In [35]:
```python
def upload():
#     if request.method == 'POST':
#         f=request.files['file'] #requesting the file
#         basepath=os.path.dirname(os.path.realpath('__file__'))#storing the file directory
#         filepath=os.path.join(basepath,"uploads",f.filename)#storing the file in uploads folder
#         f.save(filepath)#saving the file

        #Loading the saved model
        print("[INFO] loading model...")

        # Pre-process the image in the same manner we did earlier
        #image = cv2.imread(filepath)
        image = cv2.imread(r"/home/wsuser/work/dataset/testing/healthy/V01HO01.png")
        output = image.copy()

        # Load the input image, convert it to grayscale, and resize
        output = cv2.resize(output, (128, 128))
        image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        image = cv2.resize(image, (200, 200))
        image = cv2.threshold(image, 0, 255,
        cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]

        # Quantify the image and make predictions based on the extracted features using the last trained Random Forest
        features = feature.hog(image, orientations=9,
        pixels_per_cell=(10, 10), cells_per_block=(2, 2),
        transform_sqrt=True, block_norm="L1")
        preds = model.predict([features])
        print(preds)
        ls=["healthy","parkinson"]
        result = ls[preds[0]]
        return result
#     return None
```

In [36]:
```python
print (upload())
```

```
[INFO] loading model...
[0]
healthy
```

In [37]:
```python
def upload():
#     if request.method == 'POST':
#         f=request.files['file'] #requesting the file
#         basepath=os.path.dirname(os.path.realpath('__file__'))#storing the file directory
#         filepath=os.path.join(basepath,"uploads",f.filename)#storing the file in uploads folder
#         f.save(filepath)#saving the file

        #Loading the saved model
        print("[INFO] loading model...")

        # Pre-process the image in the same manner we did earlier
        #image = cv2.imread(filepath)
        image = cv2.imread(r"/home/wsuser/work/dataset/testing/parkinson/V02PO01.png")
        output = image.copy()

        # Load the input image, convert it to grayscale, and resize
        output = cv2.resize(output, (128, 128))
        image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        image = cv2.resize(image, (200, 200))
        image = cv2.threshold(image, 0, 255,
        cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]

        # Quantify the image and make predictions based on the extracted features using the last trained Random Forest
        features = feature.hog(image, orientations=9,
        pixels_per_cell=(10, 10), cells_per_block=(2, 2),
        transform_sqrt=True, block_norm="L1")
        preds = model.predict([features])
        print(preds)
        ls=["healthy","parkinson"]
        result = ls[preds[0]]
        return result
#     return None
```

In [38]:
```python
print (upload())
```

```
[INFO] loading model...
[1]
parkinson
```