```
import torch
from diffusers import StableDiffusionPipeline
import os
```

```
Flax classes are deprecated and will be removed in Diffusers v1.0.0. We recomm
Flax classes are deprecated and will be removed in Diffusers v1.0.0. We recomm
```

```
# CONFIG

MODEL_ID = "runwayml/stable-diffusion-v1-5"
OUTPUT_DIR = "brain_mri_disease_prediction_dataset"
IMAGES_PER_LABEL = 5
DEVICE = "cuda"
```

```
# LOAD MODEL (GPU)

pipe = StableDiffusionPipeline.from_pretrained(
    MODEL_ID,
    torch_dtype=torch.float16
).to(DEVICE)

pipe.enable_attention_slicing()
```

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: Use
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access pu
  warnings.warn(
```

```
model_index.json:   0%|          | 0.00/541 [00:00<?, ?B/s]
```

```
Fetching 15 files:   0%|          | 0/15 [00:00<?, ?it/s]
```

```
config.json:   0%|          | 0.00/617 [00:00<?, ?B/s]
```

```
preprocessor_config.json:   0%|          | 0.00/342 [00:00<?, ?B/s]

config.json: 0.00B [00:00, ?B/s]

scheduler_config.json:   0%|          | 0.00/308 [00:00<?, ?B/s]

special_tokens_map.json:   0%|          | 0.00/472 [00:00<?, ?B/s]

merges.txt: 0.00B [00:00, ?B/s]

text_encoder/model.safetensors:   0%|          | 0.00/492M [00:00<?, ?B/s]

safety_checker/model.safetensors:   0%|          | 0.00/1.22G [00:00<?, ?B/s]

tokenizer_config.json:   0%|          | 0.00/806 [00:00<?, ?B/s]

config.json:   0%|          | 0.00/743 [00:00<?, ?B/s]

vocab.json: 0.00B [00:00, ?B/s]

config.json:   0%|          | 0.00/547 [00:00<?, ?B/s]

unet/diffusion_pytorch_model.safetensors:   0%|          | 0.00/3.44G [00:00<
```

```
vae/diffusion_pytorch_model.safetensors:   0%|          | 0.00/335M [00:00<?,


Loading pipeline components...:   0%|          | 0/7 [00:00<?, ?it/s]


`torch_dtype` is deprecated! Use `dtype` instead!


# LABELS & PROMPTS (UPDATED)

labels_prompts = {
    "normal":
    "Brain MRI scan, axial view, healthy brain, normal anatomy, no abnormaliti

    "brain_tumor":
    "Brain MRI scan showing glioma tumor, irregular mass lesion, contrast-enha

    "stroke":
    "Brain MRI scan showing ischemic stroke, infarct region, reduced signal in

    "hemorrhage":
    "Brain MRI scan showing intracerebral hemorrhage, hyperintense bleeding re

    "alzheimer":
    "Brain MRI scan showing Alzheimer’s disease, cortical atrophy, enlarged ve

    "multiple_sclerosis":
    "Brain MRI scan showing multiple sclerosis lesions, white matter plaques,

    "edema":
    "Brain MRI scan showing cerebral edema, diffuse swelling, increased intrac

    "infection":
    "Brain MRI scan showing brain infection, abscess formation, inflammatory
}



# DATASET GENERATION

os.makedirs(OUTPUT_DIR, exist_ok=True)

for label, prompt in labels_prompts.items():
    label_dir = os.path.join(OUTPUT_DIR, label)
    os.makedirs(label_dir, exist_ok=True)

    print(f"Generating images for: {label}")
    for i in range(1, IMAGES_PER_LABEL + 1):
        image = pipe(
            prompt,
            guidance_scale=7.5,
            num_inference_steps=40
        ).images[0]

        image_path = os.path.join(label_dir, f"{label}_{i}.png")
        image.save(image_path)

        print(f"Saved: {image_path}")

print("\n Synthetic Brain MRI dataset generation completed!")
```

```
Generating images for: normal



  0%|          | 0/40 [00:00<?, ?it/s]


Saved: brain_mri_disease_prediction_dataset/normal/normal_1.png



  0%|          | 0/40 [00:00<?, ?it/s]


Saved: brain_mri_disease_prediction_dataset/normal/normal_2.png



  0%|          | 0/40 [00:00<?, ?it/s]


Saved: brain_mri_disease_prediction_dataset/normal/normal_3.png



  0%|          | 0/40 [00:00<?, ?it/s]


Saved: brain_mri_disease_prediction_dataset/normal/normal_4.png



  0%|          | 0/40 [00:00<?, ?it/s]
```

```
Saved: brain_mri_disease_prediction_dataset/normal/normal_5.png
Generating images for: brain_tumor


  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/brain_tumor/brain_tumor_1.png


  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/brain_tumor/brain_tumor_2.png


  0%|          | 0/40 [00:00<?, ?it/s]

Potential NSFW content was detected in one or more images. A black image will

Saved: brain_mri_disease_prediction_dataset/brain_tumor/brain_tumor_3.png


  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/brain_tumor/brain_tumor_4.png


  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/brain_tumor/brain_tumor_5.png
Generating images for: stroke


  0%|          | 0/40 [00:00<?, ?it/s]
```

```
Potential NSFW content was detected in one or more images. A black image will

Saved: brain_mri_disease_prediction_dataset/stroke/stroke_1.png


  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/stroke/stroke_2.png


  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/stroke/stroke_3.png


  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/stroke/stroke_4.png


  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/stroke/stroke_5.png
Generating images for: hemorrhage


  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/hemorrhage/hemorrhage_1.png


  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/hemorrhage/hemorrhage_2.png
```

```
  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/hemorrhage/hemorrhage_3.png

  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/hemorrhage/hemorrhage_4.png

  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/hemorrhage/hemorrhage_5.png
Generating images for: alzheimer

  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/alzheimer/alzheimer_1.png

  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/alzheimer/alzheimer_2.png

  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/alzheimer/alzheimer_3.png

  0%|          | 0/40 [00:00<?, ?it/s]
```

```
Saved: brain_mri_disease_prediction_dataset/alzheimer/alzheimer_4.png


  0%|          | 0/40 [00:00<?, ?it/s]


Saved: brain_mri_disease_prediction_dataset/alzheimer/alzheimer_5.png
Generating images for: multiple_sclerosis


  0%|          | 0/40 [00:00<?, ?it/s]


Saved: brain_mri_disease_prediction_dataset/multiple_sclerosis/multiple_sclero


  0%|          | 0/40 [00:00<?, ?it/s]


Saved: brain_mri_disease_prediction_dataset/multiple_sclerosis/multiple_sclero


  0%|          | 0/40 [00:00<?, ?it/s]


Saved: brain_mri_disease_prediction_dataset/multiple_sclerosis/multiple_sclero


  0%|          | 0/40 [00:00<?, ?it/s]


Saved: brain_mri_disease_prediction_dataset/multiple_sclerosis/multiple_sclero


  0%|          | 0/40 [00:00<?, ?it/s]


Saved: brain_mri_disease_prediction_dataset/multiple_sclerosis/multiple_sclero
Generating images for: edema
```

```
  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/edema/edema_1.png


  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/edema/edema_2.png


  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/edema/edema_3.png


  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/edema/edema_4.png


  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/edema/edema_5.png
Generating images for: infection


  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/infection/infection_1.png


  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/infection/infection_2.png
```

```
  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/infection/infection_3.png

  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/infection/infection_4.png

  0%|          | 0/40 [00:00<?, ?it/s]

Saved: brain_mri_disease_prediction_dataset/infection/infection_5.png

 Synthetic Brain MRI dataset generation completed!
```

```
pip install torch torchvision pillow
```

```
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packag
Requirement already satisfied: torchvision in /usr/local/lib/python3.12/dist-
Requirement already satisfied: pillow in /usr/local/lib/python3.12/dist-packag
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-pac
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/py
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-p
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dis
Requirement already satisfied: networkx>=2.5.1 in /usr/local/lib/python3.12/d
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packag
Requirement already satisfied: fsspec>=0.8.5 in /usr/local/lib/python3.12/dis
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/loca
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/li
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/l
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/l
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in /usr/local/li
Requirement already satisfied: nvidia-nccl-cu12==2.27.5 in /usr/local/lib/pyt
Requirement already satisfied: nvidia-nvshmem-cu12==3.3.20 in /usr/local/lib/
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/py
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/l
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/
Requirement already satisfied: triton==3.5.0 in /usr/local/lib/python3.12/dist
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.1
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/di
```

```
import torch
import torch.nn as nn
import torch.optim as optim
from torchvision import datasets, transforms, models
from torch.utils.data import DataLoader
from PIL import Image



# CONFIG
# ------------------------------
DATASET_DIR = "brain_mri_disease_prediction_dataset"
DEVICE = "cuda" if torch.cuda.is_available() else "cpu"
BATCH_SIZE = 8
EPOCHS = 5
IMG_SIZE = 224

# ------------------------------
# TRANSFORMS
# ------------------------------
transform = transforms.Compose([
    transforms.Resize((IMG_SIZE, IMG_SIZE)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.5], std=[0.5])
])


# ------------------------------
# DATASET & LOADER
# ------------------------------
dataset = datasets.ImageFolder(DATASET_DIR, transform=transform)
loader = DataLoader(dataset, batch_size=BATCH_SIZE, shuffle=True)

class_names = dataset.classes
num_classes = len(class_names)

print("Classes:", class_names)

# ------------------------------
# MODEL (ResNet18)
# ------------------------------
model = models.resnet18(pretrained=True)
model.fc = nn.Linear(model.fc.in_features, num_classes)
```

```
model = model.to(DEVICE)

criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.0001)
```

```
Classes: ['alzheimer', 'brain_tumor', 'edema', 'hemorrhage', 'infection', 'mu
Downloading: "https://download.pytorch.org/models/resnet18-f37072fd.pth" to /
```

```
/usr/local/lib/python3.12/dist-packages/torchvision/models/_utils.py:208: Use
  warnings.warn(
/usr/local/lib/python3.12/dist-packages/torchvision/models/_utils.py:223: Use
  warnings.warn(msg)
100%|██████████| 44.7M/44.7M [00:00<00:00, 157MB/s]
```

```
# TRAINING
# -----------------------------
model.train()
for epoch in range(EPOCHS):
    running_loss = 0.0

    for images, labels in 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}/{EPOCHS}], Loss: {running_loss:.4f}")

# -----------------------------
# SAVE MODEL
# -----------------------------
torch.save(model.state_dict(), "brain_mri_classifier.pth")
print("Model training completed & saved!")
```

```
Epoch [1/5], Loss: 11.2566
Epoch [2/5], Loss: 5.3489
Epoch [3/5], Loss: 2.6152
Epoch [4/5], Loss: 1.5216
Epoch [5/5], Loss: 1.0176
Model training completed & saved!
```

```
# -----------------------------
# LOAD MODEL

model.eval()
model.load_state_dict(torch.load("brain_mri_classifier.pth", map_location=DEVI

# -----------------------------
# LOAD IMAGE
# -----------------------------
image_path = "brain_mri_disease_prediction_dataset/stroke/stroke_1.png"

image = Image.open(image_path).convert("RGB")
image = transform(image).unsqueeze(0).to(DEVICE)

# -----------------------------
# PREDICTION
# -----------------------------
with torch.no_grad():
    output = model(image)
    predicted_class = torch.argmax(output, dim=1).item()

print("Predicted Disease:", class_names[predicted_class])

Predicted Disease: brain_tumor
```