PyTorch_Project_Muhammad_Yousif

August 5, 2024

[ ]:
```
!pip install -r requirements.txt
```

[ ]:
```
!pip install xformers
```

[87]:
```
import numpy as np
import os
import os.path as osp
import pandas as pd
import torch
import torch.nn.functional as F
from tqdm.notebook import tqdm, trange
from torch.utils.data import DataLoader, random_split
from torchvision.datasets import ImageFolder
from torchvision import transforms as torch_transforms, models
from torchvision.transforms.functional import InterpolationMode
from torchvision.utils import make_grid
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, precision_score, recall_score,␣
↪f1_score, confusion_matrix
from diffusers import StableDiffusionPipeline, EulerDiscreteScheduler
```

[88]:
```
MODEL_ID = "stabilityai/stable-diffusion-2-1-base"

def get_sd_model(args):
    dtype = torch.float32
    scheduler = EulerDiscreteScheduler.from_pretrained(MODEL_ID,␣
↪subfolder="scheduler")
    pipe = StableDiffusionPipeline.from_pretrained(MODEL_ID,␣
↪scheduler=scheduler, torch_dtype=dtype)
    pipe.enable_xformers_memory_efficient_attention()
    vae = pipe.vae
    tokenizer = pipe.tokenizer
    text_encoder = pipe.text_encoder
    unet = pipe.unet
    return vae, tokenizer, text_encoder, unet, scheduler

def get_scheduler_config():
    config = {
        "_class_name": "EulerDiscreteScheduler",
        "_diffusers_version": "0.10.2",
        "beta_end": 0.012,
        "beta_schedule": "scaled_linear",
        "beta_start": 0.00085,
        "clip_sample": False,
        "num_train_timesteps": 1000,
        "prediction_type": "epsilon",
        "set_alpha_to_one": False,
        "skip_prk_steps": True,
        "steps_offset": 1,  # todo
        "trained_betas": None
    }
    return config
```

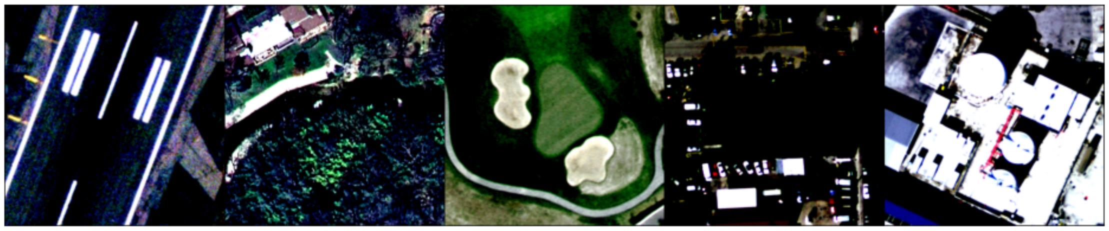
```
[ ]: device = "cuda" if torch.cuda.is_available() else "cpu"

def convert_image_to_rgb(image):
    return image.convert("RGB")

def get_image_transform(size=512):
    transform = torch_transforms.Compose([
        torch_transforms.Resize(size, interpolation=InterpolationMode.BICUBIC),
        torch_transforms.CenterCrop(size),
        convert_image_to_rgb,
        torch_transforms.ToTensor(),
        torch_transforms.Normalize((0.4914, 0.4822, 0.4465), (0.247, 0.243, 0.
 ↪261))
    ])
    return transform

def load_image_dataset(dataset_path, size=512):
    transform = get_image_transform(size=size)
    dataset = ImageFolder(dataset_path)
    train_size = int(0.8 * len(dataset))
    test_size = len(dataset) - train_size
    train_dataset, test_dataset = random_split(dataset, [train_size, test_size])
    train_dataset.dataset.transform = transform
    test_dataset.dataset.transform = transform
    return train_dataset, test_dataset

def display_images(dataset, num_images=5):
    loader = DataLoader(dataset, batch_size=num_images, shuffle=True)
    images, labels = next(iter(loader))
    grid = make_grid(images, nrow=5)
    plt.figure(figsize=(12, 12))
    plt.imshow(grid.permute(1, 2, 0))
    plt.axis('off')
    plt.show()

def calculate_metrics(y_true, y_pred):
    accuracy = accuracy_score(y_true, y_pred)
    precision = precision_score(y_true, y_pred, average='weighted')
    recall = recall_score(y_true, y_pred, average='weighted')
    f1 = f1_score(y_true, y_pred, average='weighted')
    cm = confusion_matrix(y_true, y_pred)
    return accuracy, precision, recall, f1, cm
```

[51]:
```
args = {
    'dataset_path': '/workspace/UCMerced_LandUse/Images',
    'split': 'test',
    'version': '2-1',
    'img_size': 512,
    'batch_size': 32,
    'dtype': 'float32',
    'loss': 'l2',
    'keep_counts': [5, 1],
    'num_samples': [50, 500],
    'prompt_path': 'data_prompts/uc_merced_prompts.csv',
}
```

0.1 Data

[55]:
```
train_dataset, test_dataset = load_image_dataset(args['dataset_path'],␣
  ↪size=args['img_size'])
display_images(train_dataset, num_images=5)
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers). Got range [-1.8783362..2.1308641].
```



[36]:
```
def evaluate_probability_adaptive(unet, latent, text_embeddings, scheduler,␣
  ↪args, latent_size=64, all_noise=None):
    scheduler_config = get_scheduler_config(args)
    num_timesteps = scheduler_config['num_train_timesteps'] #1000 timesteps␣
↪from Stable Diffusion 2.1 base.
    max_samples = max(args['num_samples'])
    all_noise = torch.randn((max_samples, 4, latent_size, latent_size),␣
↪device=latent.device) #(500, 4, 64, 64) noise tensor

    evaluation_data = {} #all errors for all timesteps
    evaluated_timesteps = set() #timesteps evaluated so far
    remaining_prompt_indices = list(range(len(text_embeddings))) #21
    start_step = 1
    steps_to_evaluate = list(range(1, num_timesteps, 2)) # 500 odd numbers from␣
↪0-1000

    for samples, keep_count in zip(args['num_samples'], args['keep_counts']):␣
↪#1st iteration: (50, 5) - 2nd iteration: (500, 1)

        ts = [] #list of timesteps to send for the current number of samples␣
↪(50 / 450)
        noise_indices = [] #noise indices to evaluate for current number of␣
↪samples (50 / 450)
        text_embed_indices = [] #embedding indices to evaluate for current␣
↪number of samples (50 / 450)

        current_steps_to_evaluate = steps_to_evaluate[len(steps_to_evaluate) //␣
↪samples // 2::len(steps_to_evaluate) // samples][:samples]
        current_steps_to_evaluate = [t for t in current_steps_to_evaluate if t␣
↪not in evaluated_timesteps]


        for prompt_idx in remaining_prompt_indices:

            for step_idx, t in enumerate(current_steps_to_evaluate,␣
↪start=len(evaluated_timesteps)):
                ts.extend([t]) #repeat timesteps for each timestep in␣
↪current_steps_to_evaluate
                noise_indices.extend(list(range(step_idx, step_idx + 1))) #␣
↪repeat each noise index for each timestep
                text_embed_indices.extend([prompt_idx]) #repeat each text␣
↪embedding index for each timestep

        evaluated_timesteps.update(current_steps_to_evaluate) #update list with␣
↪current number of timesteps


        predicted_errors = evaluate_error(unet, scheduler, latent, all_noise,␣
↪ts, noise_indices, text_embeddings, text_embed_indices, args['batch_size'])
```

```
        for prompt_idx in remaining_prompt_indices:

            mask = torch.tensor(text_embed_indices) == prompt_idx #creating␣
↪mask tensor to filter out errors for current prompt index from a tensor with␣
↪1050/2250 errors)
            #mask tensor looks like: tensor([ True,  True,  True,  ..., False,␣
↪False, False])

            prompt_ts = torch.tensor(ts)[mask] #timesteps for current prompt
            prompt_predicted_errors = predicted_errors[mask] #predition errors␣
↪for current prompt



            if prompt_idx not in evaluation_data:
                evaluation_data[prompt_idx] = dict(t=prompt_ts,␣
↪predicted_errors=prompt_predicted_errors) #create a dict of dicts for prompt␣
↪indices
            else:
                evaluation_data[prompt_idx]['t'] = torch.
↪cat([evaluation_data[prompt_idx]['t'], prompt_ts]) #dump all the timesteps␣
↪against each prompt
                evaluation_data[prompt_idx]['predicted_errors'] = torch.
↪cat([evaluation_data[prompt_idx]['predicted_errors'],␣
↪prompt_predicted_errors]) #dump all the errors against each prompt
                #looks like this: index 0: {'t': tensor([ 11,  31,  51,  71, ␣
↪91, 111, 131, 151, 171, 191, 211, 231, 251, 271, ...]), 'predicted_errors':␣
↪tensor([0.8067, 0.6652, 0.5988, 0.5222, 0.4832, 0.4474, 0.4130, ...])}

        errors = [-evaluation_data[prompt_idx]['predicted_errors'].mean() for␣
↪prompt_idx in remaining_prompt_indices] #take negative of means of errors␣
↪for each prompt index and store it as a tensor

        best_indices = torch.topk(torch.tensor(errors), k=keep_count, dim=0).
↪indices.tolist() #1st iteration: top 5, 2nd iteration: top 1

        remaining_prompt_indices = [remaining_prompt_indices[i] for i in␣
↪best_indices]

    assert len(remaining_prompt_indices) == 1
    predicted_index = remaining_prompt_indices[0]
    return predicted_index
```

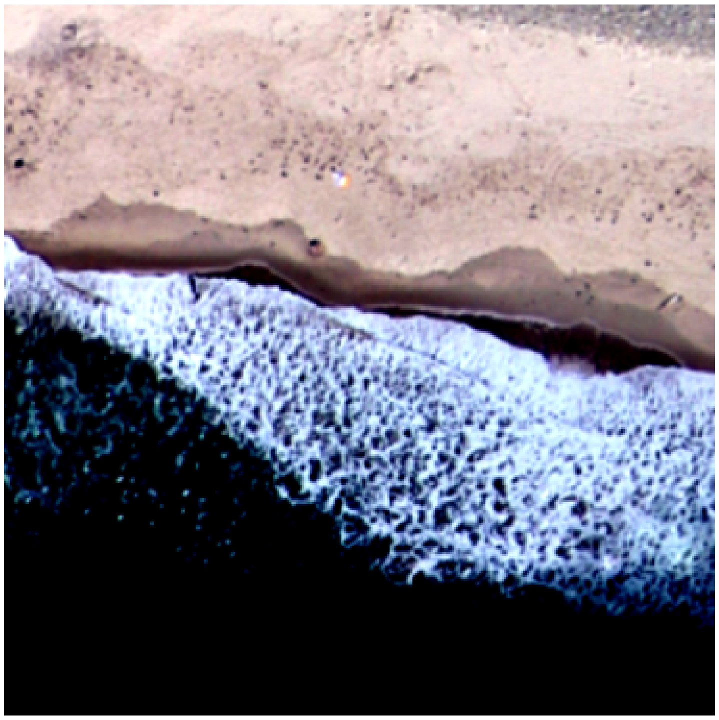
[37]:
```
def evaluate_error(unet, scheduler, latent, all_noise, ts, noise_indices,␣
↪text_embeddings, text_embed_indices, batch_size=32):
    assert len(ts) == len(noise_indices) == len(text_embed_indices)
    predicted_errors = torch.zeros(len(ts), device='cpu')
    idx = 0
    with torch.inference_mode():
        for _ in trange(len(ts) // batch_size + int(len(ts) % batch_size != 0),␣
↪leave=False):
            batch_ts = torch.tensor(ts[idx: idx + batch_size])
            noise = all_noise[noise_indices[idx: idx + batch_size]]
            noised_latent = latent * (scheduler.alphas_cumprod[batch_ts] ** 0.
↪5).view(-1, 1, 1, 1).to(device) + \
                                    noise * ((1 - scheduler.alphas_cumprod[batch_ts])␣
↪** 0.5).view(-1, 1, 1, 1).to(device) #noise at each timestep is cumulative␣
↪noise added up to that timestep.
            t_input = batch_ts.to(device)
            text_input = text_embeddings[text_embed_indices[idx: idx +␣
↪batch_size]]
            noise_prediction = unet(noised_latent, t_input,␣
↪encoder_hidden_states=text_input).sample #unet learns to predict noise added␣
↪at each specific timestep but every timestep is dependent on the previous␣
↪timesteps (thats actually what unet learns)
            error = F.mse_loss(noise, noise_prediction, reduction='none').
↪mean(dim=(1, 2, 3))
            predicted_errors[idx: idx + len(batch_ts)] = error.detach().cpu()
            idx += len(batch_ts)
    return predicted_errors
```

0.2 Classification - Latent Diffusion Classifier

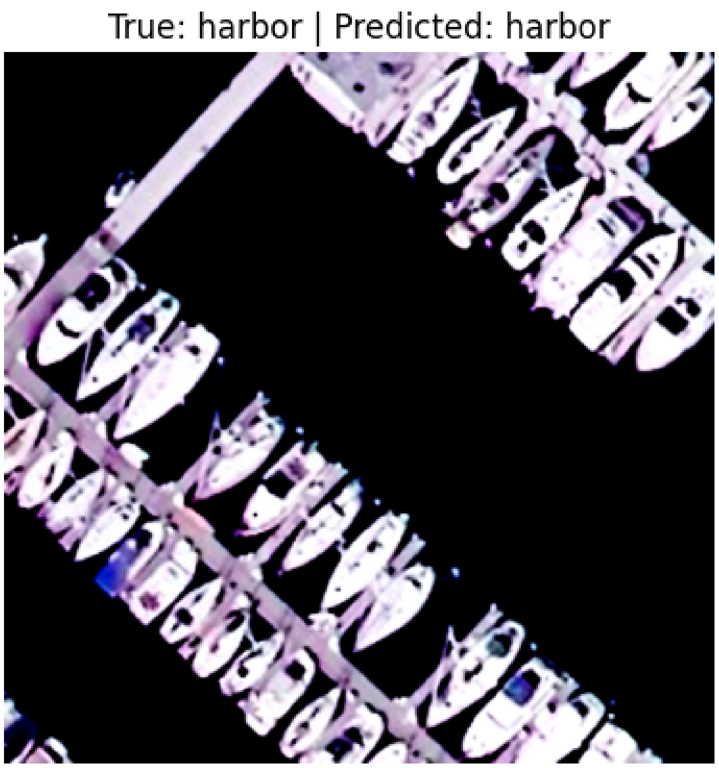
[32]:
```
train_dataset, test_dataset = load_image_dataset(args['dataset_path'],␣
 ↪size=args['img_size'])
latent_size = args['img_size'] // 8
target_dataset = test_dataset  # Ensure test set is used

prompts_df = pd.read_csv(args['prompt_path'])

vae, tokenizer, text_encoder, unet, scheduler = get_sd_model(args)
vae = vae.to(device)
text_encoder = text_encoder.to(device)
unet = unet.to(device)

all_noise = None
text_input = tokenizer(prompts_df.prompt.tolist(), padding="max_length",␣
 ↪max_length=tokenizer.model_max_length, truncation=True, return_tensors="pt")
embeddings = []
with torch.inference_mode():
    for i in range(0, len(text_input.input_ids), 100):
        text_embeddings = text_encoder(text_input.input_ids[i: i + 100].
 ↪to(device))[0]
        embeddings.append(text_embeddings)
text_embeddings = torch.cat(embeddings, dim=0)

assert len(text_embeddings) == len(prompts_df)

idxs_to_eval = list(range(len(target_dataset)))
format_str = get_formatstr(len(target_dataset) - 1)
correct_predictions = 0
total_predictions = 0
y_true = []
y_pred = []

progress_bar = tqdm(idxs_to_eval)
for i in progress_bar:
    image, label = target_dataset[i]
    y_true.append(label)
    true_label_name = class_names[label]
    with torch.no_grad():
        img_input = image.to(device).unsqueeze(0)
        latent_representation = vae.encode(img_input).latent_dist.mean
        latent_representation *= 0.18215
    predicted_index = evaluate_probability_adaptive(unet,␣
 ↪latent_representation, text_embeddings, scheduler, args, latent_size,␣
 ↪all_noise)
    predicted_class = prompts_df.classidx[predicted_index]
    predicted_label_name = class_names[predicted_class]
    y_pred.append(predicted_class)
    if predicted_class == label:
        correct_predictions += 1
    total_predictions += 1
    current_accuracy = 100 * correct_predictions / total_predictions
    progress_bar.set_description(f'Acc: {current_accuracy:.2f}%')

    if i < 5:
        plt.figure()
        plt.imshow(image.permute(1, 2, 0).numpy())
        plt.title(f'True: {true_label_name} | Predicted:␣
 ↪{predicted_label_name}')
        plt.axis('off')
        plt.show()
        print(f'Sample {i} evaluated: predicted_label={predicted_label_name},␣
 ↪true_label={true_label_name}')
```

```
Loading pipeline components…:   0%|          | 0/6 [00:00<?, ?it/s]

  0%|          | 0/421 [00:00<?, ?it/s]
  0%|          | 0/33 [00:00<?, ?it/s]

  0%|          | 0/71 [00:00<?, ?it/s]

Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers). Got range [-1.195634..1.339381].
```

True: beach | Predicted: beach



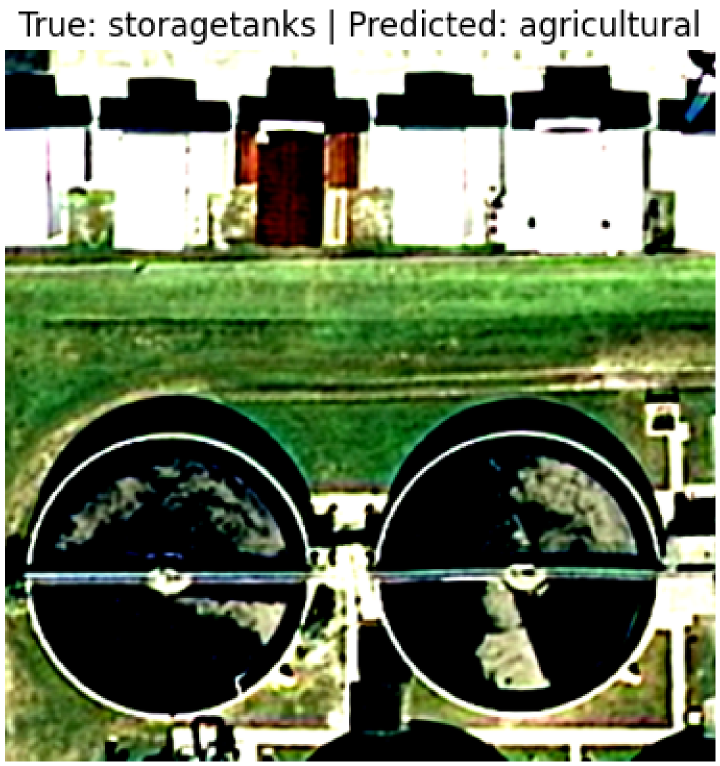
```
Sample 0 evaluated: predicted_label=beach, true_label=beach

  0%|          | 0/33 [00:00<?, ?it/s]

  0%|          | 0/71 [00:00<?, ?it/s]

Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers). Got range [-1.3549745..1.2642552].
```



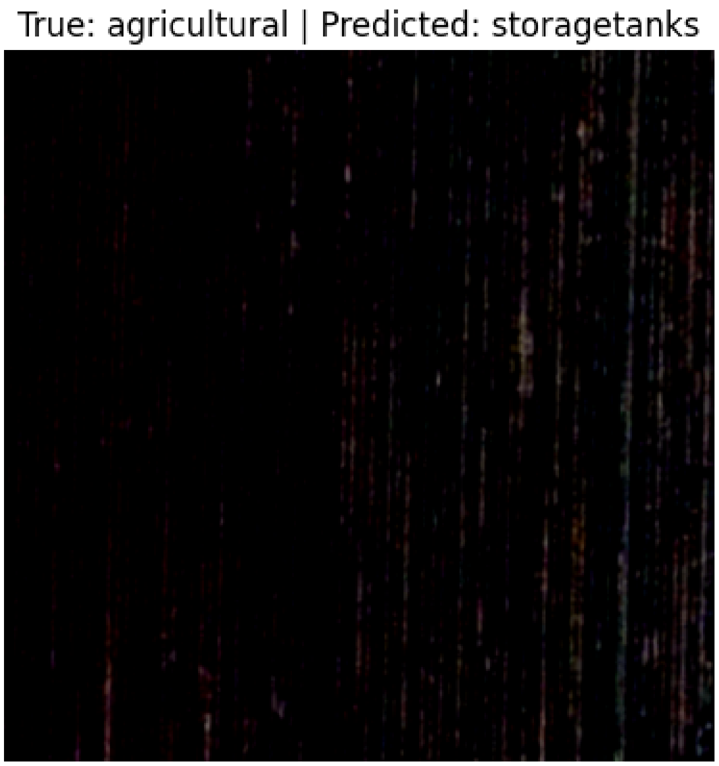
```
Sample 1 evaluated: predicted_label=harbor, true_label=harbor
  0%|          | 0/33 [00:00<?, ?it/s]
  0%|          | 0/71 [00:00<?, ?it/s]
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers). Got range [-1.9894737..2.1308641].
```

True: parkinglot | Predicted: freeway

```
Sample 2 evaluated: predicted_label=freeway, true_label=parkinglot
  0%|          | 0/33 [00:00<?, ?it/s]
  0%|          | 0/71 [00:00<?, ?it/s]
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers). Got range [-1.9894737..2.0501735].
```



```
Sample 3 evaluated: predicted_label=agricultural, true_label=storagetanks
  0%|          | 0/33 [00:00<?, ?it/s]
  0%|          | 0/71 [00:00<?, ?it/s]
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers). Got range [-0.45123854..0.455553].
```



```
Sample 4 evaluated: predicted_label=storagetanks, true_label=agricultural
  0%|          | 0/33 [00:00<?, ?it/s]
  0%|          | 0/71 [00:00<?, ?it/s]
  0%|          | 0/33 [00:00<?, ?it/s]
  0%|          | 0/71 [00:00<?, ?it/s]
  0%|          | 0/33 [00:00<?, ?it/s]
  0%|          | 0/71 [00:00<?, ?it/s]
  0%|          | 0/33 [00:00<?, ?it/s]
  0%|          | 0/71 [00:00<?, ?it/s]
  0%|          | 0/33 [00:00<?, ?it/s]
  0%|          | 0/71 [00:00<?, ?it/s]
  0%|          | 0/33 [00:00<?, ?it/s]
  0%|          | 0/71 [00:00<?, ?it/s]
  0%|          | 0/33 [00:00<?, ?it/s]
  0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
0%|          | 0/33 [00:00<?, ?it/s]
0%|          | 0/71 [00:00<?, ?it/s]
```

```
0%|          | 0/33 [00:00<?, ?it/s]

0%|          | 0/71 [00:00<?, ?it/s]

0%|          | 0/33 [00:00<?, ?it/s]

0%|          | 0/71 [00:00<?, ?it/s]

0%|          | 0/33 [00:00<?, ?it/s]

0%|          | 0/71 [00:00<?, ?it/s]

0%|          | 0/33 [00:00<?, ?it/s]

0%|          | 0/71 [00:00<?, ?it/s]

0%|          | 0/33 [00:00<?, ?it/s]

0%|          | 0/71 [00:00<?, ?it/s]

0%|          | 0/33 [00:00<?, ?it/s]

0%|          | 0/71 [00:00<?, ?it/s]

0%|          | 0/33 [00:00<?, ?it/s]

0%|          | 0/71 [00:00<?, ?it/s]

0%|          | 0/33 [00:00<?, ?it/s]

0%|          | 0/71 [00:00<?, ?it/s]
```

0.3 Evaluation

[35]:
```python
accuracy, precision, recall, f1, cm = calculate_metrics(y_true, y_pred)
print(f'Final Accuracy: {100 * correct_predictions / total_predictions:.2f}%')
print(f'Accuracy: {accuracy:.4f}')
print(f'Precision: {precision:.4f}')
print(f'Recall: {recall:.4f}')
print(f'F1 Score: {f1:.4f}')
```

```
Final Accuracy: 50.12%
Accuracy: 0.5012
Precision: 0.5349
Recall: 0.5012
F1 Score: 0.4881
```

0.4 Classification - ResNet-50

[85]:
```python
train_loader = DataLoader(train_dataset, batch_size=32, shuffle=True)
test_loader = DataLoader(test_dataset, batch_size=32, shuffle=False)

model = models.resnet18(pretrained=True)
num_features = model.fc.in_features
model.fc = torch.nn.Linear(num_features, len(train_dataset.dataset.classes))
```

```
model = model.to(device)

criterion = torch.nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)

def train_model(model, train_loader, criterion, optimizer, num_epochs):
    model.train()
    for epoch in range(num_epochs):
        running_loss = 0.0
        for images, labels in tqdm(train_loader):
            images, labels = images.to(device), labels.to(device)
            optimizer.zero_grad()
            outputs = model(images)
            loss = criterion(outputs, labels)
            loss.backward()
            optimizer.step()
            running_loss += loss.item()
        print(f"Epoch [{epoch+1}/{num_epochs}], Loss: {running_loss/
 ↪len(train_loader):.4f}")
        torch.cuda.synchronize()

def evaluate_model(model, test_loader):
    model.eval()
    y_true = []
    y_pred = []
    with torch.no_grad():
        for images, labels in test_loader:
            images, labels = images.to(device), labels.to(device)
            outputs = model(images)
            _, preds = torch.max(outputs, 1)
            y_true.extend(labels.cpu().numpy())
            y_pred.extend(preds.cpu().numpy())
    return y_true, y_pred
```

[73]:
```
train_model(model, train_loader, criterion, optimizer, 10)
```

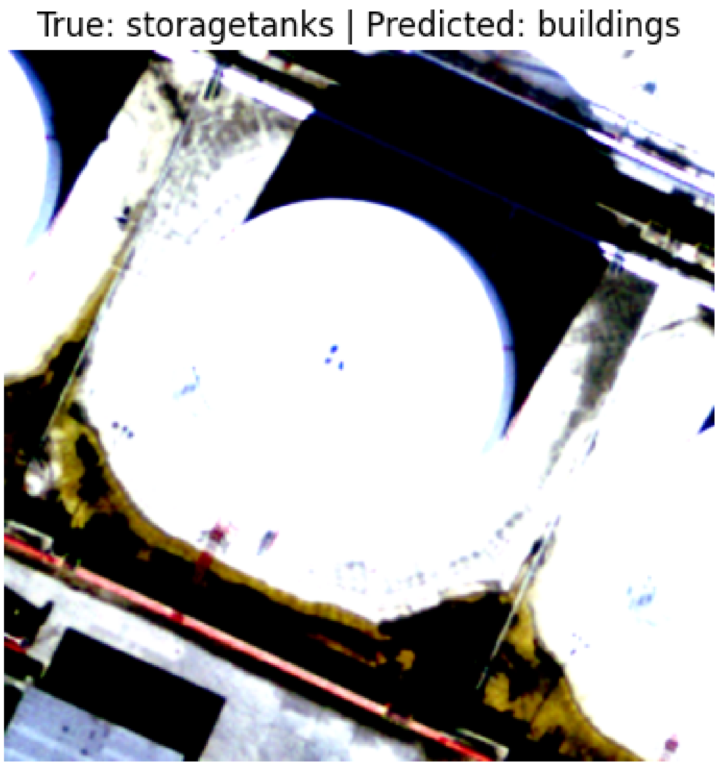
```
  0%|          | 0/53 [00:00<?, ?it/s]

Epoch [1/10], Loss: 1.2129

  0%|          | 0/53 [00:00<?, ?it/s]

Epoch [2/10], Loss: 0.6178

  0%|          | 0/53 [00:00<?, ?it/s]

Epoch [3/10], Loss: 0.4204

  0%|          | 0/53 [00:00<?, ?it/s]

Epoch [4/10], Loss: 0.4074
```

```
  0%|          | 0/53 [00:00<?, ?it/s]

Epoch [5/10], Loss: 0.2639

  0%|          | 0/53 [00:00<?, ?it/s]

Epoch [6/10], Loss: 0.2455

  0%|          | 0/53 [00:00<?, ?it/s]

Epoch [7/10], Loss: 0.2037

  0%|          | 0/53 [00:00<?, ?it/s]

Epoch [8/10], Loss: 0.1647

  0%|          | 0/53 [00:00<?, ?it/s]

Epoch [9/10], Loss: 0.1363

  0%|          | 0/53 [00:00<?, ?it/s]

Epoch [10/10], Loss: 0.1243
```

[74]:
```
y_true, y_pred = evaluate_model(model, test_loader)
```

[78]:
```
class_names = train_dataset.dataset.classes
for i in range(5):
    image, label = test_dataset[i]
    true_label_name = class_names[label]
    with torch.no_grad():
        img_input = image.to(device).unsqueeze(0)
        outputs = model(img_input)
        _, predicted_class = torch.max(outputs, 1)
    predicted_label_name = class_names[predicted_class.item()]
    plt.figure()
    plt.imshow(image.permute(1, 2, 0).numpy())
    plt.title(f'True: {true_label_name} | Predicted: {predicted_label_name}')
    plt.axis('off')
    plt.show()
    print(f'Sample {i} evaluated: predicted_label={predicted_label_name}, 
 ↪true_label={true_label_name}')
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers). Got range [-1.4679415..1.888792].
```



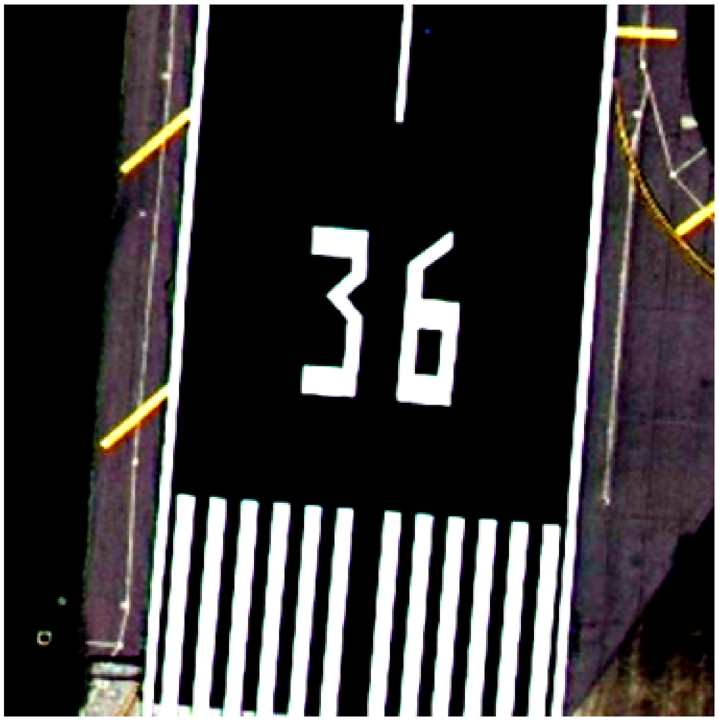
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers). Got range [-1.9894737..2.1308641].
```

```
Sample 0 evaluated: predicted_label=buildings, true_label=storagetanks
```

```
Sample 1 evaluated: predicted_label=sparseresidential,
true_label=sparseresidential
```

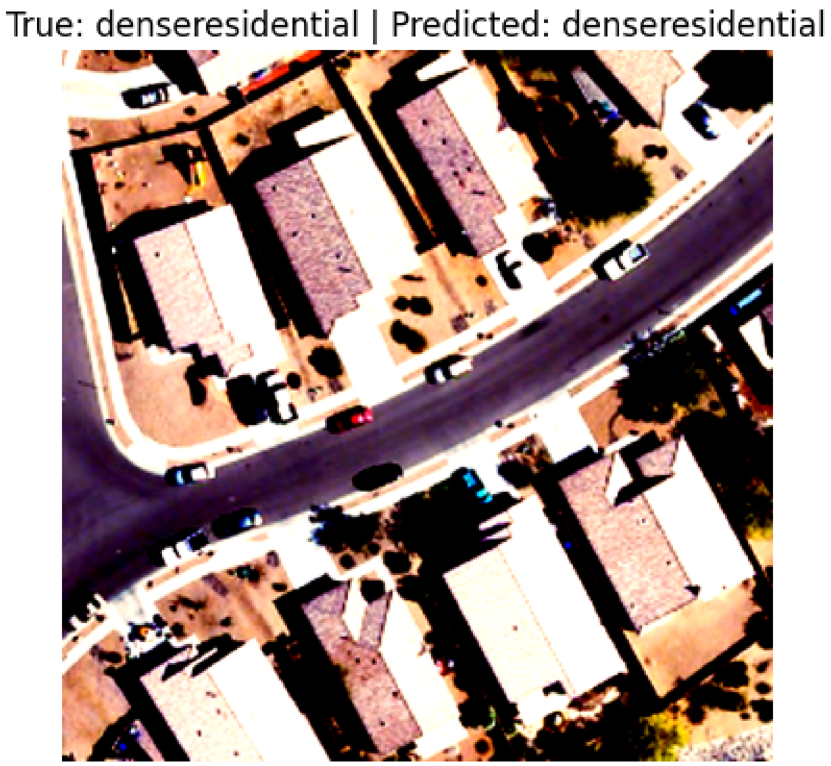
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers). Got range [-1.855257..2.1308641].
```

True: runway | Predicted: runway



```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers). Got range [-1.4252498..2.1308641].
```

```
Sample 2 evaluated: predicted_label=runway, true_label=runway
```



```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers). Got range [-1.8307058..1.7751107].
```

```
Sample 3 evaluated: predicted_label=denseresidential,
true_label=denseresidential
```

True: mediumresidential | Predicted: denseresidential

```
Sample 4 evaluated: predicted_label=denseresidential,
true_label=mediumresidential
```

[82]:
```
accuracy, precision, recall, f1, cm = calculate_metrics(y_true, y_pred)
print(f'Accuracy: {100 * accuracy:.4f}')
print(f'Precision: {100 * precision:.4f}')
print(f'Recall: {100 * recall:.4f}')
print(f'F1 Score: {100 * f1:.4f}')
```

```
Accuracy: 88.5986
Precision: 92.0790
Recall: 88.5986
F1 Score: 87.3919
```