

미니 프로젝트(Ver 2.0)

Color Image Processing

Using JavaScript & Html5

박영호

blog.naver.com/hkpyh



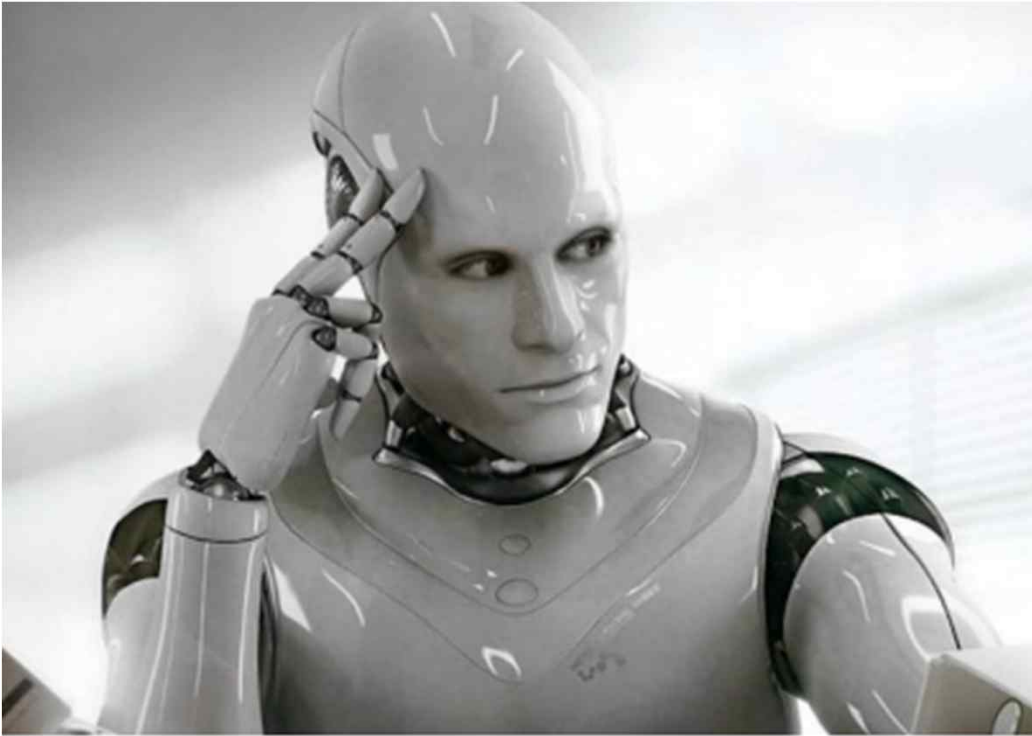
Contents

01 – Color Image Processing

02 – 개발 S/W 적용

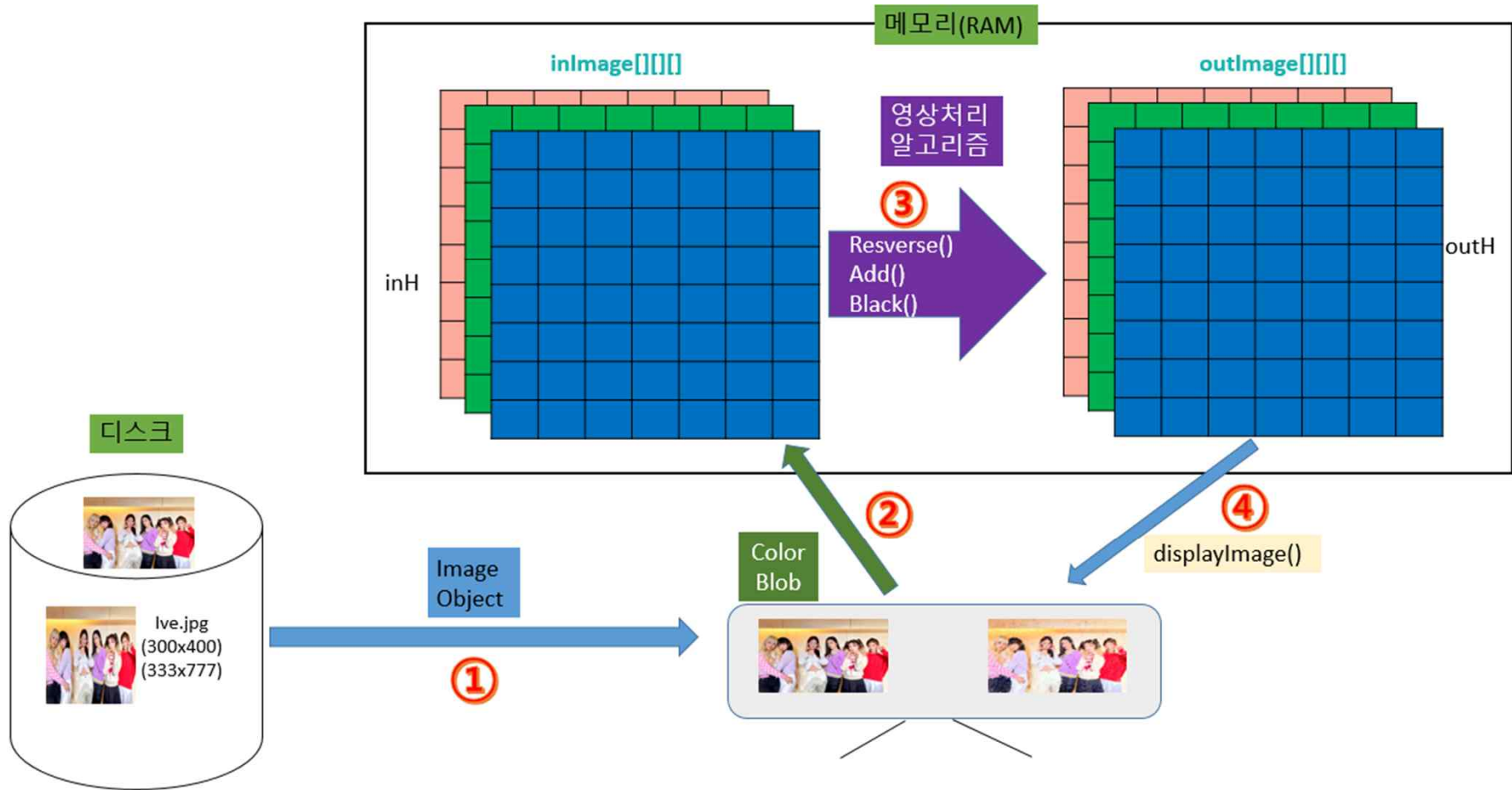
03 – 실제 활용 예시

04 – Future Plan

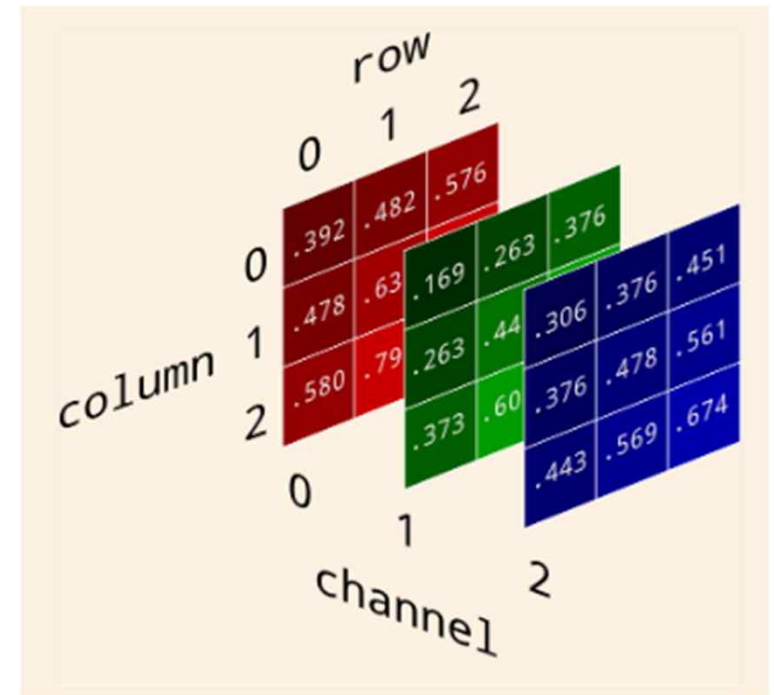


Color Image Processing

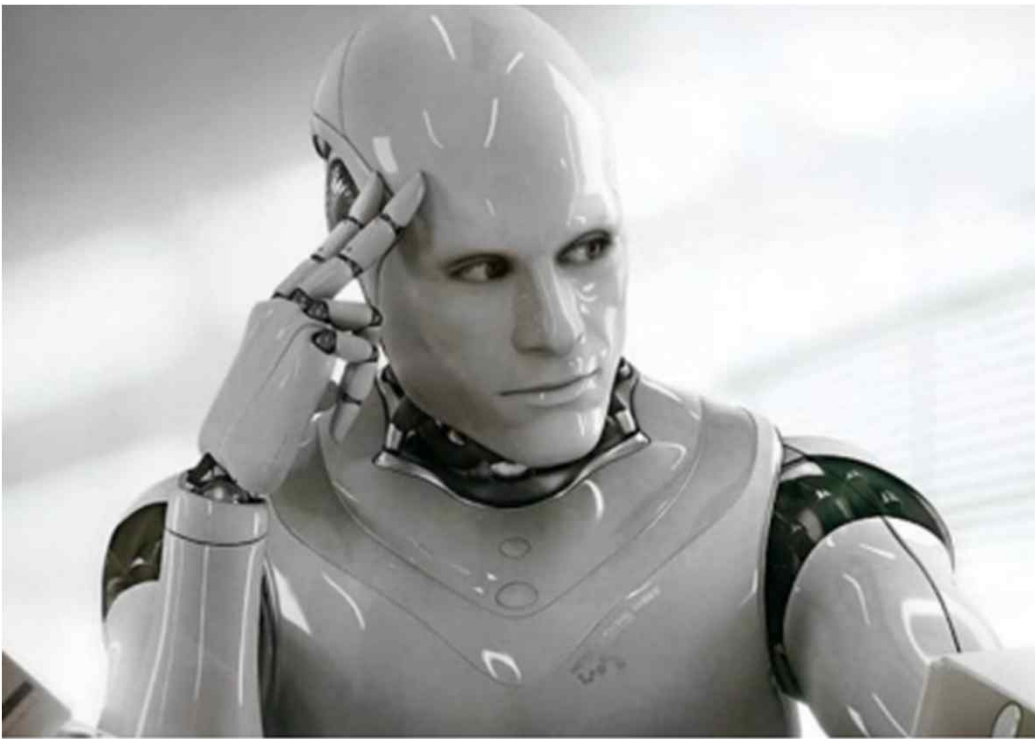
The Structure of Color Image Processing



The Structure of RGB Format

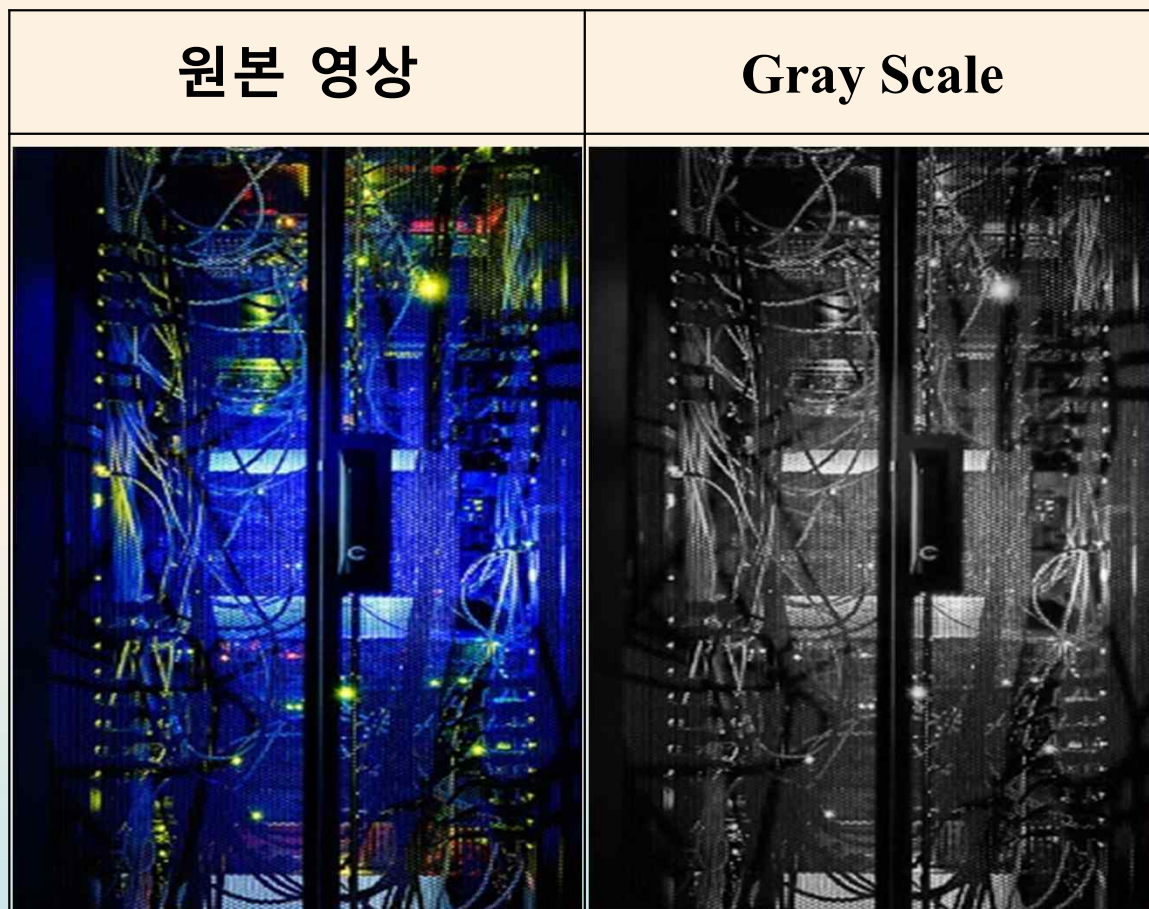


- RGB 포맷
 - 빛의 3원색 : Red, Green, Blue
 - R, G, B, 레이어에 저장된 각각의 사진 정보의 합



개발 S/W 적용

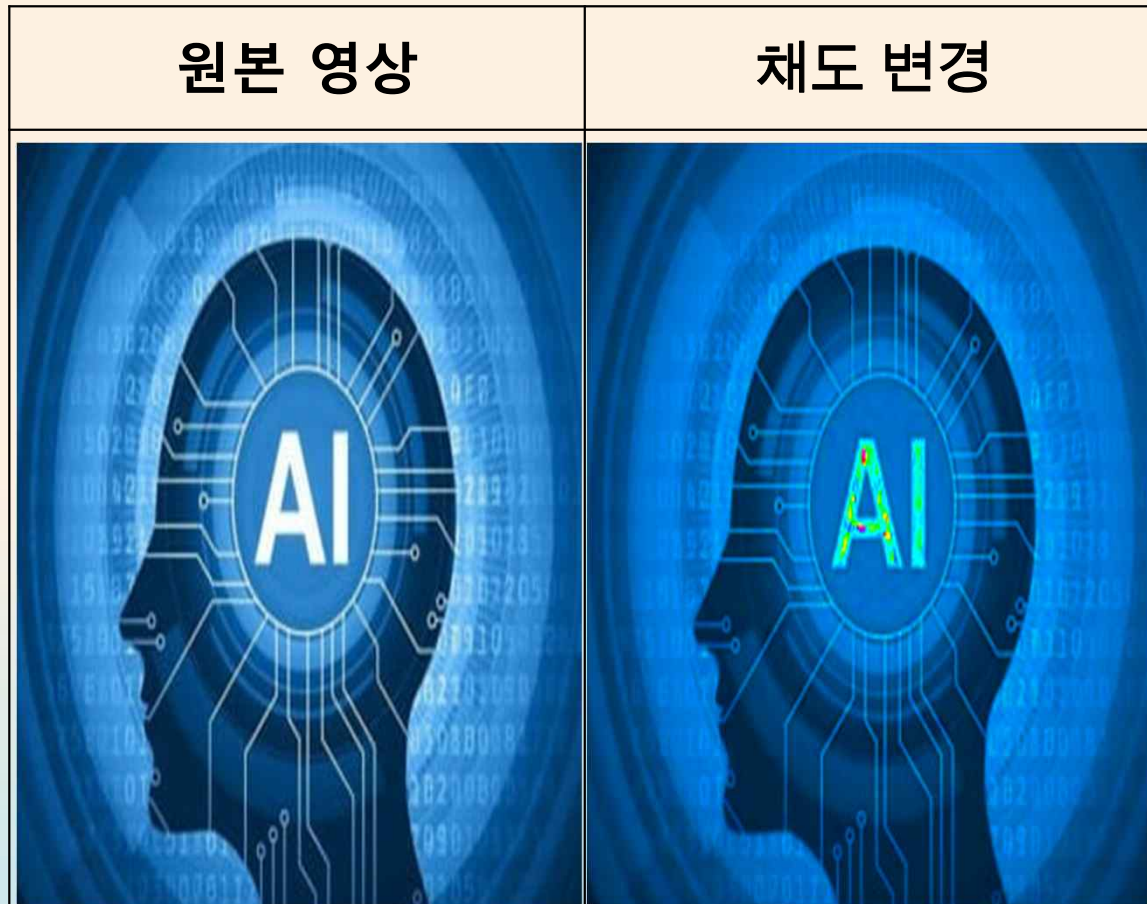
Color Image Processing(1) – Gray Scale



- Gray Scale
R,G,B의 평균 값을 출력

```
for (let i=0; i<inH; i++) {  
  for (let k=0; k<inW; k++) {  
    let sumValue = inImage[0][i][k]  
                  + inImage[1][i][k]  
                  + inImage[2][i][k]  
    let avgValue = sumValue / 3;  
  
    outImage[0][i][k] = avgValue;  
    outImage[1][i][k] = avgValue;  
    outImage[2][i][k] = avgValue;  
  }  
}
```

Color Image Processing(2) – 채도 변경



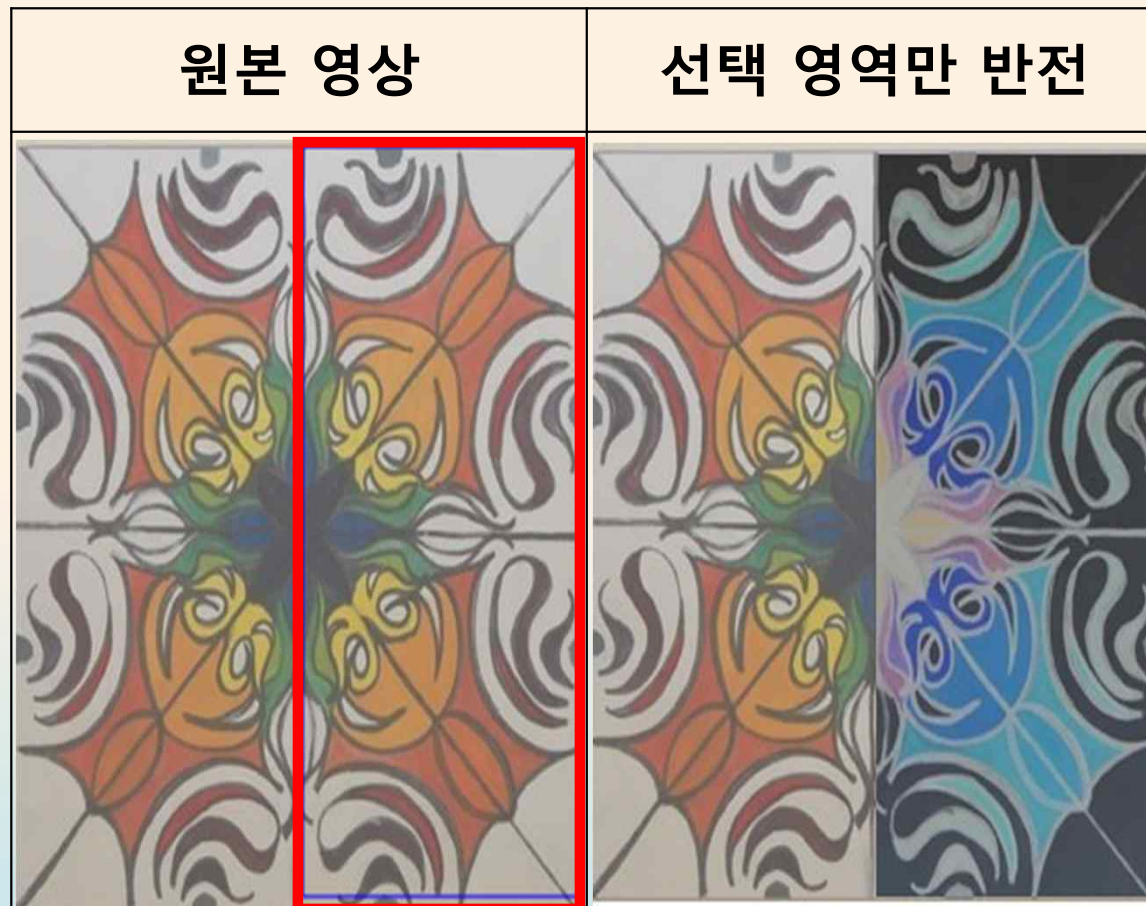
- 채도 변경
RGB → HSV로 변환 후
S 값 변경

```
let s_value = 채도 변경 값
for (let i=0; i<inH; i++) {
  for (let k=0; k<inW; k++) {
    let R = inImage[0][i][k];
    let G = inImage[1][i][k];
    let B = inImage[2][i][k];

    // RGB --> HSV
    let hsv = rgb2hsv(R,G,B);
    // 채도 변경 값 입력
    S = S + s_value;
    // HSV --> RGB
    let rgb = hsv2rgb(H,S,V);

    // 출력 영상에 넣기
    outImage[0][i][k] = R;
    outImage[1][i][k] = G;
    outImage[2][i][k] = B;
  }
}
```


Color Image Processing(3) – Event Listener



- 선택 영역(사진 우측)만 반전
Event Listener 활용

```
for (let i=0; i<inH; i++) {  
  for (let k=0; k<inW; k++) {  
    if ((startX <= k && k <= endX) &&  
        (startY <= i && i <= endY)) {  
      let sumValue = inImage[0][i][k]  
        + inImage[1][i][k] + inImage[2][i][k] ;  
      let avgValue = sumValue / 3;  
  
      if (avgValue > 127) {  
        outImage[0][i][k] = 255;  
        outImage[1][i][k] = 255;  
        outImage[2][i][k] = 255;  
      } else {  
        outImage[0][i][k] = 0;  
        outImage[1][i][k] = 0;  
        outImage[2][i][k] = 0;  
      }  
    } else {  
      outImage[0][i][k] = inImage[0][i][k];  
      outImage[1][i][k] = inImage[1][i][k];  
      outImage[2][i][k] = inImage[2][i][k];  
    }  
  }  
}
```

* Event Listener *

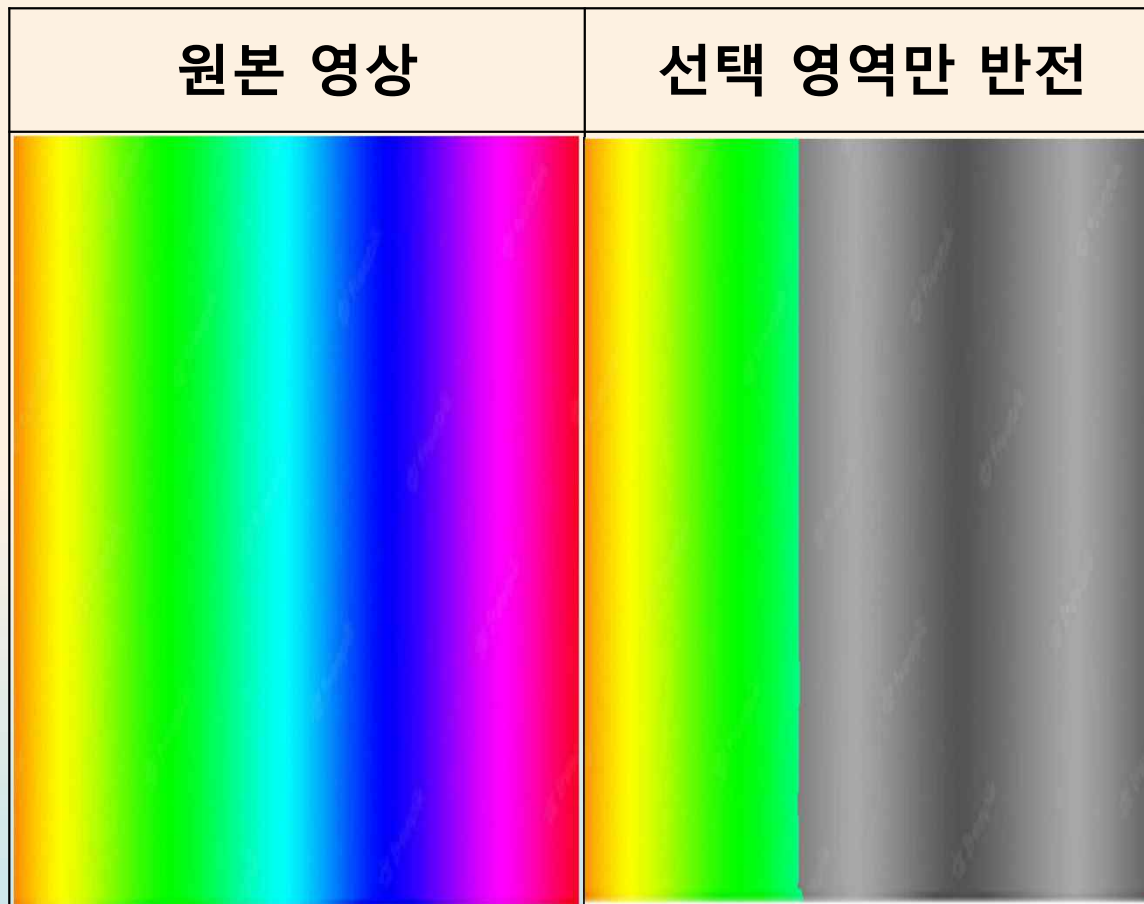
- Event Listener : 특정 이벤트에서만 동작하도록 하는 기능
 - Mouse 좌측 버튼 누를 때(down), 땄 때(up), 이동시(move) Listener 동작

```
inCanvas.addEventListener("mousedown", __downMouse, false);  
inCanvas.addEventListener("mouseup", __upMouse, false);  
inCanvas.addEventListener("mousemove", __moveMouse, false);
```

- Listener는 사용자의 입력 또는 특정 조건(시간 변화 등)이 발생할 때 개발자가 원하는 동작을 하도록 설정 가능
- (예시) 마우스를 눌렀다가 떴을 때 마우스의 좌표(offset X, Y)를 변수에 입력

```
function __upMouse(event) {  
    inCtx.putImageData(imageData, 0, 0);  
    endX = event.offsetX;  
    endY = event.offsetY;
```

Color Image Processing(4) – 이미지 추출



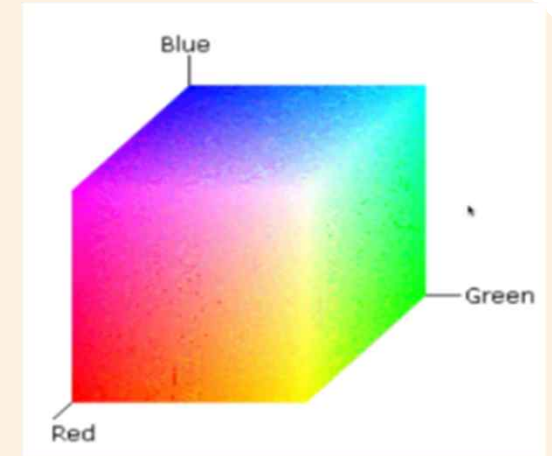
- RGB를 HSV 색상으로 변환
후 H값에 따라서 색상 추출
* 좌측은 H : 150의 예시임

```
if (col1 <= (H*360) && (H*360) <=col2) {  
    outImage[0][i][k] = R;  
    outImage[1][i][k] = G;  
    outImage[2][i][k] = B;  
} else {    // 나머지는 그레이 스케일  
    let avg = parseInt((R+G+B)/3);  
    outImage[0][i][k] = avg;  
    outImage[1][i][k] = avg;  
    outImage[2][i][k] = avg;  
}
```

* RGB & HSV 차이점 *

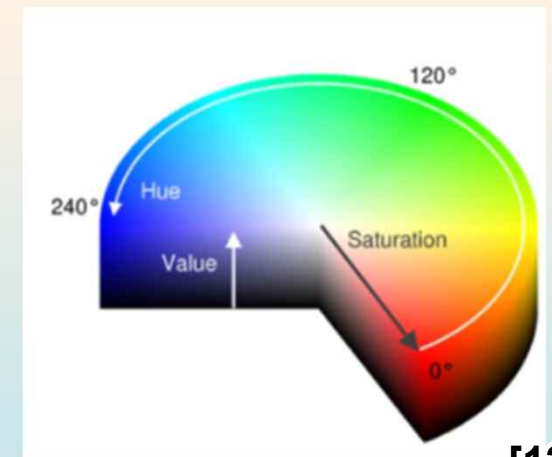
- RGB : Red(빨강), Green(초록), Blue(파랑)

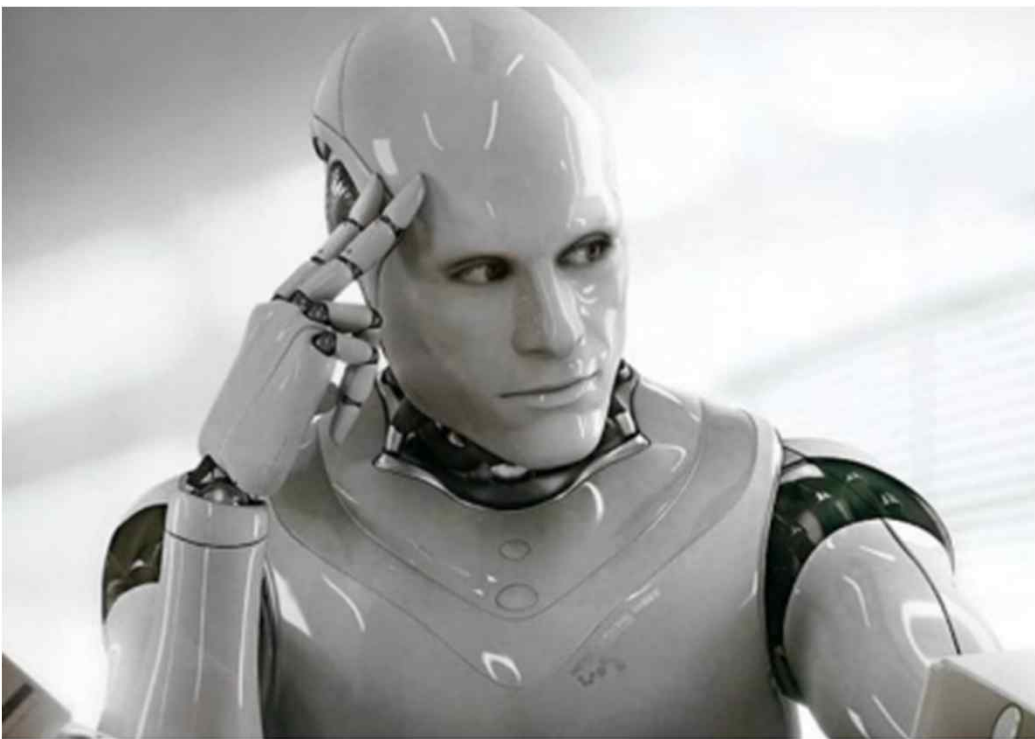
- 빛의 3원색. 색의 혼합으로 표시
- 모니터, TV 등에 표시할 때 사용
- * 인쇄 : RGB를 CMYK로 변환하는 과정 필요



- HSV : Hue(색), Saturation(진함), Value(밝기)

- 빛의 3원색. 색의 혼합으로 표시
- 본 프로젝트에서 색을 추출할 때 H값 활용
- * 0: 빨강 ~ 360: 보라





이미지 영상처리 실제 활용 예시

스마트폰 게임에서의 활용 예시(1)



- 게임명 : Dream Blast
- 게임방법
 - 같은 색깔의 아이콘 터치 시 아이템 생성
 - 정해진 회수 내로 블록 등을 처치시 성공
- 활용방법
 - 같은 색깔의 아이콘 찾기(Color 영상 처리)
 - 효율적인 방법으로 아이템 생성 및 블록 처치
(추후 AI관련 학습 이후 코딩 예정)

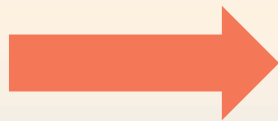
스마트폰 게임에서의 활용 예시(2)

- 각 블록 색을 추출하여 게임 화면에서 해당 블록만 표시
- 자동터치 기능 개발 및 적용 필요

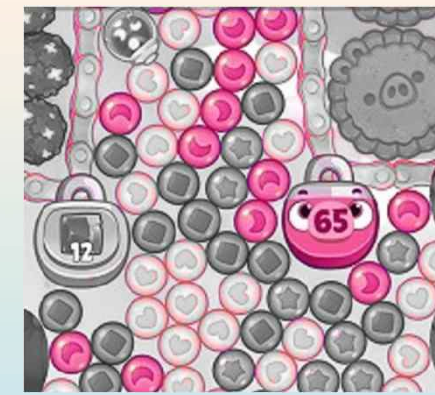
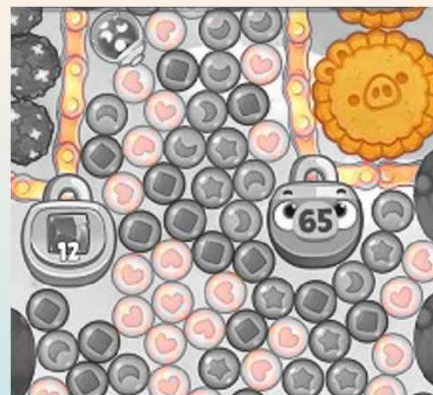
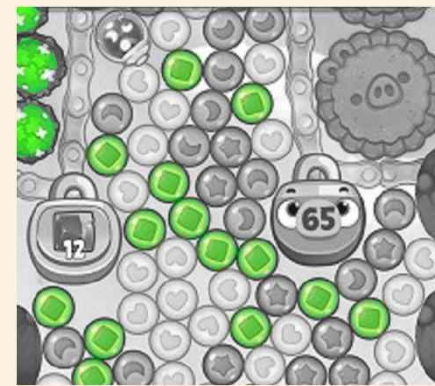
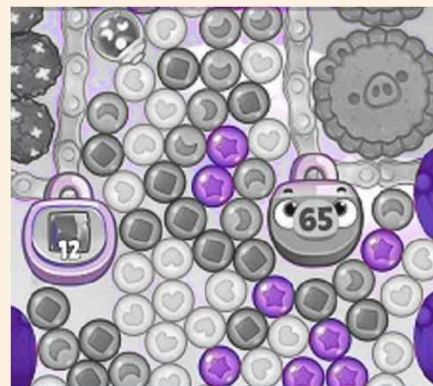


<실제 게임 화면>

picture99 선택 이후 아래 버튼을 누르세요



- 각 블록의 색깔만 인식 및 추출



Conclusion & Future Plan

'22. 9월

JavaScript
환경 구성
(완료)

'22. 9월 **현재 시점**

Color 영상
Code 작성(완료)

(추후 예정)

Web에 구현 및
상용화

'22. 9월

흑백(RAW) 영상
Code 작성(완료)

'22. 11월

동영상에
각종 효과 구현

<개선 필요 사항>

- Color 영상에 다양한 효과 적용 필요(PhotoShop 등 기존 S/W 참고)
- 홈페이지 개설 및 사용자 편의를 위한 화면 구성(JSP 학습 후 구현 예정)
- 상용화를 위한 추가 기능 개발(추출 영상 자동 터치 S/W 개발 및 구현)



THANK YOU