Nickhil Tekwani Hw3b Problem 5

```
import numpy as np
def generate_rectangles(num_rect=100):
    rectangles = []
    while len(rectangles) < num_rect:
        w = np.random.randint(5, 24)  # Max width set to 23 to ensure 28-w is positive
        h = int(130/w) + 1
        if h > 23:  # Ensuring that height is also within the bounds
            continue

        x = np.random.randint(0, 28 - w)
        y = np.random.randint(0, 28 - h)

        rectangles.append((x, y, w, h))
    return rectangles
```

```
def compute_integral_image(image):
    integral = np.zeros_like(image, dtype=np.int)
    for i in range(image.shape[0]):
        for j in range(image.shape[1]):
            integral[i, j] = image[i, j]
            if i > 0:
                integral[i, j] += integral[i-1, j]
            if j > 0:
                integral[i, j] += integral[i, j-1]
            if i > 0 and j > 0:
                integral[i, j] -= integral[i-1, j-1]
    return integral

def get_black_value(integral, x, y, w, h):
    total = integral[y+h-1, x+w-1]
    if x > 0:
        total -= integral[y+h-1, x-1]
    if y > 0:
        total -= integral[y-1, x+w-1]
    if x > 0 and y > 0:
        total += integral[y-1, x-1]
    return total

def compute_haar_features(image, rectangles):
    integral = compute_integral_image(image)
    features = []

    for rect in rectangles:
        x, y, w, h = rect
        h_mid = h // 2
        w_mid = w // 2

        vertical_feature = get_black_value(integral, x, y, w, h_mid) - get_black_value(integral, x, y+h_mid, w, h_mid)
        horizontal_feature = get_black_value(integral, x, y, w_mid, h) - get_black_value(integral, x+w_mid, y, w_mid, h)

        features.extend([vertical_feature, horizontal_feature])

    return features
```

```
from sklearn.datasets import fetch_openml
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Load MNIST data
mnist = fetch_openml('mnist_784')
X, y = mnist.data.values, mnist.target.values

# Normalize data
X = X / 255.0
X = X.reshape(-1, 28, 28)

# Generate 100 random rectangles
```

```
rectangles = generate_rectangles()

# Extract HAAR features
X_features = [compute_haar_features(img, rectangles) for img in X]

# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(X_features, y, test_size=0.2, random_state=42)

# Classification using RandomForest
clf = RandomForestClassifier()
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)

print(f"Accuracy: {accuracy:.4f}")
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/datasets/_openml.py:968: FutureWarning: The default value of `parser` will ch
  warn(
<ipython-input-2-9fdd5284ae1d>:2: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int`. To silence this
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  integral = np.zeros_like(image, dtype=np.int)
Accuracy: 0.4266
```

Colab paid products - Cancel contracts here

✓ 3m 17s completed at 11:54 PM