

```
In [ ]: import os
import numpy as np
from PIL import Image
```

```
In [ ]: file_location = '/Users/dhruvo.poddar/Downloads/My stuff/GSOC/dataset 2/t
output_location = "/Users/dhruvo.poddar/Downloads/My stuff/GSOC/new_datas
os.makedirs(output_location, exist_ok=True)

# Loop through .npy files and convert them to .png
for file in os.listdir(file_location):
    if file.endswith(".npy"):
        file_path = os.path.join(file_location, file)
        img_array = np.load(file_path) # Load .npy file

        # Ensure shape is (H, W) by removing the single channel
        if img_array.shape[0] == 1:
            img_array = img_array.squeeze(0) # Remove channel dim, now (

        # Normalize to 0-255 if needed
        img_array = (img_array - img_array.min()) / (img_array.max() - im
        img_array = img_array.astype(np.uint8) # Convert to uint8

        # Convert to PIL image
        img = Image.fromarray(img_array)

        # Save as PNG
        output_path = os.path.join(output_location, file.replace(".npy",
        img.save(output_path)

        # print(f"Converted {file} to {output_path}")
```

```
In [ ]: file_location = '/Users/dhruvo.poddar/Downloads/My stuff/GSOC/dataset 2/t
output_location = "/Users/dhruvo.poddar/Downloads/My stuff/GSOC/new_datas
os.makedirs(output_location, exist_ok=True)

# Loop through .npy files and convert them to .png
for file in os.listdir(file_location):
    if file.endswith(".npy"):
        file_path = os.path.join(file_location, file)
        img_array = np.load(file_path) # Load .npy file

        # Ensure shape is (H, W) by removing the single channel
        if img_array.shape[0] == 1:
            img_array = img_array.squeeze(0) # Remove channel dim, now (

        # Normalize to 0-255 if needed
        img_array = (img_array - img_array.min()) / (img_array.max() - im
        img_array = img_array.astype(np.uint8) # Convert to uint8

        # Convert to PIL image
        img = Image.fromarray(img_array)

        # Save as PNG
        output_path = os.path.join(output_location, file.replace(".npy",
        img.save(output_path)
```

```
In [ ]: file_location = '/Users/dhruvo.poddar/Downloads/My stuff/GSOC/dataset 2/t
output_location = "/Users/dhruvo.poddar/Downloads/My stuff/GSOC/new_datas
os.makedirs(output_location, exist_ok=True)

# Loop through .npy files and convert them to .png
for file in os.listdir(file_location):
    if file.endswith(".npy"):
        file_path = os.path.join(file_location, file)
        img_array = np.load(file_path) # Load .npy file

        # Ensure shape is (H, W) by removing the single channel
        if img_array.shape[0] == 1:
            img_array = img_array.squeeze(0) # Remove channel dim, now (

        # Normalize to 0-255 if needed
        img_array = (img_array - img_array.min()) / (img_array.max() - im
        img_array = img_array.astype(np.uint8) # Convert to uint8

        # Convert to PIL image
        img = Image.fromarray(img_array)

        # Save as PNG
        output_path = os.path.join(output_location, file.replace(".npy",
        img.save(output_path)
```

```
In [ ]: file_location = '/Users/dhruvo.poddar/Downloads/My stuff/GSOC/dataset 2/v
output_location = "/Users/dhruvo.poddar/Downloads/My stuff/GSOC/new_datas
os.makedirs(output_location, exist_ok=True)

# Loop through .npy files and convert them to .png
for file in os.listdir(file_location):
    if file.endswith(".npy"):
        file_path = os.path.join(file_location, file)
        img_array = np.load(file_path) # Load .npy file

        # Ensure shape is (H, W) by removing the single channel
        if img_array.shape[0] == 1:
            img_array = img_array.squeeze(0) # Remove channel dim, now (

        # Normalize to 0-255 if needed
        img_array = (img_array - img_array.min()) / (img_array.max() - im
        img_array = img_array.astype(np.uint8) # Convert to uint8

        # Convert to PIL image
        img = Image.fromarray(img_array)

        # Save as PNG
        output_path = os.path.join(output_location, file.replace(".npy",
        img.save(output_path)
```

```
In [ ]: file_location = '/Users/dhruvo.poddar/Downloads/My stuff/GSOC/dataset 2/v
output_location = "/Users/dhruvo.poddar/Downloads/My stuff/GSOC/new_datas
os.makedirs(output_location, exist_ok=True)

# Loop through .npy files and convert them to .png
for file in os.listdir(file_location):
    if file.endswith(".npy"):
        file_path = os.path.join(file_location, file)
        img_array = np.load(file_path) # Load .npy file
```

```

# Ensure shape is (H, W) by removing the single channel
if img_array.shape[0] == 1:
    img_array = img_array.squeeze(0) # Remove channel dim, now (

# Normalize to 0-255 if needed
img_array = (img_array - img_array.min()) / (img_array.max() - im
img_array = img_array.astype(np.uint8) # Convert to uint8

# Convert to PIL image
img = Image.fromarray(img_array)

# Save as PNG
output_path = os.path.join(output_location, file.replace(".npy",
img.save(output_path)

```

```

In [ ]: file_location = '/Users/dhruvo.poddar/Downloads/My stuff/GSOC/dataset 2/v
output_location = "/Users/dhruvo.poddar/Downloads/My stuff/GSOC/new_datas
os.makedirs(output_location, exist_ok=True)

```

```

# Loop through .npy files and convert them to .png
for file in os.listdir(file_location):
    if file.endswith(".npy"):
        file_path = os.path.join(file_location, file)
        img_array = np.load(file_path) # Load .npy file

# Ensure shape is (H, W) by removing the single channel
if img_array.shape[0] == 1:
    img_array = img_array.squeeze(0) # Remove channel dim, now (

# Normalize to 0-255 if needed
img_array = (img_array - img_array.min()) / (img_array.max() - im
img_array = img_array.astype(np.uint8) # Convert to uint8

# Convert to PIL image
img = Image.fromarray(img_array)

# Save as PNG
output_path = os.path.join(output_location, file.replace(".npy",
img.save(output_path)

```