```
!pip install diffusers transformers accelerate torch safetensors
```

```
Requirement already satisfied: diffusers in /usr/local/lib/python3.12/dist-packages (0.36.0)
Requirement already satisfied: transformers in /usr/local/lib/python3.12/dist-packages (4.57.3)
Requirement already satisfied: accelerate in /usr/local/lib/python3.12/dist-packages (1.12.0)
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages (2.9.0+cpu)
Requirement already satisfied: safetensors in /usr/local/lib/python3.12/dist-packages (0.7.0)
Requirement already satisfied: importlib_metadata in /usr/local/lib/python3.12/dist-packages (from diffusers) (8.7.1)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from diffusers) (3.20.2)
Requirement already satisfied: httpx<1.0.0 in /usr/local/lib/python3.12/dist-packages (from diffusers) (0.28.1)
Requirement already satisfied: huggingface-hub<2.0,>=0.34.0 in /usr/local/lib/python3.12/dist-packages (from diffusers) (0.36.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages (from diffusers) (2.0.2)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.12/dist-packages (from diffusers) (2025.11.3)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from diffusers) (2.32.4)
Requirement already satisfied: Pillow in /usr/local/lib/python3.12/dist-packages (from diffusers) (11.3.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (25.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-packages (from transformers) (6.0.3)
Requirement already satisfied: tokenizers<=0.23.0,>=0.22.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.22.
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.12/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: psutil in /usr/local/lib/python3.12/dist-packages (from accelerate) (5.9.5)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch) (4.15.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch) (1.14.0)
Requirement already satisfied: networkx>=2.5.1 in /usr/local/lib/python3.12/dist-packages (from torch) (3.6.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch) (3.1.6)
Requirement already satisfied: fsspec>=0.8.5 in /usr/local/lib/python3.12/dist-packages (from torch) (2025.3.0)
Requirement already satisfied: anyio in /usr/local/lib/python3.12/dist-packages (from httpx<1.0.0->diffusers) (4.12.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-packages (from httpx<1.0.0->diffusers) (2026.1.4)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.12/dist-packages (from httpx<1.0.0->diffusers) (1.0.9)
Requirement already satisfied: idna in /usr/local/lib/python3.12/dist-packages (from httpx<1.0.0->diffusers) (3.11)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.12/dist-packages (from httpcore==1.*->httpx<1.0.0->diffusers)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<2.0,>=0.34.
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->torch) (1.3.0)
Requirement already satisfied: zipp>=3.20 in /usr/local/lib/python3.12/dist-packages (from importlib_metadata->diffusers) (3.23.
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->torch) (3.0.3)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->diffusers) (3
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->diffusers) (2.5.0)
```

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

```
Flax classes are deprecated and will be removed in Diffusers v1.0.0. We recommend migrating to PyTorch classes or pinning your v
Flax classes are deprecated and will be removed in Diffusers v1.0.0. We recommend migrating to PyTorch classes or pinning your v
```

Task 1

```
model_id = "runwayml/stable-diffusion-v1-5"

pipe = StableDiffusionPipeline.from_pretrained(
    model_id,
    torch_dtype=torch.float16
)

pipe = pipe.to("cuda")
```

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
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 (https://huggingface.co/settings/tokens), set it
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
model_index.json: 100%                                  541/541 [00:00<00:00, 20.9kB/s]
Fetching 15 files: 100%                                 15/15 [00:53<00:00,  3.88s/it]
preprocessor_config.json: 100%                          342/342 [00:00<00:00, 15.7kB/s]
config.json: 100%                                       617/617 [00:00<00:00, 15.0kB/s]
special_tokens_map.json: 100%                           472/472 [00:00<00:00, 6.86kB/s]
config.json:     4.72k/? [00:00<00:00, 63.8kB/s]
scheduler_config.json: 100%                             308/308 [00:00<00:00, 5.50kB/s]
merges.txt:     525k/? [00:00<00:00, 10.3MB/s]
text_encoder/model.safetensors: 100%                    492M/492M [00:38<00:00, 83.8MB/s]
safety_checker/model.safetensors: 100%                  1.22G/1.22G [00:22<00:00, 27.1MB/s]
tokenizer_config.json: 100%                             806/806 [00:00<00:00, 67.2kB/s]
vocab.json:     1.06M/? [00:00<00:00, 32.3MB/s]
config.json: 100%                                       547/547 [00:00<00:00, 48.4kB/s]
config.json: 100%                                       743/743 [00:00<00:00, 47.2kB/s]
unet/diffusion_pytorch_model.safetensors: 100%          3.44G/3.44G [00:52<00:00, 44.5MB/s]
vae/diffusion_pytorch_model.safetensors: 100%           335M/335M [00:43<00:00, 8.92MB/s]
Loading pipeline components...: 100%                    7/7 [00:20<00:00,  3.55s/it]
`torch_dtype` is deprecated! Use `dtype` instead!
```

```
prompts = [
    "A futuristic city at night with neon lights",
    "A peaceful mountain landscape during sunrise",
    "A robot studying in a classroom",
    "A realistic portrait of a medieval warrior",
    "A cyberpunk street with rain and reflections"
]
```

```
output_dir = "synthetic_dataset"
os.makedirs(output_dir, exist_ok=True)

for idx, prompt in enumerate(prompts):
    image = pipe(prompt).images[0]
    image.save(f"{output_dir}/image_{idx+1}.png")

print("Synthetic dataset generated successfully.")
```

```
100%                          50/50 [00:10<00:00,  6.91it/s]
100%                          50/50 [00:08<00:00,  6.33it/s]
100%                          50/50 [00:07<00:00,  5.62it/s]
100%                          50/50 [00:08<00:00,  6.63it/s]
100%                          50/50 [00:07<00:00,  6.44it/s]
Synthetic dataset generated successfully.
```

```
from PIL import Image
import matplotlib.pyplot as plt

img = Image.open("synthetic_dataset/image_1.png")
plt.imshow(img)
plt.axis("off")
```

```
(np.float64(-0.5), np.float64(511.5), np.float64(511.5), np.float64(-0.5))
```

```python
from PIL import Image
import matplotlib.pyplot as plt

img = Image.open("synthetic_dataset/image_2.png")
plt.imshow(img)
plt.axis("off")
```

```
(np.float64(-0.5), np.float64(511.5), np.float64(511.5), np.float64(-0.5))
```

```python
from PIL import Image
import matplotlib.pyplot as plt

img = Image.open("synthetic_dataset/image_3.png")
plt.imshow(img)
plt.axis("off")
```

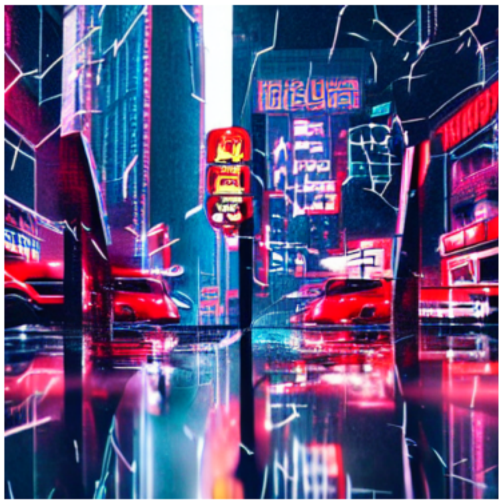
```
(np.float64(-0.5), np.float64(511.5), np.float64(511.5), np.float64(-0.5))
```

```
from PIL import Image
import matplotlib.pyplot as plt

img = Image.open("synthetic_dataset/image_4.png")
plt.imshow(img)
plt.axis("off")
```

```
(np.float64(-0.5), np.float64(511.5), np.float64(511.5), np.float64(-0.5))
```

```
from PIL import Image
import matplotlib.pyplot as plt

img = Image.open("synthetic_dataset/image_5.png")
plt.imshow(img)
plt.axis("off")
```

```
(np.float64(-0.5), np.float64(511.5), np.float64(511.5), np.float64(-0.5))
```



```
Start coding or generate with AI.
```

Task 2

```
BASE_DIR = "synthetic_chest_xray_dataset_task2"
os.makedirs(BASE_DIR, exist_ok=True)
```

```
dataset_labels = {
    "normal_anatomy": "healthy human lungs with normal anatomy chest X-ray",
    "infectious_patterns": "chest X-ray showing infectious lung disease such as pneumonia or viral infection",
    "lung_opacities": "chest X-ray showing lung opacities including ground glass and consolidations",
    "pleural_conditions": "chest X-ray showing pleural effusion or pneumothorax",
    "structural_lesions": "chest X-ray showing lung nodules masses or fibrosis",
    "cardiac_findings": "chest X-ray showing cardiomegaly and pulmonary vascular congestion",
    "medical_devices": "chest X-ray showing medical devices such as tubes catheters or pacemaker",
    "imaging_artifacts": "chest X-ray with imaging artifacts such as noise motion blur or exposure issues",
    "view_positioning": "chest X-ray with different views and patient positioning PA AP supine erect",
    "domain_shift": "chest X-ray showing domain shift due to different scanners hospitals and resolutions"
}
```

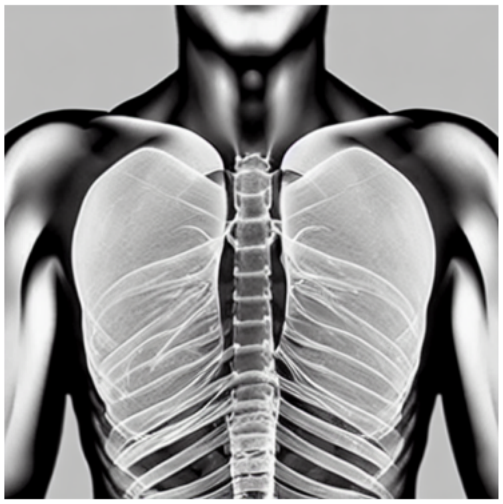
```
BASE_PROMPT = (
    "Very high quality realistic chest X-ray showing {}, "
    "medical radiology style, grayscale, diagnostic accuracy, "
    "sharp details, hospital imaging"
)
```

```
IMAGES_PER_CATEGORY = 5

for category, label in dataset_labels.items():
    category_path = os.path.join(BASE_DIR, category)
    os.makedirs(category_path, exist_ok=True)

    prompt = BASE_PROMPT.format(label)
    print(f"Generating images for {category}")

    for i in range(IMAGES_PER_CATEGORY):
        image = pipe(
            prompt,
            guidance_scale=8.5,
            num_inference_steps=40
        ).images[0]

        image.save(f"{category_path}/{category}_{i+1}.png")

print("TASK-2 synthetic dataset generation completed.")
```

```
Generating images for normal_anatomy
100%                                      40/40 [00:06<00:00,  5.72it/s]
100%                                      40/40 [00:10<00:00,  4.48it/s]
100%                                      40/40 [00:06<00:00,  6.31it/s]
100%                                      40/40 [00:06<00:00,  6.57it/s]
100%                                      40/40 [00:06<00:00,  6.72it/s]
Generating images for infectious_patterns
100%                                      40/40 [00:06<00:00,  6.80it/s]
100%                                      40/40 [00:05<00:00,  6.91it/s]
100%                                      40/40 [00:05<00:00,  6.91it/s]
100%                                      40/40 [00:05<00:00,  6.94it/s]
```

```python
sample_image = Image.open(
    f"{BASE_DIR}/normal_anatomy/normal_anatomy_1.png"
)

plt.imshow(sample_image)
plt.axis("off")
```

```
(np.float64(-0.5), np.float64(511.5), np.float64(511.5), np.float64(-0.5))
```



```
[00:06<00:00,  6.54it/s]
[00:06<00:00,  6.54it/s]
[00:06<00:00,  6.59it/s]
[00:06<00:00,  6.65it/s]
[00:06<00:00,  6.68it/s]
[00:06<00:00,  6.77it/s]
[00:06<00:00,  6.78it/s]
[00:06<00:00,  6.83it/s]
[00:06<00:00,  6.79it/s]
[00:06<00:00,  6.81it/s]
[00:06<00:00,  6.80it/s]
```

Start coding or generate with AI.

```
100%                                      40/40 [00:06<00:00,  6.70it/s]
100%                                      40/40 [00:06<00:00,  6.74it/s]
```

TASK 3

```python
import torch
import torchvision.transforms as transforms
from torchvision import models
from PIL import Image
import matplotlib.pyplot as plt
```

```python
model = models.densenet121(pretrained=True)

num_features = model.classifier.in_features
model.classifier = torch.nn.Linear(num_features, 2)  # Normal vs Abnormal
model.eval()
```

```
100%                                      40/40 [00:06<00:00,  6.77it/s]
100%                                      40/40 [00:06<00:00,  6.75it/s]
100%                                      40/40 [00:06<00:00,  6.69it/s]
Generating images for view_positioning
100%                                      40/40 [00:06<00:00,  6.66it/s]
      (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
```

```
100%                                    40/40 [00:06<00:00,  6.67it/s]
100%                                    40/40 [00:06<00:00,  6.64it/s]
100%                                    40/40 [00:06<00:00,  6.67it/s]
100%                                    40/40 [00:06<00:00,  6.65it/s]
Generating images for domain_shift
100%                                    40/40 [00:06<00:00,  6.72it/s]
100%                                    40/40 [00:06<00:00,  6.77it/s]
100%                                    40/40 [00:06<00:00,  6.79it/s]
100%                                    40/40 [00:06<00:00,  6.77it/s]
100%                                    40/40 [00:06<00:00,  6.74it/s]
TASK-2 synthetic dataset generation completed.
```

```
      )
      (denselayer2): _DenseLayer(
        (norm1): BatchNorm2d(544, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(544, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      )
      (denselayer3): _DenseLayer(
        (norm1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(576, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      )
      (denselayer4): _DenseLayer(
        (norm1): BatchNorm2d(608, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(608, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      )
      (denselayer5): _DenseLayer(
        (norm1): BatchNorm2d(640, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(640, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      )
      (denselayer6): _DenseLayer(
        (norm1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(672, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      )
      (denselayer7): _DenseLayer(
        (norm1): BatchNorm2d(704, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(704, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
```

```python
transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize(
        mean=[0.485, 0.456, 0.406],
        std=[0.229, 0.224, 0.225]
    )
])
```

```python
DATASET_DIR = "synthetic_chest_xray_dataset_task2"
```

```python
def get_true_label(folder_name):
    if folder_name == "normal_anatomy":
        return 0  # Normal
    else:
        return 1  # Abnormal
```

```python
correct = 0
total = 0

print("Image | Ground Truth | Predicted")
print("-----------------------------------")

for category in os.listdir(DATASET_DIR):
    category_path = os.path.join(DATASET_DIR, category)

    if not os.path.isdir(category_path):
        continue
    true_label = get_true_label(category)

    for img_name in os.listdir(category_path):
        img_path = os.path.join(category_path, img_name)

        image = Image.open(img_path).convert("RGB")
        input_tensor = transform(image).unsqueeze(0)

        with torch.no_grad():
            output = model(input_tensor)
            predicted_label = torch.argmax(output, dim=1).item()

        print(f"{img_name} | {true_label} | {predicted_label}")

        if predicted_label == true_label:
            correct += 1

        total += 1
```

```
Image | Ground Truth | Predicted
----------------------------------
structural_lesions_2.png | 1 | 0
structural_lesions_4.png | 1 | 1
structural_lesions_5.png | 1 | 0
structural_lesions_1.png | 1 | 0
structural_lesions_3.png | 1 | 0
medical_devices_5.png | 1 | 1
medical_devices_3.png | 1 | 1
medical_devices_4.png | 1 | 1
medical_devices_2.png | 1 | 0
medical_devices_1.png | 1 | 1
infectious_patterns_2.png | 1 | 1
infectious_patterns_4.png | 1 | 1
infectious_patterns_1.png | 1 | 0
infectious_patterns_5.png | 1 | 1
infectious_patterns_3.png | 1 | 1
normal_anatomy_5.png | 0 | 1
normal_anatomy_3.png | 0 | 0
normal_anatomy_4.png | 0 | 1
normal_anatomy_1.png | 0 | 1
normal_anatomy_2.png | 0 | 0
cardiac_findings_2.png | 1 | 1
cardiac_findings_1.png | 1 | 1
cardiac_findings_3.png | 1 | 1
cardiac_findings_4.png | 1 | 1
cardiac_findings_5.png | 1 | 0
pleural_conditions_4.png | 1 | 0
pleural_conditions_5.png | 1 | 0
pleural_conditions_2.png | 1 | 1
```