Introduction to Computer Vision: Assignment 1

편경찬
[1]Student ID: 201824607. [2]Computer Science Engineering. [3]Pusan National University.

1 Task 1: Image Patches

1.1 (a) Three image patches from grace-hopper

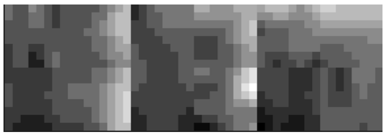

Figure 1: three image patches

이미지를 16x16 크기의 패치로 나누고, 각 패치를 정규화하여 평균이 0이고 분산이 1이 되도록 만드는 작업을 진행한 결과이다.

1.2 (b) Why is it good for the patches to have zero mean

패치의 평균을 0으로 만들면, illumination에 영향을 받지 않는 특징을 추출할 수 있다. 예를 들어, 조명이 밝은 경우와 어두운 경우에 패치의 픽셀 값이 달라질 수 있다. 평균을 0으로 만들면 이러한 조명 변화에 영향을 덜 받게 된다. 평균이 0인 패치는 내적 계산 시 조명 변화에 덜 민감하므로, 더 정확한 유사도 측정이 가능하기 때문이다.

1.3 (c) Good or bad for things for patches

패치의 장점은 이미지의 로컬 특징을 잘 표현할 수 있다는 것이다. 객체의 텍스처, 에지, 코너 등의 정보를 포함하고 있어, 객체 인식 및 매칭에 유용하다. 패치의 단점은 객체의 포즈, 크기, 조명 변화에 민감하다는 것이다. 조명이 변하면 패치의 픽셀 값이 크게 달라질 수 있다.

2 Task 2: Convolution and Gaussian Filter

2.1 (a) Proof of Gaussian filter equivalence

2D 가우시안 필터가 수직과 수평 1D 가우시안 필터의 연속 적용으로 표현될 수 있음을 증명한다. 2D 가우시안 필터는 수직(Gy)과 수평(Gx) 1D 가우시안 필터의 외적으로 표현되며, 합성곱의 연관성에 따라 순차적 적용이 가능하다.

$$G(i,j) = \frac{e^{-\frac{i^2+j^2}{2\sigma^2}}}{2\pi\sigma^2} = \underbrace{\frac{e^{-\frac{i^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma}}_{G_y(i)} \cdot \underbrace{\frac{e^{-\frac{j^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma}}_{G_x(j)}$$

이를 행렬 형태로 표현하면 다음과 같다.

$$\mathbf{G} = \mathbf{G}_y \cdot \mathbf{G}_x = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix} \begin{bmatrix} g_1 & g_2 & \cdots & g_k \end{bmatrix} = \begin{bmatrix} g_1g_1 & g_1g_2 & \cdots & g_1g_k \\ g_2g_1 & g_2g_2 & \cdots & g_2g_k \\ \vdots & \vdots & \ddots & \vdots \\ g_kg_1 & g_kg_2 & \cdots & g_kg_k \end{bmatrix}$$

따라서 2D 합성곱은 1D 필터의 순차적 적용으로 계산 가능하다.

$$I * \mathbf{G} = (I * \mathbf{G}_y) * \mathbf{G}_x$$

2.2 (b) Complete the function convolve()

주어진 커널을 뒤집어서 컨볼루션 수행한 후 Zero-padding 적용하여 입력과 동일한 크기 유지시켰다. 중첩 for-loop을 사용하여 패치별 컨볼루션 연산을 수행한다.

2.3 (c) Output and the role of Gaussian filtering

Gaussian 필터 적용 후의 이미지는 부드러워지고 노이즈가 감소한다. 고주파 성분(날카로운 경계선)이 감소하여 이미지가 흐릿해지는 효과가 있다. Gaussian 필터는 3x3, 표준 편차는 σ = 0.572 로 설정되었다.

Figure 2: Applying Gaussian filter

2.4 (d) Smoothing filter to sum up to 1

필터의 총합이 1이 아니면, 이미지의 밝기가 원래보다 어두워지거나 밝아질 수 있다. 필터의 총합을 1로 정규화하면 출력 이미지의 평균 밝기가 원본과 동일하게 유지된다.

2.5 (e) Derive the convolution kernels for derivatives

미분 연산 커널 유도

$$\mathbf{K}_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}, \quad \mathbf{K}_y = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

2.6 (f) Complete the function edge detection()

Ix (x 방향 기울기)와 Iy (y 방향 기울기)를 계산한 후, Gradient Magnitude를 계산한다.

2.7 (g) Outputs and the difference between the two images

Figure 3: q3_edge

Figure 4: q3_edge_gaussian

원본 이미지의 에지 맵은 경계가 선명하지만, 노이즈가 포함된다. Gaussian 필터 적용 후의 에지 맵은 노이즈가 줄어들었고, 경계선이 더 부드럽다.

3 Task 3: Sobel Operator

3.1 (a) Relationships of Sobel filter and Gaussian filter

다음 식이 컨볼루션의 결합 법칙에 의해 성립한다.

$$(I * G_S) * k_x = I * (G_S * k_x)$$

따라서, $G_S * k_x = S_x$ 임을 증명하면 된다. 실제로 행렬 곱을 계산하면,

$$G_S * k_x = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = S_x$$

다음 식이 성립하므로 증명이 완료된다.

$$\frac{\partial}{\partial x}(I * G_S) = I * S_x$$

3.2 (b) Complete the function sobel operator()

convolve 함수를 이용해 Gradient Magnitude를 계산한다.

3.3 (c) Output

Figure 5: q2_edge_sobel

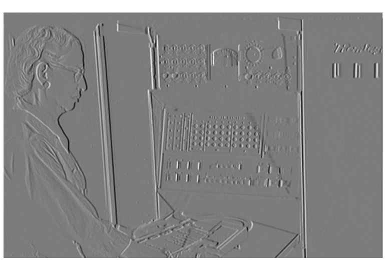

Figure 6: q2_Gx

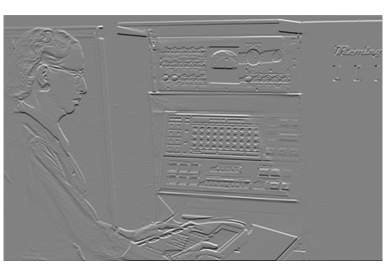

Figure 7: q2_Gy