conv-recur_genre

March 25, 2025

[1]:
```
import torch
import pandas as pd
import numpy as np
from tqdm import tqdm
import wandb
```

[2]:
```
wandb.init(entity="ameyar3103-iiit-hyderabad",project="recurrent_conv_art",␣
 ↪config={
    "epochs": 20,
    "batch_size": 64,
    "learning_rate": 0.001,
    "model": "RecurrentCNN"
})
```

```
wandb: Using wandb-core as the SDK backend.  Please refer to
https://wandb.me/wandb-core for more information.
wandb: Currently logged in as: ameyar3103
(ameyar3103-iiit-hyderabad) to https://api.wandb.ai. Use
`wandb login --relogin` to force relogin
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

[2]:
```
<wandb.sdk.wandb_run.Run at 0x744d1b98cf40>
```

0.1 Data loading

[3]:
```
df_train = pd.read_csv('wikiart_csv/genre_train.csv',header=None,␣
 ↪names=["image_path", "genre_id"])
df_val = pd.read_csv('wikiart_csv/genre_val.csv',header=None,␣
 ↪names=["image_path", "genre_id"])
```

[4]:
```
# get the number of classes
num_classes = 10 # from genre_class.txt
```

[5]:
```
# Gather input data
train_images = df_train['image_path'].values
train_labels = df_train['genre_id'].values

val_images = df_val['image_path'].values
val_labels = df_val['genre_id'].values
```

[6]:
```
from torchvision import transforms
import cv2
```

0.2 Preprocess data and create test and train dataset

[7]:
```
# create test and train dataset for dataloader

def get_image(image_path,image_size=224):
    try:
        img = cv2.imread('./wikiart/' + image_path)
        if img is None:
            raise ValueError(f"Image not loaded: ./wikiart/{image_path}")
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        h, w, _ = img.shape
        scale = 256 / min(h, w)
        new_w = int(w * scale)
        new_h = int(h * scale)
        img_resized = cv2.resize(img, (new_w, new_h))
        start_x = (new_w - image_size) // 2
        start_y = (new_h - image_size) // 2
        img_cropped = img_resized[start_y:start_y+image_size, start_x:
 ↪start_x+image_size]
        img_cropped = img_cropped.astype(np.float32) / 255.0
        img_tensor = torch.from_numpy(img_cropped).permute(2, 0, 1)
        mean = torch.tensor([0.485, 0.456, 0.406]).view(3, 1, 1)
        std  = torch.tensor([0.229, 0.224, 0.225]).view(3, 1, 1)
        img_tensor = (img_tensor - mean) / std
        return img_tensor
    except Exception as e:
        print(f"Error processing {image_path}: {e}")
        return torch.zeros(3, image_size, image_size)

class WikiArtDataset(torch.utils.data.Dataset):
    def __init__(self, images, labels):
        self.images = images
        self.labels = labels
```

```
    def __len__(self):
        return len(self.images)

    def __getitem__(self, idx):
        # image_vectors = []
        # for image in self.images:
        #     image_emb = get_image(image)
        #     image_vectors.append(image_emb)
        # image = torch.stack(image_vectors)
        image = self.images[idx]
        # label should be a one-hot encoded vector
        label = torch.zeros(num_classes)
        label[self.labels[idx]] = 1

        return image, label

train_dataset = WikiArtDataset(train_images, train_labels)
train_loader = torch.utils.data.DataLoader(train_dataset, batch_size=64,␣
 ↪shuffle=True)
val_dataset = WikiArtDataset(val_images, val_labels)
val_loader = torch.utils.data.DataLoader(val_dataset, batch_size=64,␣
 ↪shuffle=False)

for i, (images, labels) in enumerate(train_loader):
    print(images)
    print(labels)
    break
```

```
('Impressionism/robert-julian-onderdonk_goat-herder-at-the-san-antonio-
quarry-1909.jpg', 'Impressionism/arthur-verona_neagoe-basarab-study.jpg',
'Early_Renaissance/paolo-uccello_st-francis.jpg',
'High_Renaissance/michelangelo_the-ancestors-of-christ-manasseh-amon-1512.jpg',
'Baroque/adriaen-brouwer_inn-with-drunken-peasants.jpg', 'Realism/vincent-van-
gogh_farmhouses-in-loosduinen-near-the-hague-at-twilight-1883(1).jpg',
'Romanticism/jan-matejko_jadwiga.jpg', 'Post_Impressionism/bertalan-
por_brookside-1919.jpg', 'Realism/ivan-shishkin_fir.jpg',
'Impressionism/camille-pissarro_landscape-with-a-man-digging-1877.jpg',
'Romanticism/dante-gabriel-rossetti_study-for-a-vision-of-fiammetta.jpg',
'Expressionism/martiros-saryan_gohtan-mountains-1914.jpg', 'Realism/vasily-
vereshchagin_parsi-priest-fire-worshiper-bombay-1876.jpg', 'Realism/klavdy-
lebedev_spat-on-the-terrace.jpg', 'Impressionism/pierre-auguste-renoir_young-
girl-in-a-flowered-hat-1905.jpg', 'Realism/johan-hendrik-weissenbruch_figures-
on-a-country-road-a-church-in-the-distance.jpg', 'Minimalism/genevieve-
asse_blue-depth-1978.jpg', 'High_Renaissance/vittore-carpaccio_the-glory-of-st-
vidal-1514.jpg', 'Cubism/paul-klee_senecio-1922(1).jpg', 'Cubism/georges-
braque_big-nude-1908.jpg', 'Art_Nouveau_Modern/ivan-bilibin_illustration-for-
the-poem-the-tale-of-the-golden-cockerel-by-alexander-pushkin-1906-2(1).jpg',
```

```
'Romanticism/homer-watson_old-mill-and-stream-1879.jpg', 'Romanticism/gustave-
dore_don-quixote-58.jpg', 'Realism/charles-francois-daubigny_castle-gaillard-in-
andelys-eure-1877.jpg', 'Impressionism/nikolay-bogdanov-belsky_children.jpg',
'Romanticism/gustave-dore_don-quixote-55.jpg', 'Color_Field_Painting/theodoros-
stamos_olympia-sun-box-1957.jpg', 'Realism/vasily-polenov_right-hand-keeping-
the-staff.jpg', 'Mannerism_Late_Renaissance/correggio_the-assumption-of-the-
virgin-detail-1530(3).jpg', 'Impressionism/william-merritt-chase_peonies.jpg',
'High_Renaissance/andrea-del-sarto_portrait-of-baccio-bandinelli.jpg',
'New_Realism/george-luks_the-wrestlers-1905.jpg', 'Pop_Art/edward-
ruscha_standard-station-1966.jpg', 'Northern_Renaissance/mabuse_madonna-and-
child-playing-with-the-veil.jpg', 'Impressionism/william-james-glackens_lenna-
painting-the-artist-s-daughter-1918.jpg', 'Expressionism/pablo-picasso_portrait-
of-madame-patri-1918.jpg', 'Symbolism/nicholas-roerich_monhigan-
study-1922-14.jpg', 'Realism/ilya-repin_in-the-hut-1895.jpg',
'Post_Impressionism/pierre-bonnard_view-of-le-cannet-roofs-1942.jpg',
'Symbolism/mstislav-dobuzhinsky_vilno-street-1906(1).jpg',
'Impressionism/berthe-morisot_the-sewing-lesson-aka-the-artist-s-daughter-julie-
with-her-nanny.jpg', 'Impressionism/ipolit-strambu_nude-with-carpet-
background-1921.jpg', 'Post_Impressionism/salvador-dali_the-garden-at-
lyane.jpg', 'Realism/pyotr-konchalovsky_conductor-nikolai-semenovich-golovanov-
and-orchestra-1934.jpg', 'Art_Nouveau_Modern/boris-kustodiev_illustration-for-
nikolay-nekrasov-poem-uncle-jacob-1921.jpg', 'Realism/vincent-van-gogh_peasant-
woman-with-shawl-over-her-head-seen-from-the-side-2-1885.jpg',
'Expressionism/lucian-freud_girl-in-a-fur-coat-1967.jpg', 'Fauvism/maurice-de-
vlaminck_sailboats-at-chatou.jpg', 'Post_Impressionism/vincent-van-gogh_sheet-
with-sketches-of-working-people-1890-1.jpg', 'Post_Impressionism/gustave-
loiseau_trees-in-bloom.jpg', 'Rococo/vladimir-borovikovsky_portrait-of-the-
artist-dmitry-levitzky-1796.jpg', 'High_Renaissance/giovanni-bellini_st-jerome-
st-christopher-and-st-augustine-1513.jpg', 'Expressionism/dimitris-
mytaras_female-figures.jpg', 'Impressionism/eugene-boudin_the-beach-at-
trouville.jpg', 'Realism/robert-brackman_a-plate-of-fruit.jpg',
'Impressionism/joaquã\xadn-sorolla_chumberas-study.jpg', 'Romanticism/jean-leon-
gerome_a-japanese-imploring-a-divinity.jpg', 'Realism/gustave-courbet_the-
hallali-1869.jpg', 'Realism/gustave-courbet_the-source-of-the-loue-1864.jpg',
'Baroque/joseph-wright_mrs-john-ashton.jpg', 'Abstract_Expressionism/sam-
francis_pioggia-d-oro-golden-rain-1988.jpg', 'Expressionism/martiros-
saryan_landscape-with-mountains-1929.jpg', 'Art_Nouveau_Modern/alphonse-
mucha_maude-adams-as-joan-of-arc-1909.jpg', 'Impressionism/fern-coppedge_old-
house-point-pleasant.jpg')
tensor([[0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 1., 0.],
        [0., 0., 0., 0., 0., 0., 0., 1., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 1., 0., 0.],
        [0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
        [0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 1., 0.],
```

```
[0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
[0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
[0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
[0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
[0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
[0., 0., 0., 0., 0., 0., 0., 0., 1., 0.],
[0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
[1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
[0., 0., 0., 0., 0., 0., 0., 1., 0., 0.],
[0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
[0., 0., 0., 0., 0., 1., 0., 0., 0., 0.],
[0., 0., 0., 1., 0., 0., 0., 0., 0., 0.],
[0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
[0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
[0., 0., 0., 1., 0., 0., 0., 0., 0., 0.],
[1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
[0., 0., 0., 0., 0., 0., 0., 0., 1., 0.],
[0., 0., 0., 0., 0., 0., 0., 1., 0., 0.],
[0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
[0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
[0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
[0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
[0., 0., 0., 0., 0., 0., 0., 1., 0., 0.],
[0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
[0., 0., 0., 0., 0., 0., 0., 0., 1., 0.],
[0., 0., 0., 0., 0., 0., 0., 0., 1., 0.],
[0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
[0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
[0., 1., 0., 0., 0., 0., 0., 0., 0., 0.],
[0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
[0., 0., 0., 0., 0., 1., 0., 0., 0., 0.],
[0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
[0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
[0., 0., 0., 1., 0., 0., 0., 0., 0., 0.],
[0., 0., 0., 0., 0., 0., 0., 0., 1., 0.],
[0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
[0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
[0., 0., 0., 0., 0., 0., 0., 0., 1., 0.],
[0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
[0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
[0., 0., 0., 0., 0., 0., 0., 1., 0., 0.],
[0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
[0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
[0., 0., 0., 0., 0., 0., 0., 0., 0., 1.],
[0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
[0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
```

```
        [0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
        [1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 1., 0.],
        [0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
        [0., 0., 0., 0., 1., 0., 0., 0., 0., 0.]])
```

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

class RecurrentCNN(nn.Module):
    def __init__(self, num_classes, lstm_hidden_size=256, dropout_prob=0.5):
        super(RecurrentCNN, self).__init__()
        self.conv1 = nn.Conv2d(3, 32, kernel_size=3, stride=1, padding=1)
        self.pool1 = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3, stride=1, padding=1)
        self.pool2 = nn.MaxPool2d(2, 2)
        self.adaptive_pool = nn.AdaptiveAvgPool2d((14, 56))
        self.lstm_input_size = 64 * 56
        self.lstm_hidden_size = lstm_hidden_size
        self.lstm = nn.LSTM(input_size=self.lstm_input_size,␣
 ↪hidden_size=lstm_hidden_size,
                            batch_first=True, bidirectional=True)
        self.dropout = nn.Dropout(dropout_prob)
        self.fc = nn.Linear(2 * lstm_hidden_size, num_classes)

    def forward(self, x):
        x = F.relu(self.conv1(x))
        x = self.pool1(x)
        x = F.relu(self.conv2(x))
        x = self.pool2(x)
        x = self.adaptive_pool(x)
        x = x.permute(0, 2, 1, 3).contiguous()
        batch_size, seq_len, channels, width = x.shape
        x = x.view(batch_size, seq_len, channels * width)
        lstm_out, _ = self.lstm(x)
        x = lstm_out.mean(dim=1)
        x = self.dropout(x)
        x = self.fc(x)
        return x

model = RecurrentCNN(num_classes)
model.to('cuda')

# Loss and optimizer
import torch.optim as optim
```

```
wandb.watch(model, log="all")
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)
```

0.3 Training the model

[9]:
```
# Train the model
num_epochs = 20

for epoch in range(num_epochs):
    model.train()
    running_loss = 0.0
    train_bar = tqdm(train_loader, desc=f"Epoch {epoch+1}/{num_epochs}")
    for image_paths, labels in train_bar:
        image_tensors = torch.stack([get_image(image_path) for image_path in␣
 ↪image_paths])
        images = image_tensors.to('cuda')
        labels = labels.to('cuda')

        # Forward pass
        outputs = model(images)
        loss = criterion(outputs, labels)

        # Backward and optimize
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        running_loss += loss.item()
        train_bar.set_postfix(loss=loss.item())

    avg_train_loss = running_loss / len(train_loader)
    wandb.log({"epoch": epoch+1, "train_loss": avg_train_loss})

    # Validation Loop
    model.eval()
    val_loss = 0.0
    correct = 0
    total = 0
    with torch.no_grad():
        val_bar = tqdm(val_loader, desc="Validation")
        for image_paths, labels in val_bar:
            image_tensors = torch.stack([get_image(image_path) for image_path␣
 ↪in image_paths])
            image_tensors = image_tensors.to('cuda')
```

```
            labels = labels.to('cuda')
            outputs = model(image_tensors)
            loss = criterion(outputs, labels)
            val_loss += loss.item()
            _, predicted = torch.max(outputs.data, 1)
            total += labels.size(0)
            correct += (predicted == labels.argmax(dim=1)).sum().item()
            val_bar.set_postfix(loss=loss.item())

    avg_val_loss = val_loss / len(val_loader)
    val_accuracy = 100 * correct / total
    wandb.log({"val_loss": avg_val_loss, "val_accuracy": val_accuracy})
    print(f"Epoch {epoch+1}/{num_epochs} - Train Loss: {avg_train_loss:.4f},␣
↪Val Loss: {avg_val_loss:.4f}, Val Accuracy: {val_accuracy:.2f}%")
    if(epoch%5==0):
        torch.save(model.state_dict(), f"recurrent_cnn_epoch_{epoch}_genre.pth")
        torch.save(optimizer.state_dict(),␣
↪f"recurrent_cnn_optimizer_epoch_{epoch}_genre.pth")
```

```
Epoch 1/20:  53%|       | 374/711 [06:07<05:21,  1.05it/s, loss=1.49]Corrupt
JPEG data: premature end of data segment
Epoch 1/20:  98%|      | 696/711 [11:19<00:12,  1.16it/s, loss=1.68]Corrupt
JPEG data: bad Huffman code
Epoch 1/20: 100%|      | 711/711 [11:34<00:00,  1.02it/s, loss=1.7]
Validation: 100%|      | 305/305 [04:29<00:00,  1.13it/s, loss=1.2]

Epoch 1/20 - Train Loss: 1.6431, Val Loss: 1.4642, Val Accuracy: 49.14%

Epoch 2/20:  62%|       | 438/711 [07:15<03:51,  1.18it/s, loss=1.45]Corrupt
JPEG data: premature end of data segment
Epoch 2/20:  72%|       | 512/711 [08:25<03:08,  1.06it/s, loss=1.74]Corrupt
JPEG data: bad Huffman code
Epoch 2/20: 100%|      | 711/711 [11:41<00:00,  1.01it/s, loss=1.34]
Validation: 100%|      | 305/305 [04:35<00:00,  1.11it/s, loss=0.956]

Epoch 2/20 - Train Loss: 1.4466, Val Loss: 1.4048, Val Accuracy: 50.63%

Epoch 3/20:  26%|         | 185/711 [03:08<08:27,  1.04it/s, loss=1.22]Corrupt
JPEG data: bad Huffman code
Epoch 3/20:  54%|       | 385/711 [06:19<04:57,  1.10it/s, loss=1.48]Corrupt
JPEG data: premature end of data segment
Epoch 3/20: 100%|      | 711/711 [11:28<00:00,  1.03it/s, loss=1.43]
Validation: 100%|      | 305/305 [04:28<00:00,  1.14it/s, loss=1.27]

Epoch 3/20 - Train Loss: 1.3456, Val Loss: 1.3164, Val Accuracy: 54.04%

Epoch 4/20:  65%|       | 463/711 [07:16<04:06,  1.01it/s, loss=1.2]  Corrupt
JPEG data: bad Huffman code
Epoch 4/20:  83%|      | 591/711 [09:19<01:47,  1.11it/s, loss=1.46] Corrupt
JPEG data: premature end of data segment
```

```
Epoch 4/20: 100%|      | 711/711 [11:16<00:00,  1.05it/s, loss=1.2]
Validation: 100%|      | 305/305 [04:26<00:00,  1.14it/s, loss=1.36]

Epoch 4/20 - Train Loss: 1.2404, Val Loss: 1.2897, Val Accuracy: 55.05%

Epoch 5/20:  31%|         | 217/711 [03:24<08:41,  1.06s/it, loss=0.926]Corrupt
JPEG data: bad Huffman code
Epoch 5/20:  47%|        | 336/711 [05:20<05:55,  1.06it/s, loss=1.44] Corrupt
JPEG data: premature end of data segment
Epoch 5/20: 100%|      | 711/711 [11:17<00:00,  1.05it/s, loss=1.04]
Validation: 100%|      | 305/305 [04:28<00:00,  1.14it/s, loss=1.28]

Epoch 5/20 - Train Loss: 1.1248, Val Loss: 1.2582, Val Accuracy: 56.26%

Epoch 6/20:  31%|         | 218/711 [03:27<07:56,  1.03it/s, loss=0.926]Corrupt
JPEG data: premature end of data segment
Epoch 6/20:  35%|         | 246/711 [03:54<07:28,  1.04it/s, loss=0.839]Corrupt
JPEG data: bad Huffman code
Epoch 6/20: 100%|      | 711/711 [11:14<00:00,  1.05it/s, loss=1.03]
Validation: 100%|      | 305/305 [04:26<00:00,  1.15it/s, loss=1.63]

Epoch 6/20 - Train Loss: 0.9760, Val Loss: 1.2952, Val Accuracy: 56.36%

Epoch 7/20:  43%|         | 305/711 [04:49<06:21,  1.06it/s, loss=0.529]Corrupt
JPEG data: bad Huffman code
Epoch 7/20:  56%|        | 399/711 [06:17<04:40,  1.11it/s, loss=0.841]Corrupt
JPEG data: premature end of data segment
Epoch 7/20: 100%|      | 711/711 [11:12<00:00,  1.06it/s, loss=0.709]
Validation: 100%|      | 305/305 [04:29<00:00,  1.13it/s, loss=1.55]

Epoch 7/20 - Train Loss: 0.7983, Val Loss: 1.3486, Val Accuracy: 56.16%

Epoch 8/20:   0%|          | 3/711 [00:03<12:03,  1.02s/it, loss=0.813]Corrupt
JPEG data: premature end of data segment
Epoch 8/20:  38%|         | 269/711 [04:20<07:08,  1.03it/s, loss=0.511]
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Cell In[9], line 9
      7 train_bar = tqdm(train_loader, desc=f"Epoch {epoch+1}/{num_epochs}")
      8 for image_paths, labels in train_bar:
----> 9     image_tensors = torch.stack([get_image(image_path) for image_path i
 ↪image_paths])
     10     images = image_tensors.to('cuda')
     11     labels = labels.to('cuda')

Cell In[9], line 9, in <listcomp>(.0)
      7 train_bar = tqdm(train_loader, desc=f"Epoch {epoch+1}/{num_epochs}")
      8 for image_paths, labels in train_bar:
----> 9     image_tensors = torch.stack([get_image(image_path) for image_path i
 ↪image_paths])
```

```
     10     images = image_tensors.to('cuda')
     11     labels = labels.to('cuda')

Cell In[7], line 5, in get_image(image_path, image_size)
      3 def get_image(image_path,image_size=224):
      4     try:
----> 5         img = cv2.imread('./wikiart/' + image_path)
      6         if img is None:
      7             raise ValueError(f"Image not loaded: ./wikiart/{image_path}")

KeyboardInterrupt:
```

Error in callback <bound method _WandbInit._pause_backend of <wandb.sdk.wandb_init._WandbInit object at 0x744d1bc5e4d0>> (for post_run_cell):

```
---------------------------------------------------------------------------
BrokenPipeError                           Traceback (most recent call last)
File ~/.local/lib/python3.10/site-packages/wandb/sdk/wandb_init.py:565, in␣
 ↪_WandbInit._pause_backend(self, *args, **kwargs)
    563 if self.backend.interface is not None:
    564     self._logger.info("pausing backend")  # type: ignore
--> 565     self.backend.interface.publish_pause()

File ~/.local/lib/python3.10/site-packages/wandb/sdk/interface/interface.py:769␣
 ↪in InterfaceBase.publish_pause(self)
    767 def publish_pause(self) -> None:
    768     pause = pb.PauseRequest()
--> 769     self._publish_pause(pause)

File ~/.local/lib/python3.10/site-packages/wandb/sdk/interface/interface_shared
 ↪py:289, in InterfaceShared._publish_pause(self, pause)
    287 def _publish_pause(self, pause: pb.PauseRequest) -> None:
    288     rec = self._make_request(pause=pause)
--> 289     self._publish(rec)

File ~/.local/lib/python3.10/site-packages/wandb/sdk/interface/interface_sock.p
 ↪39, in InterfaceSock._publish(self, record, local)
     37 def _publish(self, record: "pb.Record", local: Optional[bool] = None) -
 ↪None:
     38     self._assign(record)
---> 39     self._sock_client.send_record_publish(record)

File ~/.local/lib/python3.10/site-packages/wandb/sdk/lib/sock_client.py:174, in
 ↪SockClient.send_record_publish(self, record)
    172 server_req.request_id = record.control.mailbox_slot
    173 server_req.record_publish.CopyFrom(record)
--> 174 self.send_server_request(server_req)
```

```
File ~/.local/lib/python3.10/site-packages/wandb/sdk/lib/sock_client.py:154, in
 ↪SockClient.send_server_request(self, msg)
    153 def send_server_request(self, msg: spb.ServerRequest) -> None:
--> 154     self._send_message(msg)

File ~/.local/lib/python3.10/site-packages/wandb/sdk/lib/sock_client.py:151, in
 ↪SockClient._send_message(self, msg)
    149 header = struct.pack("<BI", ord("W"), raw_size)
    150 with self._lock:
--> 151     self._sendall_with_error_handle(header + data)

File ~/.local/lib/python3.10/site-packages/wandb/sdk/lib/sock_client.py:130, in
 ↪SockClient._sendall_with_error_handle(self, data)
    128 start_time = time.monotonic()
    129 try:
--> 130     sent = self._sock.send(data)
    131     # sent equal to 0 indicates a closed socket
    132     if sent == 0:

BrokenPipeError: [Errno 32] Broken pipe
```