```python
import os
import torch
import torch.nn as nn
import numpy as np
import shap
from torchvision import models
from PIL import Image
import matplotlib.pyplot as plt


device = torch.device("cpu")
print(f"Using device: {device}")


model_path = r"C:\Users\Ekaansh\OneDrive\Desktop\AB\research\SHAP\
best_pneumonia_densenet121.pt"
dataset_dir = r"D:\datasets\chest_xray\val\PNEUMONIA"  # pneumonia
images
save_dir = r"C:\Users\Ekaansh\OneDrive\Desktop\AB\research\SHAP\
finding\shap_pred_1"

os.makedirs(save_dir, exist_ok=True)
class_names = ['NORMAL', 'PNEUMONIA']


model = models.densenet121(weights=None)
num_features = model.classifier.in_features
model.classifier = nn.Sequential(
    nn.Linear(num_features, 256),
    nn.ReLU(),
    nn.Dropout(0.3),
    nn.Linear(256, len(class_names))
)
model.load_state_dict(torch.load(model_path, map_location=device))
model.eval()


mean = np.array([0.485, 0.456, 0.406])
std = np.array([0.229, 0.224, 0.225])

def preprocess_input(images):
    images = images / 255.0
    images = (images - mean) / std
    images = images.transpose(0, 3, 1, 2)
    return torch.tensor(images).float()


def f(x):
    x_tensor = preprocess_input(x)
```

```python
    with torch.no_grad():
        out = model(x_tensor)
        probs = torch.softmax(out, dim=1)
    return probs.cpu().numpy()


all_images = sorted([os.path.join(dataset_dir, f) for f in
os.listdir(dataset_dir) if f.lower().endswith(('.jpg', '.jpeg',
'.png'))])
selected_images = all_images[:50]

X = np.stack([np.array(Image.open(p).convert("RGB").resize((224,
224))) for p in selected_images])


masker = shap.maskers.Image("inpaint_telea", X[0].shape)
explainer = shap.Explainer(f, masker, output_names=class_names)

for idx, img_array in enumerate(X):
    shap_values = explainer(img_array[np.newaxis, ...],
max_evals=5000, batch_size=20,
outputs=shap.Explanation.argsort.flip[:2])

    plt.figure(figsize=(5, 5))
    shap.image_plot(shap_values, show=False)

    filename = os.path.basename(selected_images[idx])
    save_path = os.path.join(save_dir, f"shap_{filename}.png")
    plt.savefig(save_path, bbox_inches='tight', pad_inches=0)
    plt.close()
    print(f"Saved: {save_path}")
```

Using device: cpu

C:\Users\Ekaansh\AppData\Local\Temp\ipykernel_24304\32912513.py:32:
FutureWarning: You are using `torch.load` with `weights_only=False`
(the current default value), which uses the default pickle module
implicitly. It is possible to construct malicious pickle data which
will execute arbitrary code during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-
models for more details). In a future release, the default value for
`weights_only` will be flipped to `True`. This limits the functions
that could be executed during unpickling. Arbitrary objects will no
longer be allowed to be loaded via this mode unless they are
explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
`weights_only=True` for any use case where you don't have full control
of the loaded file. Please open an issue on GitHub for any issues
related to this experimental feature.
  model.load_state_dict(torch.load(model_path, map_location=device))
```

{"model_id":"a2c66d2c3ab343849203f7f373e61cd8","version_major":2,"version_minor":0}

PartitionExplainer explainer: 2it [07:03, 423.67s/it]

Saved: C:\Users\Ekaansh\OneDrive\Desktop\AB\research\SHAP\finding\shap_pred_1\shap_person1946_bacteria_4874.jpeg.png

{"model_id":"e874237e52e44f25a16cdd5c62b9d851","version_major":2,"version_minor":0}

PartitionExplainer explainer: 2it [06:24, 384.69s/it]

Saved: C:\Users\Ekaansh\OneDrive\Desktop\AB\research\SHAP\finding\shap_pred_1\shap_person1946_bacteria_4875.jpeg.png

{"model_id":"e174e46cdf334c0a8f3bb4f77c41a079","version_major":2,"version_minor":0}

PartitionExplainer explainer: 2it [06:08, 368.94s/it]

Saved: C:\Users\Ekaansh\OneDrive\Desktop\AB\research\SHAP\finding\shap_pred_1\shap_person1947_bacteria_4876.jpeg.png

{"model_id":"7c7e4ee201e14916bff939ba7c6a06c5","version_major":2,"version_minor":0}

PartitionExplainer explainer: 2it [05:54, 354.41s/it]

Saved: C:\Users\Ekaansh\OneDrive\Desktop\AB\research\SHAP\finding\shap_pred_1\shap_person1949_bacteria_4880.jpeg.png

{"model_id":"8f30020bcc734b87b3a656003f341039","version_major":2,"version_minor":0}

PartitionExplainer explainer: 2it [06:04, 364.12s/it]

Saved: C:\Users\Ekaansh\OneDrive\Desktop\AB\research\SHAP\finding\shap_pred_1\shap_person1950_bacteria_4881.jpeg.png

{"model_id":"b5b6f434f3794862a2b8512fb4137e94","version_major":2,"version_minor":0}

PartitionExplainer explainer: 2it [06:14, 374.05s/it]

Saved: C:\Users\Ekaansh\OneDrive\Desktop\AB\research\SHAP\finding\shap_pred_1\shap_person1951_bacteria_4882.jpeg.png

{"model_id":"d7b615d73d294a1bb4bb4dd96f196f24","version_major":2,"version_minor":0}

PartitionExplainer explainer: 2it [07:12, 432.91s/it]

Saved: C:\Users\Ekaansh\OneDrive\Desktop\AB\research\SHAP\finding\
shap_pred_1\shap_person1952_bacteria_4883.jpeg.png

{"model_id":"f8e996ae30754942a29c1ac897e9e593","version_major":2,"version_minor":0}

PartitionExplainer explainer: 2it [08:02, 482.68s/it]

Saved: C:\Users\Ekaansh\OneDrive\Desktop\AB\research\SHAP\finding\
shap_pred_1\shap_person1954_bacteria_4886.jpeg.png

<Figure size 500x500 with 0 Axes>

<Figure size 500x500 with 0 Axes>

<Figure size 500x500 with 0 Axes>

<Figure size 500x500 with 0 Axes>

<Figure size 500x500 with 0 Axes>

<Figure size 500x500 with 0 Axes>

<Figure size 500x500 with 0 Axes>

<Figure size 500x500 with 0 Axes>