**OpenCV**

* cv2.Laplacian(img, dtype, ksize) : Laplacian filtering을 수행
  + img : filter를 적용할 원본 이미지
  + dtype : 결과 이미지 데이터 타입.
    - cv2.CV\_8U : uint8
    - cv2.CV\_16U : uint16
    - cv2.CV\_32F : float32
    - cv2.CV\_64F : float64 등
  + ksize : kernel size. Ksize x kszie kernel 사용.
* cv2.Canny(img, minVal, maxVal) : Canny edge dectection을 수행
  + img : Edge detection을 적용할 원본 이미지
  + minVal, maxVal : double thresholding에 사용되는 최저 값, 최고 값.
* 과제는 아래 페이지를 참조하세요. -

**과제 참조**

* **Gaussian의 x축 방향 1차 편미분 :**
* **Gaussian의 2차 미분 :** 
  + **이 경우에는 합이 1이되는 filtering이 아니기 때문에, 계수도 연산해야 합니다.**
* **Zero-crossing에서**
  + **사각 영역에서 원래라면 좌측이 +일 때, 우측이 –와 같이 연산 대상 픽셀에서 Zero-crossing이 발생해야 한다.**
  + **하지만 단순하게 코드를 작성하기 위함과 큰 오차가 생기지 않는다는 점을 고려해 연산 대상 픽셀을 중심으로 한 3x3 사각 영역에서, +와 – 값이 동시에 나타나면 Zero-crossing인 것으로 코드를 작성.**
  + **thresh라는 parameter를 준 것은, 실수 오차를 고려 한 것.**
* **Kernel 생성할 때, 절대 Sum으로 나누어 정규화 하지 말 것**
  + **이 Kernel은 미분 Kernel이어서 합이 1이 되지 않음**