conv-recur_style

March 25, 2025

[10]:
```
import torch
import pandas as pd
import numpy as np
from tqdm import tqdm
import wandb
```

[11]:
```
wandb.init(entity="ameyar3103-iiit-hyderabad",project="recurrent_conv_art",␣
 ↪config={
    "epochs": 20,
    "batch_size": 64,
    "learning_rate": 0.001,
    "model": "RecurrentCNN"
})
```

[11]:
```
<wandb.sdk.wandb_run.Run at 0x7f62307988e0>
```

0.1 Data loading

[12]:
```
df_train = pd.read_csv('wikiart_csv/style_train.csv',header=None,␣
 ↪names=["image_path", "style_id"])
df_val = pd.read_csv('wikiart_csv/style_val.csv',header=None,␣
 ↪names=["image_path", "style_id"])
```

[13]:
```
# get the number of classes
num_classes = 27 # from style_class.txt
```

[14]:
```
# Gather input data
train_images = df_train['image_path'].values
train_labels = df_train['style_id'].values

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

[15]:
```
from torchvision import transforms
import cv2
```

0.2 Preprocess data and create test and train dataset

[16]:
```
# 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
```

```
('Realism/ivan-shishkin_dark-forest-1890.jpg', 'Post_Impressionism/paul-
cezanne_portrait-of-madame-cezanne-in-a-red-dress.jpg',
'Post_Impressionism/maurice-prendergast_blue-mountains.jpg',
'Expressionism/m.c.-escher_not_detected_204655.jpg', 'Realism/edouard-
manet_head-of-jean-baptiste-faure.jpg', 'Art_Nouveau_Modern/alexandre-
benois_italian-comedy(1).jpg', 'Realism/vincent-van-gogh_sien-with-child-on-her-
lap-1882.jpg', 'Impressionism/claude-monet_the-seine-at-lavacourt-1880.jpg',
'Rococo/antoine-watteau_italian-comedians.jpg', 'Symbolism/william-blake_the-
goblin-1820.jpg', 'Expressionism/amedeo-modigliani_young-brunette-girl-
sitting-1918.jpg', 'Early_Renaissance/masaccio_san-giovenale-triptych-left-
panel.jpg', 'Fauvism/joan-miro_portrait-of-juanita-obrador.jpg',
'Symbolism/nicholas-roerich_himalayas-35.jpg', 'Post_Impressionism/spyros-
papaloukas_at-mount-athos.jpg', 'Baroque/rembrandt_peter-and-john-at-the-gate-
of-the-temple-1629.jpg', 'Symbolism/kuzma-petrov-vodkin_mother-1913.jpg',
'Romanticism/gustave-dore_the-punishment-of-the-simonists.jpg',
'Impressionism/marianne-north_gate-of-rajah-s-palace-benares-india-1880.jpg',
'Ukiyo_e/utagawa-toyokuni-ii_courtesan-choto-with-two-kamuro-young-attendants-
behind-her.jpg', 'Impressionism/claude-monet_snow-effect-with-setting-
sun-1875.jpg', 'Post_Impressionism/le-pho_flower-composition.jpg',
'Northern_Renaissance/joachim-wtewael_aphrodite-ares-and-eros-sun.jpg',
'Impressionism/max-liebermann_potato-gatherers.jpg',
'Naive_Art_Primitivism/niko-pirosmani_white-cow-on-a-black-background.jpg',
'Impressionism/augustus-john_vera-fearing-1931.jpg', 'Impressionism/konstantin-
makovsky_children-playing-in-the-workshop.jpg', 'Impressionism/claude-monet_the-
road-to-the-farm-of-saint-simeon.jpg', 'High_Renaissance/lorenzo-lotto_st-
jerome-in-the-desert.jpg', 'Realism/peder-severin-kroyer_the-benzon-
daughters-1897.jpg', 'Ukiyo_e/tsukioka-yoshitoshi_oda-nobunaga-fighting-with-
another-warrior-whom-he-knocks-off-a-building-into-a-raging-inferno.jpg',
'Romanticism/gustave-dore_don-quixote-81.jpg', 'Impressionism/eugene-boudin_the-
somme-near-d-abbeville-moonlight-1894.jpg', 'Realism/edgar-degas_portrait-of-
james-tissot-1868.jpg', 'Cubism/willi-baumeister_painter-with-palette-1933.jpg',
'Impressionism/claude-monet_water-lilies-16.jpg', 'Realism/thomas-
```

```
eakins_studies-for-william-rush-1876-7.jpg', 'Impressionism/edgar-degas_dance-
rehearsal-in-the-studio-of-the-opera-1895.jpg', 'Early_Renaissance/giovanni-
bellini_madonna-and-child-2.jpg', 'Impressionism/konstantin-korovin_still-life-
with-blue-vase-1922.jpg', 'Impressionism/claude-monet_rouen-cathedral-clear-
day.jpg', 'Symbolism/harry-clarke_tales-of-mystery-and-imagination-by-edgar-
allan-poe-1923-8.jpg', 'Expressionism/jean-david_le-menage-hereux.jpg',
'Expressionism/georges-braque_still-life-with-clarinet-1927.jpg',
'Impressionism/gregoire-boonzaier_tabletop-still-life-1939.jpg',
'Synthetic_Cubism/georges-braque_still-life-on-a-table-with-gillette-1914.jpg',
'Rococo/allan-ramsay_self-portrait.jpg', 'Northern_Renaissance/albrecht-
durer_saint-john-s-church-1489.jpg', 'Post_Impressionism/maxime-
maufra_landscape-1.jpg', 'Mannerism_Late_Renaissance/andrea-del-
sarto_assumption-of-the-virgin-1529.jpg', 'Romanticism/orest-kiprensky_john-the-
baptist-baptizing-people-1819.jpg', 'Abstract_Expressionism/friedel-
dzubas_untitled-77-1954.jpg', 'Impressionism/james-tissot_portrait-of-a-lady-
with-a-fan.jpg', 'Northern_Renaissance/hans-baldung_castle-weibertreu-1515.jpg',
'Impressionism/federico-zandomeneghi_the-good-book-1897.jpg',
'Expressionism/henri-matisse_red-fish-in-interior.jpg',
'Post_Impressionism/moise-kisling_sitting-nude-1930.jpg', 'Realism/vincent-van-
gogh_weed-burner-sitting-on-a-wheelbarrow-with-his-wife-1883.jpg',
'Romanticism/henry-raeburn_portrait-of-mrs-andrew.jpg', 'Expressionism/paula-
modersohn-becker_cowshed-1901.jpg', 'Post_Impressionism/ilya-mashkov_berries-on-
the-background-of-a-red-tray-1908.jpg', 'Baroque/diego-velazquez_queen-mariana-
of-austria-1653.jpg', 'Realism/john-singer-sargent_mrs-charles-e-inches-louise-
pomeroy-1887.jpg', 'Rococo/antoine-watteau_the-harlekin.jpg')
tensor([[0., 0., 0.,  …, 0., 0., 0.],
        [0., 0., 0.,  …, 0., 0., 0.],
        [0., 0., 0.,  …, 0., 0., 0.],
        …,
        [0., 0., 0.,  …, 0., 0., 0.],
        [0., 0., 0.,  …, 0., 0., 0.],
        [0., 0., 0.,  …, 0., 0., 0.]])
```

[ ]:
```python
# CNN model

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

[ ]:
```
# 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}_style.pth")
        torch.save(optimizer.state_dict(),␣
↪f"recurrent_cnn_optimizer_epoch_{epoch}_style.pth")
```

```
Epoch 1/20:   3%|          | 29/892 [00:29<15:27,  1.07s/it, loss=2.74]Corrupt
```

```
JPEG data: premature end of data segment
Epoch 1/20:  17%|         | 151/892 [02:34<12:35,  1.02s/it, loss=2.74]Corrupt
JPEG data: bad Huffman code
Epoch 1/20: 100%|      | 892/892 [15:02<00:00,  1.01s/it, loss=2.11]
Validation: 100%|      | 382/382 [05:41<00:00,  1.12it/s, loss=3.72]

Epoch 1/20 - Train Loss: 2.5116, Val Loss: 2.3391, Val Accuracy: 26.43%

Epoch 2/20:  83%|      | 744/892 [11:59<02:32,  1.03s/it, loss=2.22]Corrupt
JPEG data: bad Huffman code
Epoch 2/20:  98%|      | 873/892 [14:04<00:19,  1.00s/it, loss=2.1] Corrupt
JPEG data: premature end of data segment
Epoch 2/20: 100%|      | 892/892 [14:22<00:00,  1.03it/s, loss=6.84]
Validation: 100%|      | 382/382 [05:39<00:00,  1.13it/s, loss=2.83]

Epoch 2/20 - Train Loss: 2.2795, Val Loss: 2.2328, Val Accuracy: 29.52%

Epoch 3/20:  72%|      | 639/892 [10:06<03:45,  1.12it/s, loss=2.19]Corrupt
JPEG data: bad Huffman code
Epoch 3/20:  85%|      | 762/892 [12:07<02:30,  1.16s/it, loss=1.98]Corrupt
JPEG data: premature end of data segment
Epoch 3/20: 100%|      | 892/892 [14:15<00:00,  1.04it/s, loss=0.662]
Validation: 100%|      | 382/382 [05:39<00:00,  1.12it/s, loss=3.09]

Epoch 3/20 - Train Loss: 2.1506, Val Loss: 2.1226, Val Accuracy: 32.30%

Epoch 4/20:   0%|          | 4/892 [00:03<13:10,  1.12it/s, loss=1.96]Corrupt
JPEG data: premature end of data segment
Epoch 4/20:  68%|       | 605/892 [10:01<04:38,  1.03it/s, loss=1.9] Corrupt
JPEG data: bad Huffman code
Epoch 4/20: 100%|      | 892/892 [14:56<00:00,  1.01s/it, loss=2.31]
Validation: 100%|      | 382/382 [05:58<00:00,  1.07it/s, loss=3.31]

Epoch 4/20 - Train Loss: 2.0469, Val Loss: 2.0648, Val Accuracy: 33.97%

Epoch 5/20:  22%|         | 195/892 [03:18<10:41,  1.09it/s, loss=1.99]Corrupt
JPEG data: bad Huffman code
Epoch 5/20:  67%|       | 597/892 [10:17<07:24,  1.51s/it, loss=1.73]Corrupt
JPEG data: premature end of data segment
Epoch 5/20: 100%|      | 892/892 [15:47<00:00,  1.06s/it, loss=1.92]
Validation: 100%|      | 382/382 [06:34<00:00,  1.03s/it, loss=3.41]

Epoch 5/20 - Train Loss: 1.9385, Val Loss: 2.0295, Val Accuracy: 35.41%

Epoch 6/20:  17%|         | 150/892 [02:47<16:49,  1.36s/it, loss=1.53]Corrupt
JPEG data: bad Huffman code
Epoch 6/20:  79%|      | 708/892 [12:04<02:56,  1.04it/s, loss=1.86]Corrupt
JPEG data: premature end of data segment
Epoch 6/20: 100%|      | 892/892 [14:57<00:00,  1.01s/it, loss=1.44]
Validation: 100%|      | 382/382 [05:36<00:00,  1.14it/s, loss=3.09]

Epoch 6/20 - Train Loss: 1.8092, Val Loss: 2.0187, Val Accuracy: 36.18%
```

```
Epoch 7/20:  57%|       | 511/892 [08:08<06:00,  1.06it/s, loss=1.84]Corrupt
JPEG data: premature end of data segment
Epoch 7/20:  61%|       | 545/892 [08:40<05:53,  1.02s/it, loss=1.57]Corrupt
JPEG data: bad Huffman code
Epoch 7/20: 100%|      | 892/892 [14:11<00:00,  1.05it/s, loss=1.26]
Validation: 100%|      | 382/382 [05:49<00:00,  1.09it/s, loss=3.18]

Epoch 7/20 - Train Loss: 1.6420, Val Loss: 2.0239, Val Accuracy: 36.52%

Epoch 8/20:  62%|       | 553/892 [09:36<05:44,  1.02s/it, loss=1.42]Corrupt
JPEG data: bad Huffman code
Epoch 8/20:  97%|      | 863/892 [14:38<00:27,  1.05it/s, loss=1.64]Corrupt
JPEG data: premature end of data segment
Epoch 8/20: 100%|      | 892/892 [15:08<00:00,  1.02s/it, loss=1.28]
Validation: 100%|      | 382/382 [05:48<00:00,  1.10it/s, loss=3.17]

Epoch 8/20 - Train Loss: 1.4489, Val Loss: 2.0797, Val Accuracy: 36.31%

Epoch 9/20:  34%|       | 303/892 [04:59<09:42,  1.01it/s, loss=1.16]
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Cell In[18], 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[18], 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[16], 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 0x7f6230a66500>> (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
```

```
The Kernel crashed while executing code in the current cell or a previous cell.


Please review the code in the cell(s) to identify a possible cause of the␣
 ↪failure.

Click <a href='https://aka.ms/vscodeJupyterKernelCrash'>here</a> for more info.


View Jupyter <a href='command:jupyter.viewOutput'>log</a> for further details.
```