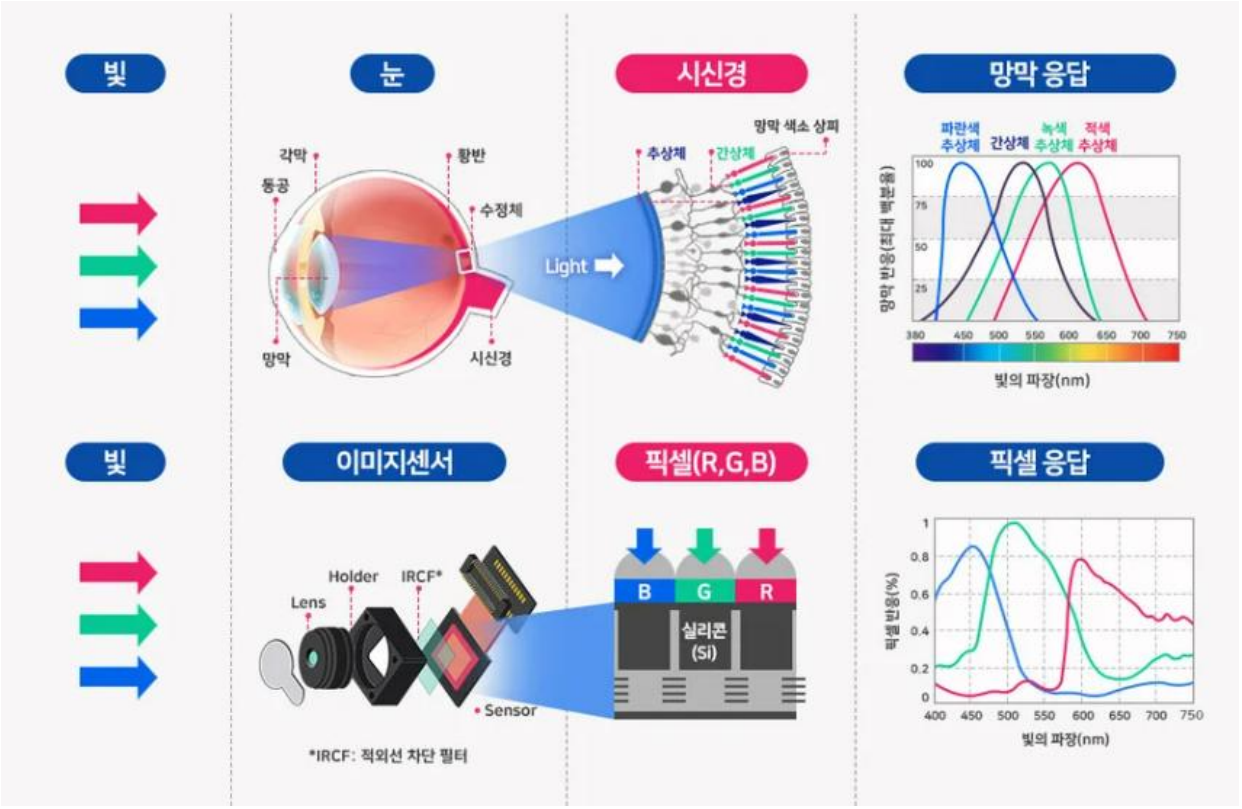
K-Digital Training

codingOn

머신 비전 기초

머신 비전 이란?

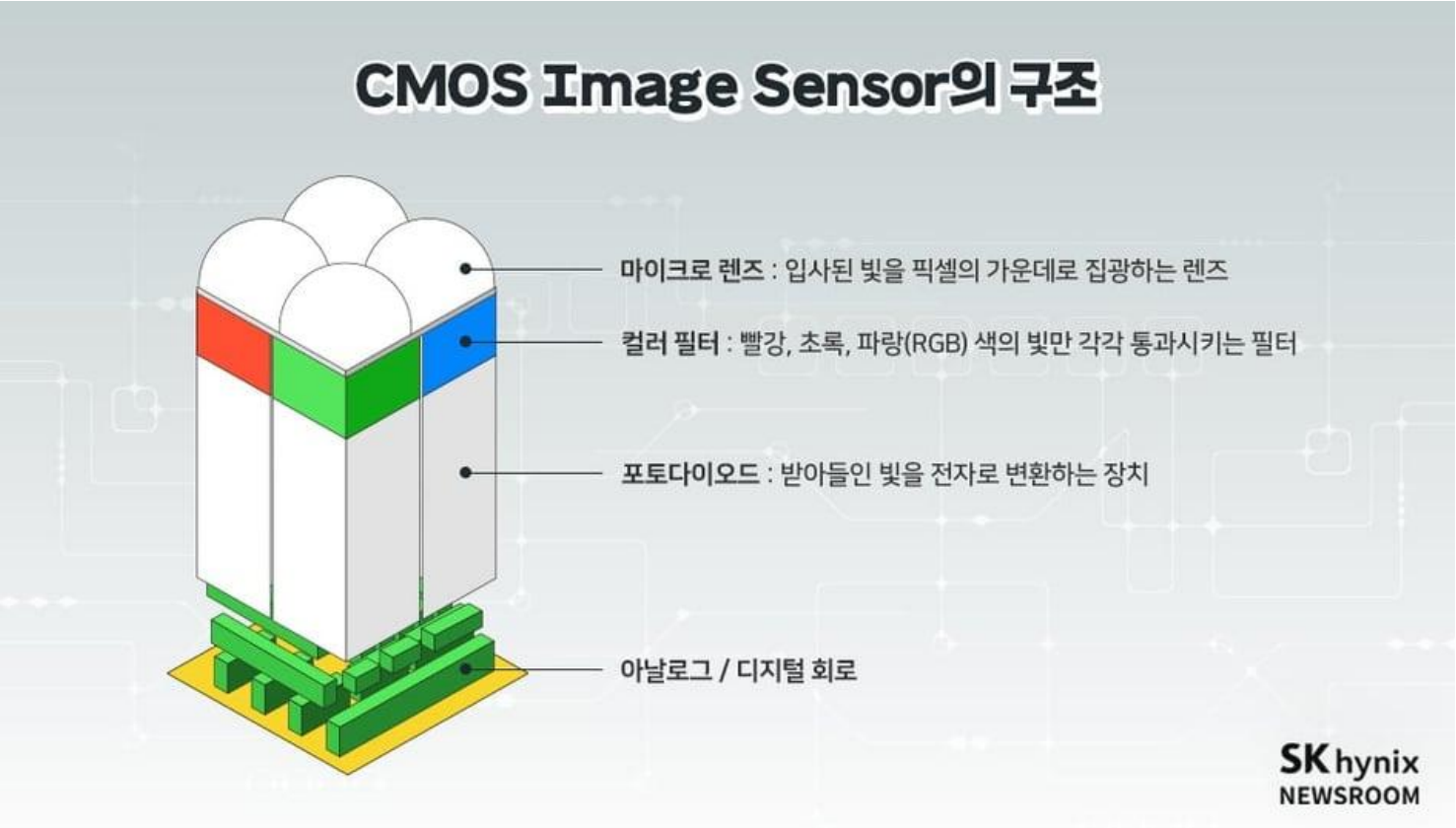
- 컴퓨터나 기계가 **시각적인 역할**을 처리할 수 있도록 연구하는 분야
- 인간이 시각정보를 보고 판단하듯, 컴퓨터는 **영상 데이터**를 보고 판단

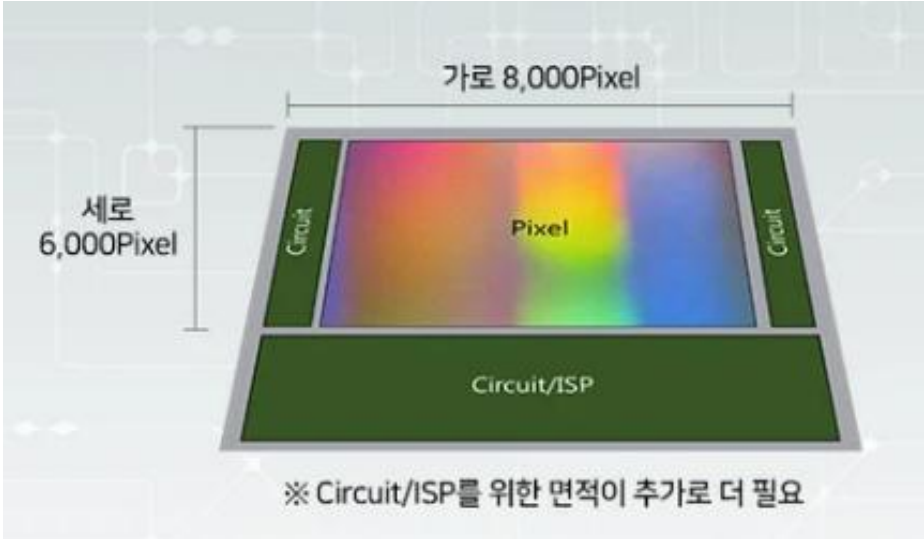


https://news.skhynix.co.kr/post/metavision-of-the-sensor

머신 비전 이란?

- 이미지 센서가 빛의 양(Intensity)을 감지하여 숫자로 변환





https://news.skhynix.co.kr/post/cmos-image-sensor-innovation-led-by-sk-hynix

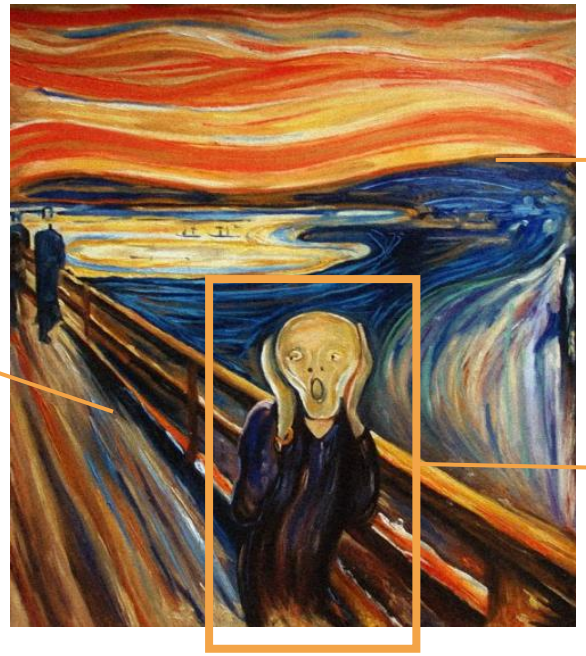
머신 비전 이란?

- 직접 각종 영상 입력 장치(주로 카메라)와 연결하고, 영상 처리(Image Processing) 기능을 하는 프로그램을 개발하는 것은 어려움
- 보통은 머신 비전용 라이브러리를 사용

이름	설명
OpenCV	무료, BSD 라이센스, 코드 공개 의무 없음, 상업적 이용 가능
Halcon	유료, 비쌈, 강력한 성능, 다양한 기능, 전문가 우대
MIL	유료, 적당히 비쌈, 국내에서 많이 사용
Vision Pro	유료, 코드 리딩 또는 패턴 매칭에 많이 사용

머신 비전 이란?

- 머신 비전은 인공지능의 한 분야
- 카메라, 스캐너와 같은 영상 매체를 통해 입력받은 이미지 또는 영상에서 **물체(Object), 전경(Foreground), 배경(Backgroud)**과 같은 유의미한 정보를 생성하는 기술



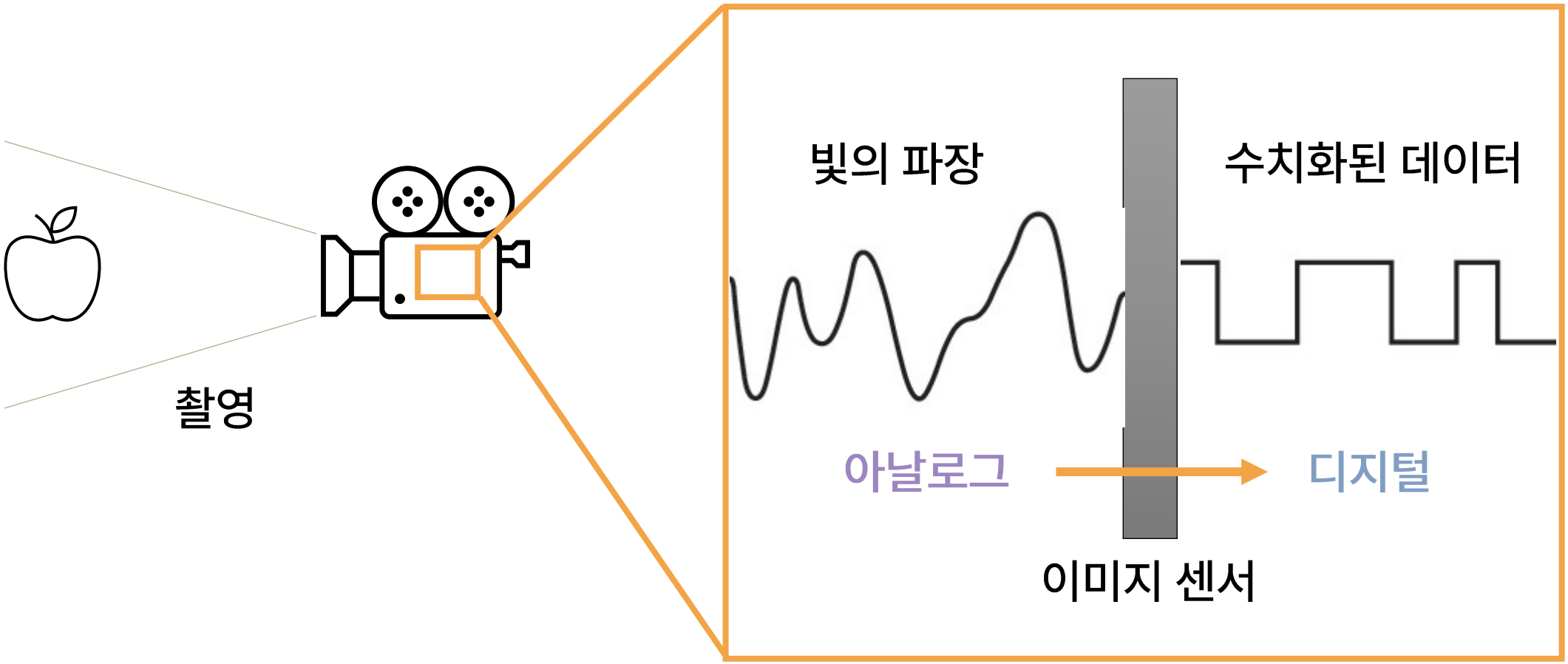
머신 비전 이란?

- 컴퓨터에게 사람과 같은 정보 해석 능력을 주는 것은 쉽지 않음

해변에 나뭇가지가 떨어져있음 → 해변에 통나무가 쓰러져있음
(사람을 보고 상대적인 크기 측정을 함)

비디오 파일이 만들어지는 과정

- 비디오 파일: 카메라를 통해 빛이 데이터로 변환된 것



아날로그/디지털

codingOn

아날로그/디지털

- 한국정보통신기술협회 용어사전

- **아날로그**
  - 연속적으로 변화하는 물리량을 표현하는 데 사용하는 용어

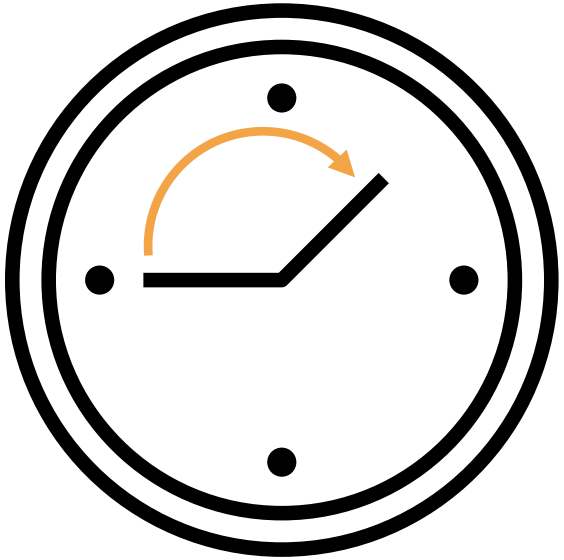
- **디지털**
  - 데이터나 물리적인 양을 0과 1이라는 2진 부호의 숫자로 표현하는 것

아날로그/디지털

codingOn

시계 바늘이 연속적으로 이동

09:00 14:00

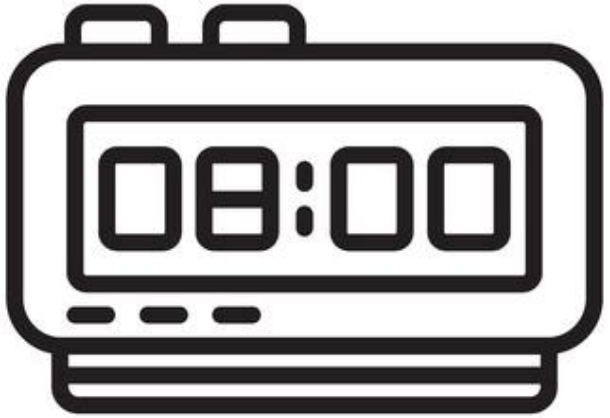


Analogue to Digital 변환

- 샘플링, Sampling
- 양자화, Quantization

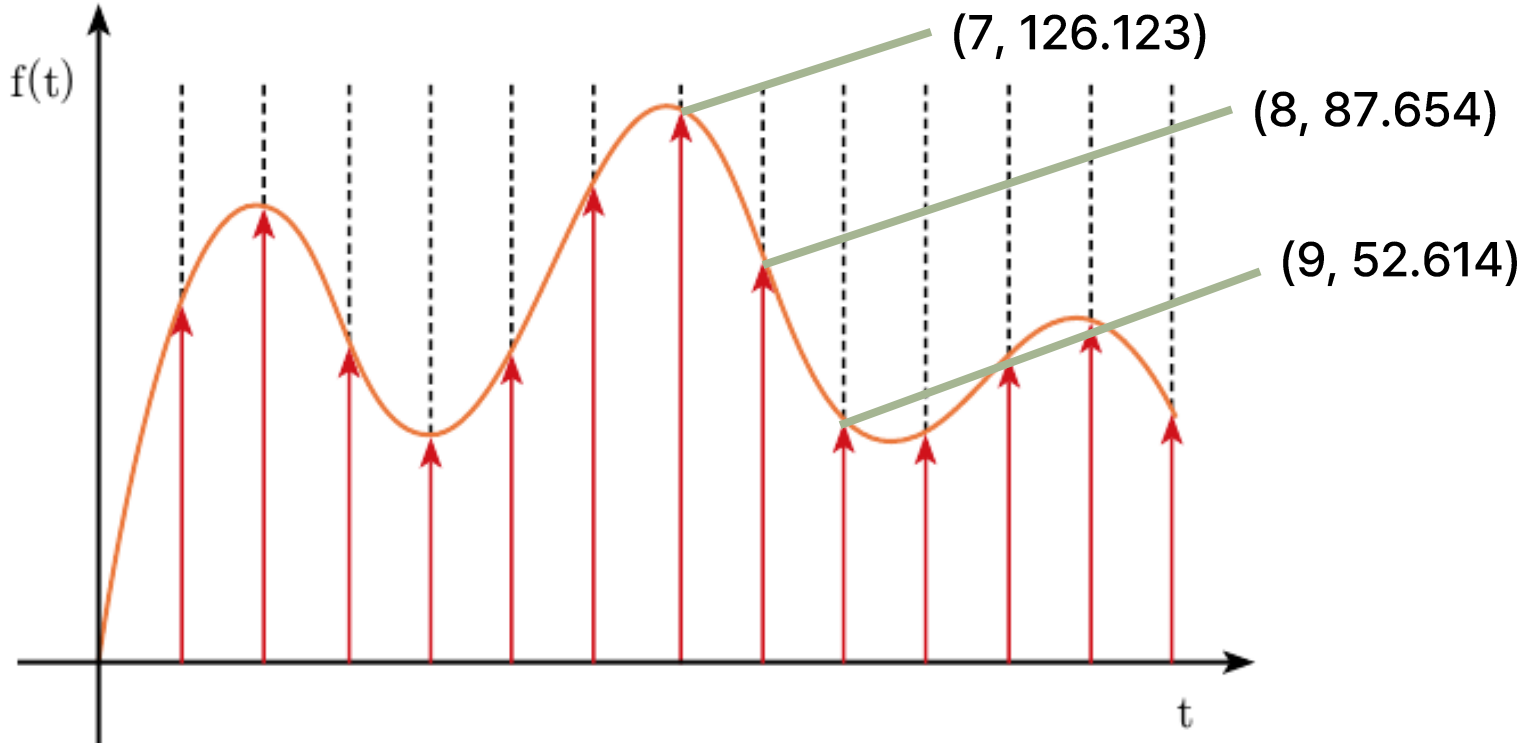
시계 숫자가 불연속적으로 변화

09:00 09:01 09:02



디지털 변환 - 샘플링, Sampling

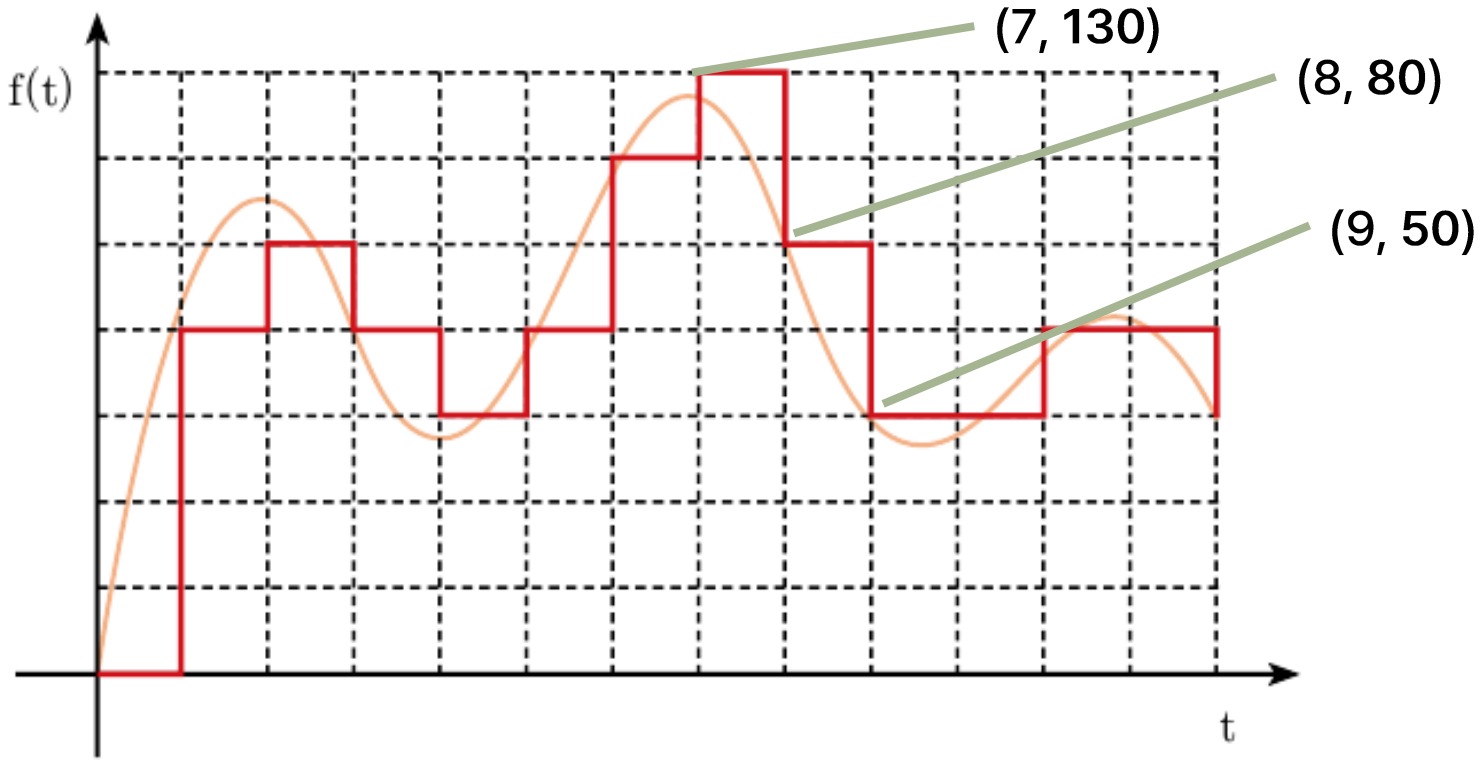
- 특정 주기(Frequency, 가로축)로 아날로그 데이터의 값을 기록
- 1초에 60번 샘플링(Sample rate) = 60Hz (FPS)
- 샘플링 더 촘촘하게 할 수록 실제 장면과 가까워짐



디지털 변환 - 양자화, Quantization

codingOn

- 특정 단위(Bit, 세로축)로 아날로그 데이터의 값을 수정
- 8bit 컬러 = 2의 8승 = 256, 24bit 컬러 = 2의 24승 = 더 자연스러움
- 양자화를 더 촘촘히 할 수록 실제 장면과 더 비슷해짐



퀴즈! Bitrate

- 아래 표는 유튜브에서 요구하는 비트 전송률(Bitrate)
- 1080p, 24FPS 영상을 업로드 한다고 했을 때 양자화 단위(Bit)는 얼마가 되어야 할까? (영상 압축은 하지 않음)

🎶 비트 전송률 (Bitrate)
= 해상도 x 초당 프레임 수 x 픽셀 단위 데이터 크기

SDR 업로드 시 권장 동영상 비트 전송률

4K 해상도의 신규 업로드 동영상을 보려면 VP9을 지원하는 기기나 브라우저를 사용하세요.

유형	동영상 비트 전송률, 표준 프레임 속도 (24, 25, 30)	동영상 비트 전송률, 높은 프레임 속도 (48, 50, 60)
8K	80~160Mbps	120~240Mbps
2160p(4K)	35~45Mbps	53~68Mbps
1440p(2K)	16Mbps	24Mbps
1080p	8Mbps	12Mbps
720p	5Mbps	7.5Mbps
480p	2.5Mbps	4Mbps
360p	1Mbps	1.5Mbps

디지털 변환

- 샘플링, 양자화를 거쳐 실제 장면이 **픽셀(Pixel)** 단위의 숫자로만 이루어진 영상 데이터로 변환됨

실습1. 영상 데이터 포맷

codingOn

- 아래 영상 데이터 포맷 관련 정보에 대해 조사 후 블로그에 정리하고 포스팅 링크 제출

- 영상 파일의 확장자가 의미하는 것 (컨테이너)
  - mp4, avi, mkv 등

- 실제 영상 데이터가 기록되는 방식 (인코딩, 디코딩)
  - h.264, xvid, hevc 등

codingOn

OpenCV 소개 및 설치

OpenCV란

codingOn

- Open Source Computer Vision Library
- Intel에서 최초 개발 (1999)
- C/C++, C#, Python, Java 지원
- 500가지가 넘는 영상 처리 알고리즘이 최적화 되어있음
- GPU 가속 모듈을 지원
- 머신러닝과 관련된 모듈을 포함
- 컴퓨터 비전 분야를 발전시키기 위한 목적

OpenCV란

codingOn

- OpenCV 2.X - iOS 및 안드로이드 지원 (2009)
- OpenCV 3.X - GPU 가속화(IPP) 지원 (2015)
- OpenCV 4.X - 안정성 및 메모리 소비 감소 (2018)
  - 사용하지 않는 OpenCV 1.X 기반 API 제거

일반적인 영상처리 과정

해결하기 원하는 문제에 대한
이미지 또는 영상 입력

각종 필터 및 마스킹을 통한
영상 노이즈 제거

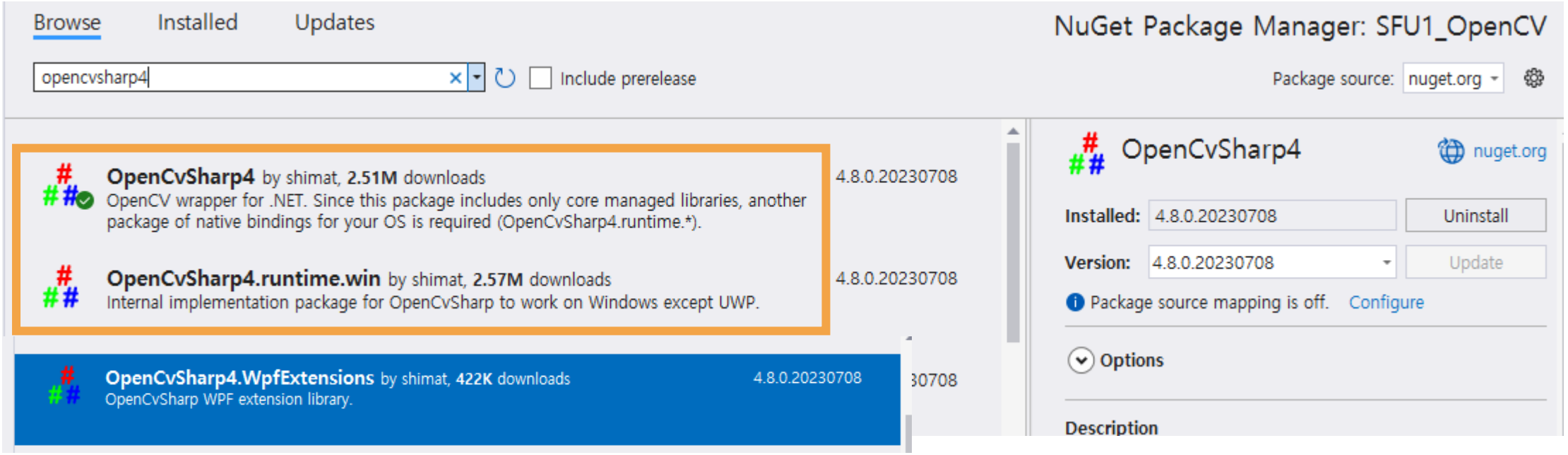
윤곽선, 코너, 특정 색 등
목표한 특징 검출

검출된 데이터를 기반으로
해석 및 문제 해결

목표한 최종 데이터 출력

OpenCVSharp 설치

- NuGet 패키지 설치
  - .Net framework 는 8.0이상
  - Visual Studio 2019 이상이 필요함
  - C#용 OpenCVSharp4 및 런타임 패키지, WpfExtensions를 설치



OpenCVSharp 설치

codingOn

- OpenCvSharp 네임스페이스 추가

```
using OpenCvSharp;

namespace SFU1_OpenCV
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    2 references
    public partial class MainWindow : System.Windows.Window
    {
        0 references
        public MainWindow()
        {
            InitializeComponent();
        }
    }
}
```

OpenCvSharp.Window와 충돌이 있으므로
네임스페이스 경로를 입력해줄 것

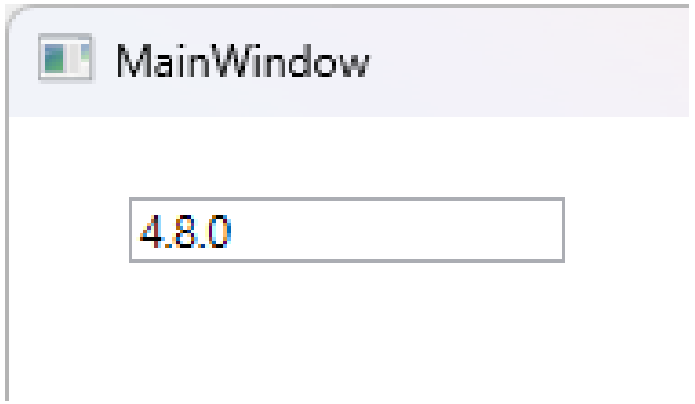
OpenCVSharp 설치

codingOn

- Cv2.GetVersionString() 메소드로 설치 확인

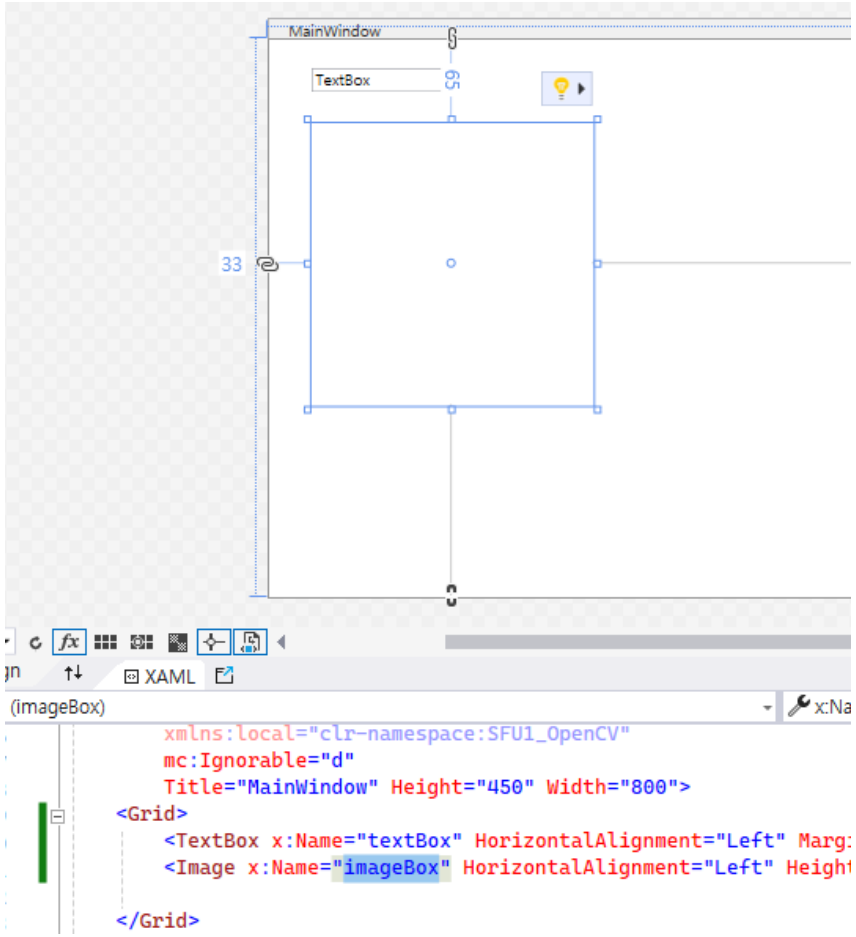
```
0 references
public MainWindow()
{
    InitializeComponent();

    textBox.Text = Cv2.GetVersionString();
}
```



OpenCV로 이미지 불러오기

- 이미지 박스 생성 및 코드 작성

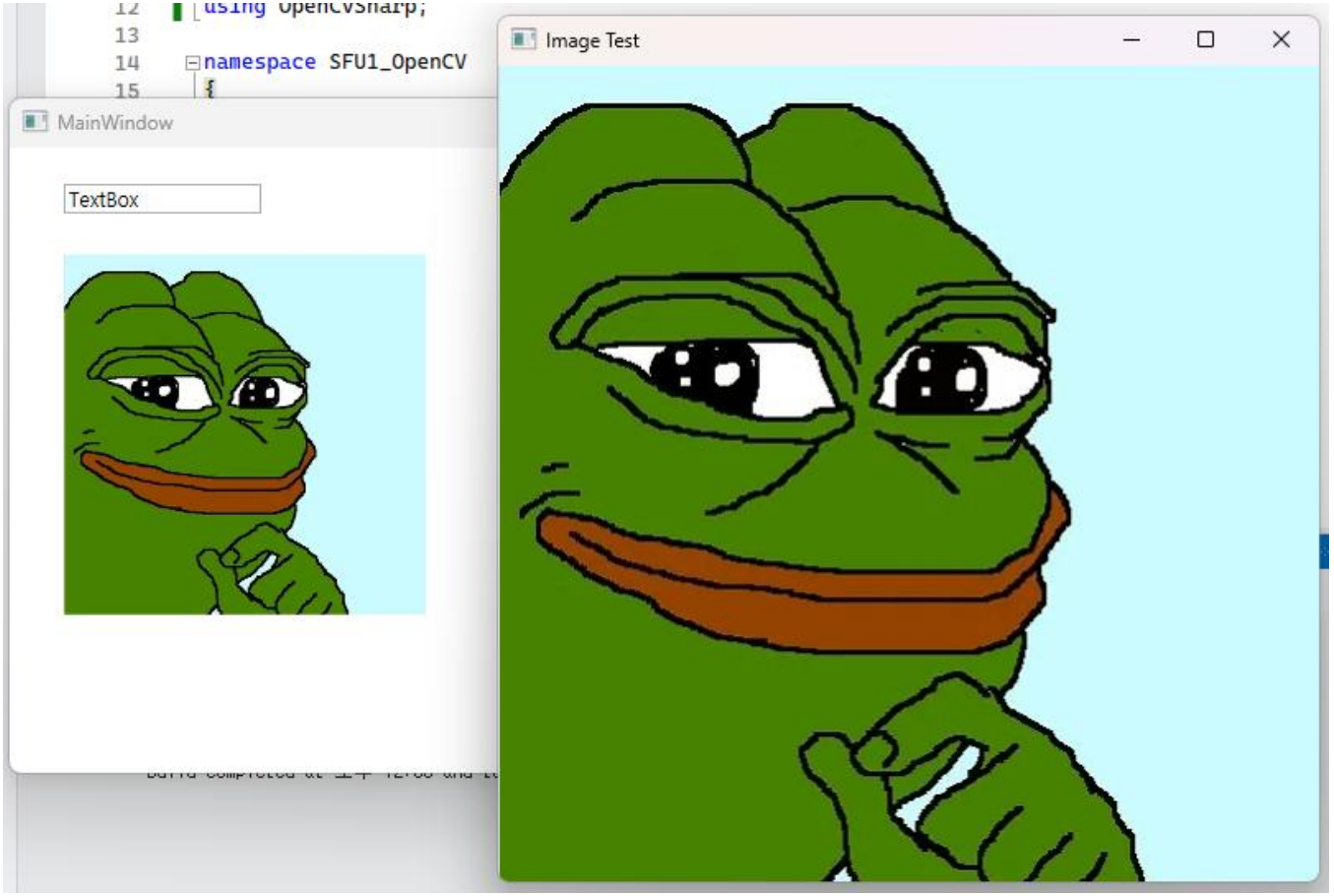



```
public partial class MainWindow : System.Windows.Window
{
    0 references
    public MainWindow()
    {
        InitializeComponent();

        Mat image = Cv2.ImRead("pepe.png");
        Cv2.ImShow("Image Test", image);
        imageBox.Source = OpenCvSharp.WpfExtensions.
            BitmapSourceConverter.ToBitmapSource(image);
    }
}
```

OpenCV로 이미지 불러오기

- Cv2.ImShow는 새창을 생성



실습2. OpenCvSharp 이미지 변환

codingOn

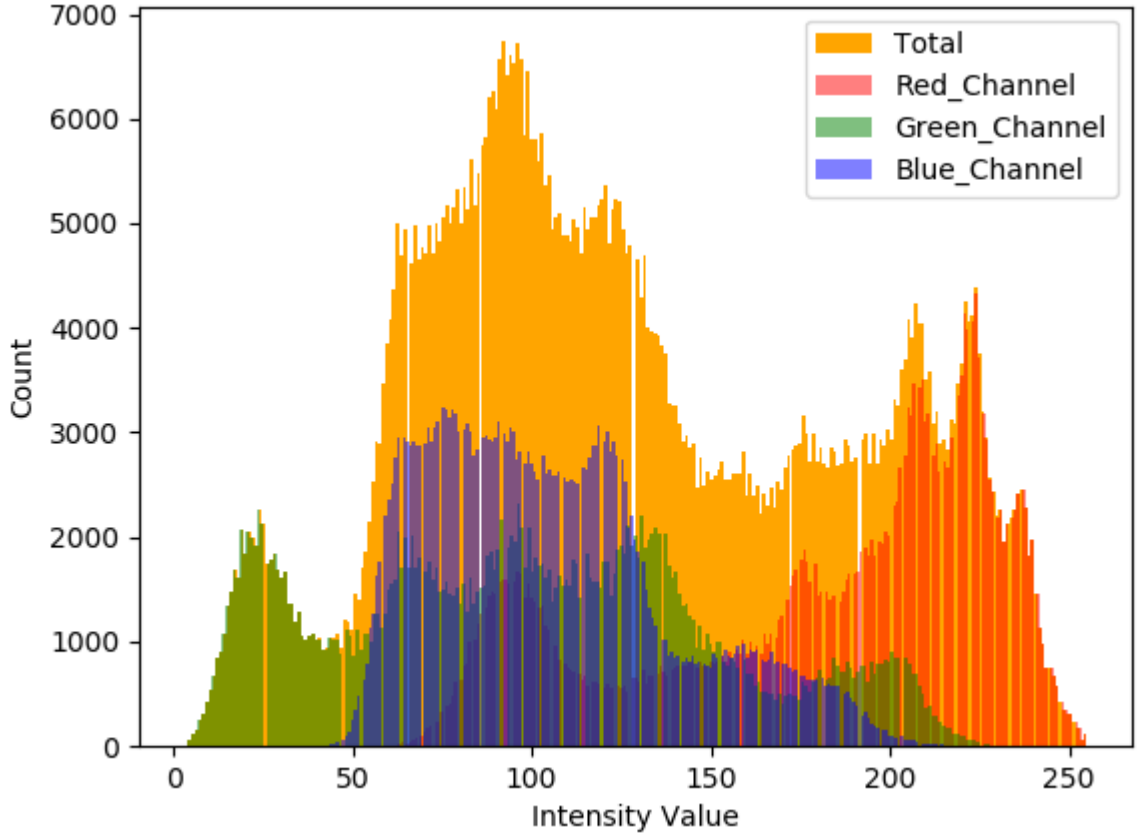
- Cv2.CvtColor(Mat source, Mat target, ColorConversionCodes) 메소드 사용하여 이미지를 간단하게 변환할 수 있음

- ColorConversionCodes의 종류를 조사하고 5개의 변환 옵션을 선택

- 이미지 박스를 6개 만들고 각각에 원본 이미지를 포함한 변환된 이미지를 표시

따라하기. 히스토그램 그리기

- 히스토그램
  - 같은 밝기를 가진 픽셀의 수를 막대 그래프로 표시한 것



따라하기. 히스토그램 그리기

codingOn

```
// Mat = Matrix = 2차원 행렬
Mat src = Cv2.ImRead(imageFilePath); // 원본 이미지
Mat gray = new Mat(); // 흑백 이미지
Mat hist = new Mat(); // 히스토그램

// 가로 256, 세로 이미지 크기 만큼 1로만 가득 차 있는 매트릭스 생성
Mat result = Mat.Ones(new OpenCvSharp.Size(256, src.Height), MatType.CV_8UC1); // CV_8UC1: 8bit 1channel 이미지
Mat dst = new Mat();

Cv2.CvtColor(src, gray, ColorConversionCodes.BGR2GRAY); // 원본 이미지 흑백으로 변환

// 히스토그램 값 생성
// 원본 매트릭스, 채널 수, 이미지 마스크, 출력 매트릭스, 차원 수, 히스토그램 크기(가로), 각 차원에 대한 색상값 범위
Cv2.CalcHist(new Mat[] { gray }, new int[] { 0 }, null, hist, 1, new int[] { 256 },
    new Rangef[] { new Rangef(0, 256) });
```

따라하기. 히스토그램 그리기

codingOn

```
// 값의 범위를 정규화 (최소값60~최대값210 이었다면, 최소값0~최대값255로 변환)
// 원본 매트릭스, 출력 매트릭스, 최소값(MinMax일 경우), 최대값(MinMax일 경우), 변환 방식
Cv2.Normalize(hist, hist, 0, 255, NormTypes.MinMax);

// 히스토그램 값을 갖고 값의 크기만큼 선의 길이를 정하고 선을 그림
for(int i = 0; i < hist.Rows; i++)
{
    Cv2.Line(result, new OpenCvSharp.Point(i, src.Height),
        new OpenCvSharp.Point(i, src.Height - hist.Get<float>(i)), Scalar.White);
}

// 가로로 매트릭스 두개를 붙이기 (높이가 같아야만 가능)
Cv2.HConcat(new Mat[] { gray, result }, dst);
Cv2.ImShow("dst", dst);

// 사용자 입력이 들어오면 창을 모두 닫기
Cv2.WaitKey(0);
Cv2.DestroyAllWindows();
```

따라하기. 히스토그램 그리기