stanowski_homework

June 1, 2025

[25]:

```
from medmnist import VesselMNIST3D
from torch.utils.data import random_split
from torch.utils.data import DataLoader
from torch.utils.data import Subset
from torchvision.transforms import v2
import torch
from torch import nn
import numpy as np
import gdown
import os
import random
import matplotlib.pyplot as plt
import sys
repo_root = os.path.abspath(os.path.join(os.getcwd(), ".."))
sys.path.append(repo_root)

def set_seed(seed=42):
    random.seed(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)
    torch.cuda.manual_seed_all(seed)
    torch.backends.cudnn.deterministic = True
    torch.backends.cudnn.benchmark = False

set_seed(1525)

def take_middle_slice(inpt: np.ndarray):
    """
    NoduleMNIST 3D contains whole nodule volumes, however for this tutorial
    we will utilize only central slice of each example.
    We repeat this slice 3 times, as model expects input to have 3 channels.
    """
    inpt = inpt.squeeze()
    X, Y, Z = inpt.shape
    slice_ = inpt[:, :, Z//2]
    slice_ = torch.Tensor(slice_).unsqueeze(dim=0).repeat(3,1,1)
    return slice_
```

```python
TRANSFORMS = v2.Compose([v2.Lambda(take_middle_slice),
                         v2.Resize(size=(224,224))
                         ])

data_dir = "./example_data"
os.makedirs(data_dir, exist_ok=True)


def download_weights(url, output_dir, filename):
    """
    Downloads weights from the given URL if they are not already downloaded.
    """
    os.makedirs(output_dir, exist_ok=True)
    output_path = os.path.join(output_dir, filename)

    if not os.path.exists(output_path):
        print(f"Downloading weights to {output_path}...")
        gdown.download(url, output_path)
    else:
        print(f"Weights already exist at {output_path}. Skipping download.")

url = "https://drive.google.com/uc?id=1xUevCbvII5yXDxVxb7bR65CPmgz2sGQA"
output_dir = "tuned_models"
filename = "lidc_dino_s8.pth"
download_weights(url, output_dir, filename)

CLASS_NAMES = ["benign", "malignant"]

LOGIT2NAME = {
    0: "benign",
    1: "malignant",
}

train_data = VesselMNIST3D(root=data_dir, split="train", size=64,␣
 ↪transform=TRANSFORMS, download=True)

zero_indices = [i for i, (_, label) in enumerate(train_data) if label == 0]
one_indices = [i for i, (_, label) in enumerate(train_data) if label == 1]

zero_indices_downsampled = random.sample(zero_indices, len(one_indices))

balanced_indices = zero_indices_downsampled + one_indices
random.shuffle(balanced_indices)

balanced_train_data = Subset(train_data, balanced_indices)
train_loader = DataLoader(balanced_train_data, batch_size=16, shuffle=True)
```

```
validation_data = VesselMNIST3D(root=data_dir, split="val", size=64,␣
 ↪transform=TRANSFORMS, download=True)
print("length of Train Datasets: ", len(train_data))
print("length of Validation Datasets: ", len(validation_data))


val_loader = DataLoader(validation_data, batch_size=16, shuffle = False)

test_dataset = VesselMNIST3D(root=data_dir, split="test", transform=TRANSFORMS)
test_loader = DataLoader(test_dataset, batch_size=16, shuffle=False)

CLASS_NAMES = ["benign", "malignant"]

LOGIT2NAME = {
    0: "benign",
    1: "malignant",
}
```

```
Weights already exist at tuned_models\lidc_dino_s8.pth. Skipping download.
length of Train Datasets:  1335
length of Validation Datasets:  191
```

[26]:

```
from transformers import ViTConfig, ViTModel

class DINO(nn.Module):
    """
    DINO Transformer model based on Huggingface implementation.
    """
    def __init__(self):
        super().__init__()
        # Backbone
        config = ViTConfig.from_pretrained('facebook/dino-vits8',␣
 ↪attn_implementation="eager") # We propose eager implementation to return att␣
 ↪scores gracefully.
        self.backbone = ViTModel(config)
        # Classfication head
        self.head = torch.nn.Linear(384, 1) # takes vector of length 384 and␣
 ↪outputs 1 number

    def forward(self, x: torch.Tensor, output_attentions:bool=False):
        out = self.backbone(x, output_attentions=output_attentions)
        x = out["pooler_output"]
        x = self.head(x)
        if output_attentions:
            att = out["attentions"]
            return x, att
```

```
        else:
            return x


DEVICE = "cuda" if torch.cuda.is_available() else "cpu"
WEIGHTS_PATH = "./tuned_models/lidc_dino_s8.pth"

MODEL = DINO()
MODEL.to(DEVICE)
```

[26]:
```
DINO(
  (backbone): ViTModel(
    (embeddings): ViTEmbeddings(
      (patch_embeddings): ViTPatchEmbeddings(
        (projection): Conv2d(3, 384, kernel_size=(8, 8), stride=(8, 8))
      )
      (dropout): Dropout(p=0.0, inplace=False)
    )
    (encoder): ViTEncoder(
      (layer): ModuleList(
        (0-11): 12 x ViTLayer(
          (attention): ViTAttention(
            (attention): ViTSelfAttention(
              (query): Linear(in_features=384, out_features=384, bias=True)
              (key): Linear(in_features=384, out_features=384, bias=True)
              (value): Linear(in_features=384, out_features=384, bias=True)
            )
            (output): ViTSelfOutput(
              (dense): Linear(in_features=384, out_features=384, bias=True)
              (dropout): Dropout(p=0.0, inplace=False)
            )
          )
          (intermediate): ViTIntermediate(
            (dense): Linear(in_features=384, out_features=1536, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): ViTOutput(
            (dense): Linear(in_features=1536, out_features=384, bias=True)
            (dropout): Dropout(p=0.0, inplace=False)
          )
          (layernorm_before): LayerNorm((384,), eps=1e-12,
elementwise_affine=True)
          (layernorm_after): LayerNorm((384,), eps=1e-12,
elementwise_affine=True)
        )
      )
    )
```

```
      (layernorm): LayerNorm((384,), eps=1e-12, elementwise_affine=True)
      (pooler): ViTPooler(
        (dense): Linear(in_features=384, out_features=384, bias=True)
        (activation): Tanh()
      )
    )
    (head): Linear(in_features=384, out_features=1, bias=True)
  )
```

[27]:
```
import torch.nn.functional as F

class FocalLoss(nn.Module):
    def __init__(self, alpha=0.25, gamma=2.0, pos_weight=None):
        super().__init__()
        self.alpha = alpha
        self.gamma = gamma
        self.pos_weight = pos_weight

    def forward(self, logits, targets):
        BCE_loss = F.binary_cross_entropy_with_logits(logits, targets,␣
 ↪pos_weight=self.pos_weight, reduction='none')
        pt = torch.exp(-BCE_loss)  # pt = prob dla prawidłowej klasy
        focal_loss = self.alpha * (1 - pt) ** self.gamma * BCE_loss
        return focal_loss.mean()
```

[29]:
```
optimizer = torch.optim.Adam(MODEL.parameters(), lr=1e-4)
num_epochs = 15

all_labels = []
for _, labels in train_loader:
    all_labels.append(labels)

all_labels = torch.cat(all_labels).float().to(DEVICE)



num_pos = (all_labels == 1).sum().item()
num_neg = (all_labels == 0).sum().item()

pos_weight = torch.tensor(num_neg / num_pos).to(DEVICE)
criterion = FocalLoss(alpha=0.25, gamma=2.0, pos_weight=pos_weight)  # lub inne␣
 ↪wartości
print(f"num_pos: {num_pos}, num_neg: {num_neg}, pos_weight: {pos_weight.item():.
 ↪4f}")
val_accuracies = []

for epoch in range(num_epochs):
    MODEL.train()
```

```
    total_loss = 0

    for batch_idx, (imgs, labels) in enumerate(train_loader):
        imgs = imgs.to(DEVICE)
        labels = labels.to(DEVICE).float().squeeze(1)
        optimizer.zero_grad()
        logits = MODEL(imgs).squeeze(1)
        loss = criterion(logits, labels)
        loss.backward()
        optimizer.step()

        total_loss += loss.item()

        print("loss:", loss.item())

    avg_loss = total_loss / (batch_idx + 1)
    print(f"Epoch {epoch + 1}, Loss: {avg_loss:.4f}")

    MODEL.eval()
    correct = 0
    total = 0
    with torch.no_grad():
        for imgs, labels in val_loader:
            imgs = imgs.to(DEVICE)
            labels = labels.to(DEVICE).float()

            logits = MODEL(imgs)
            probs = torch.sigmoid(logits)
            preds = probs.round().squeeze().cpu().numpy()
            labels = labels.squeeze().cpu().numpy()
            correct += (preds == labels).sum()
            total += labels.shape[0]
            print("preds (val):", preds)
            print("labels (val):", labels)

    val_acc = 100 * correct / total
    val_accuracies.append(val_acc)
    print(f"Validation accuracy: {val_acc:.2f}%")

plt.figure(figsize=(8,5))
plt.plot(range(1, num_epochs+1), val_accuracies, marker='o')
plt.title("Validation accuracy per E]epoch")
plt.xlabel("Epoch")
plt.ylabel("Validation accuracy (%)")
plt.grid(True)
plt.show()
```

```python
MODEL.eval()
correct_test = 0
total_test = 0
with torch.no_grad():
    for imgs, labels in test_loader:
        imgs = imgs.to(DEVICE)
        labels = labels.to(DEVICE).float()

        logits = MODEL(imgs)
        probs = torch.sigmoid(logits)
        preds = probs.round().squeeze().cpu().numpy()
        labels = labels.squeeze().cpu().numpy()
        correct_test += (preds == labels).sum()
        total_test += labels.shape[0]

test_acc = 100 * correct_test / total_test
print(f"Test accuracy: {test_acc:.2f}%")
```

```
num_pos: 150, num_neg: 150, pos_weight: 1.0000
loss: 0.14075657725334167
loss: 0.06351222842931747
loss: 0.043826259672641754
loss: 0.03918594866991043
loss: 0.05615861713886261
loss: 0.05191811919212341
loss: 0.044362787157297134
loss: 0.042200468480587006
loss: 0.04675618186593056
loss: 0.039542678743600845
loss: 0.048038892447948456
loss: 0.0636490136384964
loss: 0.04167595133185387
loss: 0.041553787887096405
loss: 0.04357944801449776
loss: 0.042718324810266495
loss: 0.04907320439815521
loss: 0.03754698485136032
loss: 0.04241284355521202
Epoch 1, Loss: 0.0515
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

```
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Validation accuracy: 11.52%
loss: 0.04158097505569458
loss: 0.04684554040431976
loss: 0.0502932034432888
loss: 0.044244684278964996
loss: 0.04164588078856468
loss: 0.04273136332631111
loss: 0.04195341467857361
loss: 0.04187668859958649
loss: 0.04626242816448212
loss: 0.03947575390338898
loss: 0.047290362417697906
loss: 0.04420217499136925
loss: 0.048100925981998444
loss: 0.04257012531161308
loss: 0.043724365532398224
loss: 0.044874463230371475
loss: 0.041094932705163956
loss: 0.04506208747625351
loss: 0.045299772173166275
Epoch 2, Loss: 0.0442
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1. 1. 1.]
labels (val): [1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 0. 0. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

```
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 0. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Validation accuracy: 20.42%
loss: 0.04508793726563454
loss: 0.04287930577993393
loss: 0.04237480089068413
loss: 0.0439719557762146
loss: 0.049027979373931885
loss: 0.04273757338523865
loss: 0.049408551305532455
loss: 0.04565347731113434
loss: 0.04262344539165497
loss: 0.04452013596892357
loss: 0.04522779583930969
loss: 0.04290347173810005
loss: 0.046248096972703934
loss: 0.03853914886713028
loss: 0.0484711155295372
loss: 0.04464155063033104
loss: 0.0403461679816246
loss: 0.051775794476270676
loss: 0.05407834053039551
Epoch 3, Loss: 0.0453
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

```
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Validation accuracy: 88.48%
loss: 0.05262657254934311
loss: 0.05063309520483017
loss: 0.04432445019483566
loss: 0.041348449885845184
loss: 0.046444837003946304
loss: 0.04876819998025894
loss: 0.04259558767080307
loss: 0.046157434582710266
loss: 0.04051939770579338
loss: 0.044475022703409195
loss: 0.040413998067379
loss: 0.04194027930498123
loss: 0.044518645852804184
loss: 0.040455412119627
loss: 0.03909440338611603
loss: 0.04155981168150902
loss: 0.05527450516819954
loss: 0.039549507200717926
loss: 0.05015914514660835
Epoch 4, Loss: 0.0448
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

```
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Validation accuracy: 87.43%
loss: 0.04805659130215645
loss: 0.04265624284744263
loss: 0.03845047950744629
loss: 0.05105288326740265
loss: 0.03456864133477211
loss: 0.050780393183231354
loss: 0.05124438926577568
loss: 0.04017280787229538
loss: 0.04397096112370491
loss: 0.04066793620586395
loss: 0.0412658266723156
loss: 0.03870716691017151
loss: 0.0408342108130455
loss: 0.046741195023059845
loss: 0.043171849101781845
loss: 0.037105005234479904
loss: 0.03873463720083237
loss: 0.043576762080192566
loss: 0.051413994282484055
Epoch 5, Loss: 0.0433
preds (val): [0. 0. 1. 0. 1. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
preds (val): [1. 1. 0. 0. 1. 0. 0. 1. 0. 0. 0. 0. 0. 1. 1. 0.]
labels (val): [1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

```
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Validation accuracy: 82.72%
loss: 0.03801870346069336
loss: 0.041511934250593185
loss: 0.04510176181793213
loss: 0.04338895156979561
loss: 0.03852447122335434
loss: 0.035342585295438766
loss: 0.03884278982877731
loss: 0.054835762828588486
loss: 0.03511139750480652
loss: 0.048431072384119034
loss: 0.04594090208411217
loss: 0.04136810824275017
loss: 0.03951428830623627
loss: 0.043208327144384384
loss: 0.05295830965042114
loss: 0.04337477311491966
loss: 0.0470111221075058
loss: 0.04836561158299446
loss: 0.04077548533678055
Epoch 6, Loss: 0.0432
preds (val): [0. 0. 1. 0. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
preds (val): [1. 1. 0. 0. 1. 0. 0. 1. 0. 0. 0. 0. 0. 1. 1. 0.]
labels (val): [1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 1. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 1. 1. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

```
Validation accuracy: 75.92%
loss: 0.04374665394425392
loss: 0.04095173627138138
loss: 0.04428928345441818
loss: 0.043194208294153214
loss: 0.044178154319524765
loss: 0.04285283386707306
loss: 0.0400061272084713
loss: 0.043193891644477844
loss: 0.0414276160299778
loss: 0.04063017666339874
loss: 0.041905228048563004
loss: 0.042484577745199203
loss: 0.03954632580280304
loss: 0.045593906193971634
loss: 0.03837363421916962
loss: 0.04621327295899391
loss: 0.04571560025215149
loss: 0.03633960708975792
loss: 0.03551841527223587
Epoch 7, Loss: 0.0419
preds (val): [0. 0. 1. 0. 1. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
preds (val): [1. 1. 0. 0. 1. 0. 0. 1. 0. 0. 0. 0. 0. 1. 1. 0.]
labels (val): [1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 1. 1. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Validation accuracy: 79.06%
loss: 0.04584737494587898
loss: 0.0521291084587574
```

```
loss: 0.03694257512688637
loss: 0.04675525426864624
loss: 0.040550172328948975
loss: 0.046277228742837906
loss: 0.04415375739336014
loss: 0.0436103492975235
loss: 0.046476710587739944
loss: 0.054645851254463196
loss: 0.04041486233472824
loss: 0.03600252419710159
loss: 0.038512300699949265
loss: 0.03454121574759483
loss: 0.048805929720401764
loss: 0.0495651550590992
loss: 0.0451933816075325
loss: 0.045496076345443726
loss: 0.04683726653456688
Epoch 8, Loss: 0.0444
preds (val): [0. 1. 1. 0. 1. 1. 1. 1. 1. 0. 1. 0. 0. 0. 1. 1.]
labels (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
preds (val): [1. 1. 0. 0. 1. 0. 0. 1. 0. 0. 0. 0. 0. 1. 1. 0.]
labels (val): [1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 1. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 1. 1. 0. 1. 0. 0. 1. 1. 0. 0. 0. 0. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 0. 1. 0. 0. 0. 0. 1. 0. 1. 0. 0. 1. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 1. 0. 0. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 0. 1. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 1. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Validation accuracy: 73.30%
loss: 0.034873463213443756
loss: 0.04220857471227646
loss: 0.037717998027801514
loss: 0.042069267481565475
loss: 0.04061528295278549
```

```
loss: 0.040906425565481186
loss: 0.04444725438952446
loss: 0.045253388583660126
loss: 0.03765987232327461
loss: 0.0433889776468277
loss: 0.03928259760141373
loss: 0.03835313022136688
loss: 0.04180965572595596
loss: 0.048668570816516876
loss: 0.04100678488612175
loss: 0.04305335879325867
loss: 0.04479554295539856
loss: 0.037086233496665955
loss: 0.03796388581395149
Epoch 9, Loss: 0.0411
preds (val): [0. 0. 1. 0. 1. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
preds (val): [1. 1. 0. 0. 1. 0. 0. 1. 0. 0. 0. 0. 0. 1. 1. 0.]
labels (val): [1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Validation accuracy: 81.68%
loss: 0.04846752807497978
loss: 0.04634765535593033
loss: 0.038801707327365875
loss: 0.03692525625228882
loss: 0.04157141596078873
loss: 0.03942066431045532
loss: 0.036144860088825226
loss: 0.03323439881205559
```

```
loss: 0.04964040592312813
loss: 0.03634417802095413
loss: 0.05426150560379028
loss: 0.044141802936792374
loss: 0.03016185387969017
loss: 0.036769893020391464
loss: 0.048678863793611526
loss: 0.04394804313778877
loss: 0.04303784295916557
loss: 0.040589213371276855
loss: 0.05683170258998871
Epoch 10, Loss: 0.0424
preds (val): [0. 0. 1. 0. 1. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
preds (val): [1. 1. 0. 0. 1. 0. 0. 1. 0. 0. 0. 0. 0. 1. 1. 0.]
labels (val): [1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Validation accuracy: 82.20%
loss: 0.047591082751750946
loss: 0.035583190619945526
loss: 0.029955189675092697
loss: 0.039771970361471176
loss: 0.03990701586008072
loss: 0.046172551810741425
loss: 0.042587410658597946
loss: 0.03730021417140961
loss: 0.04730115085840225
loss: 0.04002366214990616
loss: 0.045881487429142
```

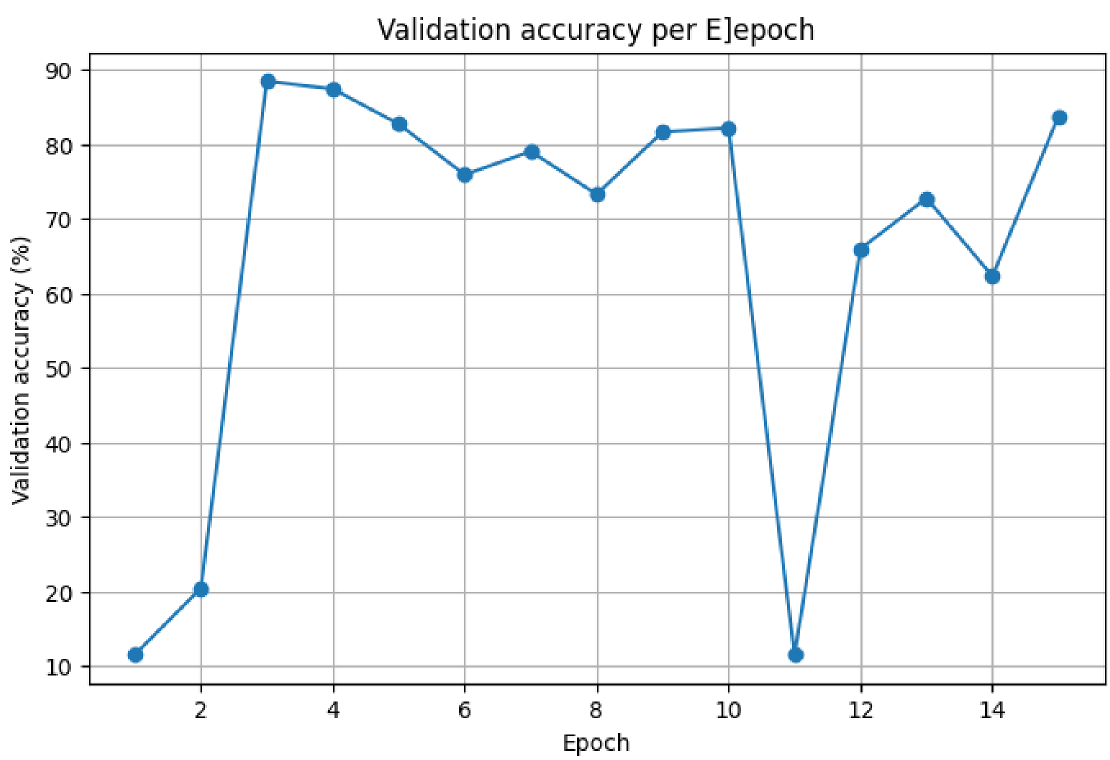
```
loss: 0.04311409592628479
loss: 0.0386342778801918
loss: 0.0360533744096756
loss: 0.04644332826137543
loss: 0.04440363496541977
loss: 0.04152946174144745
loss: 0.04742872342467308
loss: 0.04214690998196602
Epoch 11, Loss: 0.0417
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Validation accuracy: 11.52%
loss: 0.04277776926755905
loss: 0.04520551860332489
loss: 0.03896603733301163
loss: 0.044026657938957214
loss: 0.03832508251070976
loss: 0.047818854451179504
loss: 0.05093594267964363
loss: 0.04345042258501053
loss: 0.04066222161054611
loss: 0.04045538976788521
loss: 0.03837883472442627
loss: 0.041701916605234146
loss: 0.035322874784469604
loss: 0.03823874518275261
```

```
loss: 0.03821010887622833
loss: 0.04899520426988602
loss: 0.039673782885074615
loss: 0.04459426552057266
loss: 0.04435146972537041
Epoch 12, Loss: 0.0422
preds (val): [0. 1. 1. 0. 1. 1. 1. 1. 1. 0. 1. 0. 0. 0. 1. 1.]
labels (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
preds (val): [1. 1. 0. 0. 1. 1. 0. 1. 0. 0. 0. 0. 0. 1. 1. 1.]
labels (val): [1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 1. 1. 0. 0. 1. 0. 0. 0. 1. 0. 1. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 1. 1. 0. 1. 0. 1. 1. 1. 0. 0. 0. 0. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 0. 1. 0. 0. 0. 0. 1. 0. 1. 1. 0. 1. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 0. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 1. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 0. 0. 0. 0. 0. 1. 1. 0. 0. 1. 0. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 0. 1. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 0. 0. 0. 0. 0. 1. 0. 0. 0. 1. 0. 1. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 0. 1. 0. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Validation accuracy: 65.97%
loss: 0.03682113066315651
loss: 0.035695359110832214
loss: 0.03966609016060829
loss: 0.04311579093337059
loss: 0.044457849115133286
loss: 0.04081512615084648
loss: 0.03829460218548775
loss: 0.03958725929260254
loss: 0.04632265120744705
loss: 0.043383412063121796
loss: 0.04000889137387276
loss: 0.03972284495830536
loss: 0.05071886628866196
loss: 0.04332438483834267
loss: 0.04591420292854309
loss: 0.03723292425274849
loss: 0.04102782905101776
```

```
loss: 0.04699540138244629
loss: 0.03452787175774574
Epoch 13, Loss: 0.0415
preds (val): [0. 1. 1. 0. 1. 1. 1. 1. 1. 0. 1. 0. 0. 0. 1. 1.]
labels (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
preds (val): [1. 1. 0. 0. 1. 0. 0. 1. 0. 0. 0. 0. 0. 1. 1. 0.]
labels (val): [1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 1. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 1. 1. 0. 1. 0. 0. 1. 1. 0. 0. 0. 0. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 0. 1. 0. 0. 0. 0. 1. 0. 1. 0. 0. 1. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 1. 0. 0. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 0. 1. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 1. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 0. 1. 0. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Validation accuracy: 72.77%
loss: 0.04053866118192673
loss: 0.04007831960916519
loss: 0.039664171636104584
loss: 0.036850400269031525
loss: 0.03884793445467949
loss: 0.035477012395858765
loss: 0.046595487743616104
loss: 0.042428310960531235
loss: 0.04129050672054291
loss: 0.040984343737363815
loss: 0.03942945599555969
loss: 0.04501169174909592
loss: 0.04033627733588219
loss: 0.04660087451338768
loss: 0.04082568734884262
loss: 0.04407734051346779
loss: 0.04086275026202202
loss: 0.04138583317399025
loss: 0.04986537992954254
Epoch 14, Loss: 0.0416
```

```
preds (val): [1. 1. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 0. 0. 1. 1.]
labels (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
preds (val): [1. 1. 1. 0. 1. 1. 0. 1. 0. 0. 0. 0. 0. 1. 1. 1.]
labels (val): [1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 1. 1. 0. 0. 1. 0. 0. 0. 1. 0. 1. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 1. 1. 0. 1. 0. 1. 1. 1. 0. 0. 0. 0. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 0. 1. 0. 0. 0. 0. 1. 0. 1. 1. 0. 1. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 0. 0. 1. 0. 0. 0. 1. 1. 1. 0. 0. 1. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 1. 1. 0. 0. 0. 1. 1. 0. 0. 1. 0. 1. 1. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 1. 0. 1. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [1. 0. 1. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1. 1. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 1. 0. 1. 0. 1. 0. 1. 0. 0. 0. 0. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Validation accuracy: 62.30%
loss: 0.043654754757881165
loss: 0.03846138343214989
loss: 0.04308188706636429
loss: 0.040102556347846985
loss: 0.04585262015461922
loss: 0.03904949501156807
loss: 0.03935644030570984
loss: 0.04386238381266594
loss: 0.03984368219971657
loss: 0.0446937121450901
loss: 0.03914562612771988
loss: 0.03603755682706833
loss: 0.0456714890897274
loss: 0.04613157734274864
loss: 0.04028534144163132
loss: 0.040796007961034775
loss: 0.0351550318300724
loss: 0.042891766875982285
loss: 0.041660111397504807
Epoch 15, Loss: 0.0414
preds (val): [0. 0. 1. 0. 1. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
preds (val): [1. 1. 0. 0. 1. 0. 0. 1. 0. 0. 0. 0. 0. 1. 1. 0.]
```

```
labels (val): [1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
preds (val): [0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
labels (val): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Validation accuracy: 83.77%
```



```
Test accuracy: 77.75%
```

[30]:
```
#### When loading Obz, you may need to do:
## from obz.data_inspector.extractor import FirstOrderExtractor

# Setup OutlierDetector
from obzai.data_inspector.extractor import FirstOrderExtractor
from obzai.data_inspector.detector import GMMDetector
# Choose desired feature extractor. Chosen extractor will be used for␣
↪monitoring.
first_order_extrc = FirstOrderExtractor()

# Pass choosen extractor(s) to chosen OutlierDetector. Below we utilize outlier␣
↪detector based on Gaussian Mixture Models.
gmm_detector = GMMDetector(extractors=[first_order_extrc], n_components=3,␣
↪outlier_quantile=0.01)
# Call .fit() method with passed reference dataloader.
# Method will extract desired image features and fit outlier detection model␣
↪(in that case GMM).
gmm_detector.fit(val_loader)
```

[31]:
```
# Setup XAI Tools
## from obz.xai_tool import ...
from obzai.xai.xai_tool import CDAM, AttentionMap

# Choose desired XAI Tools
cdam_tool = CDAM(model=MODEL,
                 mode='vanilla',                      # CDAM mode
                 gradient_type="from_logits",  # Whether backpropagate␣
↪gradients from logits or probabilities.
                 gradient_reduction="average",        # Gradient reduction␣
↪method.
                 activation_type="sigmoid")           # Activation function␣
↪applied on logits. (Needed when gradients are backpropagated from␣
↪probabilities.)
# In CDAM you need to specify on which layer you want to create hooks.
cdam_tool.create_hooks(layer_name="backbone.encoder.layer.11.layernorm_before")

attention_tool = AttentionMap(model=MODEL,
                              attention_layer_id = -1,# ID of an attention␣
↪layer from which to extract attention weights
                              head = None             # ID of attention head to␣
↪choose. If None, attention scores are averaged.
                              )
```

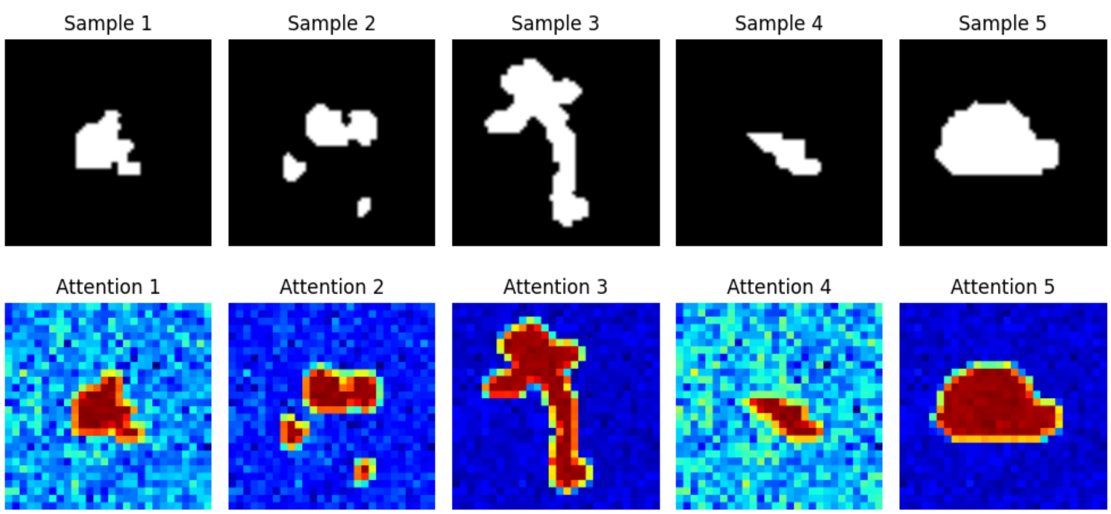
[32]:
```
samples, labels = next(iter(val_loader))
attention_maps = attention_tool.explain(samples)
```

```
# Visualize samples and attention maps
fig, axes = plt.subplots(2, 5, figsize=(10, 5))

# First row: Original samples
for i in range(5):
    original_image = samples[i].permute(1, 2, 0).cpu().numpy()  # Convert␣
 ↪tensor to numpy array and rearrange dimensions
    axes[0, i].imshow(original_image, cmap='gray')
    axes[0, i].set_title(f"Sample {i + 1}")
    axes[0, i].axis('off')

# Second row: Attention maps
for i in range(5):
    attention_map = attention_maps[i].cpu().numpy()
    axes[1, i].imshow(attention_map, cmap='jet')
    axes[1, i].set_title(f"Attention {i + 1}")
    axes[1, i].axis('off')

plt.tight_layout()
plt.show()
```



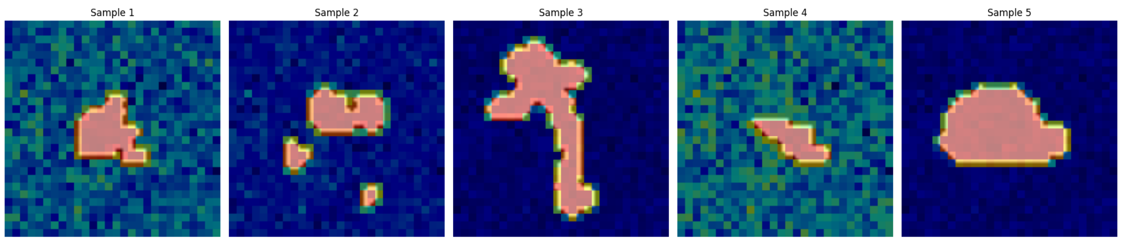
[33]:

```
fig, axes = plt.subplots(1, 5, figsize=(20, 5))

for i in range(5):
    original_image = samples[i].permute(1, 2, 0).cpu().numpy()  # Convert␣
 ↪tensor to numpy array and rearrange dimensions
    attention_map = attention_maps[i].cpu().numpy()  # Convert attention map to␣
 ↪numpy array
```

```
    axes[i].imshow(original_image, cmap='gray')
    axes[i].imshow(attention_map, cmap='jet', alpha=0.5)  # Use alpha for␣
↪transparency
    axes[i].set_title(f"Sample {i + 1}")
    axes[i].axis('off')

plt.tight_layout()
plt.show()
```



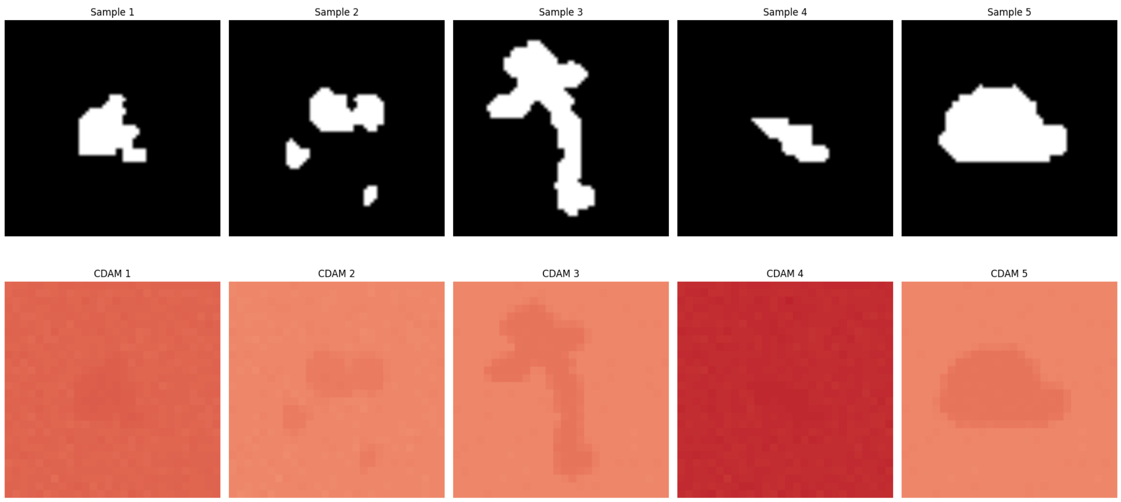
[34]:
```
cdam_maps = cdam_tool.explain(samples, target_idx=[0]*len(samples))
```

[36]:
```
fig, axes = plt.subplots(2, 5, figsize=(20, 10))

for i in range(5):
    original_image = samples[i].permute(1, 2, 0).cpu().numpy()  # Convert␣
↪tensor to numpy array and rearrange dimensions
    axes[0, i].imshow(original_image, cmap='gray')
    axes[0, i].set_title(f"Sample {i + 1}")
    axes[0, i].axis('off')

for i in range(5):
    cdam_map = cdam_maps[i].squeeze().cpu().numpy()  # Convert CDAM map to␣
↪numpy array
    axes[1, i].imshow(cdam_map, cmap='coolwarm', vmin=-cdam_maps.abs().max(),␣
↪vmax=cdam_maps.abs().max())  # Diverging colormap
    axes[1, i].set_title(f"CDAM {i + 1}")
    axes[1, i].axis('off')

plt.tight_layout()
plt.show()
```



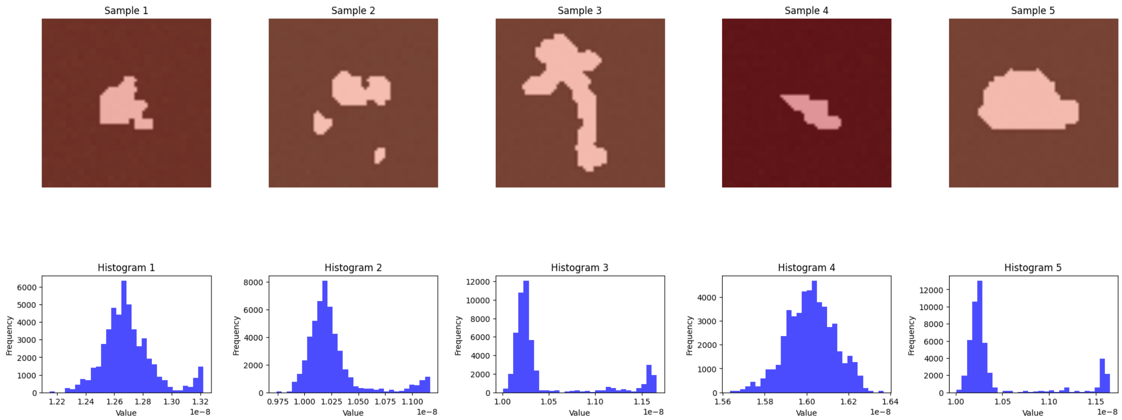
[37]:
```
fig, axes = plt.subplots(2, 5, figsize=(20, 10), gridspec_kw={'height_ratios':␣
↪[4, 1]})

for i in range(5):
    original_image = samples[i].permute(1, 2, 0).cpu().numpy()
    cdam_map = cdam_maps[i].squeeze().cpu().numpy()

    im = axes[0, i].imshow(original_image, cmap='gray')
    im = axes[0, i].imshow(cdam_map, cmap='coolwarm', alpha=0.5,␣
↪vmin=-cdam_maps.abs().max(), vmax=cdam_maps.abs().max())  # Use alpha for␣
↪transparency
    axes[0, i].set_title(f"Sample {i + 1}")
    axes[0, i].axis('off')

for i in range(5):
    cdam_map = cdam_maps[i].squeeze().cpu().numpy()
    axes[1, i].hist(cdam_map.ravel(), bins=30, color='blue', alpha=0.7)
    axes[1, i].set_title(f"Histogram {i + 1}")
    axes[1, i].set_xlabel('Value')
    axes[1, i].set_ylabel('Frequency')

plt.tight_layout()
plt.show()
```


Sample 1
Sample 2
Sample 3
Sample 4
Sample 5
Histogram 1
Histogram 2
Histogram 3
Histogram 4
Histogram 5
Frequency
Value
1e−8