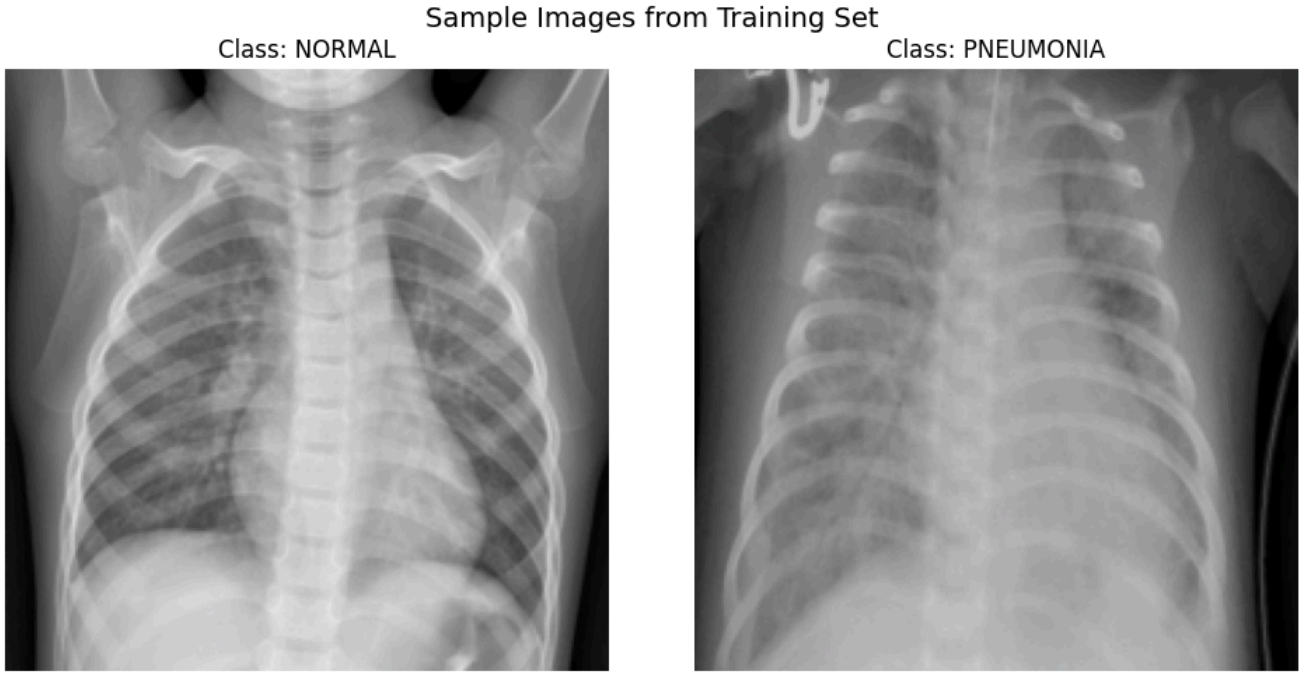
In [6]:
```python
# Data preprocessing pipeline
print("Transform pipeline:")
print(transform)
```

```
Transform pipeline:
Compose(
    Resize(size=(256, 256), interpolation=bilinear, max_size=None, antialias=True)
    CenterCrop(size=(224, 224))
    ToTensor()
    Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
)
```

In [7]:
```python
train_loader = get_dataloader(
    data_dir=DATA_DIR,
    split="train",
    batch_size=BATCH_SIZE,
    shuffle=True,
    num_workers=NUM_WORKERS,
    pin_memory=DEVICE.type == "cuda",
    transform=transform,
)
val_loader = get_dataloader(
    data_dir=DATA_DIR,
    split="val",
    batch_size=BATCH_SIZE,
    shuffle=False,
    num_workers=NUM_WORKERS,
    pin_memory=DEVICE.type == "cuda",
    transform=transform,
)
test_loader = get_dataloader(
    data_dir=DATA_DIR,
    split="test",
    batch_size=BATCH_SIZE,
    shuffle=False,
    num_workers=NUM_WORKERS,
    pin_memory=DEVICE.type == "cuda",
    transform=transform,
)
```

In [8]:
```python
# Helper function to denormalize image for visualization
def denormalize_image(tensor_img):
    """Convert normalized tensor image back to displayable format."""
    img = tensor_img.permute(1, 2, 0).numpy()
    img = img * np.array(IMAGENET_STD) + np.array(IMAGENET_MEAN)
    return np.clip(img, 0, 1)

# Sanity checks
print("Dataset Information:")
print("-" * 50)
print(f"Training examples:   {len(train_loader.dataset):4d}")
print(f"Validation examples: {len(val_loader.dataset):4d}")
print(f"Test examples:       {len(test_loader.dataset):4d}")
print(f"Classes: {train_loader.dataset.classes}")
print(f"Class to index: {train_loader.dataset.class_to_idx}")

# Plot sample images from each class
normal_idx = np.where(np.array(train_loader.dataset.targets) == 0)[0][0]
pneumonia_idx = np.where(np.array(train_loader.dataset.targets) == 1)[0][0]
normal_img, normal_label = train_loader.dataset[normal_idx]
pneumonia_img, pneumonia_label = train_loader.dataset[pneumonia_idx]

fig, axs = plt.subplots(1, 2, figsize=(10, 5))
axs[0].imshow(denormalize_image(normal_img))
axs[0].set_title(f"Class: {train_loader.dataset.classes[normal_label]}")
axs[0].axis("off")
axs[1].imshow(denormalize_image(pneumonia_img))
axs[1].set_title(f"Class: {train_loader.dataset.classes[pneumonia_label]}")
axs[1].axis("off")
plt.suptitle("Sample Images from Training Set", fontsize=14)
plt.tight_layout()
plt.show()

# Verify class mapping
assert train_loader.dataset.class_to_idx == {"NORMAL": 0, "PNEUMONIA": 1}, (
    "Class to index mapping is incorrect. Expected {'NORMAL': 0, 'PNEUMONIA': 1}"
)
```

```
Dataset Information:
--------------------------------------------------
Training examples:   5216
Validation examples:   16
Test examples:        624
Classes: ['NORMAL', 'PNEUMONIA']
Class to index: {'NORMAL': 0, 'PNEUMONIA': 1}
```



3. `Task 2` - Model Training, Comparison & Evaluation (5%)

3.1. Model Summary

In [9]:
```python
# Initialize all models
models_config = {
    "simple_cnn": {"model_name": "simplecnn", "freeze_features": None},
    "resnet152_frozen": {"model_name": "resnet152", "freeze_features": True},
    "resnet152_unfrozen": {"model_name": "resnet152", "freeze_features": False},
    "vgg16_frozen": {"model_name": "vgg16", "freeze_features": True},
    "vgg16_unfrozen": {"model_name": "vgg16", "freeze_features": False},
}

models = {}
for name, config in models_config.items():
    kwargs = {"num_classes": 1}
    if config["freeze_features"] is not None:
        kwargs["freeze_features"] = config["freeze_features"]
    models[name] = get_model(model_name=config["model_name"], **kwargs).to(DEVICE)
    print(f"Initialized {name}")

# Keep individual variables for backward compatibility
simple_cnn = models["simple_cnn"]
resnet152_frozen = models["resnet152_frozen"]
resnet152_unfrozen = models["resnet152_unfrozen"]
vgg16_frozen = models["vgg16_frozen"]
vgg16_unfrozen = models["vgg16_unfrozen"]
```

```
Initialized simple_cnn
Initialized resnet152_frozen
Initialized resnet152_unfrozen
Initialized vgg16_frozen
Initialized vgg16_unfrozen
```

In [10]:
```python
# Helper function to print model summary
def print_model_summary(model, model_name=None):
    """Print model architecture summary."""
    name = model_name or model._get_name()
    print(f"\n{'='*70}")
    print(f"Model: {name}")
    print(f"{'='*70}")
    torchsummary.summary(model.to(DEVICE), (3, IMG_SIZE, IMG_SIZE), device=str(DEVICE))

print_model_summary(simple_cnn, "SimpleCNN")
```

```
================================================================
Model: SimpleCNN
================================================================
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 32, 224, 224]             896
              ReLU-2         [-1, 32, 224, 224]               0
         MaxPool2d-3         [-1, 32, 112, 112]               0
            Conv2d-4         [-1, 64, 112, 112]          18,496
              ReLU-5         [-1, 64, 112, 112]               0
         MaxPool2d-6           [-1, 64, 56, 56]               0
            Conv2d-7          [-1, 128, 56, 56]          73,856
              ReLU-8          [-1, 128, 56, 56]               0
         MaxPool2d-9          [-1, 128, 28, 28]               0
AdaptiveAvgPool2d-10            [-1, 128, 7, 7]               0
          Flatten-11                 [-1, 6272]               0
           Linear-12                  [-1, 256]       1,605,888
             ReLU-13                  [-1, 256]               0
          Dropout-14                  [-1, 256]               0
           Linear-15                    [-1, 1]             257
================================================================
Total params: 1,699,393
Trainable params: 1,699,393
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 48.34
Params size (MB): 6.48
Estimated Total Size (MB): 55.39
----------------------------------------------------------------
```

In [11]:
```
print_model_summary(resnet152_frozen, "ResNet152 (frozen features)")
print_model_summary(resnet152_unfrozen, "ResNet152 (unfrozen features)")
```

```
================================================================
Model: ResNet152 (frozen features)
================================================================
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 64, 112, 112]           9,408
       BatchNorm2d-2         [-1, 64, 112, 112]             128
              ReLU-3         [-1, 64, 112, 112]               0
         MaxPool2d-4           [-1, 64, 56, 56]               0
            Conv2d-5           [-1, 64, 56, 56]           4,096
       BatchNorm2d-6           [-1, 64, 56, 56]             128
              ReLU-7           [-1, 64, 56, 56]               0
            Conv2d-8           [-1, 64, 56, 56]          36,864
       BatchNorm2d-9           [-1, 64, 56, 56]             128
             ReLU-10           [-1, 64, 56, 56]               0
           Conv2d-11          [-1, 256, 56, 56]          16,384
      BatchNorm2d-12          [-1, 256, 56, 56]             512
           Conv2d-13          [-1, 256, 56, 56]          16,384
      BatchNorm2d-14          [-1, 256, 56, 56]             512
             ReLU-15          [-1, 256, 56, 56]               0
       Bottleneck-16          [-1, 256, 56, 56]               0
           Conv2d-17           [-1, 64, 56, 56]          16,384
      BatchNorm2d-18           [-1, 64, 56, 56]             128
             ReLU-19           [-1, 64, 56, 56]               0
           Conv2d-20           [-1, 64, 56, 56]          36,864
      BatchNorm2d-21           [-1, 64, 56, 56]             128
             ReLU-22           [-1, 64, 56, 56]               0
           Conv2d-23          [-1, 256, 56, 56]          16,384
      BatchNorm2d-24          [-1, 256, 56, 56]             512
             ReLU-25          [-1, 256, 56, 56]               0
       Bottleneck-26          [-1, 256, 56, 56]               0
           Conv2d-27           [-1, 64, 56, 56]          16,384
      BatchNorm2d-28           [-1, 64, 56, 56]             128
             ReLU-29           [-1, 64, 56, 56]               0
           Conv2d-30           [-1, 64, 56, 56]          36,864
      BatchNorm2d-31           [-1, 64, 56, 56]             128
             ReLU-32           [-1, 64, 56, 56]               0
           Conv2d-33          [-1, 256, 56, 56]          16,384
      BatchNorm2d-34          [-1, 256, 56, 56]             512
             ReLU-35          [-1, 256, 56, 56]               0
       Bottleneck-36          [-1, 256, 56, 56]               0
           Conv2d-37          [-1, 128, 56, 56]          32,768
      BatchNorm2d-38          [-1, 128, 56, 56]             256
             ReLU-39          [-1, 128, 56, 56]               0
           Conv2d-40          [-1, 128, 28, 28]         147,456
      BatchNorm2d-41          [-1, 128, 28, 28]             256
             ReLU-42          [-1, 128, 28, 28]               0
           Conv2d-43          [-1, 512, 28, 28]          65,536
      BatchNorm2d-44          [-1, 512, 28, 28]           1,024
           Conv2d-45          [-1, 512, 28, 28]         131,072
      BatchNorm2d-46          [-1, 512, 28, 28]           1,024
             ReLU-47          [-1, 512, 28, 28]               0
       Bottleneck-48          [-1, 512, 28, 28]               0
           Conv2d-49          [-1, 128, 28, 28]          65,536
      BatchNorm2d-50          [-1, 128, 28, 28]             256
             ReLU-51          [-1, 128, 28, 28]               0
           Conv2d-52          [-1, 128, 28, 28]         147,456
      BatchNorm2d-53          [-1, 128, 28, 28]             256
             ReLU-54          [-1, 128, 28, 28]               0
           Conv2d-55          [-1, 512, 28, 28]          65,536
      BatchNorm2d-56          [-1, 512, 28, 28]           1,024
             ReLU-57          [-1, 512, 28, 28]               0
       Bottleneck-58          [-1, 512, 28, 28]               0
           Conv2d-59          [-1, 128, 28, 28]          65,536
      BatchNorm2d-60          [-1, 128, 28, 28]             256
             ReLU-61          [-1, 128, 28, 28]               0
           Conv2d-62          [-1, 128, 28, 28]         147,456
      BatchNorm2d-63          [-1, 128, 28, 28]             256
             ReLU-64          [-1, 128, 28, 28]               0
           Conv2d-65          [-1, 512, 28, 28]          65,536
      BatchNorm2d-66          [-1, 512, 28, 28]           1,024
             ReLU-67          [-1, 512, 28, 28]               0
       Bottleneck-68          [-1, 512, 28, 28]               0
           Conv2d-69          [-1, 128, 28, 28]          65,536
      BatchNorm2d-70          [-1, 128, 28, 28]             256
             ReLU-71          [-1, 128, 28, 28]               0
           Conv2d-72          [-1, 128, 28, 28]         147,456
      BatchNorm2d-73          [-1, 128, 28, 28]             256
             ReLU-74          [-1, 128, 28, 28]               0
           Conv2d-75          [-1, 512, 28, 28]          65,536
      BatchNorm2d-76          [-1, 512, 28, 28]           1,024
             ReLU-77          [-1, 512, 28, 28]               0
       Bottleneck-78          [-1, 512, 28, 28]               0
           Conv2d-79          [-1, 128, 28, 28]          65,536
      BatchNorm2d-80          [-1, 128, 28, 28]             256
             ReLU-81          [-1, 128, 28, 28]               0
           Conv2d-82          [-1, 128, 28, 28]         147,456
```

```
 BatchNorm2d-83          [-1, 128, 28, 28]             256
        ReLU-84          [-1, 128, 28, 28]               0
      Conv2d-85          [-1, 512, 28, 28]          65,536
 BatchNorm2d-86          [-1, 512, 28, 28]           1,024
        ReLU-87          [-1, 512, 28, 28]               0
  Bottleneck-88          [-1, 512, 28, 28]               0
      Conv2d-89          [-1, 128, 28, 28]          65,536
 BatchNorm2d-90          [-1, 128, 28, 28]             256
        ReLU-91          [-1, 128, 28, 28]               0
      Conv2d-92          [-1, 128, 28, 28]         147,456
 BatchNorm2d-93          [-1, 128, 28, 28]             256
        ReLU-94          [-1, 128, 28, 28]               0
      Conv2d-95          [-1, 512, 28, 28]          65,536
 BatchNorm2d-96          [-1, 512, 28, 28]           1,024
        ReLU-97          [-1, 512, 28, 28]               0
  Bottleneck-98          [-1, 512, 28, 28]               0
      Conv2d-99          [-1, 128, 28, 28]          65,536
BatchNorm2d-100          [-1, 128, 28, 28]             256
       ReLU-101          [-1, 128, 28, 28]               0
     Conv2d-102          [-1, 128, 28, 28]         147,456
BatchNorm2d-103          [-1, 128, 28, 28]             256
       ReLU-104          [-1, 128, 28, 28]               0
     Conv2d-105          [-1, 512, 28, 28]          65,536
BatchNorm2d-106          [-1, 512, 28, 28]           1,024
       ReLU-107          [-1, 512, 28, 28]               0
 Bottleneck-108          [-1, 512, 28, 28]               0
     Conv2d-109          [-1, 128, 28, 28]          65,536
BatchNorm2d-110          [-1, 128, 28, 28]             256
       ReLU-111          [-1, 128, 28, 28]               0
     Conv2d-112          [-1, 128, 28, 28]         147,456
BatchNorm2d-113          [-1, 128, 28, 28]             256
       ReLU-114          [-1, 128, 28, 28]               0
     Conv2d-115          [-1, 512, 28, 28]          65,536
BatchNorm2d-116          [-1, 512, 28, 28]           1,024
       ReLU-117          [-1, 512, 28, 28]               0
 Bottleneck-118          [-1, 512, 28, 28]               0
     Conv2d-119          [-1, 256, 28, 28]         131,072
BatchNorm2d-120          [-1, 256, 28, 28]             512
       ReLU-121          [-1, 256, 28, 28]               0
     Conv2d-122          [-1, 256, 14, 14]         589,824
BatchNorm2d-123          [-1, 256, 14, 14]             512
       ReLU-124          [-1, 256, 14, 14]               0
     Conv2d-125         [-1, 1024, 14, 14]         262,144
BatchNorm2d-126         [-1, 1024, 14, 14]           2,048
     Conv2d-127         [-1, 1024, 14, 14]         524,288
BatchNorm2d-128         [-1, 1024, 14, 14]           2,048
       ReLU-129         [-1, 1024, 14, 14]               0
 Bottleneck-130         [-1, 1024, 14, 14]               0
     Conv2d-131          [-1, 256, 14, 14]         262,144
BatchNorm2d-132          [-1, 256, 14, 14]             512
       ReLU-133          [-1, 256, 14, 14]               0
     Conv2d-134          [-1, 256, 14, 14]         589,824
BatchNorm2d-135          [-1, 256, 14, 14]             512
       ReLU-136          [-1, 256, 14, 14]               0
     Conv2d-137         [-1, 1024, 14, 14]         262,144
BatchNorm2d-138         [-1, 1024, 14, 14]           2,048
       ReLU-139         [-1, 1024, 14, 14]               0
 Bottleneck-140         [-1, 1024, 14, 14]               0
     Conv2d-141          [-1, 256, 14, 14]         262,144
BatchNorm2d-142          [-1, 256, 14, 14]             512
       ReLU-143          [-1, 256, 14, 14]               0
     Conv2d-144          [-1, 256, 14, 14]         589,824
BatchNorm2d-145          [-1, 256, 14, 14]             512
       ReLU-146          [-1, 256, 14, 14]               0
     Conv2d-147         [-1, 1024, 14, 14]         262,144
BatchNorm2d-148         [-1, 1024, 14, 14]           2,048
       ReLU-149         [-1, 1024, 14, 14]               0
 Bottleneck-150         [-1, 1024, 14, 14]               0
     Conv2d-151          [-1, 256, 14, 14]         262,144
BatchNorm2d-152          [-1, 256, 14, 14]             512
       ReLU-153          [-1, 256, 14, 14]               0
     Conv2d-154          [-1, 256, 14, 14]         589,824
BatchNorm2d-155          [-1, 256, 14, 14]             512
       ReLU-156          [-1, 256, 14, 14]               0
     Conv2d-157         [-1, 1024, 14, 14]         262,144
BatchNorm2d-158         [-1, 1024, 14, 14]           2,048
       ReLU-159         [-1, 1024, 14, 14]               0
 Bottleneck-160         [-1, 1024, 14, 14]               0
     Conv2d-161          [-1, 256, 14, 14]         262,144
BatchNorm2d-162          [-1, 256, 14, 14]             512
       ReLU-163          [-1, 256, 14, 14]               0
     Conv2d-164          [-1, 256, 14, 14]         589,824
BatchNorm2d-165          [-1, 256, 14, 14]             512
       ReLU-166          [-1, 256, 14, 14]               0
     Conv2d-167         [-1, 1024, 14, 14]         262,144
BatchNorm2d-168         [-1, 1024, 14, 14]           2,048
       ReLU-169         [-1, 1024, 14, 14]               0
 Bottleneck-170         [-1, 1024, 14, 14]               0
```

```
     Conv2d-171         [-1, 256, 14, 14]         262,144
BatchNorm2d-172         [-1, 256, 14, 14]             512
       ReLU-173         [-1, 256, 14, 14]               0
     Conv2d-174         [-1, 256, 14, 14]         589,824
BatchNorm2d-175         [-1, 256, 14, 14]             512
       ReLU-176         [-1, 256, 14, 14]               0
     Conv2d-177        [-1, 1024, 14, 14]         262,144
BatchNorm2d-178        [-1, 1024, 14, 14]           2,048
       ReLU-179        [-1, 1024, 14, 14]               0
 Bottleneck-180        [-1, 1024, 14, 14]               0
     Conv2d-181         [-1, 256, 14, 14]         262,144
BatchNorm2d-182         [-1, 256, 14, 14]             512
       ReLU-183         [-1, 256, 14, 14]               0
     Conv2d-184         [-1, 256, 14, 14]         589,824
BatchNorm2d-185         [-1, 256, 14, 14]             512
       ReLU-186         [-1, 256, 14, 14]               0
     Conv2d-187        [-1, 1024, 14, 14]         262,144
BatchNorm2d-188        [-1, 1024, 14, 14]           2,048
       ReLU-189        [-1, 1024, 14, 14]               0
 Bottleneck-190        [-1, 1024, 14, 14]               0
     Conv2d-191         [-1, 256, 14, 14]         262,144
BatchNorm2d-192         [-1, 256, 14, 14]             512
       ReLU-193         [-1, 256, 14, 14]               0
     Conv2d-194         [-1, 256, 14, 14]         589,824
BatchNorm2d-195         [-1, 256, 14, 14]             512
       ReLU-196         [-1, 256, 14, 14]               0
     Conv2d-197        [-1, 1024, 14, 14]         262,144
BatchNorm2d-198        [-1, 1024, 14, 14]           2,048
       ReLU-199        [-1, 1024, 14, 14]               0
 Bottleneck-200        [-1, 1024, 14, 14]               0
     Conv2d-201         [-1, 256, 14, 14]         262,144
BatchNorm2d-202         [-1, 256, 14, 14]             512
       ReLU-203         [-1, 256, 14, 14]               0
     Conv2d-204         [-1, 256, 14, 14]         589,824
BatchNorm2d-205         [-1, 256, 14, 14]             512
       ReLU-206         [-1, 256, 14, 14]               0
     Conv2d-207        [-1, 1024, 14, 14]         262,144
BatchNorm2d-208        [-1, 1024, 14, 14]           2,048
       ReLU-209        [-1, 1024, 14, 14]               0
 Bottleneck-210        [-1, 1024, 14, 14]               0
     Conv2d-211         [-1, 256, 14, 14]         262,144
BatchNorm2d-212         [-1, 256, 14, 14]             512
       ReLU-213         [-1, 256, 14, 14]               0
     Conv2d-214         [-1, 256, 14, 14]         589,824
BatchNorm2d-215         [-1, 256, 14, 14]             512
       ReLU-216         [-1, 256, 14, 14]               0
     Conv2d-217        [-1, 1024, 14, 14]         262,144
BatchNorm2d-218        [-1, 1024, 14, 14]           2,048
       ReLU-219        [-1, 1024, 14, 14]               0
 Bottleneck-220        [-1, 1024, 14, 14]               0
     Conv2d-221         [-1, 256, 14, 14]         262,144
BatchNorm2d-222         [-1, 256, 14, 14]             512
       ReLU-223         [-1, 256, 14, 14]               0
     Conv2d-224         [-1, 256, 14, 14]         589,824
BatchNorm2d-225         [-1, 256, 14, 14]             512
       ReLU-226         [-1, 256, 14, 14]               0
     Conv2d-227        [-1, 1024, 14, 14]         262,144
BatchNorm2d-228        [-1, 1024, 14, 14]           2,048
       ReLU-229        [-1, 1024, 14, 14]               0
 Bottleneck-230        [-1, 1024, 14, 14]               0
     Conv2d-231         [-1, 256, 14, 14]         262,144
BatchNorm2d-232         [-1, 256, 14, 14]             512
       ReLU-233         [-1, 256, 14, 14]               0
     Conv2d-234         [-1, 256, 14, 14]         589,824
BatchNorm2d-235         [-1, 256, 14, 14]             512
       ReLU-236         [-1, 256, 14, 14]               0
     Conv2d-237        [-1, 1024, 14, 14]         262,144
BatchNorm2d-238        [-1, 1024, 14, 14]           2,048
       ReLU-239        [-1, 1024, 14, 14]               0
 Bottleneck-240        [-1, 1024, 14, 14]               0
     Conv2d-241         [-1, 256, 14, 14]         262,144
BatchNorm2d-242         [-1, 256, 14, 14]             512
       ReLU-243         [-1, 256, 14, 14]               0
     Conv2d-244         [-1, 256, 14, 14]         589,824
BatchNorm2d-245         [-1, 256, 14, 14]             512
       ReLU-246         [-1, 256, 14, 14]               0
     Conv2d-247        [-1, 1024, 14, 14]         262,144
BatchNorm2d-248        [-1, 1024, 14, 14]           2,048
       ReLU-249        [-1, 1024, 14, 14]               0
 Bottleneck-250        [-1, 1024, 14, 14]               0
     Conv2d-251         [-1, 256, 14, 14]         262,144
BatchNorm2d-252         [-1, 256, 14, 14]             512
       ReLU-253         [-1, 256, 14, 14]               0
     Conv2d-254         [-1, 256, 14, 14]         589,824
BatchNorm2d-255         [-1, 256, 14, 14]             512
       ReLU-256         [-1, 256, 14, 14]               0
     Conv2d-257        [-1, 1024, 14, 14]         262,144
BatchNorm2d-258        [-1, 1024, 14, 14]           2,048
```

```
       ReLU-259        [-1, 1024, 14, 14]               0
 Bottleneck-260        [-1, 1024, 14, 14]               0
     Conv2d-261         [-1, 256, 14, 14]         262,144
BatchNorm2d-262         [-1, 256, 14, 14]             512
       ReLU-263         [-1, 256, 14, 14]               0
     Conv2d-264         [-1, 256, 14, 14]         589,824
BatchNorm2d-265         [-1, 256, 14, 14]             512
       ReLU-266         [-1, 256, 14, 14]               0
     Conv2d-267        [-1, 1024, 14, 14]         262,144
BatchNorm2d-268        [-1, 1024, 14, 14]           2,048
       ReLU-269        [-1, 1024, 14, 14]               0
 Bottleneck-270        [-1, 1024, 14, 14]               0
     Conv2d-271         [-1, 256, 14, 14]         262,144
BatchNorm2d-272         [-1, 256, 14, 14]             512
       ReLU-273         [-1, 256, 14, 14]               0
     Conv2d-274         [-1, 256, 14, 14]         589,824
BatchNorm2d-275         [-1, 256, 14, 14]             512
       ReLU-276         [-1, 256, 14, 14]               0
     Conv2d-277        [-1, 1024, 14, 14]         262,144
BatchNorm2d-278        [-1, 1024, 14, 14]           2,048
       ReLU-279        [-1, 1024, 14, 14]               0
 Bottleneck-280        [-1, 1024, 14, 14]               0
     Conv2d-281         [-1, 256, 14, 14]         262,144
BatchNorm2d-282         [-1, 256, 14, 14]             512
       ReLU-283         [-1, 256, 14, 14]               0
     Conv2d-284         [-1, 256, 14, 14]         589,824
BatchNorm2d-285         [-1, 256, 14, 14]             512
       ReLU-286         [-1, 256, 14, 14]               0
     Conv2d-287        [-1, 1024, 14, 14]         262,144
BatchNorm2d-288        [-1, 1024, 14, 14]           2,048
       ReLU-289        [-1, 1024, 14, 14]               0
 Bottleneck-290        [-1, 1024, 14, 14]               0
     Conv2d-291         [-1, 256, 14, 14]         262,144
BatchNorm2d-292         [-1, 256, 14, 14]             512
       ReLU-293         [-1, 256, 14, 14]               0
     Conv2d-294         [-1, 256, 14, 14]         589,824
BatchNorm2d-295         [-1, 256, 14, 14]             512
       ReLU-296         [-1, 256, 14, 14]               0
     Conv2d-297        [-1, 1024, 14, 14]         262,144
BatchNorm2d-298        [-1, 1024, 14, 14]           2,048
       ReLU-299        [-1, 1024, 14, 14]               0
 Bottleneck-300        [-1, 1024, 14, 14]               0
     Conv2d-301         [-1, 256, 14, 14]         262,144
BatchNorm2d-302         [-1, 256, 14, 14]             512
       ReLU-303         [-1, 256, 14, 14]               0
     Conv2d-304         [-1, 256, 14, 14]         589,824
BatchNorm2d-305         [-1, 256, 14, 14]             512
       ReLU-306         [-1, 256, 14, 14]               0
     Conv2d-307        [-1, 1024, 14, 14]         262,144
BatchNorm2d-308        [-1, 1024, 14, 14]           2,048
       ReLU-309        [-1, 1024, 14, 14]               0
 Bottleneck-310        [-1, 1024, 14, 14]               0
     Conv2d-311         [-1, 256, 14, 14]         262,144
BatchNorm2d-312         [-1, 256, 14, 14]             512
       ReLU-313         [-1, 256, 14, 14]               0
     Conv2d-314         [-1, 256, 14, 14]         589,824
BatchNorm2d-315         [-1, 256, 14, 14]             512
       ReLU-316         [-1, 256, 14, 14]               0
     Conv2d-317        [-1, 1024, 14, 14]         262,144
BatchNorm2d-318        [-1, 1024, 14, 14]           2,048
       ReLU-319        [-1, 1024, 14, 14]               0
 Bottleneck-320        [-1, 1024, 14, 14]               0
     Conv2d-321         [-1, 256, 14, 14]         262,144
BatchNorm2d-322         [-1, 256, 14, 14]             512
       ReLU-323         [-1, 256, 14, 14]               0
     Conv2d-324         [-1, 256, 14, 14]         589,824
BatchNorm2d-325         [-1, 256, 14, 14]             512
       ReLU-326         [-1, 256, 14, 14]               0
     Conv2d-327        [-1, 1024, 14, 14]         262,144
BatchNorm2d-328        [-1, 1024, 14, 14]           2,048
       ReLU-329        [-1, 1024, 14, 14]               0
 Bottleneck-330        [-1, 1024, 14, 14]               0
     Conv2d-331         [-1, 256, 14, 14]         262,144
BatchNorm2d-332         [-1, 256, 14, 14]             512
       ReLU-333         [-1, 256, 14, 14]               0
     Conv2d-334         [-1, 256, 14, 14]         589,824
BatchNorm2d-335         [-1, 256, 14, 14]             512
       ReLU-336         [-1, 256, 14, 14]               0
     Conv2d-337        [-1, 1024, 14, 14]         262,144
BatchNorm2d-338        [-1, 1024, 14, 14]           2,048
       ReLU-339        [-1, 1024, 14, 14]               0
 Bottleneck-340        [-1, 1024, 14, 14]               0
     Conv2d-341         [-1, 256, 14, 14]         262,144
BatchNorm2d-342         [-1, 256, 14, 14]             512
       ReLU-343         [-1, 256, 14, 14]               0
     Conv2d-344         [-1, 256, 14, 14]         589,824
BatchNorm2d-345         [-1, 256, 14, 14]             512
       ReLU-346         [-1, 256, 14, 14]               0
```

```
     Conv2d-347        [-1, 1024, 14, 14]         262,144
BatchNorm2d-348        [-1, 1024, 14, 14]           2,048
       ReLU-349        [-1, 1024, 14, 14]               0
 Bottleneck-350        [-1, 1024, 14, 14]               0
     Conv2d-351         [-1, 256, 14, 14]         262,144
BatchNorm2d-352         [-1, 256, 14, 14]             512
       ReLU-353         [-1, 256, 14, 14]               0
     Conv2d-354         [-1, 256, 14, 14]         589,824
BatchNorm2d-355         [-1, 256, 14, 14]             512
       ReLU-356         [-1, 256, 14, 14]               0
     Conv2d-357        [-1, 1024, 14, 14]         262,144
BatchNorm2d-358        [-1, 1024, 14, 14]           2,048
       ReLU-359        [-1, 1024, 14, 14]               0
 Bottleneck-360        [-1, 1024, 14, 14]               0
     Conv2d-361         [-1, 256, 14, 14]         262,144
BatchNorm2d-362         [-1, 256, 14, 14]             512
       ReLU-363         [-1, 256, 14, 14]               0
     Conv2d-364         [-1, 256, 14, 14]         589,824
BatchNorm2d-365         [-1, 256, 14, 14]             512
       ReLU-366         [-1, 256, 14, 14]               0
     Conv2d-367        [-1, 1024, 14, 14]         262,144
BatchNorm2d-368        [-1, 1024, 14, 14]           2,048
       ReLU-369        [-1, 1024, 14, 14]               0
 Bottleneck-370        [-1, 1024, 14, 14]               0
     Conv2d-371         [-1, 256, 14, 14]         262,144
BatchNorm2d-372         [-1, 256, 14, 14]             512
       ReLU-373         [-1, 256, 14, 14]               0
     Conv2d-374         [-1, 256, 14, 14]         589,824
BatchNorm2d-375         [-1, 256, 14, 14]             512
       ReLU-376         [-1, 256, 14, 14]               0
     Conv2d-377        [-1, 1024, 14, 14]         262,144
BatchNorm2d-378        [-1, 1024, 14, 14]           2,048
       ReLU-379        [-1, 1024, 14, 14]               0
 Bottleneck-380        [-1, 1024, 14, 14]               0
     Conv2d-381         [-1, 256, 14, 14]         262,144
BatchNorm2d-382         [-1, 256, 14, 14]             512
       ReLU-383         [-1, 256, 14, 14]               0
     Conv2d-384         [-1, 256, 14, 14]         589,824
BatchNorm2d-385         [-1, 256, 14, 14]             512
       ReLU-386         [-1, 256, 14, 14]               0
     Conv2d-387        [-1, 1024, 14, 14]         262,144
BatchNorm2d-388        [-1, 1024, 14, 14]           2,048
       ReLU-389        [-1, 1024, 14, 14]               0
 Bottleneck-390        [-1, 1024, 14, 14]               0
     Conv2d-391         [-1, 256, 14, 14]         262,144
BatchNorm2d-392         [-1, 256, 14, 14]             512
       ReLU-393         [-1, 256, 14, 14]               0
     Conv2d-394         [-1, 256, 14, 14]         589,824
BatchNorm2d-395         [-1, 256, 14, 14]             512
       ReLU-396         [-1, 256, 14, 14]               0
     Conv2d-397        [-1, 1024, 14, 14]         262,144
BatchNorm2d-398        [-1, 1024, 14, 14]           2,048
       ReLU-399        [-1, 1024, 14, 14]               0
 Bottleneck-400        [-1, 1024, 14, 14]               0
     Conv2d-401         [-1, 256, 14, 14]         262,144
BatchNorm2d-402         [-1, 256, 14, 14]             512
       ReLU-403         [-1, 256, 14, 14]               0
     Conv2d-404         [-1, 256, 14, 14]         589,824
BatchNorm2d-405         [-1, 256, 14, 14]             512
       ReLU-406         [-1, 256, 14, 14]               0
     Conv2d-407        [-1, 1024, 14, 14]         262,144
BatchNorm2d-408        [-1, 1024, 14, 14]           2,048
       ReLU-409        [-1, 1024, 14, 14]               0
 Bottleneck-410        [-1, 1024, 14, 14]               0
     Conv2d-411         [-1, 256, 14, 14]         262,144
BatchNorm2d-412         [-1, 256, 14, 14]             512
       ReLU-413         [-1, 256, 14, 14]               0
     Conv2d-414         [-1, 256, 14, 14]         589,824
BatchNorm2d-415         [-1, 256, 14, 14]             512
       ReLU-416         [-1, 256, 14, 14]               0
     Conv2d-417        [-1, 1024, 14, 14]         262,144
BatchNorm2d-418        [-1, 1024, 14, 14]           2,048
       ReLU-419        [-1, 1024, 14, 14]               0
 Bottleneck-420        [-1, 1024, 14, 14]               0
     Conv2d-421         [-1, 256, 14, 14]         262,144
BatchNorm2d-422         [-1, 256, 14, 14]             512
       ReLU-423         [-1, 256, 14, 14]               0
     Conv2d-424         [-1, 256, 14, 14]         589,824
BatchNorm2d-425         [-1, 256, 14, 14]             512
       ReLU-426         [-1, 256, 14, 14]               0
     Conv2d-427        [-1, 1024, 14, 14]         262,144
BatchNorm2d-428        [-1, 1024, 14, 14]           2,048
       ReLU-429        [-1, 1024, 14, 14]               0
 Bottleneck-430        [-1, 1024, 14, 14]               0
     Conv2d-431         [-1, 256, 14, 14]         262,144
BatchNorm2d-432         [-1, 256, 14, 14]             512
       ReLU-433         [-1, 256, 14, 14]               0
     Conv2d-434         [-1, 256, 14, 14]         589,824
```

```
      BatchNorm2d-435          [-1, 256, 14, 14]             512
             ReLU-436          [-1, 256, 14, 14]               0
           Conv2d-437         [-1, 1024, 14, 14]         262,144
      BatchNorm2d-438         [-1, 1024, 14, 14]           2,048
             ReLU-439         [-1, 1024, 14, 14]               0
       Bottleneck-440         [-1, 1024, 14, 14]               0
           Conv2d-441          [-1, 256, 14, 14]         262,144
      BatchNorm2d-442          [-1, 256, 14, 14]             512
             ReLU-443          [-1, 256, 14, 14]               0
           Conv2d-444          [-1, 256, 14, 14]         589,824
      BatchNorm2d-445          [-1, 256, 14, 14]             512
             ReLU-446          [-1, 256, 14, 14]               0
           Conv2d-447         [-1, 1024, 14, 14]         262,144
      BatchNorm2d-448         [-1, 1024, 14, 14]           2,048
             ReLU-449         [-1, 1024, 14, 14]               0
       Bottleneck-450         [-1, 1024, 14, 14]               0
           Conv2d-451          [-1, 256, 14, 14]         262,144
      BatchNorm2d-452          [-1, 256, 14, 14]             512
             ReLU-453          [-1, 256, 14, 14]               0
           Conv2d-454          [-1, 256, 14, 14]         589,824
      BatchNorm2d-455          [-1, 256, 14, 14]             512
             ReLU-456          [-1, 256, 14, 14]               0
           Conv2d-457         [-1, 1024, 14, 14]         262,144
      BatchNorm2d-458         [-1, 1024, 14, 14]           2,048
             ReLU-459         [-1, 1024, 14, 14]               0
       Bottleneck-460         [-1, 1024, 14, 14]               0
           Conv2d-461          [-1, 256, 14, 14]         262,144
      BatchNorm2d-462          [-1, 256, 14, 14]             512
             ReLU-463          [-1, 256, 14, 14]               0
           Conv2d-464          [-1, 256, 14, 14]         589,824
      BatchNorm2d-465          [-1, 256, 14, 14]             512
             ReLU-466          [-1, 256, 14, 14]               0
           Conv2d-467         [-1, 1024, 14, 14]         262,144
      BatchNorm2d-468         [-1, 1024, 14, 14]           2,048
             ReLU-469         [-1, 1024, 14, 14]               0
       Bottleneck-470         [-1, 1024, 14, 14]               0
           Conv2d-471          [-1, 256, 14, 14]         262,144
      BatchNorm2d-472          [-1, 256, 14, 14]             512
             ReLU-473          [-1, 256, 14, 14]               0
           Conv2d-474          [-1, 256, 14, 14]         589,824
      BatchNorm2d-475          [-1, 256, 14, 14]             512
             ReLU-476          [-1, 256, 14, 14]               0
           Conv2d-477         [-1, 1024, 14, 14]         262,144
      BatchNorm2d-478         [-1, 1024, 14, 14]           2,048
             ReLU-479         [-1, 1024, 14, 14]               0
       Bottleneck-480         [-1, 1024, 14, 14]               0
           Conv2d-481          [-1, 512, 14, 14]         524,288
      BatchNorm2d-482          [-1, 512, 14, 14]           1,024
             ReLU-483          [-1, 512, 14, 14]               0
           Conv2d-484            [-1, 512, 7, 7]       2,359,296
      BatchNorm2d-485            [-1, 512, 7, 7]           1,024
             ReLU-486            [-1, 512, 7, 7]               0
           Conv2d-487           [-1, 2048, 7, 7]       1,048,576
      BatchNorm2d-488           [-1, 2048, 7, 7]           4,096
           Conv2d-489           [-1, 2048, 7, 7]       2,097,152
      BatchNorm2d-490           [-1, 2048, 7, 7]           4,096
             ReLU-491           [-1, 2048, 7, 7]               0
       Bottleneck-492           [-1, 2048, 7, 7]               0
           Conv2d-493            [-1, 512, 7, 7]       1,048,576
      BatchNorm2d-494            [-1, 512, 7, 7]           1,024
             ReLU-495            [-1, 512, 7, 7]               0
           Conv2d-496            [-1, 512, 7, 7]       2,359,296
      BatchNorm2d-497            [-1, 512, 7, 7]           1,024
             ReLU-498            [-1, 512, 7, 7]               0
           Conv2d-499           [-1, 2048, 7, 7]       1,048,576
      BatchNorm2d-500           [-1, 2048, 7, 7]           4,096
             ReLU-501           [-1, 2048, 7, 7]               0
       Bottleneck-502           [-1, 2048, 7, 7]               0
           Conv2d-503            [-1, 512, 7, 7]       1,048,576
      BatchNorm2d-504            [-1, 512, 7, 7]           1,024
             ReLU-505            [-1, 512, 7, 7]               0
           Conv2d-506            [-1, 512, 7, 7]       2,359,296
      BatchNorm2d-507            [-1, 512, 7, 7]           1,024
             ReLU-508            [-1, 512, 7, 7]               0
           Conv2d-509           [-1, 2048, 7, 7]       1,048,576
      BatchNorm2d-510           [-1, 2048, 7, 7]           4,096
             ReLU-511           [-1, 2048, 7, 7]               0
       Bottleneck-512           [-1, 2048, 7, 7]               0
AdaptiveAvgPool2d-513           [-1, 2048, 1, 1]               0
           Linear-514                  [-1, 1]           2,049
           ResNet-515                  [-1, 1]               0
================================================================
Total params: 58,145,857
Trainable params: 2,049
Non-trainable params: 58,143,808
----------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 606.58
```

```
Params size (MB): 221.81
Estimated Total Size (MB): 828.97
----------------------------------------------------------------

================================================================
Model: ResNet152 (unfrozen features)
================================================================
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 64, 112, 112]           9,408
       BatchNorm2d-2         [-1, 64, 112, 112]             128
              ReLU-3         [-1, 64, 112, 112]               0
         MaxPool2d-4           [-1, 64, 56, 56]               0
            Conv2d-5           [-1, 64, 56, 56]           4,096
       BatchNorm2d-6           [-1, 64, 56, 56]             128
              ReLU-7           [-1, 64, 56, 56]               0
            Conv2d-8           [-1, 64, 56, 56]          36,864
       BatchNorm2d-9           [-1, 64, 56, 56]             128
             ReLU-10           [-1, 64, 56, 56]               0
           Conv2d-11          [-1, 256, 56, 56]          16,384
      BatchNorm2d-12          [-1, 256, 56, 56]             512
           Conv2d-13          [-1, 256, 56, 56]          16,384
      BatchNorm2d-14          [-1, 256, 56, 56]             512
             ReLU-15          [-1, 256, 56, 56]               0
       Bottleneck-16          [-1, 256, 56, 56]               0
           Conv2d-17           [-1, 64, 56, 56]          16,384
      BatchNorm2d-18           [-1, 64, 56, 56]             128
             ReLU-19           [-1, 64, 56, 56]               0
           Conv2d-20           [-1, 64, 56, 56]          36,864
      BatchNorm2d-21           [-1, 64, 56, 56]             128
             ReLU-22           [-1, 64, 56, 56]               0
           Conv2d-23          [-1, 256, 56, 56]          16,384
      BatchNorm2d-24          [-1, 256, 56, 56]             512
             ReLU-25          [-1, 256, 56, 56]               0
       Bottleneck-26          [-1, 256, 56, 56]               0
           Conv2d-27           [-1, 64, 56, 56]          16,384
      BatchNorm2d-28           [-1, 64, 56, 56]             128
             ReLU-29           [-1, 64, 56, 56]               0
           Conv2d-30           [-1, 64, 56, 56]          36,864
      BatchNorm2d-31           [-1, 64, 56, 56]             128
             ReLU-32           [-1, 64, 56, 56]               0
           Conv2d-33          [-1, 256, 56, 56]          16,384
      BatchNorm2d-34          [-1, 256, 56, 56]             512
             ReLU-35          [-1, 256, 56, 56]               0
       Bottleneck-36          [-1, 256, 56, 56]               0
           Conv2d-37          [-1, 128, 56, 56]          32,768
      BatchNorm2d-38          [-1, 128, 56, 56]             256
             ReLU-39          [-1, 128, 56, 56]               0
           Conv2d-40          [-1, 128, 28, 28]         147,456
      BatchNorm2d-41          [-1, 128, 28, 28]             256
             ReLU-42          [-1, 128, 28, 28]               0
           Conv2d-43          [-1, 512, 28, 28]          65,536
      BatchNorm2d-44          [-1, 512, 28, 28]           1,024
           Conv2d-45          [-1, 512, 28, 28]         131,072
      BatchNorm2d-46          [-1, 512, 28, 28]           1,024
             ReLU-47          [-1, 512, 28, 28]               0
       Bottleneck-48          [-1, 512, 28, 28]               0
           Conv2d-49          [-1, 128, 28, 28]          65,536
      BatchNorm2d-50          [-1, 128, 28, 28]             256
             ReLU-51          [-1, 128, 28, 28]               0
           Conv2d-52          [-1, 128, 28, 28]         147,456
      BatchNorm2d-53          [-1, 128, 28, 28]             256
             ReLU-54          [-1, 128, 28, 28]               0
           Conv2d-55          [-1, 512, 28, 28]          65,536
      BatchNorm2d-56          [-1, 512, 28, 28]           1,024
             ReLU-57          [-1, 512, 28, 28]               0
       Bottleneck-58          [-1, 512, 28, 28]               0
           Conv2d-59          [-1, 128, 28, 28]          65,536
      BatchNorm2d-60          [-1, 128, 28, 28]             256
             ReLU-61          [-1, 128, 28, 28]               0
           Conv2d-62          [-1, 128, 28, 28]         147,456
      BatchNorm2d-63          [-1, 128, 28, 28]             256
             ReLU-64          [-1, 128, 28, 28]               0
           Conv2d-65          [-1, 512, 28, 28]          65,536
      BatchNorm2d-66          [-1, 512, 28, 28]           1,024
             ReLU-67          [-1, 512, 28, 28]               0
       Bottleneck-68          [-1, 512, 28, 28]               0
           Conv2d-69          [-1, 128, 28, 28]          65,536
      BatchNorm2d-70          [-1, 128, 28, 28]             256
             ReLU-71          [-1, 128, 28, 28]               0
           Conv2d-72          [-1, 128, 28, 28]         147,456
      BatchNorm2d-73          [-1, 128, 28, 28]             256
             ReLU-74          [-1, 128, 28, 28]               0
           Conv2d-75          [-1, 512, 28, 28]          65,536
      BatchNorm2d-76          [-1, 512, 28, 28]           1,024
             ReLU-77          [-1, 512, 28, 28]               0
       Bottleneck-78          [-1, 512, 28, 28]               0
```

```
      Conv2d-79         [-1, 128, 28, 28]          65,536
 BatchNorm2d-80         [-1, 128, 28, 28]             256
        ReLU-81         [-1, 128, 28, 28]               0
      Conv2d-82         [-1, 128, 28, 28]         147,456
 BatchNorm2d-83         [-1, 128, 28, 28]             256
        ReLU-84         [-1, 128, 28, 28]               0
      Conv2d-85         [-1, 512, 28, 28]          65,536
 BatchNorm2d-86         [-1, 512, 28, 28]           1,024
        ReLU-87         [-1, 512, 28, 28]               0
  Bottleneck-88         [-1, 512, 28, 28]               0
      Conv2d-89         [-1, 128, 28, 28]          65,536
 BatchNorm2d-90         [-1, 128, 28, 28]             256
        ReLU-91         [-1, 128, 28, 28]               0
      Conv2d-92         [-1, 128, 28, 28]         147,456
 BatchNorm2d-93         [-1, 128, 28, 28]             256
        ReLU-94         [-1, 128, 28, 28]               0
      Conv2d-95         [-1, 512, 28, 28]          65,536
 BatchNorm2d-96         [-1, 512, 28, 28]           1,024
        ReLU-97         [-1, 512, 28, 28]               0
  Bottleneck-98         [-1, 512, 28, 28]               0
      Conv2d-99         [-1, 128, 28, 28]          65,536
BatchNorm2d-100         [-1, 128, 28, 28]             256
       ReLU-101         [-1, 128, 28, 28]               0
     Conv2d-102         [-1, 128, 28, 28]         147,456
BatchNorm2d-103         [-1, 128, 28, 28]             256
       ReLU-104         [-1, 128, 28, 28]               0
     Conv2d-105         [-1, 512, 28, 28]          65,536
BatchNorm2d-106         [-1, 512, 28, 28]           1,024
       ReLU-107         [-1, 512, 28, 28]               0
 Bottleneck-108         [-1, 512, 28, 28]               0
     Conv2d-109         [-1, 128, 28, 28]          65,536
BatchNorm2d-110         [-1, 128, 28, 28]             256
       ReLU-111         [-1, 128, 28, 28]               0
     Conv2d-112         [-1, 128, 28, 28]         147,456
BatchNorm2d-113         [-1, 128, 28, 28]             256
       ReLU-114         [-1, 128, 28, 28]               0
     Conv2d-115         [-1, 512, 28, 28]          65,536
BatchNorm2d-116         [-1, 512, 28, 28]           1,024
       ReLU-117         [-1, 512, 28, 28]               0
 Bottleneck-118         [-1, 512, 28, 28]               0
     Conv2d-119         [-1, 256, 28, 28]         131,072
BatchNorm2d-120         [-1, 256, 28, 28]             512
       ReLU-121         [-1, 256, 28, 28]               0
     Conv2d-122         [-1, 256, 14, 14]         589,824
BatchNorm2d-123         [-1, 256, 14, 14]             512
       ReLU-124         [-1, 256, 14, 14]               0
     Conv2d-125        [-1, 1024, 14, 14]         262,144
BatchNorm2d-126        [-1, 1024, 14, 14]           2,048
     Conv2d-127        [-1, 1024, 14, 14]         524,288
BatchNorm2d-128        [-1, 1024, 14, 14]           2,048
       ReLU-129        [-1, 1024, 14, 14]               0
 Bottleneck-130        [-1, 1024, 14, 14]               0
     Conv2d-131         [-1, 256, 14, 14]         262,144
BatchNorm2d-132         [-1, 256, 14, 14]             512
       ReLU-133         [-1, 256, 14, 14]               0
     Conv2d-134         [-1, 256, 14, 14]         589,824
BatchNorm2d-135         [-1, 256, 14, 14]             512
       ReLU-136         [-1, 256, 14, 14]               0
     Conv2d-137        [-1, 1024, 14, 14]         262,144
BatchNorm2d-138        [-1, 1024, 14, 14]           2,048
       ReLU-139        [-1, 1024, 14, 14]               0
 Bottleneck-140        [-1, 1024, 14, 14]               0
     Conv2d-141         [-1, 256, 14, 14]         262,144
BatchNorm2d-142         [-1, 256, 14, 14]             512
       ReLU-143         [-1, 256, 14, 14]               0
     Conv2d-144         [-1, 256, 14, 14]         589,824
BatchNorm2d-145         [-1, 256, 14, 14]             512
       ReLU-146         [-1, 256, 14, 14]               0
     Conv2d-147        [-1, 1024, 14, 14]         262,144
BatchNorm2d-148        [-1, 1024, 14, 14]           2,048
       ReLU-149        [-1, 1024, 14, 14]               0
 Bottleneck-150        [-1, 1024, 14, 14]               0
     Conv2d-151         [-1, 256, 14, 14]         262,144
BatchNorm2d-152         [-1, 256, 14, 14]             512
       ReLU-153         [-1, 256, 14, 14]               0
     Conv2d-154         [-1, 256, 14, 14]         589,824
BatchNorm2d-155         [-1, 256, 14, 14]             512
       ReLU-156         [-1, 256, 14, 14]               0
     Conv2d-157        [-1, 1024, 14, 14]         262,144
BatchNorm2d-158        [-1, 1024, 14, 14]           2,048
       ReLU-159        [-1, 1024, 14, 14]               0
 Bottleneck-160        [-1, 1024, 14, 14]               0
     Conv2d-161         [-1, 256, 14, 14]         262,144
BatchNorm2d-162         [-1, 256, 14, 14]             512
       ReLU-163         [-1, 256, 14, 14]               0
     Conv2d-164         [-1, 256, 14, 14]         589,824
BatchNorm2d-165         [-1, 256, 14, 14]             512
       ReLU-166         [-1, 256, 14, 14]               0
```

```
     Conv2d-167        [-1, 1024, 14, 14]         262,144
BatchNorm2d-168        [-1, 1024, 14, 14]           2,048
       ReLU-169        [-1, 1024, 14, 14]               0
 Bottleneck-170        [-1, 1024, 14, 14]               0
     Conv2d-171         [-1, 256, 14, 14]         262,144
BatchNorm2d-172         [-1, 256, 14, 14]             512
       ReLU-173         [-1, 256, 14, 14]               0
     Conv2d-174         [-1, 256, 14, 14]         589,824
BatchNorm2d-175         [-1, 256, 14, 14]             512
       ReLU-176         [-1, 256, 14, 14]               0
     Conv2d-177        [-1, 1024, 14, 14]         262,144
BatchNorm2d-178        [-1, 1024, 14, 14]           2,048
       ReLU-179        [-1, 1024, 14, 14]               0
 Bottleneck-180        [-1, 1024, 14, 14]               0
     Conv2d-181         [-1, 256, 14, 14]         262,144
BatchNorm2d-182         [-1, 256, 14, 14]             512
       ReLU-183         [-1, 256, 14, 14]               0
     Conv2d-184         [-1, 256, 14, 14]         589,824
BatchNorm2d-185         [-1, 256, 14, 14]             512
       ReLU-186         [-1, 256, 14, 14]               0
     Conv2d-187        [-1, 1024, 14, 14]         262,144
BatchNorm2d-188        [-1, 1024, 14, 14]           2,048
       ReLU-189        [-1, 1024, 14, 14]               0
 Bottleneck-190        [-1, 1024, 14, 14]               0
     Conv2d-191         [-1, 256, 14, 14]         262,144
BatchNorm2d-192         [-1, 256, 14, 14]             512
       ReLU-193         [-1, 256, 14, 14]               0
     Conv2d-194         [-1, 256, 14, 14]         589,824
BatchNorm2d-195         [-1, 256, 14, 14]             512
       ReLU-196         [-1, 256, 14, 14]               0
     Conv2d-197        [-1, 1024, 14, 14]         262,144
BatchNorm2d-198        [-1, 1024, 14, 14]           2,048
       ReLU-199        [-1, 1024, 14, 14]               0
 Bottleneck-200        [-1, 1024, 14, 14]               0
     Conv2d-201         [-1, 256, 14, 14]         262,144
BatchNorm2d-202         [-1, 256, 14, 14]             512
       ReLU-203         [-1, 256, 14, 14]               0
     Conv2d-204         [-1, 256, 14, 14]         589,824
BatchNorm2d-205         [-1, 256, 14, 14]             512
       ReLU-206         [-1, 256, 14, 14]               0
     Conv2d-207        [-1, 1024, 14, 14]         262,144
BatchNorm2d-208        [-1, 1024, 14, 14]           2,048
       ReLU-209        [-1, 1024, 14, 14]               0
 Bottleneck-210        [-1, 1024, 14, 14]               0
     Conv2d-211         [-1, 256, 14, 14]         262,144
BatchNorm2d-212         [-1, 256, 14, 14]             512
       ReLU-213         [-1, 256, 14, 14]               0
     Conv2d-214         [-1, 256, 14, 14]         589,824
BatchNorm2d-215         [-1, 256, 14, 14]             512
       ReLU-216         [-1, 256, 14, 14]               0
     Conv2d-217        [-1, 1024, 14, 14]         262,144
BatchNorm2d-218        [-1, 1024, 14, 14]           2,048
       ReLU-219        [-1, 1024, 14, 14]               0
 Bottleneck-220        [-1, 1024, 14, 14]               0
     Conv2d-221         [-1, 256, 14, 14]         262,144
BatchNorm2d-222         [-1, 256, 14, 14]             512
       ReLU-223         [-1, 256, 14, 14]               0
     Conv2d-224         [-1, 256, 14, 14]         589,824
BatchNorm2d-225         [-1, 256, 14, 14]             512
       ReLU-226         [-1, 256, 14, 14]               0
     Conv2d-227        [-1, 1024, 14, 14]         262,144
BatchNorm2d-228        [-1, 1024, 14, 14]           2,048
       ReLU-229        [-1, 1024, 14, 14]               0
 Bottleneck-230        [-1, 1024, 14, 14]               0
     Conv2d-231         [-1, 256, 14, 14]         262,144
BatchNorm2d-232         [-1, 256, 14, 14]             512
       ReLU-233         [-1, 256, 14, 14]               0
     Conv2d-234         [-1, 256, 14, 14]         589,824
BatchNorm2d-235         [-1, 256, 14, 14]             512
       ReLU-236         [-1, 256, 14, 14]               0
     Conv2d-237        [-1, 1024, 14, 14]         262,144
BatchNorm2d-238        [-1, 1024, 14, 14]           2,048
       ReLU-239        [-1, 1024, 14, 14]               0
 Bottleneck-240        [-1, 1024, 14, 14]               0
     Conv2d-241         [-1, 256, 14, 14]         262,144
BatchNorm2d-242         [-1, 256, 14, 14]             512
       ReLU-243         [-1, 256, 14, 14]               0
     Conv2d-244         [-1, 256, 14, 14]         589,824
BatchNorm2d-245         [-1, 256, 14, 14]             512
       ReLU-246         [-1, 256, 14, 14]               0
     Conv2d-247        [-1, 1024, 14, 14]         262,144
BatchNorm2d-248        [-1, 1024, 14, 14]           2,048
       ReLU-249        [-1, 1024, 14, 14]               0
 Bottleneck-250        [-1, 1024, 14, 14]               0
     Conv2d-251         [-1, 256, 14, 14]         262,144
BatchNorm2d-252         [-1, 256, 14, 14]             512
       ReLU-253         [-1, 256, 14, 14]               0
     Conv2d-254         [-1, 256, 14, 14]         589,824
```

```
BatchNorm2d-255         [-1, 256, 14, 14]             512
       ReLU-256         [-1, 256, 14, 14]               0
     Conv2d-257        [-1, 1024, 14, 14]         262,144
BatchNorm2d-258        [-1, 1024, 14, 14]           2,048
       ReLU-259        [-1, 1024, 14, 14]               0
 Bottleneck-260        [-1, 1024, 14, 14]               0
     Conv2d-261         [-1, 256, 14, 14]         262,144
BatchNorm2d-262         [-1, 256, 14, 14]             512
       ReLU-263         [-1, 256, 14, 14]               0
     Conv2d-264         [-1, 256, 14, 14]         589,824
BatchNorm2d-265         [-1, 256, 14, 14]             512
       ReLU-266         [-1, 256, 14, 14]               0
     Conv2d-267        [-1, 1024, 14, 14]         262,144
BatchNorm2d-268        [-1, 1024, 14, 14]           2,048
       ReLU-269        [-1, 1024, 14, 14]               0
 Bottleneck-270        [-1, 1024, 14, 14]               0
     Conv2d-271         [-1, 256, 14, 14]         262,144
BatchNorm2d-272         [-1, 256, 14, 14]             512
       ReLU-273         [-1, 256, 14, 14]               0
     Conv2d-274         [-1, 256, 14, 14]         589,824
BatchNorm2d-275         [-1, 256, 14, 14]             512
       ReLU-276         [-1, 256, 14, 14]               0
     Conv2d-277        [-1, 1024, 14, 14]         262,144
BatchNorm2d-278        [-1, 1024, 14, 14]           2,048
       ReLU-279        [-1, 1024, 14, 14]               0
 Bottleneck-280        [-1, 1024, 14, 14]               0
     Conv2d-281         [-1, 256, 14, 14]         262,144
BatchNorm2d-282         [-1, 256, 14, 14]             512
       ReLU-283         [-1, 256, 14, 14]               0
     Conv2d-284         [-1, 256, 14, 14]         589,824
BatchNorm2d-285         [-1, 256, 14, 14]             512
       ReLU-286         [-1, 256, 14, 14]               0
     Conv2d-287        [-1, 1024, 14, 14]         262,144
BatchNorm2d-288        [-1, 1024, 14, 14]           2,048
       ReLU-289        [-1, 1024, 14, 14]               0
 Bottleneck-290        [-1, 1024, 14, 14]               0
     Conv2d-291         [-1, 256, 14, 14]         262,144
BatchNorm2d-292         [-1, 256, 14, 14]             512
       ReLU-293         [-1, 256, 14, 14]               0
     Conv2d-294         [-1, 256, 14, 14]         589,824
BatchNorm2d-295         [-1, 256, 14, 14]             512
       ReLU-296         [-1, 256, 14, 14]               0
     Conv2d-297        [-1, 1024, 14, 14]         262,144
BatchNorm2d-298        [-1, 1024, 14, 14]           2,048
       ReLU-299        [-1, 1024, 14, 14]               0
 Bottleneck-300        [-1, 1024, 14, 14]               0
     Conv2d-301         [-1, 256, 14, 14]         262,144
BatchNorm2d-302         [-1, 256, 14, 14]             512
       ReLU-303         [-1, 256, 14, 14]               0
     Conv2d-304         [-1, 256, 14, 14]         589,824
BatchNorm2d-305         [-1, 256, 14, 14]             512
       ReLU-306         [-1, 256, 14, 14]               0
     Conv2d-307        [-1, 1024, 14, 14]         262,144
BatchNorm2d-308        [-1, 1024, 14, 14]           2,048
       ReLU-309        [-1, 1024, 14, 14]               0
 Bottleneck-310        [-1, 1024, 14, 14]               0
     Conv2d-311         [-1, 256, 14, 14]         262,144
BatchNorm2d-312         [-1, 256, 14, 14]             512
       ReLU-313         [-1, 256, 14, 14]               0
     Conv2d-314         [-1, 256, 14, 14]         589,824
BatchNorm2d-315         [-1, 256, 14, 14]             512
       ReLU-316         [-1, 256, 14, 14]               0
     Conv2d-317        [-1, 1024, 14, 14]         262,144
BatchNorm2d-318        [-1, 1024, 14, 14]           2,048
       ReLU-319        [-1, 1024, 14, 14]               0
 Bottleneck-320        [-1, 1024, 14, 14]               0
     Conv2d-321         [-1, 256, 14, 14]         262,144
BatchNorm2d-322         [-1, 256, 14, 14]             512
       ReLU-323         [-1, 256, 14, 14]               0
     Conv2d-324         [-1, 256, 14, 14]         589,824
BatchNorm2d-325         [-1, 256, 14, 14]             512
       ReLU-326         [-1, 256, 14, 14]               0
     Conv2d-327        [-1, 1024, 14, 14]         262,144
BatchNorm2d-328        [-1, 1024, 14, 14]           2,048
       ReLU-329        [-1, 1024, 14, 14]               0
 Bottleneck-330        [-1, 1024, 14, 14]               0
     Conv2d-331         [-1, 256, 14, 14]         262,144
BatchNorm2d-332         [-1, 256, 14, 14]             512
       ReLU-333         [-1, 256, 14, 14]               0
     Conv2d-334         [-1, 256, 14, 14]         589,824
BatchNorm2d-335         [-1, 256, 14, 14]             512
       ReLU-336         [-1, 256, 14, 14]               0
     Conv2d-337        [-1, 1024, 14, 14]         262,144
BatchNorm2d-338        [-1, 1024, 14, 14]           2,048
       ReLU-339        [-1, 1024, 14, 14]               0
 Bottleneck-340        [-1, 1024, 14, 14]               0
     Conv2d-341         [-1, 256, 14, 14]         262,144
BatchNorm2d-342         [-1, 256, 14, 14]             512
```

```
       ReLU-343         [-1, 256, 14, 14]               0
     Conv2d-344         [-1, 256, 14, 14]         589,824
BatchNorm2d-345         [-1, 256, 14, 14]             512
       ReLU-346         [-1, 256, 14, 14]               0
     Conv2d-347        [-1, 1024, 14, 14]         262,144
BatchNorm2d-348        [-1, 1024, 14, 14]           2,048
       ReLU-349        [-1, 1024, 14, 14]               0
 Bottleneck-350        [-1, 1024, 14, 14]               0
     Conv2d-351         [-1, 256, 14, 14]         262,144
BatchNorm2d-352         [-1, 256, 14, 14]             512
       ReLU-353         [-1, 256, 14, 14]               0
     Conv2d-354         [-1, 256, 14, 14]         589,824
BatchNorm2d-355         [-1, 256, 14, 14]             512
       ReLU-356         [-1, 256, 14, 14]               0
     Conv2d-357        [-1, 1024, 14, 14]         262,144
BatchNorm2d-358        [-1, 1024, 14, 14]           2,048
       ReLU-359        [-1, 1024, 14, 14]               0
 Bottleneck-360        [-1, 1024, 14, 14]               0
     Conv2d-361         [-1, 256, 14, 14]         262,144
BatchNorm2d-362         [-1, 256, 14, 14]             512
       ReLU-363         [-1, 256, 14, 14]               0
     Conv2d-364         [-1, 256, 14, 14]         589,824
BatchNorm2d-365         [-1, 256, 14, 14]             512
       ReLU-366         [-1, 256, 14, 14]               0
     Conv2d-367        [-1, 1024, 14, 14]         262,144
BatchNorm2d-368        [-1, 1024, 14, 14]           2,048
       ReLU-369        [-1, 1024, 14, 14]               0
 Bottleneck-370        [-1, 1024, 14, 14]               0
     Conv2d-371         [-1, 256, 14, 14]         262,144
BatchNorm2d-372         [-1, 256, 14, 14]             512
       ReLU-373         [-1, 256, 14, 14]               0
     Conv2d-374         [-1, 256, 14, 14]         589,824
BatchNorm2d-375         [-1, 256, 14, 14]             512
       ReLU-376         [-1, 256, 14, 14]               0
     Conv2d-377        [-1, 1024, 14, 14]         262,144
BatchNorm2d-378        [-1, 1024, 14, 14]           2,048
       ReLU-379        [-1, 1024, 14, 14]               0
 Bottleneck-380        [-1, 1024, 14, 14]               0
     Conv2d-381         [-1, 256, 14, 14]         262,144
BatchNorm2d-382         [-1, 256, 14, 14]             512
       ReLU-383         [-1, 256, 14, 14]               0
     Conv2d-384         [-1, 256, 14, 14]         589,824
BatchNorm2d-385         [-1, 256, 14, 14]             512
       ReLU-386         [-1, 256, 14, 14]               0
     Conv2d-387        [-1, 1024, 14, 14]         262,144
BatchNorm2d-388        [-1, 1024, 14, 14]           2,048
       ReLU-389        [-1, 1024, 14, 14]               0
 Bottleneck-390        [-1, 1024, 14, 14]               0
     Conv2d-391         [-1, 256, 14, 14]         262,144
BatchNorm2d-392         [-1, 256, 14, 14]             512
       ReLU-393         [-1, 256, 14, 14]               0
     Conv2d-394         [-1, 256, 14, 14]         589,824
BatchNorm2d-395         [-1, 256, 14, 14]             512
       ReLU-396         [-1, 256, 14, 14]               0
     Conv2d-397        [-1, 1024, 14, 14]         262,144
BatchNorm2d-398        [-1, 1024, 14, 14]           2,048
       ReLU-399        [-1, 1024, 14, 14]               0
 Bottleneck-400        [-1, 1024, 14, 14]               0
     Conv2d-401         [-1, 256, 14, 14]         262,144
BatchNorm2d-402         [-1, 256, 14, 14]             512
       ReLU-403         [-1, 256, 14, 14]               0
     Conv2d-404         [-1, 256, 14, 14]         589,824
BatchNorm2d-405         [-1, 256, 14, 14]             512
       ReLU-406         [-1, 256, 14, 14]               0
     Conv2d-407        [-1, 1024, 14, 14]         262,144
BatchNorm2d-408        [-1, 1024, 14, 14]           2,048
       ReLU-409        [-1, 1024, 14, 14]               0
 Bottleneck-410        [-1, 1024, 14, 14]               0
     Conv2d-411         [-1, 256, 14, 14]         262,144
BatchNorm2d-412         [-1, 256, 14, 14]             512
       ReLU-413         [-1, 256, 14, 14]               0
     Conv2d-414         [-1, 256, 14, 14]         589,824
BatchNorm2d-415         [-1, 256, 14, 14]             512
       ReLU-416         [-1, 256, 14, 14]               0
     Conv2d-417        [-1, 1024, 14, 14]         262,144
BatchNorm2d-418        [-1, 1024, 14, 14]           2,048
       ReLU-419        [-1, 1024, 14, 14]               0
 Bottleneck-420        [-1, 1024, 14, 14]               0
     Conv2d-421         [-1, 256, 14, 14]         262,144
BatchNorm2d-422         [-1, 256, 14, 14]             512
       ReLU-423         [-1, 256, 14, 14]               0
     Conv2d-424         [-1, 256, 14, 14]         589,824
BatchNorm2d-425         [-1, 256, 14, 14]             512
       ReLU-426         [-1, 256, 14, 14]               0
     Conv2d-427        [-1, 1024, 14, 14]         262,144
BatchNorm2d-428        [-1, 1024, 14, 14]           2,048
       ReLU-429        [-1, 1024, 14, 14]               0
 Bottleneck-430        [-1, 1024, 14, 14]               0
```

```
          Conv2d-431          [-1, 256, 14, 14]         262,144
     BatchNorm2d-432          [-1, 256, 14, 14]             512
            ReLU-433          [-1, 256, 14, 14]               0
          Conv2d-434          [-1, 256, 14, 14]         589,824
     BatchNorm2d-435          [-1, 256, 14, 14]             512
            ReLU-436          [-1, 256, 14, 14]               0
          Conv2d-437         [-1, 1024, 14, 14]         262,144
     BatchNorm2d-438         [-1, 1024, 14, 14]           2,048
            ReLU-439         [-1, 1024, 14, 14]               0
      Bottleneck-440         [-1, 1024, 14, 14]               0
          Conv2d-441          [-1, 256, 14, 14]         262,144
     BatchNorm2d-442          [-1, 256, 14, 14]             512
            ReLU-443          [-1, 256, 14, 14]               0
          Conv2d-444          [-1, 256, 14, 14]         589,824
     BatchNorm2d-445          [-1, 256, 14, 14]             512
            ReLU-446          [-1, 256, 14, 14]               0
          Conv2d-447         [-1, 1024, 14, 14]         262,144
     BatchNorm2d-448         [-1, 1024, 14, 14]           2,048
            ReLU-449         [-1, 1024, 14, 14]               0
      Bottleneck-450         [-1, 1024, 14, 14]               0
          Conv2d-451          [-1, 256, 14, 14]         262,144
     BatchNorm2d-452          [-1, 256, 14, 14]             512
            ReLU-453          [-1, 256, 14, 14]               0
          Conv2d-454          [-1, 256, 14, 14]         589,824
     BatchNorm2d-455          [-1, 256, 14, 14]             512
            ReLU-456          [-1, 256, 14, 14]               0
          Conv2d-457         [-1, 1024, 14, 14]         262,144
     BatchNorm2d-458         [-1, 1024, 14, 14]           2,048
            ReLU-459         [-1, 1024, 14, 14]               0
      Bottleneck-460         [-1, 1024, 14, 14]               0
          Conv2d-461          [-1, 256, 14, 14]         262,144
     BatchNorm2d-462          [-1, 256, 14, 14]             512
            ReLU-463          [-1, 256, 14, 14]               0
          Conv2d-464          [-1, 256, 14, 14]         589,824
     BatchNorm2d-465          [-1, 256, 14, 14]             512
            ReLU-466          [-1, 256, 14, 14]               0
          Conv2d-467         [-1, 1024, 14, 14]         262,144
     BatchNorm2d-468         [-1, 1024, 14, 14]           2,048
            ReLU-469         [-1, 1024, 14, 14]               0
      Bottleneck-470         [-1, 1024, 14, 14]               0
          Conv2d-471          [-1, 256, 14, 14]         262,144
     BatchNorm2d-472          [-1, 256, 14, 14]             512
            ReLU-473          [-1, 256, 14, 14]               0
          Conv2d-474          [-1, 256, 14, 14]         589,824
     BatchNorm2d-475          [-1, 256, 14, 14]             512
            ReLU-476          [-1, 256, 14, 14]               0
          Conv2d-477         [-1, 1024, 14, 14]         262,144
     BatchNorm2d-478         [-1, 1024, 14, 14]           2,048
            ReLU-479         [-1, 1024, 14, 14]               0
      Bottleneck-480         [-1, 1024, 14, 14]               0
          Conv2d-481          [-1, 512, 14, 14]         524,288
     BatchNorm2d-482          [-1, 512, 14, 14]           1,024
            ReLU-483          [-1, 512, 14, 14]               0
          Conv2d-484            [-1, 512, 7, 7]       2,359,296
     BatchNorm2d-485            [-1, 512, 7, 7]           1,024
            ReLU-486            [-1, 512, 7, 7]               0
          Conv2d-487           [-1, 2048, 7, 7]       1,048,576
     BatchNorm2d-488           [-1, 2048, 7, 7]           4,096
          Conv2d-489           [-1, 2048, 7, 7]       2,097,152
     BatchNorm2d-490           [-1, 2048, 7, 7]           4,096
            ReLU-491           [-1, 2048, 7, 7]               0
      Bottleneck-492           [-1, 2048, 7, 7]               0
          Conv2d-493            [-1, 512, 7, 7]       1,048,576
     BatchNorm2d-494            [-1, 512, 7, 7]           1,024
            ReLU-495            [-1, 512, 7, 7]               0
          Conv2d-496            [-1, 512, 7, 7]       2,359,296
     BatchNorm2d-497            [-1, 512, 7, 7]           1,024
            ReLU-498            [-1, 512, 7, 7]               0
          Conv2d-499           [-1, 2048, 7, 7]       1,048,576
     BatchNorm2d-500           [-1, 2048, 7, 7]           4,096
            ReLU-501           [-1, 2048, 7, 7]               0
      Bottleneck-502           [-1, 2048, 7, 7]               0
          Conv2d-503            [-1, 512, 7, 7]       1,048,576
     BatchNorm2d-504            [-1, 512, 7, 7]           1,024
            ReLU-505            [-1, 512, 7, 7]               0
          Conv2d-506            [-1, 512, 7, 7]       2,359,296
     BatchNorm2d-507            [-1, 512, 7, 7]           1,024
            ReLU-508            [-1, 512, 7, 7]               0
          Conv2d-509           [-1, 2048, 7, 7]       1,048,576
     BatchNorm2d-510           [-1, 2048, 7, 7]           4,096
            ReLU-511           [-1, 2048, 7, 7]               0
      Bottleneck-512           [-1, 2048, 7, 7]               0
AdaptiveAvgPool2d-513           [-1, 2048, 1, 1]               0
          Linear-514                  [-1, 1]           2,049
          ResNet-515                  [-1, 1]               0
================================================================
Total params: 58,145,857
Trainable params: 58,145,857
```

```
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 606.58
Params size (MB): 221.81
Estimated Total Size (MB): 828.97
----------------------------------------------------------------
```

In [12]:

```
print_model_summary(vgg16_frozen, "VGG16 (frozen features)")
print_model_summary(vgg16_unfrozen, "VGG16 (unfrozen features)")
```

```
================================================================
Model: VGG16 (frozen features)
================================================================
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 64, 224, 224]           1,792
       BatchNorm2d-2         [-1, 64, 224, 224]             128
              ReLU-3         [-1, 64, 224, 224]               0
            Conv2d-4         [-1, 64, 224, 224]          36,928
       BatchNorm2d-5         [-1, 64, 224, 224]             128
              ReLU-6         [-1, 64, 224, 224]               0
         MaxPool2d-7         [-1, 64, 112, 112]               0
            Conv2d-8        [-1, 128, 112, 112]          73,856
       BatchNorm2d-9        [-1, 128, 112, 112]             256
             ReLU-10        [-1, 128, 112, 112]               0
           Conv2d-11        [-1, 128, 112, 112]         147,584
      BatchNorm2d-12        [-1, 128, 112, 112]             256
             ReLU-13        [-1, 128, 112, 112]               0
        MaxPool2d-14          [-1, 128, 56, 56]               0
           Conv2d-15          [-1, 256, 56, 56]         295,168
      BatchNorm2d-16          [-1, 256, 56, 56]             512
             ReLU-17          [-1, 256, 56, 56]               0
           Conv2d-18          [-1, 256, 56, 56]         590,080
      BatchNorm2d-19          [-1, 256, 56, 56]             512
             ReLU-20          [-1, 256, 56, 56]               0
           Conv2d-21          [-1, 256, 56, 56]         590,080
      BatchNorm2d-22          [-1, 256, 56, 56]             512
             ReLU-23          [-1, 256, 56, 56]               0
        MaxPool2d-24          [-1, 256, 28, 28]               0
           Conv2d-25          [-1, 512, 28, 28]       1,180,160
      BatchNorm2d-26          [-1, 512, 28, 28]           1,024
             ReLU-27          [-1, 512, 28, 28]               0
           Conv2d-28          [-1, 512, 28, 28]       2,359,808
      BatchNorm2d-29          [-1, 512, 28, 28]           1,024
             ReLU-30          [-1, 512, 28, 28]               0
           Conv2d-31          [-1, 512, 28, 28]       2,359,808
      BatchNorm2d-32          [-1, 512, 28, 28]           1,024
             ReLU-33          [-1, 512, 28, 28]               0
        MaxPool2d-34          [-1, 512, 14, 14]               0
           Conv2d-35          [-1, 512, 14, 14]       2,359,808
      BatchNorm2d-36          [-1, 512, 14, 14]           1,024
             ReLU-37          [-1, 512, 14, 14]               0
           Conv2d-38          [-1, 512, 14, 14]       2,359,808
      BatchNorm2d-39          [-1, 512, 14, 14]           1,024
             ReLU-40          [-1, 512, 14, 14]               0
           Conv2d-41          [-1, 512, 14, 14]       2,359,808
      BatchNorm2d-42          [-1, 512, 14, 14]           1,024
             ReLU-43          [-1, 512, 14, 14]               0
        MaxPool2d-44            [-1, 512, 7, 7]               0
AdaptiveAvgPool2d-45            [-1, 512, 7, 7]               0
           Linear-46                 [-1, 4096]     102,764,544
             ReLU-47                 [-1, 4096]               0
          Dropout-48                 [-1, 4096]               0
           Linear-49                 [-1, 4096]      16,781,312
             ReLU-50                 [-1, 4096]               0
          Dropout-51                 [-1, 4096]               0
           Linear-52                    [-1, 1]           4,097
              VGG-53                    [-1, 1]               0
================================================================
Total params: 134,273,089
Trainable params: 119,549,953
Non-trainable params: 14,723,136
----------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 322.13
Params size (MB): 512.21
Estimated Total Size (MB): 834.92
----------------------------------------------------------------

================================================================
Model: VGG16 (unfrozen features)
================================================================
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 64, 224, 224]           1,792
       BatchNorm2d-2         [-1, 64, 224, 224]             128
              ReLU-3         [-1, 64, 224, 224]               0
            Conv2d-4         [-1, 64, 224, 224]          36,928
       BatchNorm2d-5         [-1, 64, 224, 224]             128
              ReLU-6         [-1, 64, 224, 224]               0
         MaxPool2d-7         [-1, 64, 112, 112]               0
            Conv2d-8        [-1, 128, 112, 112]          73,856
       BatchNorm2d-9        [-1, 128, 112, 112]             256
             ReLU-10        [-1, 128, 112, 112]               0
           Conv2d-11        [-1, 128, 112, 112]         147,584
      BatchNorm2d-12        [-1, 128, 112, 112]             256
```

```
              ReLU-13       [-1, 128, 112, 112]               0
         MaxPool2d-14         [-1, 128, 56, 56]               0
            Conv2d-15         [-1, 256, 56, 56]         295,168
       BatchNorm2d-16         [-1, 256, 56, 56]             512
              ReLU-17         [-1, 256, 56, 56]               0
            Conv2d-18         [-1, 256, 56, 56]         590,080
       BatchNorm2d-19         [-1, 256, 56, 56]             512
              ReLU-20         [-1, 256, 56, 56]               0
            Conv2d-21         [-1, 256, 56, 56]         590,080
       BatchNorm2d-22         [-1, 256, 56, 56]             512
              ReLU-23         [-1, 256, 56, 56]               0
         MaxPool2d-24         [-1, 256, 28, 28]               0
            Conv2d-25         [-1, 512, 28, 28]       1,180,160
       BatchNorm2d-26         [-1, 512, 28, 28]           1,024
              ReLU-27         [-1, 512, 28, 28]               0
            Conv2d-28         [-1, 512, 28, 28]       2,359,808
       BatchNorm2d-29         [-1, 512, 28, 28]           1,024
              ReLU-30         [-1, 512, 28, 28]               0
            Conv2d-31         [-1, 512, 28, 28]       2,359,808
       BatchNorm2d-32         [-1, 512, 28, 28]           1,024
              ReLU-33         [-1, 512, 28, 28]               0
         MaxPool2d-34         [-1, 512, 14, 14]               0
            Conv2d-35         [-1, 512, 14, 14]       2,359,808
       BatchNorm2d-36         [-1, 512, 14, 14]           1,024
              ReLU-37         [-1, 512, 14, 14]               0
            Conv2d-38         [-1, 512, 14, 14]       2,359,808
       BatchNorm2d-39         [-1, 512, 14, 14]           1,024
              ReLU-40         [-1, 512, 14, 14]               0
            Conv2d-41         [-1, 512, 14, 14]       2,359,808
       BatchNorm2d-42         [-1, 512, 14, 14]           1,024
              ReLU-43         [-1, 512, 14, 14]               0
         MaxPool2d-44           [-1, 512, 7, 7]               0
AdaptiveAvgPool2d-45           [-1, 512, 7, 7]               0
           Linear-46                 [-1, 4096]     102,764,544
             ReLU-47                 [-1, 4096]               0
          Dropout-48                 [-1, 4096]               0
           Linear-49                 [-1, 4096]      16,781,312
             ReLU-50                 [-1, 4096]               0
          Dropout-51                 [-1, 4096]               0
           Linear-52                    [-1, 1]           4,097
              VGG-53                    [-1, 1]               0
================================================================
Total params: 134,273,089
Trainable params: 134,273,089
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 322.13
Params size (MB): 512.21
Estimated Total Size (MB): 834.92
----------------------------------------------------------------
```

3.2. Model Training

In [13]:
```
!wqf7009-a3-train
```

```
usage: wqf7009-a3-train [-h] --model {simplecnn,vgg16,resnet152} [--freeze]
                        [--epochs EPOCHS] [--batch-size BATCH_SIZE]
                        [--data-dir DATA_DIR] [--lr LR]
                        [--num-workers NUM_WORKERS]
wqf7009-a3-train: error: the following arguments are required: --model
```

In [14]:
```
# Train the models with these commands:
# !wqf7009-a3-train --model=simplecnn --epochs=10 --batch-size=128 --data-dir=../data/chest_xray --lr=1e-4 --num-workers=4
# !wqf7009-a3-train --model=vgg16 --epochs=10 --batch-size=128 --data-dir=../data/chest_xray --lr=1e-4 --num-workers=4
# !wqf7009-a3-train --model=resnet152 --epochs=10 --batch-size=128 --data-dir=../data/chest_xray --lr=1e-4 --num-workers=4
# !wqf7009-a3-train --model=vgg16 --freeze --epochs=10 --batch-size=128 --data-dir=../data/chest_xray --lr=1e-4 --num-workers=4
# !wqf7009-a3-train --model=resnet152 --freeze --epochs=10 --batch-size=128 --data-dir=../data/chest_xray --lr=1e-4 --num-workers=4
```

I've trained the models on A100 GPUs somewhere else, we can load the trained models directly for evaluation in Task 3.

3.3. Model Evaluation

In [15]:
```
!wqf7009-a3-eval
```

```
usage: wqf7009-a3-eval [-h] --model {simplecnn,vgg16,resnet152} --checkpoint
                       CHECKPOINT [--data-dir DATA_DIR]
                       [--batch-size BATCH_SIZE] [--split {train,val,test}]
wqf7009-a3-eval: error: the following arguments are required: --model, --checkpoint
```

SimpleCNN

In [16]:
```
!wqf7009-a3-eval --model=simplecnn \
    --checkpoint=../models/simplecnn_baseline_10ep_1767264997_best.pth \
    --data-dir=../data/chest_xray \
```

```
    --batch-size=32 \
    --split=test
```

```
Using device: cuda
⁘ Loading test dataset...

Loading weights from: simplecnn_baseline_10ep_1767264997_best.pth
Running inference...
```

```
╭──────────────────────────────────────────────────────────────────────────────╮
│ Evaluation Config: Model=simplecnn | Split=test                              │
╰──────────────────────────────────────────────────────────────────────────────╯
```

Detailed Evaluation Metrics

Metric	Value
Accuracy	0.8093
Precision	0.7841
Recall (Sensitivity)	0.9590
F1 Score	0.8627

Confusion Matrix

Actual \ Predicted	Pred NORMAL	Pred PNEUMONIA
Actual NORMAL	131	103
Actual PNEUMONIA	16	374

```
Evaluating:   0%|          | 0/20 [00:00<?, ?batch/s]
Evaluating:   5%|▌         | 1/20 [00:03<01:00,  3.16s/batch]
Evaluating:  25%|██▌       | 5/20 [00:03<00:07,  1.92batch/s]
Evaluating:  40%|████      | 8/20 [00:03<00:03,  3.45batch/s]
Evaluating:  50%|█████     | 10/20 [00:03<00:02,  4.29batch/s]
Evaluating:  65%|██████▌   | 13/20 [00:04<00:01,  5.44batch/s]
Evaluating:  75%|███████▌  | 15/20 [00:04<00:00,  6.59batch/s]
Evaluating:  85%|████████▌ | 17/20 [00:04<00:00,  7.28batch/s]
Evaluating: 100%|██████████| 20/20 [00:05<00:00,  3.96batch/s]
```

ResNet152 (Freeze Backbone)

In [17]:
```
!wqf7009-a3-eval --model=resnet152 \
    --checkpoint=../models/resnet152_frozen_10ep_1767266526_best.pth \
    --data-dir=../data/chest_xray \
    --batch-size=32 \
    --split=test
```

```
Using device: cuda
⁘ Loading test dataset...

Loading weights from: resnet152_frozen_10ep_1767266526_best.pth
Running inference...
```

```
╭──────────────────────────────────────────────────────────────────────────────╮
│ Evaluation Config: Model=resnet152 | Split=test                              │
╰──────────────────────────────────────────────────────────────────────────────╯
```

Detailed Evaluation Metrics

Metric	Value
Accuracy	0.8125
Precision	0.8124
Recall (Sensitivity)	0.9103
F1 Score	0.8585

Confusion Matrix

Actual \ Predicted	Pred NORMAL	Pred PNEUMONIA
Actual NORMAL	152	82
Actual PNEUMONIA	35	355

```
Evaluating:   0%|          | 0/20 [00:00<?, ?batch/s]
Evaluating:   5%|▌         | 1/20 [00:03<00:59,  3.11s/batch]
Evaluating:  10%|▉         | 2/20 [00:03<00:24,  1.34s/batch]
Evaluating:  20%|██        | 4/20 [00:03<00:09,  1.77batch/s]
Evaluating:  25%|██▌       | 5/20 [00:03<00:06,  2.37batch/s]
Evaluating:  35%|███▌      | 7/20 [00:03<00:03,  3.69batch/s]
Evaluating:  40%|████      | 8/20 [00:03<00:02,  4.38batch/s]
Evaluating:  45%|████▌     | 9/20 [00:03<00:02,  5.13batch/s]
Evaluating:  50%|█████     | 10/20 [00:04<00:01,  5.91batch/s]
Evaluating:  55%|█████▌    | 11/20 [00:04<00:01,  6.66batch/s]
Evaluating:  60%|██████    | 12/20 [00:04<00:01,  7.34batch/s]
Evaluating:  65%|██████▌   | 13/20 [00:04<00:00,  7.93batch/s]
Evaluating:  75%|███████▌  | 15/20 [00:04<00:00,  8.75batch/s]
Evaluating:  85%|████████▌ | 17/20 [00:04<00:00,  9.19batch/s]
Evaluating:  90%|█████████ | 18/20 [00:04<00:00,  9.33batch/s]
Evaluating: 100%|██████████| 20/20 [00:04<00:00, 10.59batch/s]
Evaluating: 100%|██████████| 20/20 [00:05<00:00,  3.59batch/s]
```

ResNet152 (Fine-tune All Layers)

In [18]:
```
!wqf7009-a3-eval --model=resnet152 \
    --checkpoint=../models/resnet152_unfrozen_10ep_1767265644_best.pth \
    --data-dir=../data/chest_xray \
    --batch-size=32 \
    --split=test
```

```
Using device: cuda
⁘ Loading test dataset...

Loading weights from: resnet152_unfrozen_10ep_1767265644_best.pth
Running inference...
```

```
╭──────────────────────────────────────────────────────────────────────────────╮
│ Evaluation Config: Model=resnet152 | Split=test                              │
╰──────────────────────────────────────────────────────────────────────────────╯
```

Detailed Evaluation Metrics

Metric	Value
Accuracy	0.8622
Precision	0.8248
Recall (Sensitivity)	0.9897
F1 Score	0.8998

Confusion Matrix

Actual \ Predicted	Pred NORMAL	Pred PNEUMONIA
Actual NORMAL	152	82
Actual PNEUMONIA	4	386

```
Evaluating:   0%|          | 0/20 [00:00<?, ?batch/s]
Evaluating:   5%|▌         | 1/20 [00:03<00:58,  3.09s/batch]
Evaluating:  10%|█         | 2/20 [00:03<00:23,  1.33s/batch]
Evaluating:  15%|█▌        | 3/20 [00:03<00:13,  1.30batch/s]
Evaluating:  25%|██▌       | 5/20 [00:03<00:05,  2.56batch/s]
Evaluating:  30%|███       | 6/20 [00:03<00:04,  3.25batch/s]
Evaluating:  35%|███▌      | 7/20 [00:03<00:03,  4.05batch/s]
Evaluating:  45%|████▌     | 9/20 [00:03<00:01,  5.55batch/s]
Evaluating:  50%|█████     | 10/20 [00:03<00:01,  6.22batch/s]
Evaluating:  60%|██████    | 12/20 [00:04<00:01,  7.40batch/s]
Evaluating:  70%|███████   | 14/20 [00:04<00:00,  8.26batch/s]
Evaluating:  75%|███████▌  | 15/20 [00:04<00:00,  8.56batch/s]
Evaluating:  85%|████████▌ | 17/20 [00:04<00:00,  9.02batch/s]
Evaluating:  90%|█████████ | 18/20 [00:04<00:00,  9.20batch/s]
Evaluating:  95%|█████████▌| 19/20 [00:04<00:00,  9.36batch/s]
Evaluating: 100%|██████████| 20/20 [00:05<00:00,  3.61batch/s]
```

VGG16 (Freeze Backbone)

In [19]:
```
!wqf7009-a3-eval --model=vgg16 \
    --checkpoint=../models/vgg16_frozen_10ep_1767266050_best.pth \
    --data-dir=../data/chest_xray \
    --batch-size=32 \
    --split=test
```

```
Using device: cuda
⁘ Loading test dataset...

Loading weights from: vgg16_frozen_10ep_1767266050_best.pth
Running inference...
```

```
╭──────────────────────────────────────────────────────────────────────────────╮
│ Evaluation Config: Model=vgg16 | Split=test                                  │
╰──────────────────────────────────────────────────────────────────────────────╯
```

Detailed Evaluation Metrics

Metric	Value
Accuracy	0.8221
Precision	0.7807
Recall (Sensitivity)	0.9949
F1 Score	0.8749

Confusion Matrix

Actual \ Predicted	Pred NORMAL	Pred PNEUMONIA
Actual NORMAL	125	109
Actual PNEUMONIA	2	388

```
Evaluating:   0%|          | 0/20 [00:00<?, ?batch/s]
Evaluating:   5%|▌         | 1/20 [00:03<00:59,  3.14s/batch]
Evaluating:  10%|█         | 2/20 [00:03<00:24,  1.36s/batch]
Evaluating:  15%|█▌        | 3/20 [00:03<00:13,  1.27batch/s]
Evaluating:  20%|██        | 4/20 [00:03<00:08,  1.93batch/s]
Evaluating:  25%|██▌       | 5/20 [00:03<00:05,  2.72batch/s]
Evaluating:  30%|███       | 6/20 [00:03<00:03,  3.60batch/s]
Evaluating:  35%|███▌      | 7/20 [00:03<00:02,  4.50batch/s]
Evaluating:  40%|████      | 8/20 [00:03<00:02,  5.41batch/s]
Evaluating:  45%|████▌     | 9/20 [00:03<00:01,  6.25batch/s]
Evaluating:  50%|█████     | 10/20 [00:04<00:01,  7.03batch/s]
Evaluating:  55%|█████▌    | 11/20 [00:04<00:01,  7.62batch/s]
Evaluating:  60%|██████    | 12/20 [00:04<00:00,  8.06batch/s]
Evaluating:  65%|██████▌   | 13/20 [00:04<00:00,  8.50batch/s]
Evaluating:  70%|███████   | 14/20 [00:04<00:00,  8.76batch/s]
Evaluating:  75%|███████▌  | 15/20 [00:04<00:00,  9.00batch/s]
Evaluating:  80%|████████  | 16/20 [00:04<00:00,  9.18batch/s]
Evaluating:  85%|████████▌ | 17/20 [00:04<00:00,  9.28batch/s]
Evaluating:  90%|█████████ | 18/20 [00:04<00:00,  9.35batch/s]
Evaluating:  95%|█████████▌| 19/20 [00:05<00:00,  9.44batch/s]
Evaluating: 100%|██████████| 20/20 [00:05<00:00,  3.53batch/s]
```

VGG16 (Fine-tune All Layers)

In [20]:
```
!wqf7009-a3-eval --model=vgg16 \
    --checkpoint=../models/vgg16_unfrozen_10ep_1767265306_best.pth \
    --data-dir=../data/chest_xray \
    --batch-size=32 \
    --split=test
```

```
Using device: cuda
⁘ Loading test dataset...

Loading weights from: vgg16_unfrozen_10ep_1767265306_best.pth
Running inference...
╭──────────────────────────────────────────────────────────────────────────────╮
│ Evaluation Config: Model=vgg16 | Split=test                                  │
╰──────────────────────────────────────────────────────────────────────────────╯
   Detailed Evaluation Metrics
┌──────────────────────┬──────────┐
│               Metric │ Value    │
├──────────────────────┼──────────┤
│             Accuracy │ 0.8317   │
│            Precision │ 0.7890   │
│ Recall (Sensitivity) │ 0.9974   │
│             F1 Score │ 0.8811   │
└──────────────────────┴──────────┘
                  Confusion Matrix
┌──────────────────────┬──────────────┬──────────────┐
│ Actual \ Predicted   │ Pred NORMAL  │ Pred PNEUMONIA │
├──────────────────────┼──────────────┼──────────────┤
│      Actual NORMAL   │     130      │      104     │
│   Actual PNEUMONIA   │      1       │      389     │
└──────────────────────┴──────────────┴──────────────┘
```

```
Evaluating:   0%|          | 0/20 [00:00<?, ?batch/s]
Evaluating:   5%|▌         | 1/20 [00:03<00:59,  3.11s/batch]
Evaluating:  10%|█         | 2/20 [00:03<00:24,  1.35s/batch]
Evaluating:  15%|█▌        | 3/20 [00:03<00:13,  1.28batch/s]
Evaluating:  20%|██        | 4/20 [00:03<00:08,  1.94batch/s]
Evaluating:  25%|██▌       | 5/20 [00:03<00:05,  2.73batch/s]
Evaluating:  30%|███       | 6/20 [00:03<00:03,  3.59batch/s]
Evaluating:  35%|███▌      | 7/20 [00:03<00:02,  4.51batch/s]
Evaluating:  40%|████      | 8/20 [00:03<00:02,  5.42batch/s]
Evaluating:  45%|████▌     | 9/20 [00:03<00:01,  6.28batch/s]
Evaluating:  50%|█████     | 10/20 [00:04<00:01,  6.97batch/s]
Evaluating:  55%|█████▌    | 11/20 [00:04<00:01,  7.57batch/s]
Evaluating:  60%|██████    | 12/20 [00:04<00:00,  8.05batch/s]
Evaluating:  65%|██████▌   | 13/20 [00:04<00:00,  8.42batch/s]
Evaluating:  70%|███████   | 14/20 [00:04<00:00,  8.73batch/s]
Evaluating:  75%|███████▌  | 15/20 [00:04<00:00,  8.95batch/s]
Evaluating:  80%|████████  | 16/20 [00:04<00:00,  9.17batch/s]
Evaluating:  85%|████████▌ | 17/20 [00:04<00:00,  9.25batch/s]
Evaluating:  90%|█████████ | 18/20 [00:04<00:00,  9.39batch/s]
Evaluating:  95%|█████████▌| 19/20 [00:05<00:00,  9.41batch/s]
Evaluating: 100%|██████████| 20/20 [00:05<00:00,  3.54batch/s]
```

3.4. Model Comparison

First load all the trained weights into the respective models.

In [21]:
```
# Model checkpoint paths
checkpoints = {
    "simple_cnn": "simplecnn_baseline_10ep_1767264997_best.pth",
    "resnet152_frozen": "resnet152_frozen_10ep_1767266526_best.pth",
    "resnet152_unfrozen": "resnet152_unfrozen_10ep_1767265644_best.pth",
    "vgg16_frozen": "vgg16_frozen_10ep_1767266050_best.pth",
```

```
    "vgg16_unfrozen": "vgg16_unfrozen_10ep_1767265306_best.pth",
}

# Load trained model weights
print("Loading trained model weights...")
for name, checkpoint_file in checkpoints.items():
    checkpoint_path = MODELS_DIR / checkpoint_file
    if checkpoint_path.exists():
        models[name].load_state_dict(torch.load(checkpoint_path, map_location=DEVICE))
        print(f"Loaded {name} from {checkpoint_file}")
    else:
        print(f"Warning: {checkpoint_file} not found")
```

```
Loading trained model weights...
Loaded simple_cnn from simplecnn_baseline_10ep_1767264997_best.pth
Loaded resnet152_frozen from resnet152_frozen_10ep_1767266526_best.pth
Loaded resnet152_unfrozen from resnet152_unfrozen_10ep_1767265644_best.pth
Loaded vgg16_frozen from vgg16_frozen_10ep_1767266050_best.pth
Loaded vgg16_unfrozen from vgg16_unfrozen_10ep_1767265306_best.pth
```

In [22]:
```
# Setup loss function with class weighting
criterion = get_weighted_bce_loss(train_loader.dataset).to(DEVICE)
pos_weight = criterion.pos_weight.cpu().numpy()[0]
print(f"Loss function: {criterion}")
print(f"Positive class weight: {pos_weight:.4f} (for class 'PNEUMONIA')")
print()

# Evaluate all models on test set
print("Evaluating models on test set...")
print("-" * 50)
model_names_display = {
    "simple_cnn": "SimpleCNN",
    "resnet152_frozen": "ResNet152 (frozen)",
    "resnet152_unfrozen": "ResNet152 (unfrozen)",
    "vgg16_frozen": "VGG16 (frozen)",
    "vgg16_unfrozen": "VGG16 (unfrozen)",
}

results = {}
for name, display_name in model_names_display.items():
    print(f"Evaluating {display_name}...", end=" ")
    results[display_name] = evaluate(models[name], test_loader, criterion, DEVICE)
    print(f"(F1: {results[display_name][2]:.4f})")
```

```
Loss function: BCEWithLogitsLoss()
Positive class weight: 0.3461 (for class 'PNEUMONIA')

Evaluating models on test set...
--------------------------------------------------
Evaluating SimpleCNN...
Evaluating :   0%|          | 0/20 [00:07<?, ?it/s]
(F1: 0.8627)
Evaluating ResNet152 (frozen)...
Evaluating :   0%|          | 0/20 [00:27<?, ?it/s]
(F1: 0.8585)
Evaluating ResNet152 (unfrozen)...
Evaluating :   0%|          | 0/20 [00:07<?, ?it/s]
(F1: 0.8998)
Evaluating VGG16 (frozen)...
Evaluating :   0%|          | 0/20 [00:07<?, ?it/s]
(F1: 0.8749)
Evaluating VGG16 (unfrozen)...
Evaluating :   0%|          | 0/20 [00:07<?, ?it/s]
(F1: 0.8811)
```

In [23]:
```
# Display results in a formatted table
print("\nModel Performance Summary:")
print("=" * 70)
results_df = pd.DataFrame(results, index=["Loss", "Accuracy", "F1 Score"]).T
results_df.round(4)
```

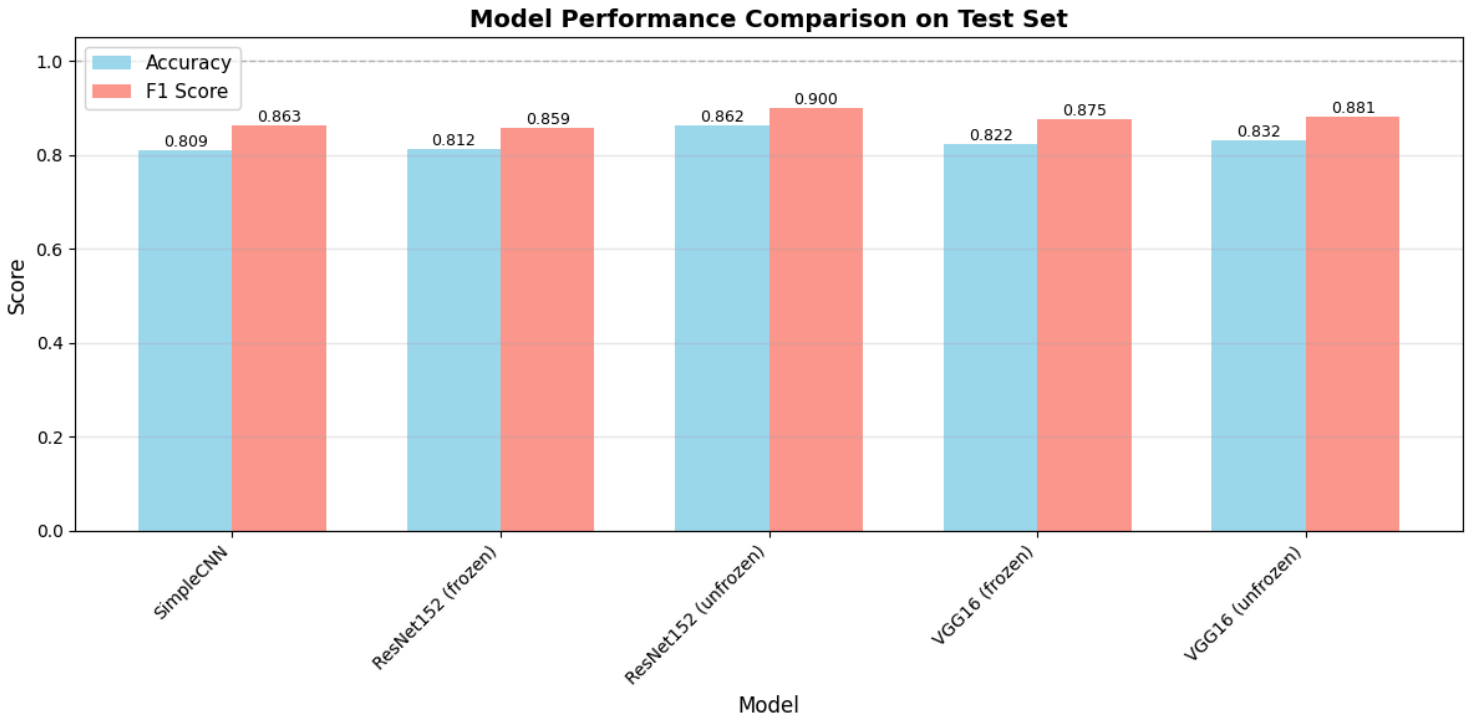
```
Model Performance Summary:
======================================================================
```

Out[23]:

	Loss	Accuracy	F1 Score
SimpleCNN	0.4575	0.8093	0.8627
ResNet152 (frozen)	0.3216	0.8125	0.8585
ResNet152 (unfrozen)	0.5864	0.8622	0.8998
VGG16 (frozen)	0.5849	0.8221	0.8749
VGG16 (unfrozen)	2.0633	0.8317	0.8811

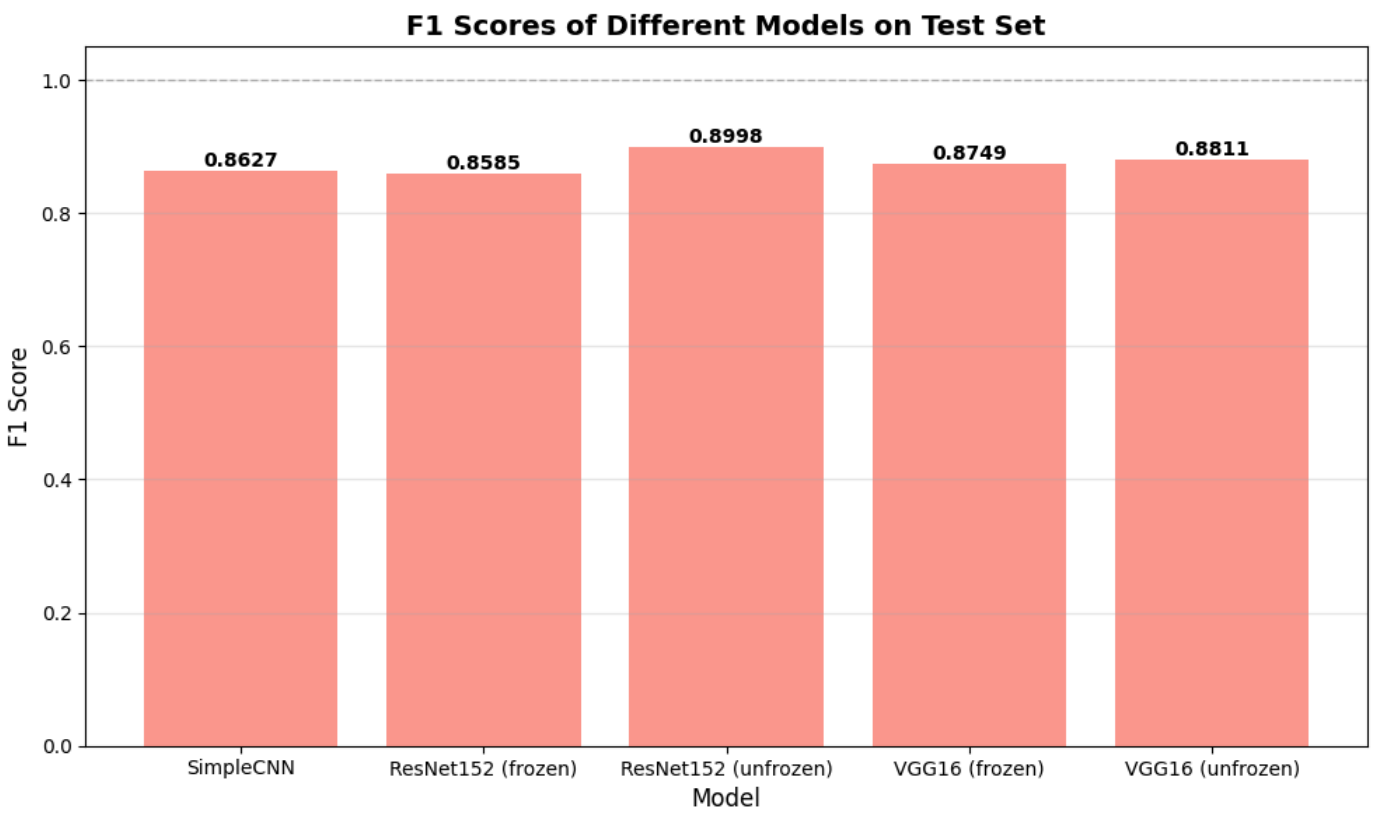
In [24]:
```
# Create comparison dataframe
df = pd.DataFrame(results, index=["Loss", "Accuracy", "F1 Score"]).T
df = df.drop(columns=["Loss"])
```

```
# Plot comparison of Accuracy and F1 Score
width = 0.35
fig, ax = plt.subplots(figsize=(12, 6))
x = np.arange(len(df))
bars1 = ax.bar(x - width/2, df["Accuracy"], width, label="Accuracy", color="skyblue", alpha=0.8)
bars2 = ax.bar(x + width/2, df["F1 Score"], width, label="F1 Score", color="salmon", alpha=0.8)

# Add value labels on bars
for bars in [bars1, bars2]:
    for bar in bars:
        height = bar.get_height()
        ax.text(bar.get_x() + bar.get_width()/2., height,
                f'{height:.3f}', ha='center', va='bottom', fontsize=9)

ax.axhline(y=1.0, color="gray", linestyle="--", linewidth=1, alpha=0.5)
ax.set_title("Model Performance Comparison on Test Set", fontsize=14, fontweight="bold")
ax.set_xlabel("Model", fontsize=12)
ax.set_ylabel("Score", fontsize=12)
ax.set_xticks(x)
ax.set_xticklabels(df.index, rotation=45, ha="right")
ax.legend(fontsize=11)
ax.grid(axis="y", alpha=0.3)
plt.tight_layout()
plt.savefig(FIG_DIR / "model_performance.png", dpi=150, bbox_inches="tight")
plt.show()
```



In [25]:

```
# Plot F1 scores comparison
f1_scores = df["F1 Score"].values
fig, ax = plt.subplots(figsize=(10, 6))
bars = ax.bar(df.index, f1_scores, color="salmon", alpha=0.8, linewidth=1.5)

# Add value labels on bars
for bar in bars:
    height = bar.get_height()
    ax.text(bar.get_x() + bar.get_width()/2., height,
            f'{height:.4f}', ha='center', va='bottom', fontsize=10, fontweight="bold")

ax.axhline(y=1.0, color="gray", linestyle="--", linewidth=1, alpha=0.5)
ax.set_title("F1 Scores of Different Models on Test Set", fontsize=14, fontweight="bold")
ax.set_xlabel("Model", fontsize=12)
ax.set_ylabel("F1 Score", fontsize=12)
ax.grid(axis="y", alpha=0.3)
plt.tight_layout()
plt.savefig(FIG_DIR / "f1_scores.png", dpi=150, bbox_inches="tight")
plt.show()
```



4. XAI Method 1 - Grad-CAM

In [26]:
```
# Find a batch with a good mix of pneumonia examples for visualization
print("Searching for a batch with pneumonia examples...")
images = None
labels = None
target_pneumonia_count = BATCH_SIZE // 2  # At least half the batch

for i, (batch_images, batch_labels) in enumerate(test_loader):
    pneumonia_count = (batch_labels.numpy() == 1).sum()
    print(f"  Batch {i}: {pneumonia_count} pneumonia examples")
    if pneumonia_count >= target_pneumonia_count:
        images = batch_images
        labels = batch_labels
        break

if images is None:
    print("Warning: No suitable batch found, using first batch")
    images, labels = next(iter(test_loader))

images = images.to(DEVICE)
labels = labels.numpy()
print(f"\n-> Selected batch with {len(labels)} examples ({labels.sum()} pneumonia cases)")
```

```
Searching for a batch with pneumonia examples...
  Batch 0: 0 pneumonia examples
  Batch 1: 0 pneumonia examples
  Batch 2: 0 pneumonia examples
  Batch 3: 0 pneumonia examples
  Batch 4: 0 pneumonia examples
  Batch 5: 0 pneumonia examples
  Batch 6: 0 pneumonia examples
  Batch 7: 22 pneumonia examples

-> Selected batch with 32 examples (22 pneumonia cases)
```

In [27]:
```
# Get predictions from the best performing model (ResNet152 unfrozen)
preds, probs = predict(images, model=resnet152_unfrozen)
preds, probs = preds.cpu().numpy(), probs.cpu().numpy()

# Calculate batch metrics
print("Batch Performance Metrics:")
print("-" * 50)
print(f"Accuracy:  {accuracy_score(labels, preds):.4f}")
print(f"F1 Score:  {f1_score(labels, preds):.4f}")
print(f"Precision: {precision_score(labels, preds):.4f}")
print(f"Recall:    {recall_score(labels, preds):.4f}")
print()

# Create dataframe with predictions sorted by confidence
df = pd.DataFrame({
```

```
    "labels": labels,
    "preds": preds,
    "probs": probs,
    "correct": (labels == preds).astype(int)
})
df = df.sort_values("probs", ascending=False)

print("Predictions sorted by confidence:")
df
```

```
Batch Performance Metrics:
--------------------------------------------------
Accuracy:  0.8750
F1 Score:  0.9167
Precision: 0.8462
Recall:    1.0000

Predictions sorted by confidence:
```

Out[27]:

	labels	preds	probs	correct
22	1	1	0.999979	1
23	1	1	0.999906	1
15	1	1	0.999904	1
13	1	1	0.999867	1
31	1	1	0.999847	1
25	1	1	0.999807	1
11	1	1	0.999789	1
28	1	1	0.999756	1
14	1	1	0.999717	1
26	1	1	0.999688	1
12	1	1	0.999668	1
16	1	1	0.999653	1
18	1	1	0.999649	1
29	1	1	0.999464	1
21	1	1	0.999420	1
27	1	1	0.999293	1
17	1	1	0.999273	1
20	1	1	0.999096	1
24	1	1	0.998902	1
19	1	1	0.998727	1
6	0	1	0.998645	0
4	0	1	0.998531	0
10	1	1	0.998155	1
2	0	1	0.997262	0
30	1	1	0.995750	1
7	0	1	0.991810	0
9	0	0	0.336039	1
3	0	0	0.120561	1
8	0	0	0.070298	1
1	0	0	0.008140	1
5	0	0	0.005114	1
0	0	0	0.000004	1

In [28]:

```
# Select examples for visualization: top 3 most confident and bottom 3 least confident
top_conf_inds = df.head(3).index.tolist()  # Top 3 most confident predictions
bot_conf_inds = df.tail(3).index.tolist()[
    ::-1
]  # Bottom 3 least confident (reversed for display)
all_conf_inds = top_conf_inds + bot_conf_inds

print(f"Selected {len(all_conf_inds)} examples for Grad-CAM visualization:")
print(f"  - Top 3 most confident: indices {top_conf_inds}")
print(f"  - Bottom 3 least confident: indices {bot_conf_inds}")
print("\nDetails:")
```

```
for idx in all_conf_inds:
    row = df.loc[idx]
    status = "->" if row["correct"] else "x"
    print(
        f"  {status} Index {idx}: Label={row['labels']}, Pred={row['preds']}, "
        f"Prob={row['probs']:.4f}"
    )
```

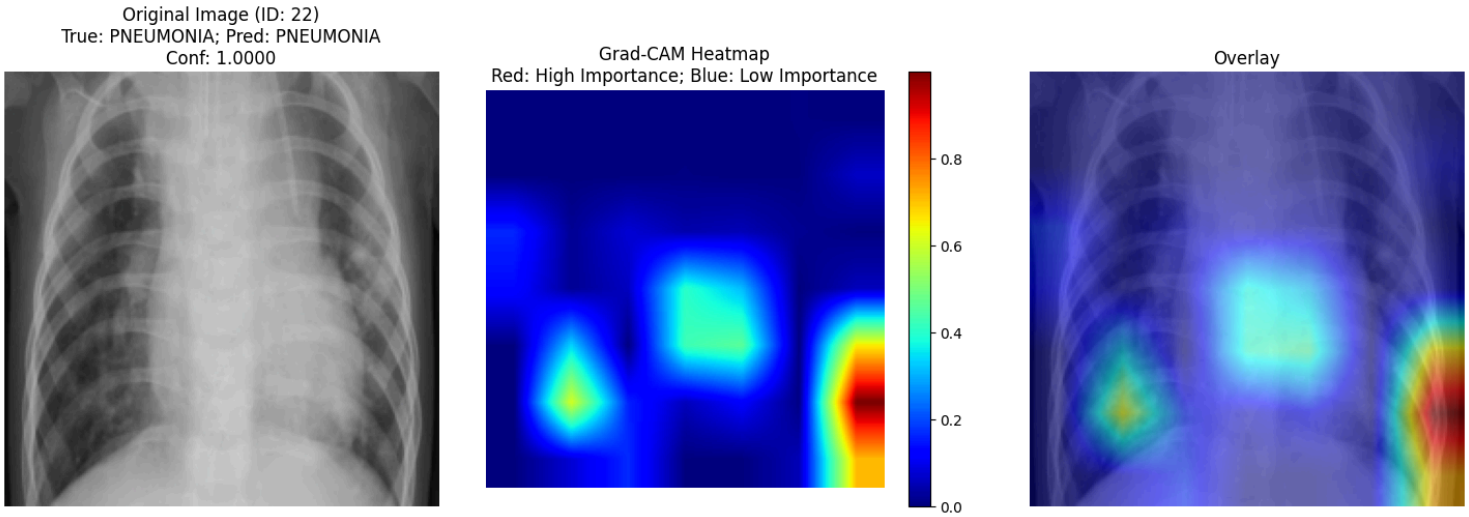
```
Selected 6 examples for Grad-CAM visualization:
  - Top 3 most confident: indices [22, 23, 15]
  - Bottom 3 least confident: indices [0, 5, 1]

Details:
  -> Index 22: Label=1.0, Pred=1.0, Prob=1.0000
  -> Index 23: Label=1.0, Pred=1.0, Prob=0.9999
  -> Index 15: Label=1.0, Pred=1.0, Prob=0.9999
  -> Index 0: Label=0.0, Pred=0.0, Prob=0.0000
  -> Index 5: Label=0.0, Pred=0.0, Prob=0.0051
  -> Index 1: Label=0.0, Pred=0.0, Prob=0.0081
```

4.1. `Task 3` - Implementation (5%)

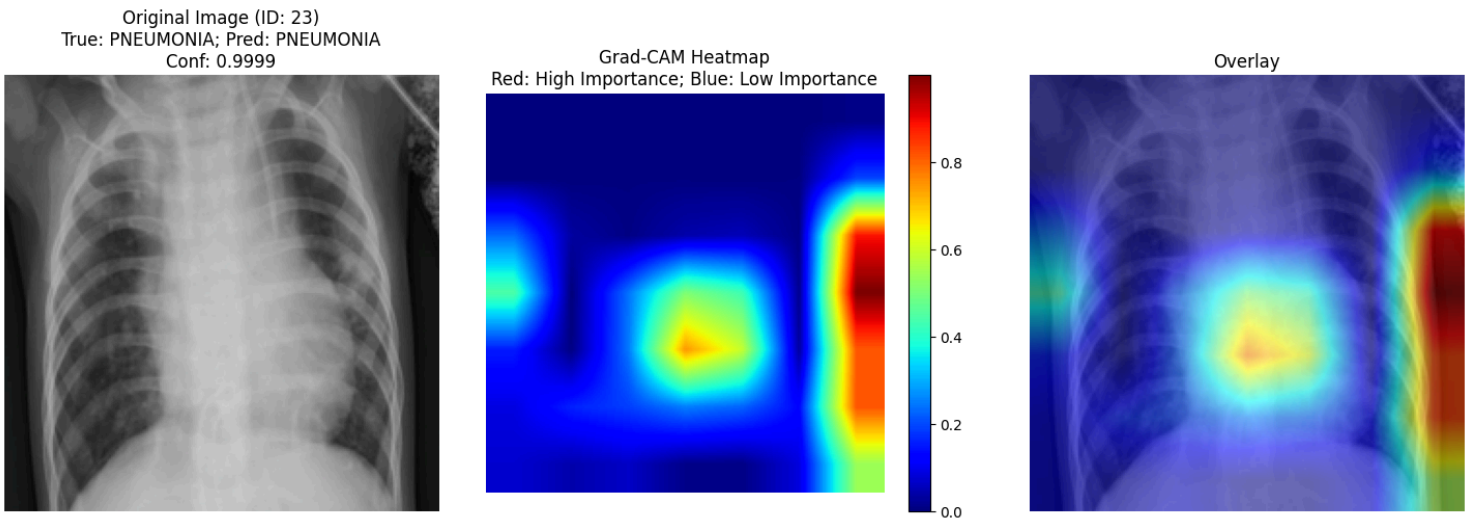
In [29]:
```
# Initialize Grad-CAM for ResNet152
# Target the last layer of the final ResNet block (Layer4) for feature visualization
target_layer = resnet152_unfrozen.model.layer4[-1]
cam = GradCAM(model=resnet152_unfrozen, target_layers=[target_layer])
```

In [30]:
```
# Generate and visualize Grad-CAM heatmaps
print(f"\nGenerating Grad-CAM visualizations for {len(all_conf_inds)} examples...")
print("-" * 70)

for i, idx in enumerate(all_conf_inds):
    row = df.loc[idx]
    print(
        f"[{i + 1}/{len(all_conf_inds)}] Processing example {idx}: "
        f"Label={row['labels']}, Pred={row['preds']}, Prob={row['probs']:.4f}"
    )

    visualize_gradcam(
        image=images[idx],
        label=labels[idx],
        output=preds[idx],
        conf=probs[idx],
        image_id=idx,
        idx_to_classname=test_loader.dataset.classes,
        gradcam=cam,
        save_path=FIG_DIR
        / f"gradcam_{test_loader.dataset.classes[labels[idx]].lower()}_id{idx}.png",
    )
```

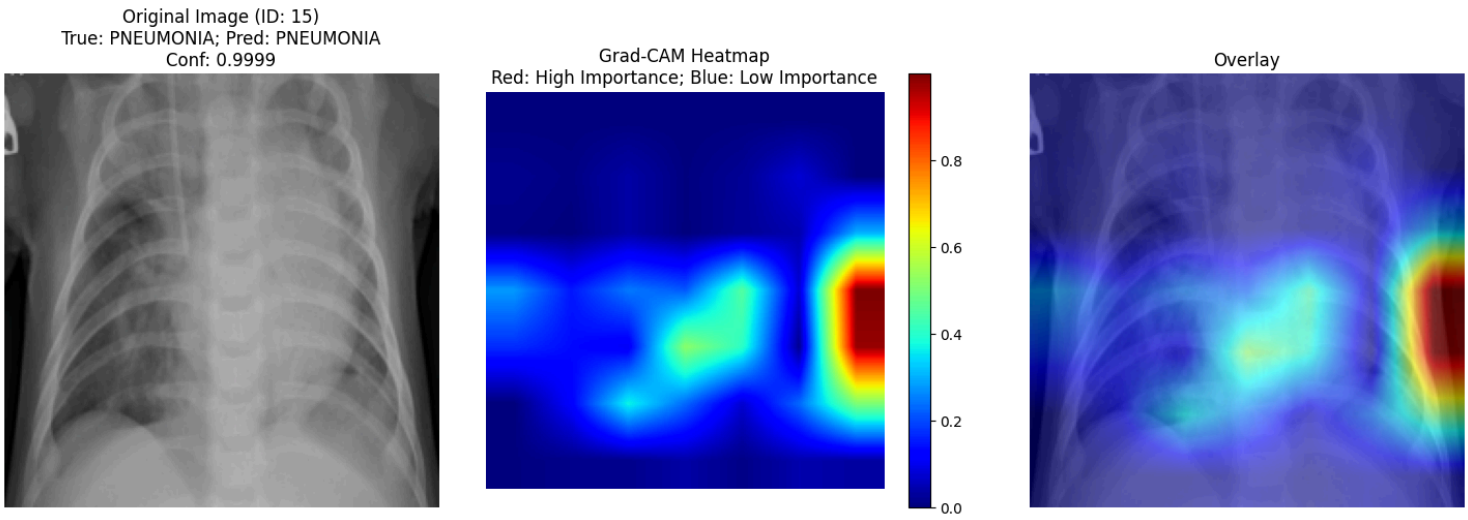
```
Generating Grad-CAM visualizations for 6 examples...
----------------------------------------------------------------------
[1/6] Processing example 22: Label=1.0, Pred=1.0, Prob=1.0000
```



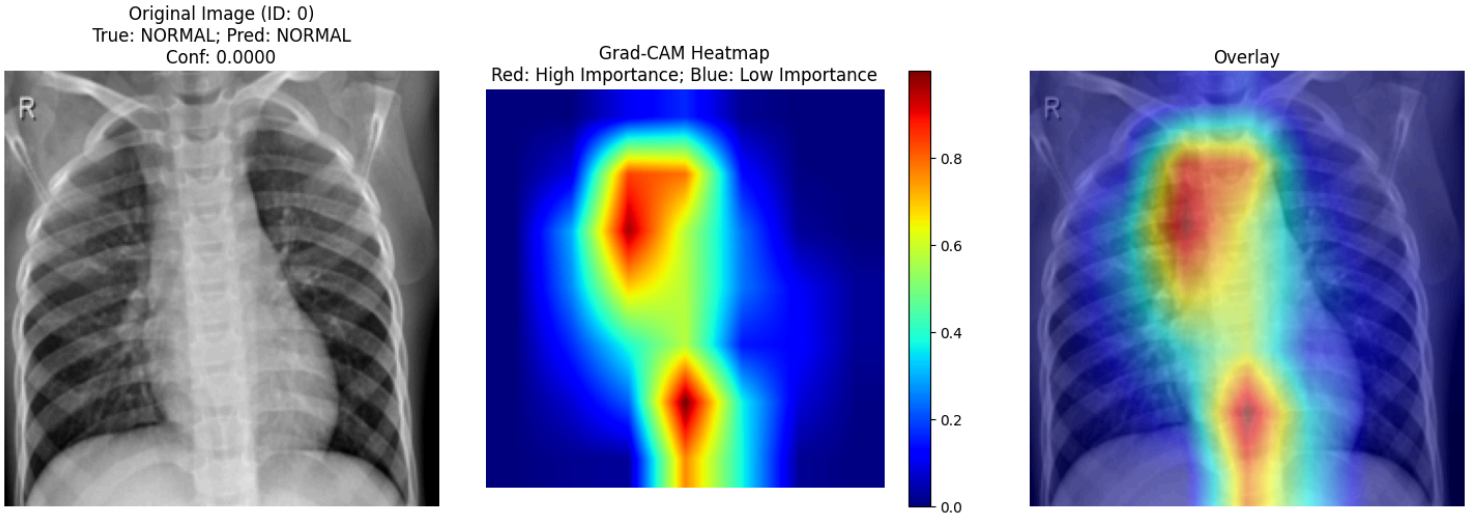

```
Saved figure to: d:\jherng\Workspace\university\masters\courses\year2526_sem1\wqf7009_explainable_artificial_intelligence\assignment3\wqf7009-a3\docs\figures\gradcam_pneumonia_id22.png
[2/6] Processing example 23: Label=1.0, Pred=1.0, Prob=0.9999
```



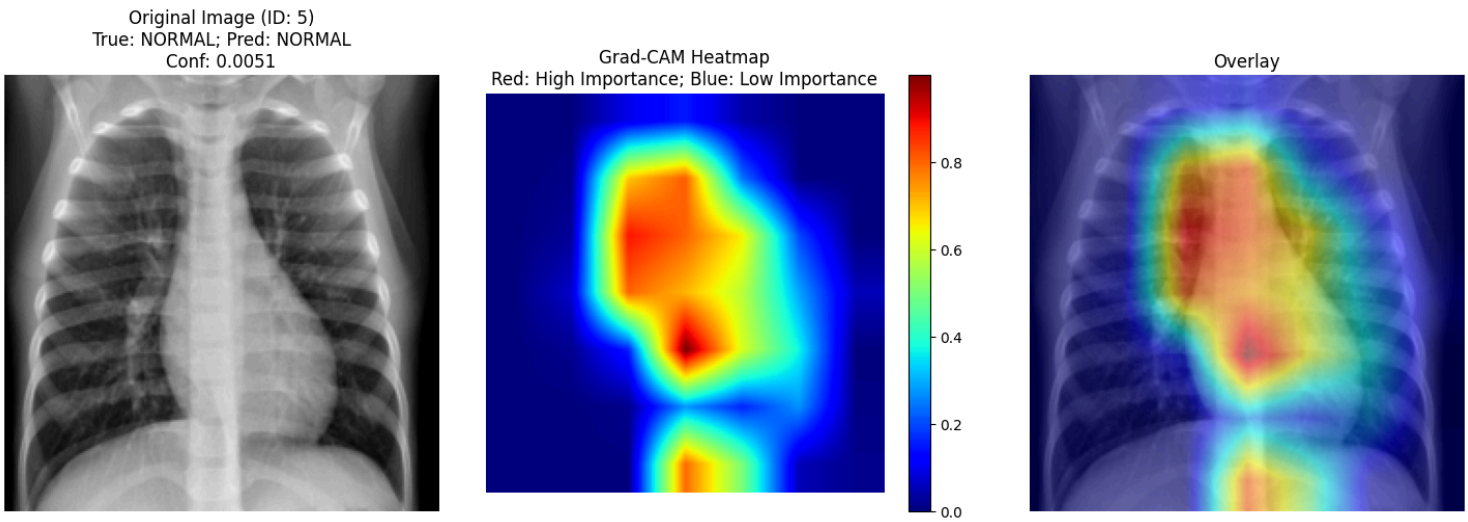

```
Saved figure to: d:\jherng\Workspace\university\masters\courses\year2526_sem1\wqf7009_explainable_artificial_intelligence\assignment3\wqf7009-a3\docs\figures\gradcam_pneumonia_id23.png
[3/6] Processing example 15: Label=1.0, Pred=1.0, Prob=0.9999
```



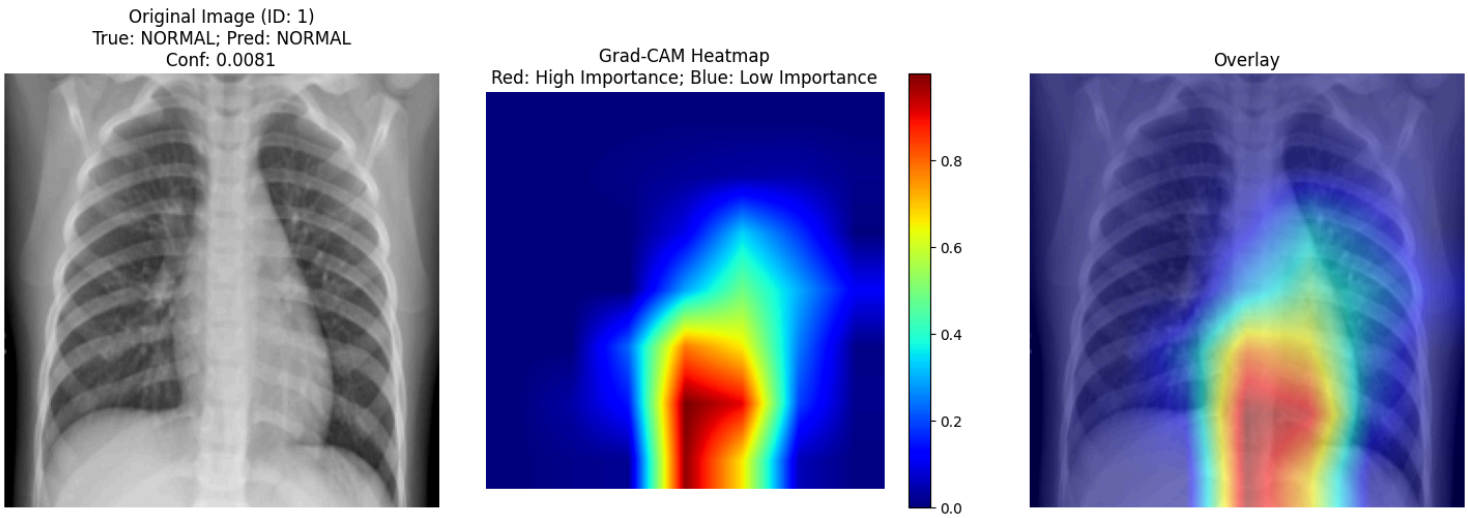

```
Saved figure to: d:\jherng\Workspace\university\masters\courses\year2526_sem1\wqf7009_explainable_artificial_intelligence\assignment3\wqf7009-a3\docs\figures\gradcam_pneumonia_id15.png
[4/6] Processing example 0: Label=0.0, Pred=0.0, Prob=0.0000
```




```
Saved figure to: d:\jherng\Workspace\university\masters\courses\year2526_sem1\wqf7009_explainable_artificial_intelligence\assignment3\wqf7009-a3\docs\figures\gradcam_normal_id0.png
[5/6] Processing example 5: Label=0.0, Pred=0.0, Prob=0.0051
```




```
Saved figure to: d:\jherng\Workspace\university\masters\courses\year2526_sem1\wqf7009_explainable_artificial_intelligence\assignment3\wqf7009-a3\docs\figures\gradcam_normal_id5.png
[6/6] Processing example 1: Label=0.0, Pred=0.0, Prob=0.0081
```




```
Saved figure to: d:\jherng\Workspace\university\masters\courses\year2526_sem1\wqf7009_explainable_artificial_intelligence\assignment3\wqf7009-a3\docs\figures\gradcam_normal_id1.png
```

4.2. `Task 4` - Explanation (5%)

1. Analysis of Normal Cases (IDs 0, 1, 5) In the correctly classified Normal cases, the Grad-CAM heatmaps predominantly focus on the **mediastinum** (the central white region containing the heart and spine) and the clear lung fields.

- **Medical Interpretation:** This suggests the model is employing negative reasoning. By focusing on the sharp edges of the **cardiac silhouette** (the heart border), the model is confirming the **absence of the "Silhouette Sign."** In pneumonia patients, the heart border is often obscured by fluid. By verifying the heart border is crisp and defined, the model correctly deduces the lung is normal.
- **Conclusion:** The model correctly identifies "Normal" by recognizing the structural integrity of the central chest anatomy, rather than just looking at empty space.

2. Analysis of Pneumonia Cases (IDs 15, 22, 23) For the Pneumonia cases, the model behavior is mixed, revealing both correct pathological detection and significant reliability issues.

- **Correct Detection:** The heatmap correctly highlights areas of the "fluffy," radiopaque (white) patches where fluid has accumulated in the lung lobes. This indicates the model has learned to identify the primary visual feature of bacterial pneumonia.
- **Shortcut Learning:** However, a concerning pattern is observed where the highest confidence regions (the "hot" red blobs) appear **outside the thoracic cage** (e.g., the background on the right side).
  - **Root Cause:** This is an example of **"Shortcut Learning"**. The model likely memorized background artifacts, such as hospital tags, cables, or specific scanner noise, that are more common in the Pneumonia dataset than in the Normal dataset.
  - **Class Imbalance:** Although we utilized a **Weighted Binary Cross Entropy Loss** (assigning a lower weight of 0.3461 to the majority Pneumonia class) to prevent the model from biased guessing, this loss function does not prevent the model from exploiting non-medical correlations in the background.

3. Conclusion on Reliability

While the model achieves high F1-score (0.8998), the XAI analysis tells it is **not fully reliable** for clinical deployment. It effectively detects prominent consolidation in severe cases, but its tendency to rely on background artifacts suggests that its high performance may partly stem from memorizing spurious correlations rather than understanding lung pathology. Future work may involve preprocessing to crop background artifacts to force the model to focus solely on the lung fields.

5. XAI Method 2 - RISE

RISE (Randomized Input Sampling for Explanation) is a model-agnostic explanation method that generates saliency maps by randomly masking input images and weighting the masks by their corresponding model predictions. Unlike Grad-CAM, RISE does not require gradient computation and can work with any black-box model.

We'll use the same batch and examples from Section 4 for consistency in comparison.

Parameters Used for RISE:

- **N=2000**: Number of masks (higher = better quality but slower computation)
- **s=8**: Grid size for mask generation (8x8 grid creates smoother masks)
- **p1=0.2**: Probability of masking (20% occlusion density balances between detail and coverage)

5.1. `Task 5` - Implementation (5%)

In [31]:
```
# Initialize RISE explainer and generate random masks
# Parameters:
# - N=2000: Number of random masks to generate (more masks = better quality but slower)
# - s=8: Grid size for mask generation (8x8 grid creates smoother masks)
# - p1=0.2: Probability of masking (20% occlusion density balances between detail and coverage)
explainer = RISE(
    model=resnet152_unfrozen, input_size=(IMG_SIZE, IMG_SIZE), batch_size=100
)
explainer.generate_masks(N=2000, s=8, p1=0.2)
```

```
Generating RISE masks:   0%|          | 0/20 [00:00<?, ?it/s]
Generated 2000 masks of size (224, 224)
```

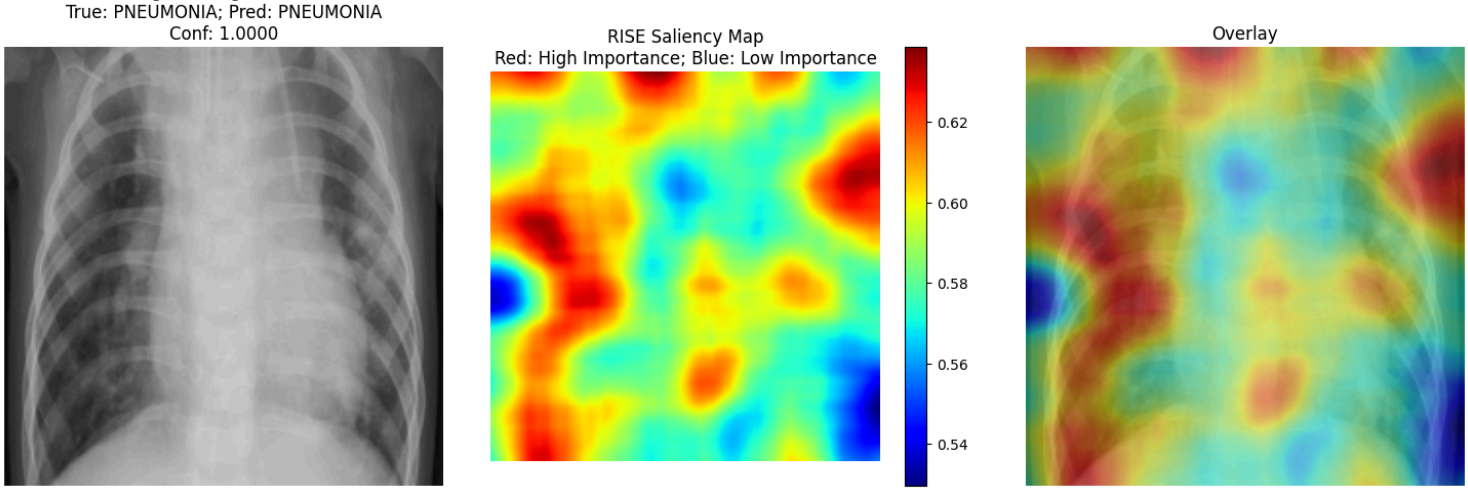
In [32]:
```
# Generate RISE saliency maps for the same examples used in Grad-CAM visualization
# We'll use the examples selected in Section 4: top 3 most confident and bottom 3 least confident
print(f"\nGenerating RISE saliency maps for {len(all_conf_inds)} examples...")
print("-" * 70)

# Store saliency maps for visualization
saliency_maps = {}

for i, idx in enumerate(all_conf_inds):
    row = df.loc[idx]
    print(
        f"[{i + 1}/{len(all_conf_inds)}] Processing example {idx}: "
        f"Label={row['labels']}, Pred={row['preds']}, Prob={row['probs']:.4f}"
    )

    # Determine target class based on prediction
    target_class = int(row["labels"])

    # Generate saliency map
    saliency_map = explainer(images[idx], target_class=target_class)
    saliency_maps[idx] = saliency_map
```

```
Generating RISE saliency maps for 6 examples...
----------------------------------------------------------------------
[1/6] Processing example 22: Label=1.0, Pred=1.0, Prob=1.0000
Computing RISE saliency:   0%|          | 0/20 [00:00<?, ?it/s]
[2/6] Processing example 23: Label=1.0, Pred=1.0, Prob=0.9999
Computing RISE saliency:   0%|          | 0/20 [00:00<?, ?it/s]
[3/6] Processing example 15: Label=1.0, Pred=1.0, Prob=0.9999
Computing RISE saliency:   0%|          | 0/20 [00:00<?, ?it/s]
[4/6] Processing example 0: Label=0.0, Pred=0.0, Prob=0.0000
Computing RISE saliency:   0%|          | 0/20 [00:00<?, ?it/s]
[5/6] Processing example 5: Label=0.0, Pred=0.0, Prob=0.0051
Computing RISE saliency:   0%|          | 0/20 [00:00<?, ?it/s]
[6/6] Processing example 1: Label=0.0, Pred=0.0, Prob=0.0081
Computing RISE saliency:   0%|          | 0/20 [00:00<?, ?it/s]
```

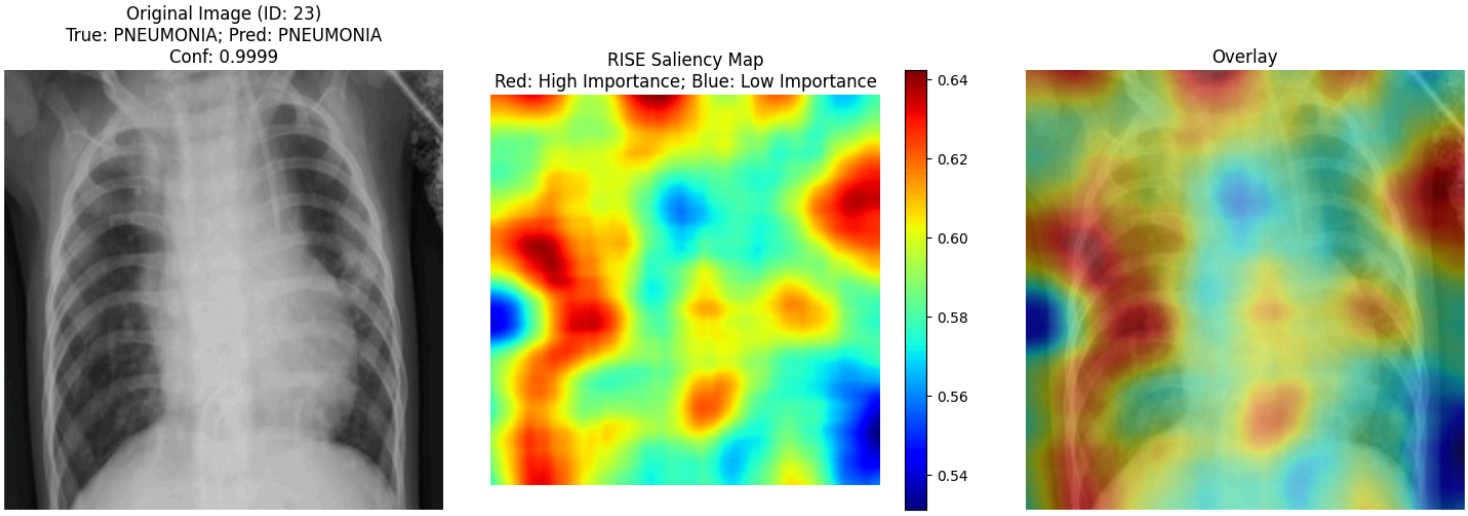
In [33]:
```
# Visualize RISE saliency maps for all selected examples
print(f"\nVisualizing RISE saliency maps for {len(all_conf_inds)} examples...")
print("-" * 70)

for i, idx in enumerate(all_conf_inds):
    row = df.loc[idx]
    print(
        f"[{i + 1}/{len(all_conf_inds)}] Visualizing example {idx}: "
        f"Label={row['labels']}, Pred={row['preds']}, Prob={row['probs']:.4f}"
    )

    visualize_rise(
        image=images[idx],
        saliency_map=saliency_maps[idx],
        label=labels[idx],
        output=preds[idx],
        conf=probs[idx],
        image_id=idx,
        idx_to_classname=test_loader.dataset.classes,
        save_path=FIG_DIR
```

```
        / f"rise_{test_loader.dataset.classes[labels[idx]].lower()}_id{idx}.png",
    )
```

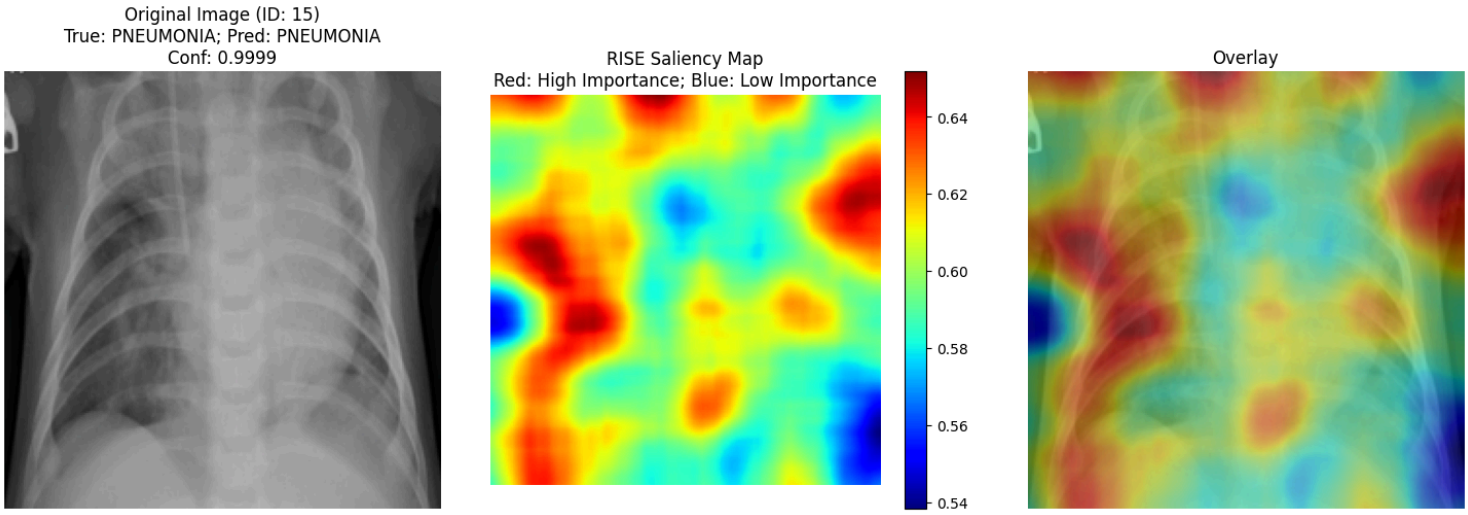
```
Visualizing RISE saliency maps for 6 examples...
--------------------------------------------------------------------------
[1/6] Visualizing example 22: Label=1.0, Pred=1.0, Prob=1.0000
```



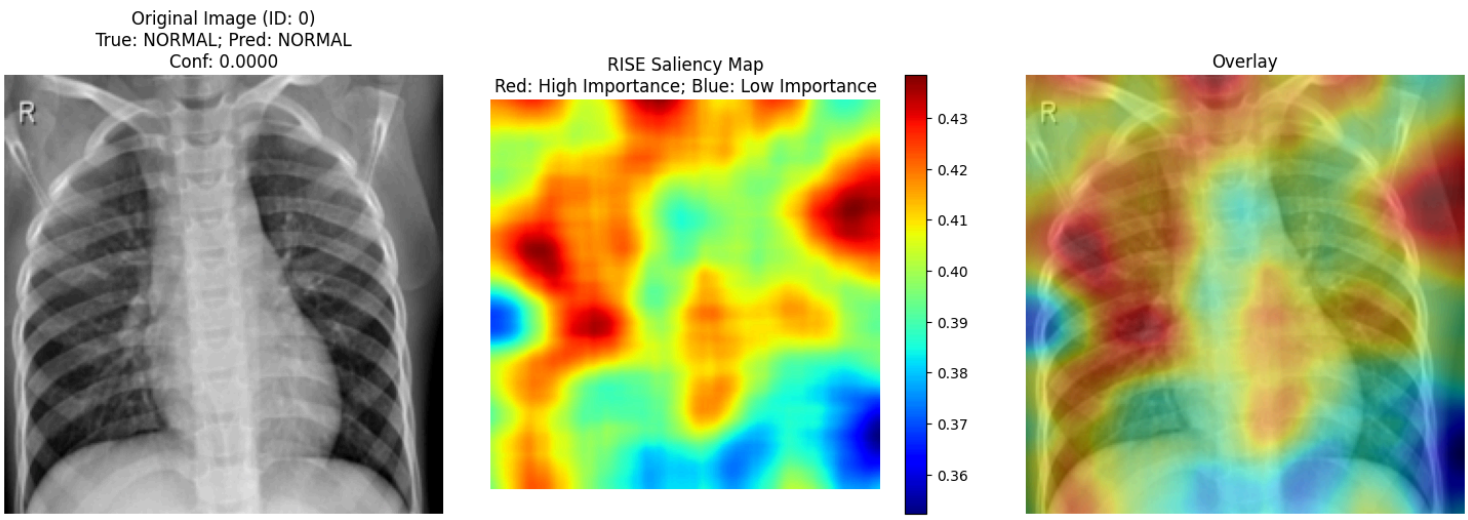

```
Saved figure to: d:\jherng\Workspace\university\masters\courses\year2526_sem1\wqf7009_explainable_artificial_intelligence\assignment3\wqf7009-a3\docs\figures\rise_pneumonia_id22.png
[2/6] Visualizing example 23: Label=1.0, Pred=1.0, Prob=0.9999
```



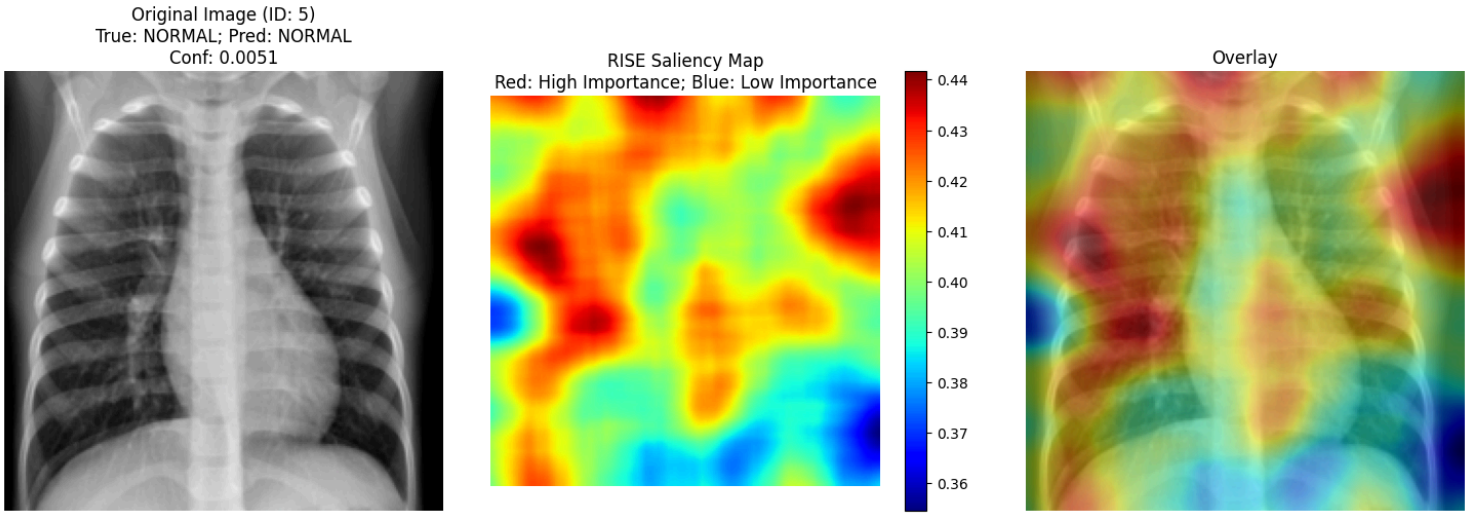

```
Saved figure to: d:\jherng\Workspace\university\masters\courses\year2526_sem1\wqf7009_explainable_artificial_intelligence\assignment3\wqf7009-a3\docs\figures\rise_pneumonia_id23.png
[3/6] Visualizing example 15: Label=1.0, Pred=1.0, Prob=0.9999
```



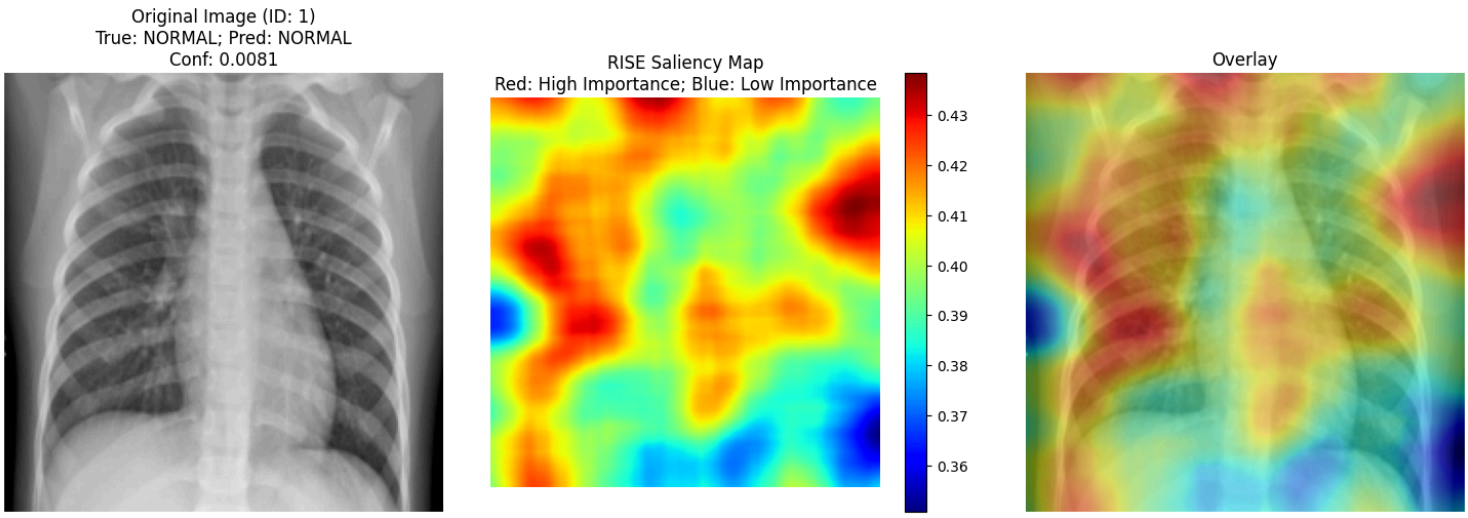

```
Saved figure to: d:\jherng\Workspace\university\masters\courses\year2526_sem1\wqf7009_explainable_artificial_intelligence\assignment3\wqf7009-a3\docs\figures\rise_pneumonia_id15.png
[4/6] Visualizing example 0: Label=0.0, Pred=0.0, Prob=0.0000
```




```
Saved figure to: d:\jherng\Workspace\university\masters\courses\year2526_sem1\wqf7009_explainable_artificial_intelligence\assignment3\wqf7009-a3\docs\figures\rise_normal_id0.png
[5/6] Visualizing example 5: Label=0.0, Pred=0.0, Prob=0.0051
```




```
Saved figure to: d:\jherng\Workspace\university\masters\courses\year2526_sem1\wqf7009_explainable_artificial_intelligence\assignment3\wqf7009-a3\docs\figures\rise_normal_id5.png
[6/6] Visualizing example 1: Label=0.0, Pred=0.0, Prob=0.0081
```




```
Saved figure to: d:\jherng\Workspace\university\masters\courses\year2526_sem1\wqf7009_explainable_artificial_intelligence\assignment3\wqf7009-a3\docs\figures\rise_normal_id1.png
```

5.2. `Task 6` - Explanation (5%)

1. Interpretation of RISE Heatmaps Unlike Grad-CAM, which uses gradients to identify discriminative regions, RISE uses randomized masking (perturbation) to test the model's sensitivity to every part of the image.

- **Normal Cases (IDs 0, 1, 5):** The RISE heatmaps display broad, continuous areas of high importance (red regions) covering the clear lung fields (right lung, left side of the image) and the **mediastinum** (heart/spine).
    - *Medical Insight:* This indicates that the model relies on the **uniformity of the texture** in healthy lungs. The continuous red signal confirms that the model is detecting the "radiolucency" (blackness of air) across the entire lung field to predict "Normal," rather than looking for a specific focal point.

- **Pneumonia Cases (IDs 15, 22, 23):** The heatmaps are more fragmented, showing scattered "blobs" of high importance focused on areas of **consolidation** (the "fluffy" white patches).
    - *Medical Insight:* This fragmentation suggests that the model is reacting to specific localized irregularities in the lung texture caused by fluid or infection, rather than a single global shape.

2. Comparison with Grad-CAM (Task 3) Comparing the two methods reveals significant differences in how the model's reasoning is visualized:

- **Localization vs. Granularity:** Grad-CAM produced smooth, cohesive heatmaps because it operates on low-resolution feature maps and upsampled to the image size. In contrast, RISE produces **fine-grained, pixel-level sensitivity maps**. While Grad-CAM showed us *where* the model was looking generally, RISE reveals exactly *which pixels* caused the confidence score to drop when masked.
- **Artifacts and Noise:** RISE contains more "artifacts" (scattered red spots in the background corners). This reveals that the RISE method's fragility. Because RISE tests random occlusions, it reveals that the model is sensitive to background noise—changing pixels in the top corners (left and right) *does* affect the probability score, which Grad-CAM smoothed over.

3. Reflection on Model Trustworthiness The RISE analysis reinforces the "Shortcut Learning" hypothesis raised in the Grad-CAM analysis, but makes it even more apparent.

- **Trustworthiness Issue:** The fact that RISE highlights scattered pixels outside the body (top left/right corners) confirms that the model is **over-sensitive to background noise**. While Grad-CAM showed the model *looked* at the background, RISE proves that the background *actively drives the decision*.
- **Method Evaluation:** While Grad-CAM appears "cleaner" and easier for a clinician to read rapidly, RISE provides a more honest, i.e., noisier, view of the model's pixel-level reliance. For this specific model, **Grad-CAM is the superior visualization for clinical interpretability**, as the extreme granularity of RISE makes it difficult to distinguish between true pathology detection and random model noise.