

# Rapport : Extraction de Caractéristiques avec des Filtres de Gabor pour la Détection de Pneumonie

Ce rapport présente le processus de réétiquetage d'un jeu de données d'images médicales, l'application de filtres de Gabor pour extraire des caractéristiques de texture, et la sauvegarde des données traitées dans un fichier CSV pour une analyse ultérieure. En utilisant des filtres de Gabor, reconnus pour leur efficacité dans la capture des motifs de texture, le code crée un ensemble de caractéristiques à partir de chaque image du jeu de données afin de soutenir les futures tâches de classification. Ce rapport fait référence à un tutoriel scikit-image et à un article de Medium, qui expliquent en détail les capacités des filtres de Gabor dans le traitement des images et la reconnaissance de textures.

## Cas d'utilisation : Détection de Pneumonie à partir de Radiographies Thoraciques

Le but du code est de prétraiter un jeu de données de radiographies thoraciques, séparées en deux classes : "NORMAL" et "PNEUMONIA". Le code effectue trois tâches principales :

- Réétiquetage du jeu de données** : Chaque fichier image du jeu de données est renommé selon un format standardisé, assurant des identifiants uniques dans tout le jeu de données.
- Extraction des caractéristiques de texture avec des filtres de Gabor** : Les caractéristiques de texture de chaque image sont capturées à l'aide de filtres de Gabor à différentes orientations et échelles.
- Sauvegarde des données extraites dans un fichier CSV** : Les caractéristiques traitées et les étiquettes sont enregistrées dans un fichier CSV pour une intégration facile dans des flux de travail de machine learning.

## Explication du Code

### 1. Réétiquetage du Jeu de Données

Le jeu de données se trouve sur le bureau de l'utilisateur, avec des sous-répertoires pour chaque classe ("NORMAL" et "PNEUMONIA"). Chaque image est renommée avec un identifiant unique qui inclut l'étiquette de classe et un compteur global, garantissant des noms de fichiers uniques et simplifiant l'organisation du jeu de données.

```
# Renommer les images avec des identifiants uniques
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. Définition et Application des Filtres de Gabor

Le code définit plusieurs filtres de Gabor en faisant varier :

- **Theta (orientation)** : Quatre orientations (0, 45, 90 et 135 degrés) pour capturer la texture sous différents angles.
- **Sigma (échelle)** : Deux valeurs de sigma (1 et 3) pour contrôler la sensibilité du filtre à différentes échelles.
- **Fréquence** : Deux valeurs de fréquence (0,05 et 0,25) permettant au filtre de réagir aux motifs à différents niveaux de granularité.

Ces filtres extraient des informations de texture sensibles à l'orientation des images, essentielles pour distinguer les textures des poumons sains et atteints de pneumonie.

```
# Création de filtres de Gabor avec divers paramètres
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. Extraction des Caractéristiques et Sauvegarde des Données dans un Fichier CSV

Chaque image du jeu de données est chargée en niveaux de gris, et la fonction `compute_feats` applique chaque filtre de Gabor pour extraire deux caractéristiques clés :

- **Moyenne** : L'intensité moyenne de l'image filtrée, qui représente la présence générale de la texture détectée.
- **Variance** : Mesure de la force avec laquelle la texture ou le motif varie dans l'image.

Les caractéristiques extraites sont ensuite enregistrées dans un fichier CSV, avec le numéro de l'image et l'étiquette ("NORMAL" ou "PNEUMONIA"), permettant une

intégration facile dans des tâches de machine learning ultérieures.

```
# Extraction des caractéristiques et sauvegarde dans un fichier CSV
with open(csv_file_path, mode='w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(["numéro 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])
```

## Résumé

---

Le code organise et prétraite efficacement un jeu de données de radiographies thoraciques pour la détection de pneumonie en :

1. Renommant chaque image avec un identifiant unique,
2. Extrayant des caractéristiques de texture à l'aide de filtres de Gabor qui mettent en évidence les motifs et les textures à différentes orientations et échelles, et
3. Enregistrant ces caractéristiques extraites ainsi que les étiquettes dans un fichier CSV.

Ce jeu de données traité, maintenant riche en caractéristiques de texture spécifiques à l'orientation et à l'échelle, peut être directement utilisé dans des pipelines de machine learning pour améliorer les tâches de classification entre les cas "NORMAL" et "PNEUMONIA".