

Image Processing

실습 14주차

김 대 현

Department of Computer Science and Engineering
Chungnam National University, Korea



- 과목 홈페이지

- 충남대학교 사이버 캠퍼스 (<https://dcs-lcms.cnu.ac.kr/login?redirectUrl=https://dcs-lcms.cnu.ac.kr/>)

- TA 연락처

- 김대현
- 공대 5호관 531호 컴퓨터비전 연구실
- Email: seven776484@gmail.com
 - [IP]을 이메일 제목에 붙여주세요
 - 과제 질문은 메일 또는 사전에 미리 연락하고 연구실 방문 가능

- Tutor 연락처

- 정주헌
- Email: 201802015@o.cnu.ac.kr

공지사항

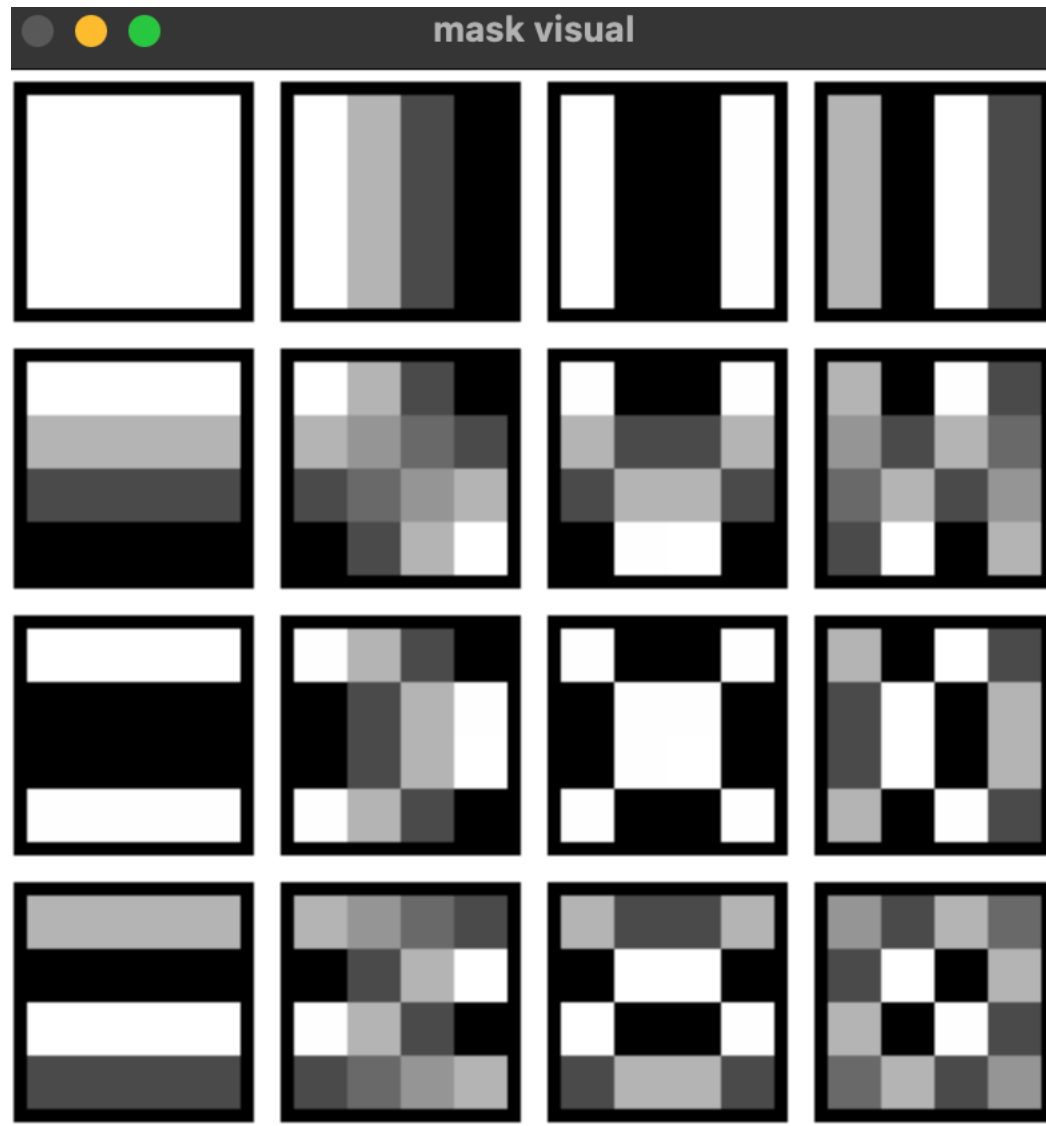
- 공지사항
- DCT 과제 리뷰 및 채점 기준 공개
- 실습
- 과제 (Color noise reduction 구현)

목차

• 공지사항

- 영상처리 02·03분반 과제 채점 공개
 - 채점은 기본적으로 각 주차 별 채점 기준표에 의거하여 각 분반 조교가 채점
 - 과제 채점에 대한 문의 사항은 각 분반 조교에게 문의
- 영상처리 02·03분반 과제 채점 기준 공개
 - DCT 채점 기준 공개
- 과제 Copy 관련 공지
 - 과제 Copy와 관련하여 과제를 함께 진행하였을 경우 보고서에 같이 과제를 진행한 학부생 학번 기입

- DCT(Discrete cosine transform) mask 완성



DCT 채점 기준

• 영상처리 02 · 03분반 과제 채점 기준 공개

• DCT 채점 기준 공개

- 02 · 03 분반 코드 모두 동일
- 코드 기준
 - 총 2가지 TODO 항목에 대하여 평가
 - Spatial2Frequency_mask 함수
 - my_transform 함수

```
#####
# TODO
# TODO mask 만들기
# TODO sub mask shape : 4 x 4
# TODO full mask shape = 16 x 16
# TODO DCT에서 사용된 mask는 4 x 4 mask가 16개 있음 (u, v) 별로 1개씩 있음 u=4, v=4
# TODO submask 마다 0 ~ 255의 범위를 갖도록 변환 (my_transform 함수 사용)
# TODO full mask는 각 sub mask로 구성되어있음
#####
submask = ???|
```

```
#####
# TODO
# TODO my_normalize
# TODO mask를 normalization(0 ~ 1)후 (0 ~ 255)의 값을 갖도록 변환
#####
???
return dst
```

DCT 채점 기준

- 영상처리 02 · 03분반 과제 채점 기준 공개
 - DCT 채점 기준 공개
 - 코드 기준
 - Spatial2Frequency_mask 함수
 - Submask 잘못 구현 시 4점 감점(Normalization은 따로 평가)
 - 부분 점수 없음

```
#####
# TODO
# TODO mask 만들기
# TODO sub mask shape : 4 x 4
# TODO full mask shape = 16 x 16
# TODO DCT에서 사용된 mask는 4 x 4 mask가 16개 있음 (u, v) 별로 1개씩 있음 u=4, v=4
# TODO submask 마다 0 ~ 255의 범위를 갖도록 변환 (my_transform 함수 사용)
# TODO full mask는 각 sub mask로 구성되어있음
#####
submask = ???|
```

DCT 채점 기준

• 영상처리 02 · 03분반 과제 채점 기준 공개

• DCT 채점 기준 공개

• 코드 기준

• my_normalize 함수

• 잘못 구현 시 3점 감점

• 부분 점수

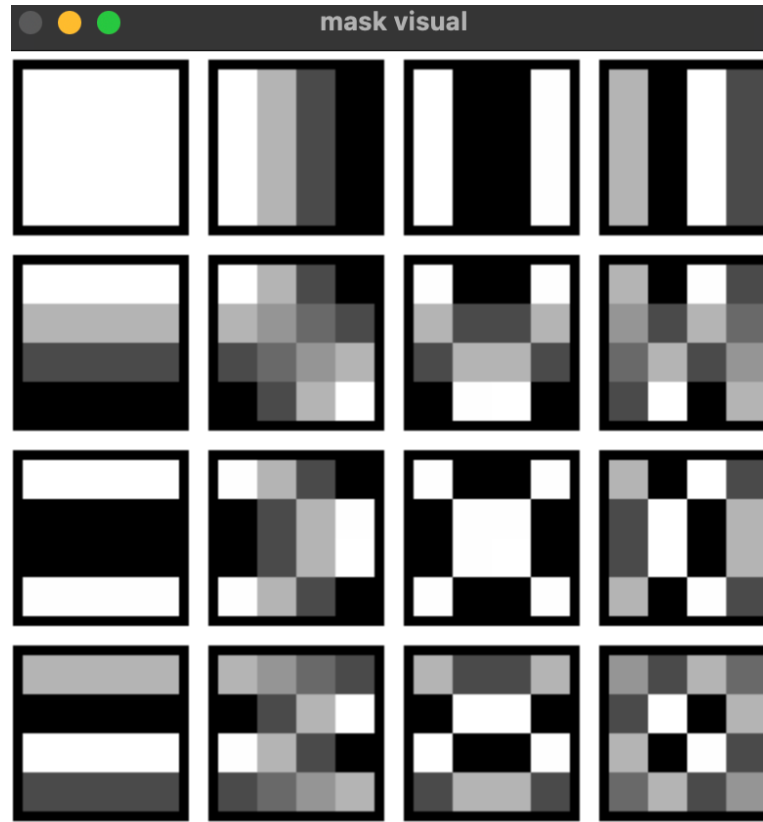
- Mask의 최솟값과 최댓값이 같은 경우와 그 외의 영역 경우로 나누어서 처리한 경우
-> 감점 없음
- 2가지 경우로 나누지 않고 한 가지 경우에만 처리한 경우 2점 감점

```
#####
# TODO
# TODO mask 만들기
# TODO sub mask shape : 4 x 4
# TODO full mask shape = 16 x 16
# TODO DCT에서 사용된 mask는 4 x 4 mask가 16개 있음 (u, v) 별로 1개씩 있음 u=4, v=4
# TODO submask 마다 0 ~ 255의 범위를 갖도록 변환 (my_transform 함수 사용)
# TODO full mask는 각 sub mask로 구성되어있음
#####
submask = ???|
```


DCT 채점 기준

• 영상처리 02 · 03분반 과제 채점 기준 공개

- Canny Edge Detection 채점 기준 공개
 - 보고서 기준
 - 제출 이미지 누락 시 1점 감점
 - 코드 누락 시 1점 감점



- RGB to YIQ

- RGB to YIQ

- Y: Luminance을 표현 -> 밝기 정보를 의미
 - I and Q: Chrominance을 표현 -> 색 정보를 의미

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

```
def BGR2YIQ(src):  
    """  
    :param src: BGR 이미지  
    :return: YIQ 이미지  
    """  
    src = src.astype(np.float32)  
  
    B = src[:, :, 0]  
    G = src[:, :, 1]  
    R = src[:, :, 2]  
  
    Y = 0.299 * R + 0.587 * G + 0.114 * B  
    I = 0.596 * R + (-0.274) * G + (-0.322) * B  
    Q = 0.211 * R + (-0.523) * G + (0.312) * B  
  
    YIQ = np.clip(np.round(np.dstack((Y, I, Q))), 0, 255).astype(np.uint8)  
  
    return YIQ
```

- YIQ to RGB

- YIQ to RGB

- Y: Luminance을 표현 -> 밝기 정보를 의미
 - I and Q: Chrominance을 표현 -> 색 정보를 의미

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.106 & 1.703 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

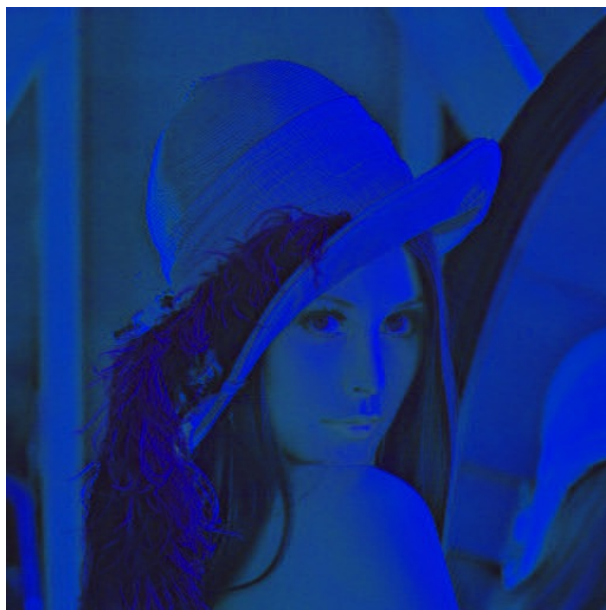
```
def YIQ2BGR(src):  
    """  
    :param src: YIQ 이미지  
    :return: BGR 이미  
    """  
    src = src.astype(np.float32)  
  
    Y = src[:, :, 0]  
    I = src[:, :, 1]  
    Q = src[:, :, 2]  
  
    R = 1 * Y + 0.956 * I + 0.621 * Q  
    G = 1 * Y + (-0.272) * I + (-0.647) * Q  
    B = 1 * Y + (-1.106) * I + (1.703) * Q  
  
    BGR = np.clip(np.round(np.dstack((B, G, R))), 0, 255).astype(np.uint8)  
  
    return BGR
```

실습

- 결과 이미지



BGR



BGR -> YIQ

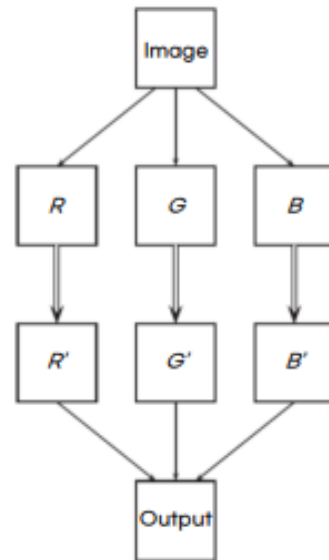


YIQ -> BGR

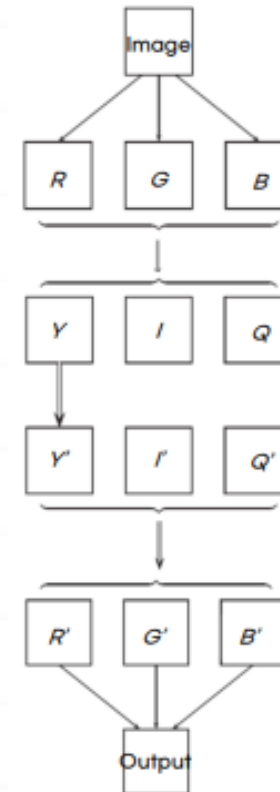
- **Noise reduction**
 - RGB, YUV 이미지에 대한 noise reduction 결과 비교하기
 - Gaussian, median filtering을 진행
 - 보고서에 원본과 결과 사진 비교

- **Processing of Color Images**
 - YIQ와 RGB 이미지에서 filtering 진행하기

1. Independent processing



2. Intensity Only



- Noise reduction
 - RGB 결과



Gaussian



Median

- Noise reduction
 - YUV 결과



Gaussian



Median

- **Noise reduction**
 - 코드 첨부 및 결과 분석 이미지 첨부
 - Page 14, 15 결과 이미지 첨부
 - 과제 기간 : 6월 2일 ~ 6월 23일 23시59분까지

Q & A