

영상정보처리 실습

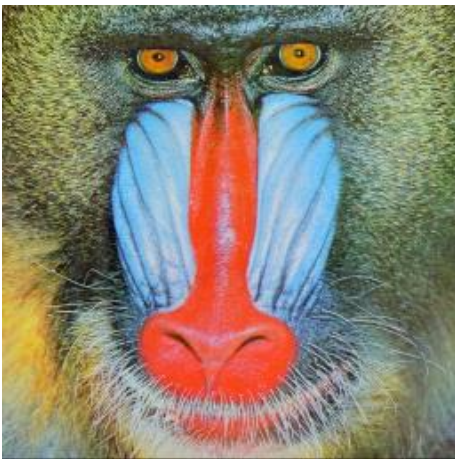
Lab1 Pixel 단위 Operation

1. 실습

- Color 분리
- 역상 이미지 획득
- Histogram Equalization
- Otsu Binarization

1.1 Color 분리

- 1) Red 값만 추출하여 저장
- 2) Green 값만 추출하여 저장
- 3) Blue 값만 추출하여 저장
- 입력 영상: Mandrill.bmp



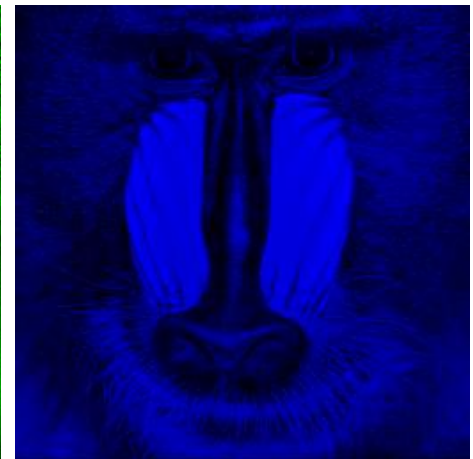
원본영상



Red 영상



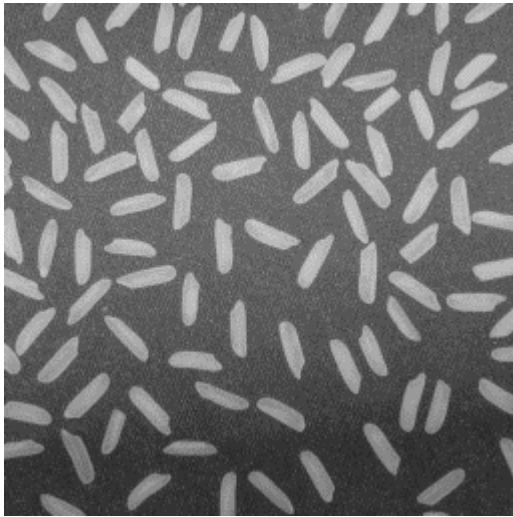
Green 영상



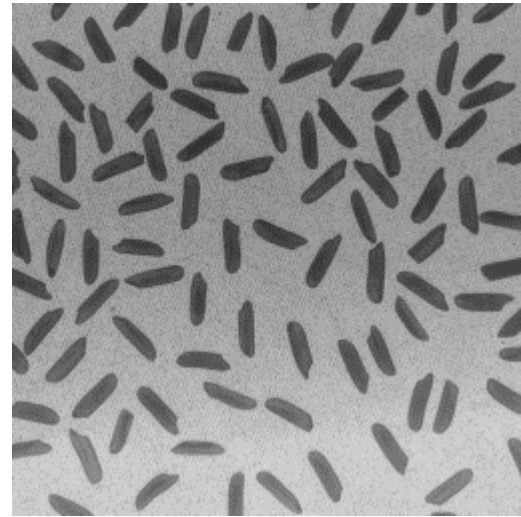
Blue 영상

1.2 흑백영상 역상 이미지 만들기

- 최고 계조 (255) – 입력 픽셀 값
- 입력 영상: rice.bmp (흑백영상)



원본영상



역상영상

1.3 Histogram Equalization

- 최고 계조 (255) – 입력 픽셀 값
- 입력 영상: citrus.bmp (흑백영상)



Citrus.bmp

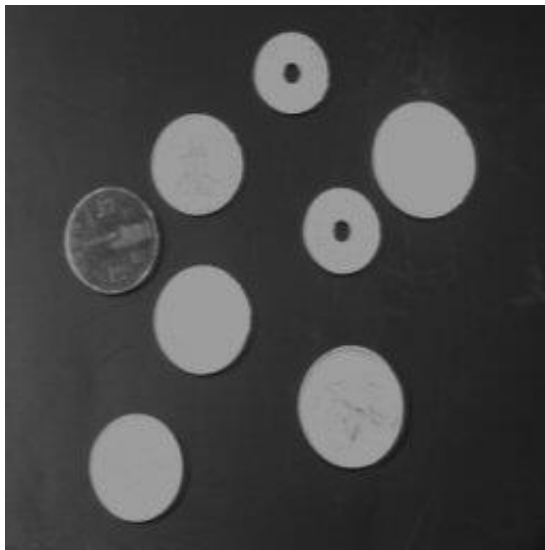
원본 영상



HistogramEq.bmp

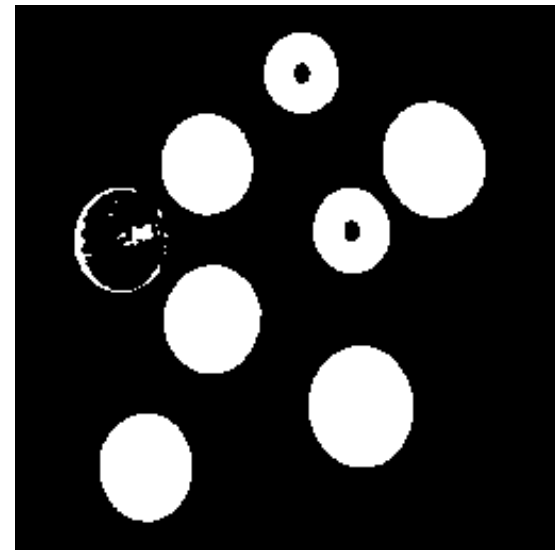
1.4 Otsu Binarization

- 최고 계조 (255) – 입력 픽셀 값
- 입력 영상: coin.bmp (흑백영상)



coin.bmp

원본영상



Otsu.bmp

결과영상

2. 설명

- Color 분리
- 역상 이미지 획득
- Histogram Equalization
- Otsu Binarization

2.0 코드 설명

#Color 영상 읽기

```
src =cv2.imread("Mandrill.bmp", IMREAD_COLOR);
```

#흑백 영상 읽기

```
src =cv2.imread("rice.png", IMREAD_GRAYSCALE);
```

#입력과 같은 타입의 영상 만들기

```
Import numpy as np  
  
H,W,C = src.shape[:]  
  
dst =np.zeros((H,W,C), src.dtype);
```

모든 요소의 값이 0으로 채워진 행렬을 만든다.

2.0 코드 설명

#영상 저장하기

```
dst =cv2.imwrite("원하는 경로/원하는 파일이름",src);
```

2.0 코드 설명

Color 영상 픽셀별로 접근

```
For y in range(h):  
    for x in range(w) :  
        for c in range(3):  
            src[y,x,c] = 255;
```

#2-2 흑백 영상 픽셀별로 접근

```
For y in range(h):  
    for x in range(w) :  
        src[y,x] = 255;
```

2.3 Histogram Equalization

1. calculate histogram

```
    loop over i ROWS of input image
      loop over j COLS of input image
        k = input_image[i][j]
        hist[k] = hist[k] + 1
      end loop over j
    end loop over i
```
2. calculate the sum of hist

```
    loop over i gray levels
      sum = sum + hist[i]
      sum_of_hist[i] = sum
    end loop over i
```
3. transform input image to output image

```
    area = area of image (ROWS x COLS)
    Dm = number of gray levels in output image
    loop over i ROWS
      loop over j COLS
        k = input_image[i][j]
        out_image[i][j] = (Dm/area) x sum_of_hist[k]
      end loop over j
    end loop over i
```

2.4 Otsu Binarization

$$\arg \min_t \sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

$$\begin{aligned} q_1(t) &= \sum_{i=1}^t P(i) & \mu_1(t) &= \sum_{i=1}^t \frac{iP(i)}{q_1(t)} & \sigma_1^2(t) &= \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \\ q_2(t) &= \sum_{i=t+1}^I P(i) & \mu_2(t) &= \sum_{i=t+1}^I \frac{iP(i)}{q_2(t)} & \sigma_2^2(t) &= \sum_{i=t+1}^I [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)} \end{aligned}$$

Otsu Algorithm

1. histogram $\bar{I} = \{h_{ij}\}$ ($h_{ij}[i]$)

2. $P[i]$ $\bar{I} = \{h_{ij}\}$

for i I

$$P[i] = \frac{h_{ij}[i]}{ROWS \cdot COLS}$$

I : gray level

3-어떤 $P[i]$ 값을 사용하든 무관하게 계산

3. otsu algorithm

for t I

for i t

$$f_1[t] = f_1[t] + P[i]$$

for z $[t+1, I]$

$$f_2[t] = f_2[t] + P[i]$$

1) $f_1(t), f_2(t)$ 계산

for i t

$$mu_1[t] = mu_1[t] + i \cdot \frac{P[i]}{f_1(t)}$$

for i $[t+1, I]$

$$mu_2[t] = mu_2[t] + i \cdot \frac{P[i]}{f_2(t)}$$

2) $M_1(t), M_2(t)$ 계산

for i t

$$sigma_1[t] = sigma_1[t] + (i - mu_1[t])^2 \cdot \frac{P[i]}{f_1[t]}$$

for i $[t+1, I]$

$$sigma_2[t] = sigma_2[t] + (i - mu_2[t])^2 \cdot \frac{P[i]}{f_2[t]}$$

3) $\sigma_1^2(t), \sigma_2^2(t)$ 계산

4. $\text{sigma}_w[t]$ $\hat{\sigma}_w^2$ ($\sigma_w^2(t)$)
for $t \in I$

$$\text{sigma}_w[t] = f_1[t] \text{sigma}_w[t] + f_2[t] \text{sigma}_w[t]$$

5. $\sigma_w^2(t)$ 를 최소로 만드는 t 값 \hat{t}

$$\arg \min_t \sigma_w^2(t)$$

// 신호 이동리듬 고민해보기

6. $\hat{\sigma}_w^2$ 와 \hat{t} 를 이용하여 이진화

3. 과제 제출방법

- 과제 제출함에

Lab#_본인학번_본인이름.pdf 제출.

- 보고서 작성

- 보고서 제목 및 형식 준수: Lab#_학번_이름.pdf

- 보고서에 포함되어야 하는 항목 :

- ✓ 문제에서 요구하는 각 세부 기능 설명
(문제당 소스코드 제외 1페이지 안으로 작성)
- ✓ 기능별 실행 화면 캡처
- ✓ 소스코드