|  |  |
| --- | --- |
| 교육 제목 | AI 프로젝트 기반 S/W 전문가 양성과정 |
| 교육 일시 | 21.12.16 |
| 교육 장소 | 집 |
| **교육 내용** | |
| 오전 | 1. 1215 review    1. shape 이퀄리제이션    2. range 스케일링 2. 화소처리  수학 수식을 이해해보자! 3. 엠보싱 embossing 영상을 양각형태로 표현 4. 블러링 blurring 영상의 세밀한 부분을 제거하여 흐리게 하거나 부드럽게 세밀한 부분은 주파수 영역에서 보면 고주파인데, 블러링은 고주파를 제거함 사용하는 가중치의 회선 마스크는 low pass filter – 높은건 못 지나가! 5. 샤프닝 sharpening 블러링과 반대 개념 저주파 성분을 제거함 사용하는 가중치의 회선 마스크는 high pass filter – 낮은건 못 지나가! 6. 경계선 검출 경계선을 찾아내는 개념 7. 회선처리의 원리  경계선에서 문제가 생김 방법들:  1. 경계선을 연산 안하기 - 원래 영상에서 그대로 갖다 넣기, focus는 중앙에 있으니까 2. zero padding – padding이 0이면 경계선은 이론상 이상한 값이 나올 수 있음 3. 중첩 부분에서만 회선 수행 – 마스크가 3x3이면 1,1에서 회선 시작 4. 영상의 크기를 조정하여 회선 수행 – 영상의 시작과 끝부분이 연결된 것으로 처리하는 방법 5. 회선 마스크의 특징 – 주변 화소의 값을 각 방향에서 대칭적으로 고려해야 함, 홀수의 크기를 사용해야 함, 출력 밝기 에너지를 보존해야 하므로 영상이 평균 밝기를 원 영상과 똑같이 유지해야 함 8. 엠보싱2 경계선만 도드라지게 처리, 적절하게 구분된 경계선만 값으로 출력하기 때문에 영상이 볼록한 느낌을 갖게 됨 9. 블러링2 블러링 회선 마스크는 모든 계수가 양수이고 모든 값을 더한 것은 1 10. 샤프닝2 샤프닝 회선 마스크는 중앙값이 크고 주위는 -1이거나 0, shape 맞춰줘야 함 11. 가우시안 필터 – 부드럽게, 원본의 색감을 많이 유지   소문자 시그마 값이 클수록 높이는 낮지만 폭은 넓어져서 많은 저주파 성분을 통과시킴, 반대는 반대 12. edge – 디지털 영상의 밝기가 낮은 값에서 높은 값으로 or 반대로 변하는 지점 유사 연산자 기법 – 화소를 감산한 값에서 최대값을 결정하여 에지를 검출  미분을 이용한 방법 – edge가 화소의 밝기 “변화율”에 관여하는 것을 차용 13. 수학적으로 들어가니까 너무 어려움… 14. 영상을 논리연산 합, 차, 곱, 분 15. 동영상 – 정지영상을 순차적으로 보여주어 움직이는 것처럼 지각하게 한 것 |
| 오후 | 1. 1215 review 2. cv2.copyTo(src, mask, dst) # 결과 연산이 인풋이었다가 아웃풋으로 나옴 3. img2 = cv2.resize(img2, (w, h), cv2.INTER\_AREA) = INTER\_AREA 축소할 때 부드럽게 4. 격자 있는 것은 알파채널이 있는 것 mask와 BGR이 합쳐진 것 cv2.imread("파일", cv2.IMREAD\_UNCHANGED)  cv2.IMREAD\_UNCHANGED = 알파 채널까지 읽어오기 5. 실습 6. cv2.line(img, pt1, pt2, color[, thickness[, lineType[, shift]]]) 선 그리기 cv2.line(img, (50, 50), (300, 50), (0, 0, 255), 5, cv2.LINE\_8) 소스, 좌표, 컬러, 두께 등 7. cv2.arrowedLine(img, pt1, pt2, color[, thickness[, line\_type[, shift[, tipLength]]]]) cv2.arrowedLine(img, (50, 40), (300, 40), (255, 0, 0), 5, cv2.LINE\_8) 8. rectangle(img, pt1, pt2, color[, thickness[, lineType[, shift]]]) cv2.rectangle(img, (100, 220), (380, 380), (255, 0, 0), 5) cv2.rectangle(img, (50, 350, 350, 150), (0, 255, 0), -1) 이렇게 쓰기도 함 cv2.circle(img, (600, 200), 100, (255, 255, 0), 5, cv2.LINE\_AA) # LINE\_AA 안티 앨리어싱 – 이미지 계단현상 방지 9. cv2.putText(img, text, (800, 100), cv2.FONT\_HERSHEY\_SIMPLEX, 0.8, (0, 0, 0), 1, cv2.LINE\_AA) image에 text 뿌려줌, 나중에 네모칸이 인식하고 그 위에 글자를 뿌려줘야할 때 사용함 |