

## Image.c

```
struct image
-   int w, h, c
-   float *data
```

### 1. 영상 할당

- a. `image make_image(int w, int h, int c)`
  - i. 역할: 메모리 할당 및 해당 pixel 0으로 채우기
- b. `image make_random_image(int w, int h, int c)`
  - i. 역할: 메모리 할당 0.5를 기준으로 random하게 채우기

### 2. 영상 데이터 변환

- a. `image float_to_image(int w, int h, int c, float *data)`
  - i. 역할: 배열을 이미지 데이터로 바꾸는 과정
  - ii. 위험 (w, h, c 수에 맞게 배열 사이즈가 맞아야 에러 발생 안함)

### 3. 이미지 복사

- a. `image copy_image(image p)`
  - i. 이미지 복사(w, h, c)까지
- b. `void copy_image_into(image src, image dest)`
  - i. 실제 데이터 복사 (src.h \* src.w \* src.c)
  - ii. 위험 (assert 도 없음) - 권장 안함

### 4. 영상 변환 함수

- a. `void transpose_image(image im)`
  - i. 역할 : transpose 됨
- b. `void rotate_image_cw(image im, int times)`
  - i. 역할: times 만큼 회전
  - ii. 이미지가 w, h 일치해야 함
- c. `image rotate_image(image im, float rad)`
  - i. 역할: 영상을 rad 단위로 회전시
- d. `void flip_image(image a)`
  - i. 역할: 이미지 좌우 반전
- e. `image image_distance(image a, image b)`
  - i. 역할: pixel 별 유클리디안 거리
  - ii. 위험 (이미지 사이즈가 다른 경우에도 연산 가능)
- f. `void fill_image(image m, float s)`
  - i. 역할: 영상을 픽셀값을 s 으로 settting 함
- g. `void translate_image(image m, float s)`
  - i. 역할: 영상을 픽셀값을 +s 만큼 증가
- h. `void scale_image(image m, float s)`
  - i. 역할: 영상을 픽셀값을 \*s 만큼 증가
- i. `void scale_image_channel(image im, int c, float v)`
  - i. 역할: 채널 c 에 해당하는 영상 맵에 \*v
- j. `void translate_image_channel(image im, int c, float v)`
  - i. 역할: 채널 c 에 해당하는 영상 맵에 +v

- k. void normalize\_image(image p): 0~1
- l. void normalize\_image2(image p): ??/
- m. void image threshold\_image(image im, float thresh)

#### 5. pixel 단위의 연산

- a. static float get\_pixel(image m, int x, int y, int c)
  - i. 위험: 범위 내 접근 아니면 assert 걸려 있음
- b. static float get\_pixel\_extend(image m, int x, int y, int c)
  - i. 안전함
- c. static void set\_pixel(image m, int x, int y, int c, float val)
  - i. 안전함
- d. static void add\_pixel(image m, int x, int y, int c, float val)
  - i. 위험: 범위내 접근이 아니면 assert 걸려있음
- e. static float bilinear\_interpolate(image im, float x, float y, int c)
  - i. 안전함
  - ii. 이미지가 연속 좌표를 가질 때, 가장 가까운 정수 좌표의 pixel value를 biliner로 유추함

#### 6. 색공간 변화

- a. void yuv\_to\_rgb(image im)
- b. void rgb\_to\_yuv(image im)
- c. void rgb\_to\_hsv(image im)
- d. void hsv\_to\_rgb(image im)
- e. void grayscale\_image\_3c (가중치 곱해진 형태)
  - i. 3채널 그레이 스케일로 바꿈
- f. image grayscale\_image()
  - i. 1채널 그레이 스케일 영상 반환
- g. image binarize\_image(image im)
  - i. 0.5 이하 0, 아니면 1

/

#### 7. 이미지 로딩함수

- a. image load\_image(char \*filename, int w, int h, int c)
  - i. 이미지 resize해서 읽어옴.

#### 8. 이미지 출력함수

- a. void print\_image(image m)
- b. void show\_image(image p, const char \*name)
  - i. opencv 컴파일 된경우만
  - ii. 그렇지 않으면 save 함수 호출하게됨

#### 9. 영상 저장

- a. void save\_image(image im, const char \*name)