This report outlines the process of relabeling a medical image dataset, applying Gabor filters to extract texture characteristics, and saving the processed data into a CSV file for further analysis. Using Gabor filters, renowned for their effectiveness in capturing texture patterns, the code creates a feature set from each image in the dataset to support future classification tasks. This report references a scikit-image tutorial and a Medium article, which elaborate on Gabor filters' capabilities in image processing and texture recognition.

Use Case: Pneumonia Detection from Chest X-rays

The purpose of the code is to preprocess a dataset of chest X-rays, separated into "NORMAL" and "PNEUMONIA" classes. The code performs three main tasks:

1. **Relabeling the Dataset**: Each image file in the dataset is renamed to a standardized format, ensuring unique identifiers across the dataset.
2. **Extracting Texture Features with Gabor Filters**: Texture characteristics from each image are captured using Gabor filters at various orientations and scales.
3. **Saving Extracted Data to CSV**: The processed features and labels are saved in a CSV file for easy integration into machine learning workflows.

Code Explanation

1. Relabeling the Dataset

The dataset is located on the user’s desktop, with subdirectories for each class ("NORMAL" and "PNEUMONIA"). Each image is renamed with a unique identifier that includes the class label and a global counter, which ensures unique filenames and simplifies dataset organization.

```
# Renaming images with unique identifiers
for sub_dir in sub_dirs:
    folder_path = os.path.join(base_dir, sub_dir)
    if os.path.exists(folder_path):
        for filename in os.listdir(folder_path):
            if filename.endswith(('.png', '.jpg', '.jpeg')):
                new_name = f"{sub_dir}_{global_count}.jpg"
                old_path = os.path.join(folder_path, filename)
                new_path = os.path.join(folder_path, new_name)
                os.rename(old_path, new_path)
                global_count += 1
```

2. Defining and Applying Gabor Filters

The code defines multiple Gabor filters by varying:

- **Theta (Orientation)**: Four orientations (0, 45, 90, and 135 degrees) capture texture at different angles.
- **Sigma (Scale)**: Two values of sigma (1 and 3) control the filter's sensitivity to different scales.
- **Frequency**: Two frequency values (0.05 and 0.25) allow the filter to respond to patterns at different granularities.

These filters extract orientation-sensitive texture information from the images, essential for distinguishing between healthy and pneumonia-affected lung textures.

```
# Creating Gabor filters with various parameters
kernels = []
for theta in range(4):
    theta = theta / 4.0 * np.pi
    for sigma in (1, 3):
        for frequency in (0.05, 0.25):
            kernel = np.real(gabor_kernel(frequency, theta=theta, sigma_x=sigma, sigma_y=sigma))
            kernels.append(kernel)
```

3. Feature Extraction and Saving Data to CSV

Each image in the dataset is loaded in grayscale, and the `compute_feats` function applies each Gabor filter to extract two key features:

- **Mean**: Average intensity of the filtered image, which represents the general presence of the detected texture.
- **Variance**: Measures how strongly the texture or pattern varies across the image.

The extracted features are then saved to a CSV file, along with the image number and label ("NORMAL" or "PNEUMONIA"), allowing for easy integration into further machine learning tasks.

```
# Feature extraction and saving to CSV
with open(csv_file_path, mode='w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(["numero d'image", "caractéristique", "label"])

    for sub_dir in sub_dirs:
        folder_path = os.path.join(base_dir, sub_dir)

        if os.path.exists(folder_path):
            for filename in os.listdir(folder_path):
                if filename.endswith(('.png', '.jpg', '.jpeg')):
                    image_path = os.path.join(folder_path, filename)
                    image = img_as_float(io.imread(image_path,
as_gray=True))
                    feats = compute_feats(image, kernels)
                    numero_image =
filename.split('_')[-1].split('.')[0]
                    label = filename.split('_')[0]
                    feats_str = str(feats.tolist())
                    writer.writerow([numero_image, feats_str,
label])
```

Summary

The code effectively organizes and preprocesses a dataset of chest X-rays for pneumonia detection by:

1. **Relabeling** each image with a unique identifier,
2. **Extracting texture features** using Gabor filters that highlight patterns and textures at different orientations and scales, and
3. **Saving** these extracted features along with labels into a CSV file.

This processed dataset, now rich in orientation-specific and scale-specific texture features, can be directly used in machine learning pipelines to enhance classification tasks between "NORMAL" and "PNEUMONIA" cases.