

3차시: AI 현미경을 만나다.

1. 탐구 목표

- ✓ 배율과 분해능의 개념과 현미경의 구조를 설명할 수 있다.
- ✓ 라즈베리파이를 이용하여 디지털 현미경을 제작할 수 있다.
- ✓ 직접 제작한 현미경을 이용하여 백혈구 세포를 분류하는 인공지능 디지털 현미경을 구현할 수 있다.

2. 현미경에 대해 알아보까요?

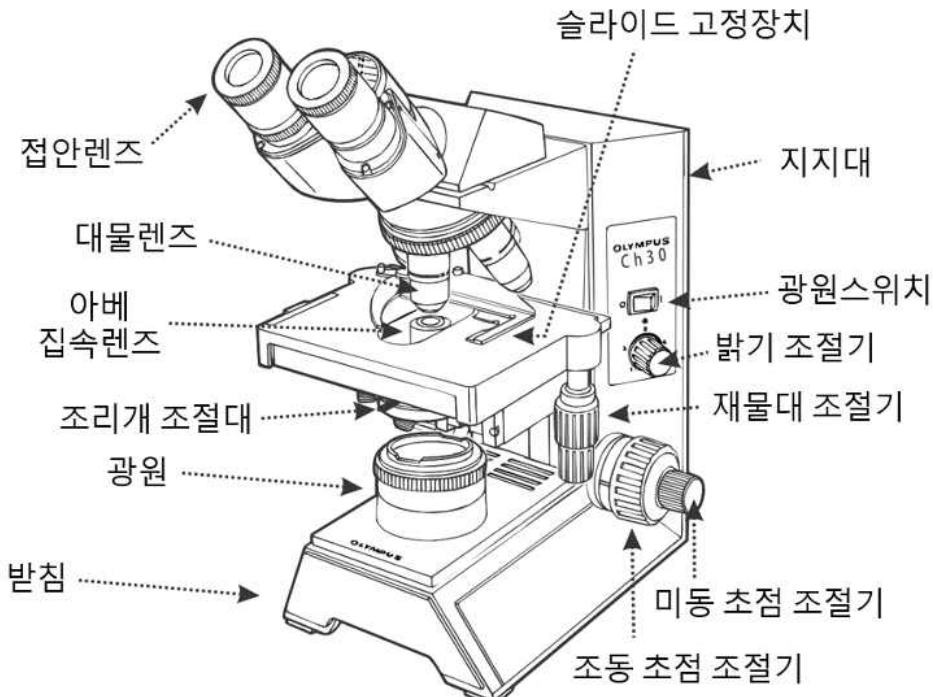
2-1. 현미경이란?

- 아주 작은 물체를 뚜렷하게 볼 수 있게 물체를 크게 보이도록하는 도구를 말합니다.
- 한 개의 렌즈나 여러 개의 렌즈를 조합하여 만들어진 광학장치로 작은 물체를 확대하여 보여줍니다.



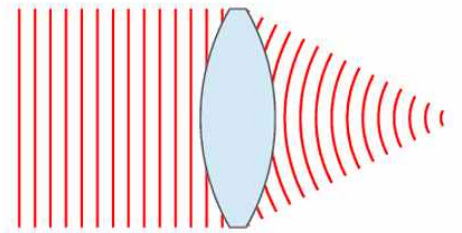
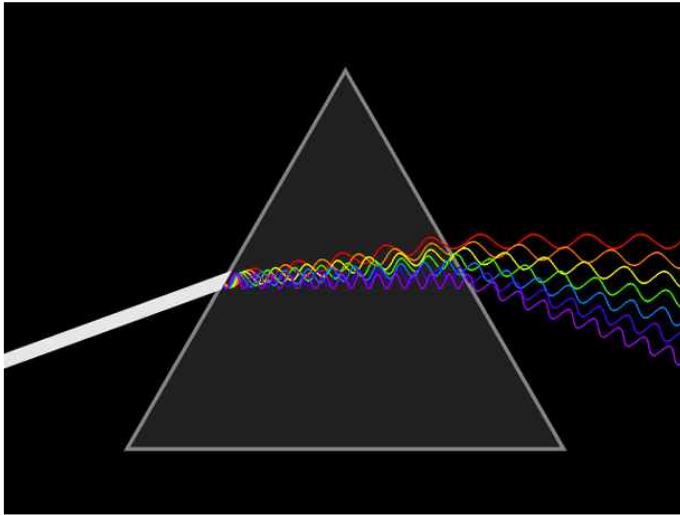
17세기 현미경(출처: Leica)

2-2. 현미경의 구조

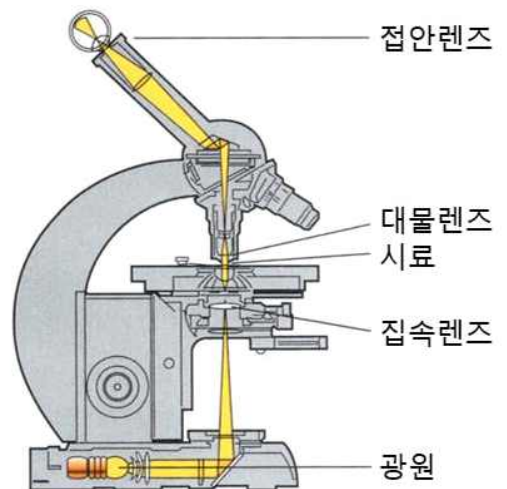
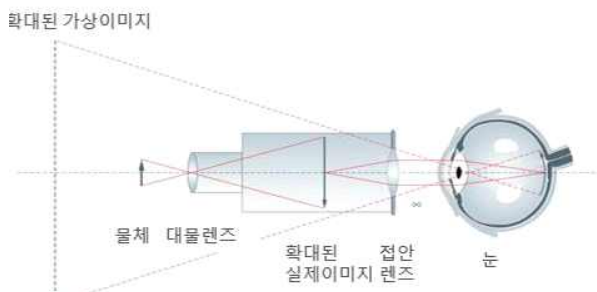
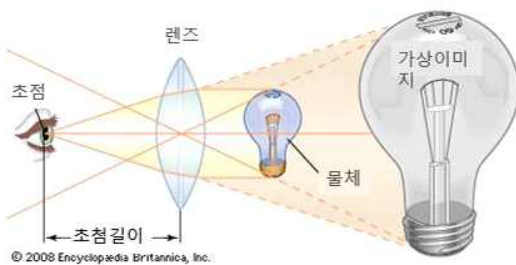


2-3. 렌즈와 빛의 굴절

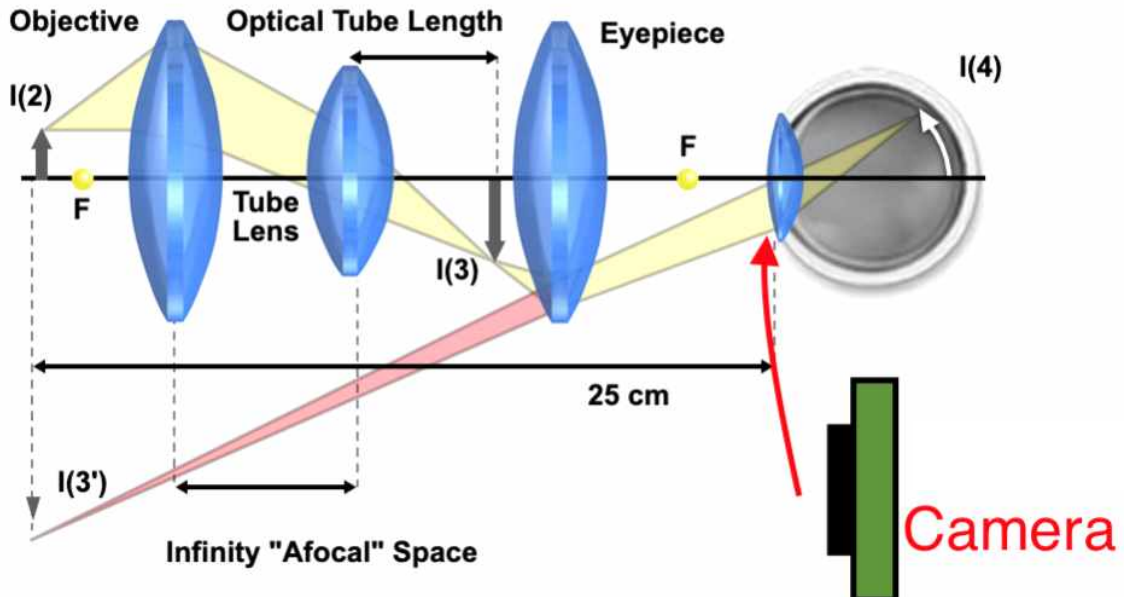
- 빛이 프리즘에 들어오면 빛의 파장에 따라 굴절되는 정도가 달라 다양한 색깔의 스펙트럼으로 나누어집니다.
- 볼록 렌즈를 통과한 빛은 렌즈의 곡률 때문에 빛이 한 곳으로 모아집니다. 이러한 특징을 이용하면 물체를 확대하고자 특정 초점에 빛을 모을 수도 있습니다.



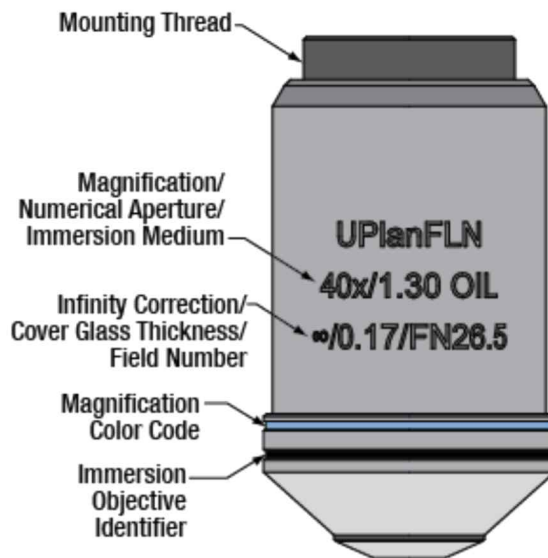
2-4. 렌즈를 이용한 확대 원리



2-5. 광학 현미경의 기본 구조

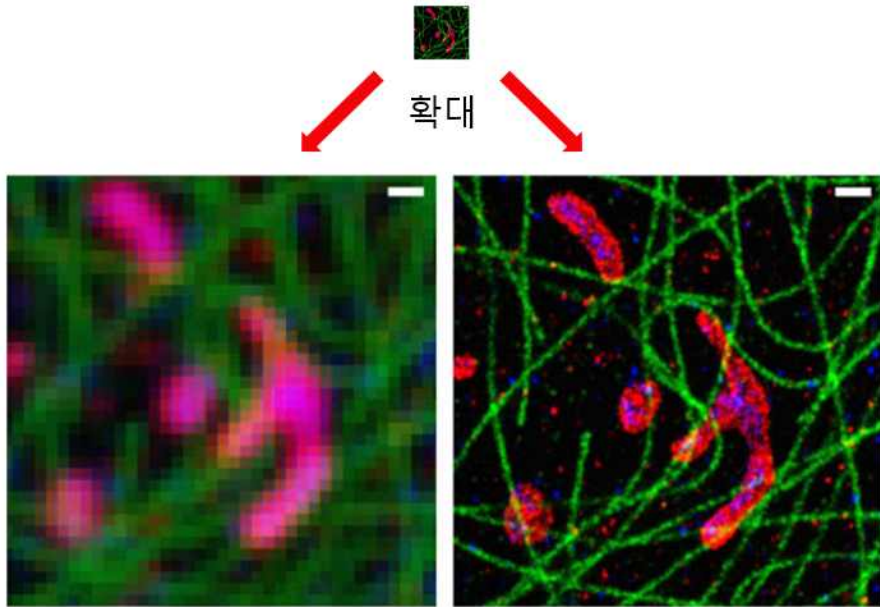


2-6. 대물렌즈 알아보기

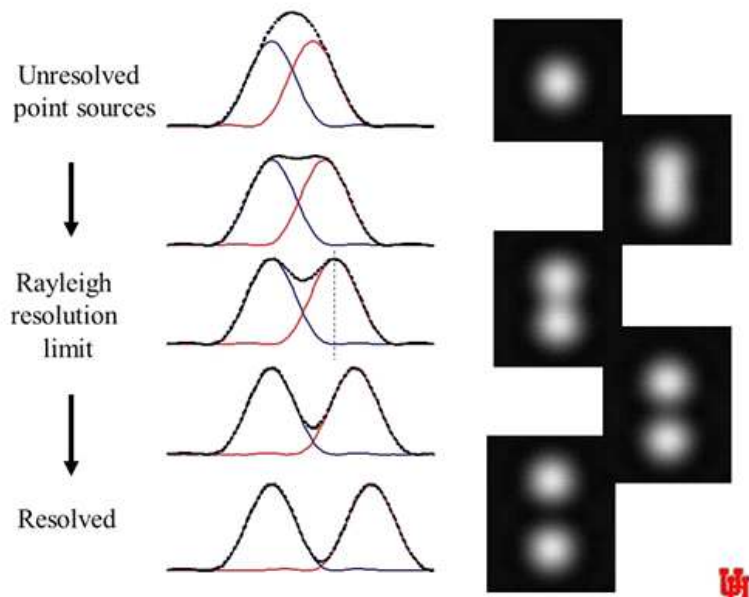


2-7. 배율과 분해능

- 배율: 단순히 이미지가 얼마나 크게 확대되었는지를 알려줍니다. 예를 들어, 배율이 10배인 현미경은 우리가 보는 물체를 실제 크기보다 10배 더 크게 보여줍니다.

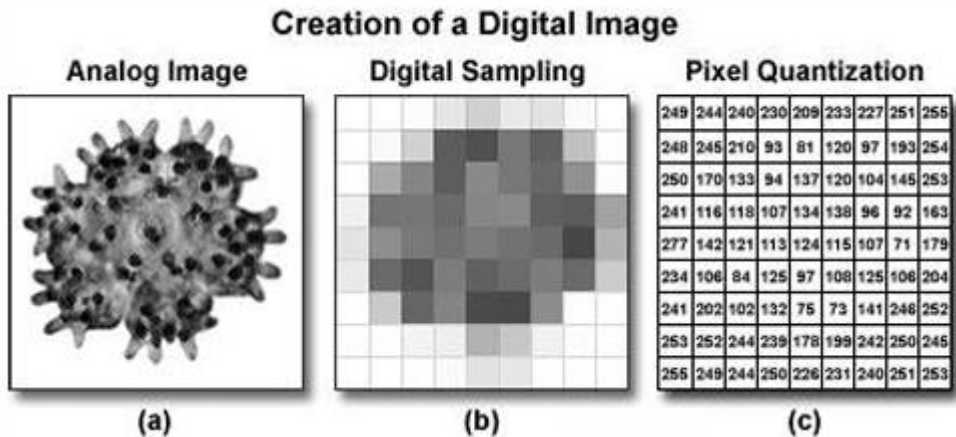


- 분해능: 가까이 있는 두 점을 얼마나 뚜렷하게 구별할 수 있는지를 알려줍니다. 예를 들어, 두 개의 작은 점이 너무 가까이 있으면 하나의 점으로 보일 수 있는데, 분해능이 좋으면 이 두 점을 분리해서 볼 수 있습니다.

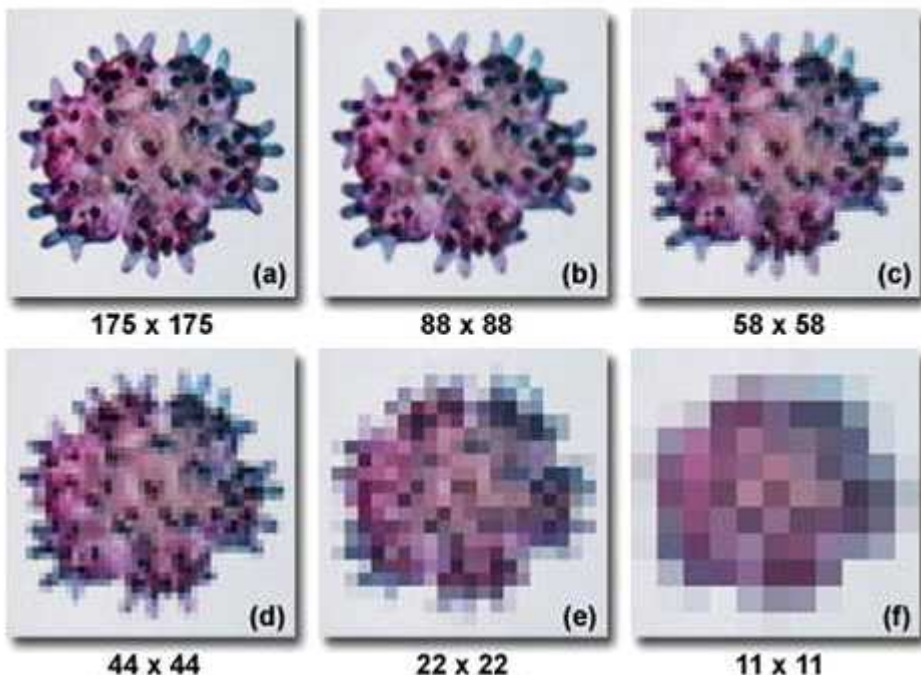


2-8. 디지털 이미지

- 우리가 보는 모든 사진이나 화면은 픽셀이라고 부르는 점들로 이루어져 있습니다. 픽셀(Pixel)은 Picture Element의 줄임말로 이미지를 구성하는 아주 작은 점을 말합니다. 디지털 이미지는 각 픽셀에 숫자가 쓰여져 있는데, 그 값은 0부터 255까지의 범위를 가질 수 있습니다.

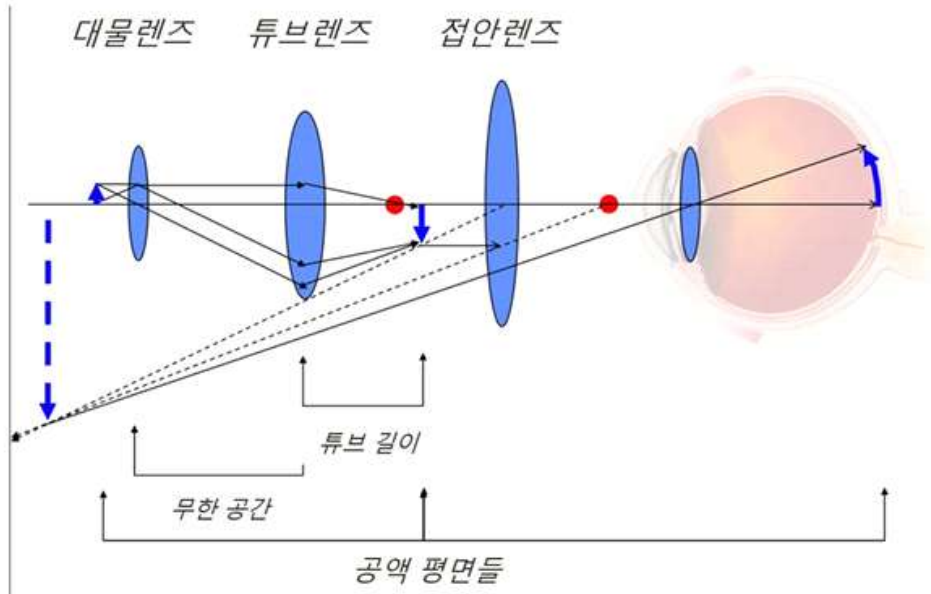


- 픽셀이 많을수록 사진이 더 선명하게 보입니다. 각 픽셀의 숫자값은 픽셀의 밝기에 대한 정보를 가지고 있습니다.
- 컬러 이미지의 경우는 각 픽셀에 대해 RGB라는 세 가지 채널을 별도로 가지고 있습니다. RGB는 빨강(Red), 초록(Green), 파랑(Blue)를 의미합니다.

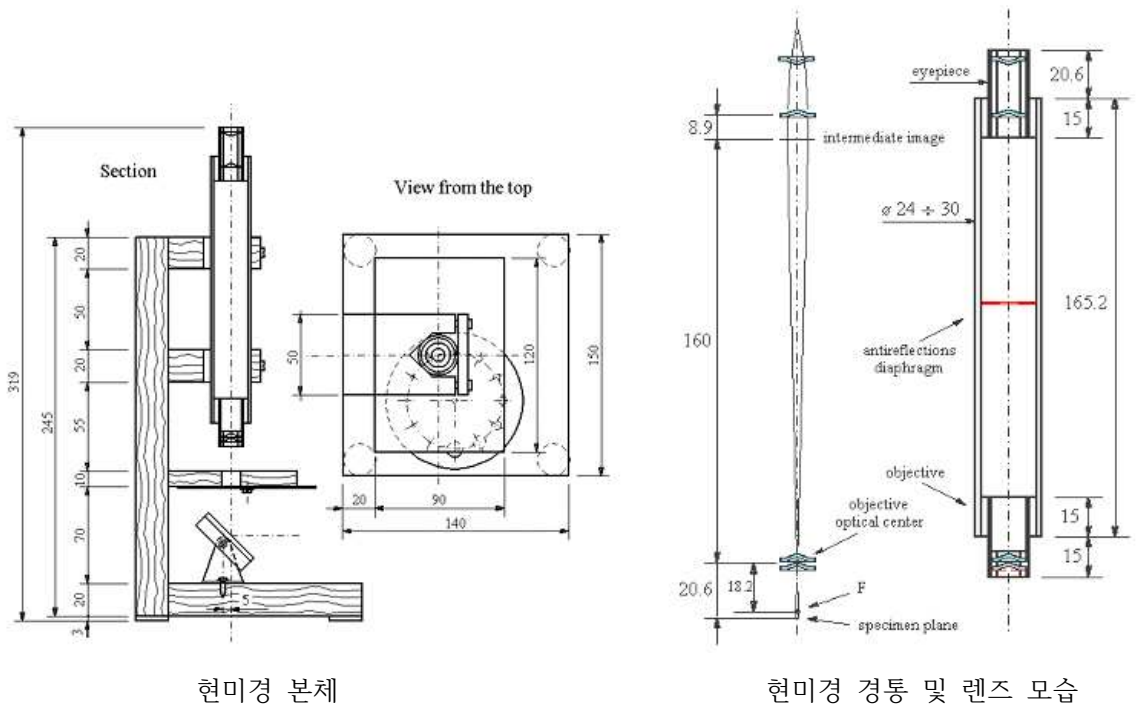


3. 라즈베리파이 디지털 현미경을 직접 만들어 볼까요?

3-1. 현미경의 렌즈 구성



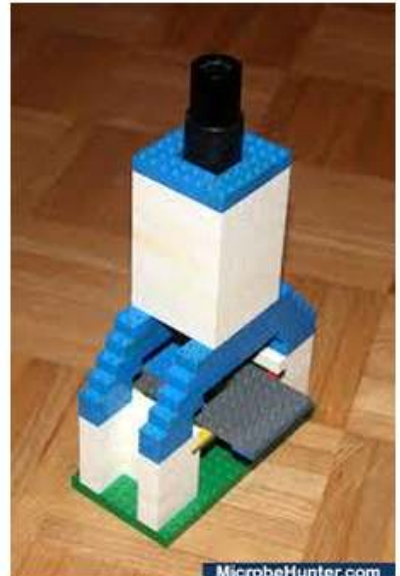
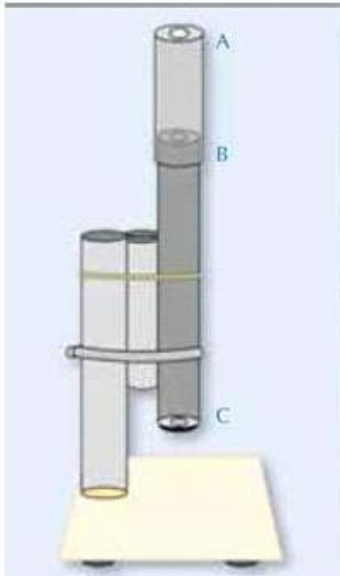
3-2. 현미경 구조도



현미경 본체

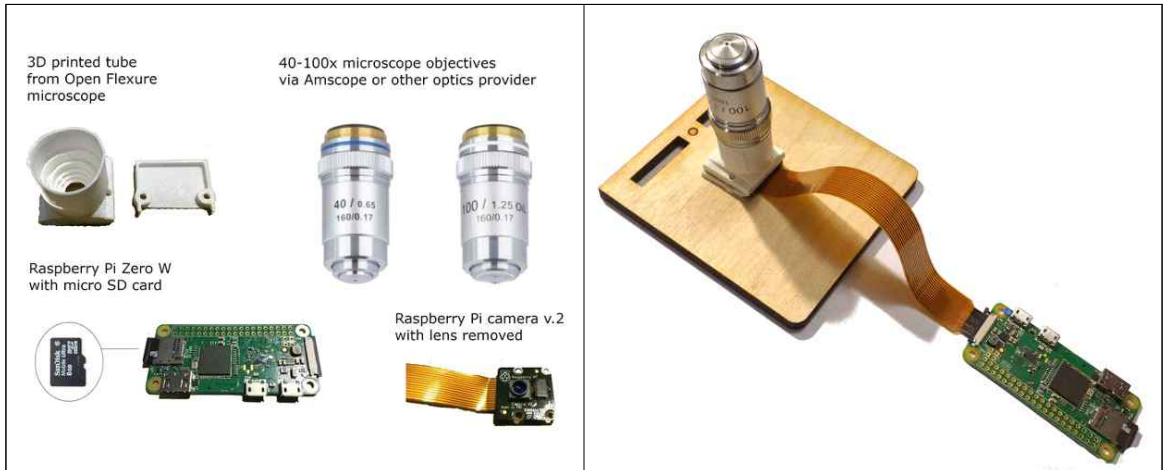
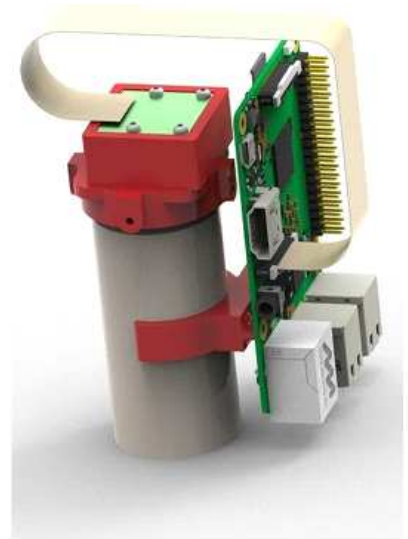
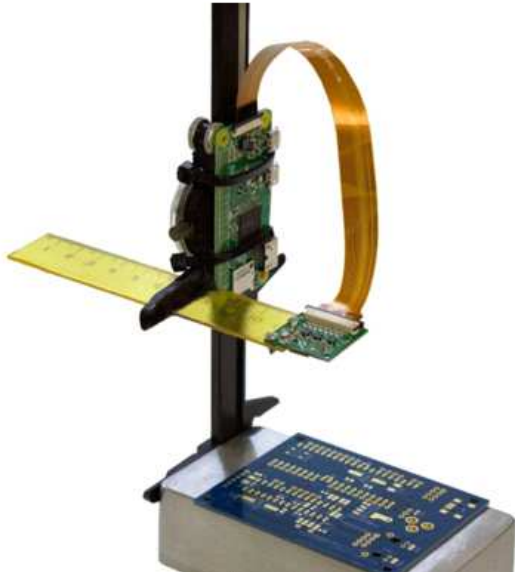
현미경 경통 및 렌즈 모습

3-3. DIY 현미경 예시



3-4. DIY 디지털 현미경 예시

라즈베리파이를 이용한 현미경



3-5. 라즈베리파이 DIY 디지털 현미경 만들기

(1) 부품확인하기

- 현미경 박스를 열어 아래 그림과 같이 5개의 부품 모듈을 꺼낸 후, 상자를 안전한 곳에 보관합니다.



(2) 현미경 스탠드 조립하기

- 위 그림에서 붉은 색 박스 안에 있는 상자를 열어 현미경 스탠드 부품을 꺼내어 그림과 같이 조립합니다.
- 안전 보조링은 스탠드 바닥에서 약 10cm 정도 높이에 고정한다. 보조링은 실수로 경통 조임 뭉치가 떨어지는 것을 방지해줍니다.



(3) 경통 조립하기

- 왼쪽 그림과 같이 경통 조임 뭉치를 조립합니다.
- 플라스틱 박스에서 4배, 10배 렌즈와 경통을 꺼내어 오른쪽 그림과 같이 조립합니다.
- 경통 조임 뭉치를 조립할 때 조동 나사가 오른쪽에 오도록 끼웁니다. 만일 왼손이 주 손인 경우에는 조동 나사를 왼쪽에 위치하도록 끼워줍니다.
- 모두 세 개의 나사가 있는데 잘 조이지 않는 경우 경통이 떨어질 수 있으니 주의해서 체결해줍니다.

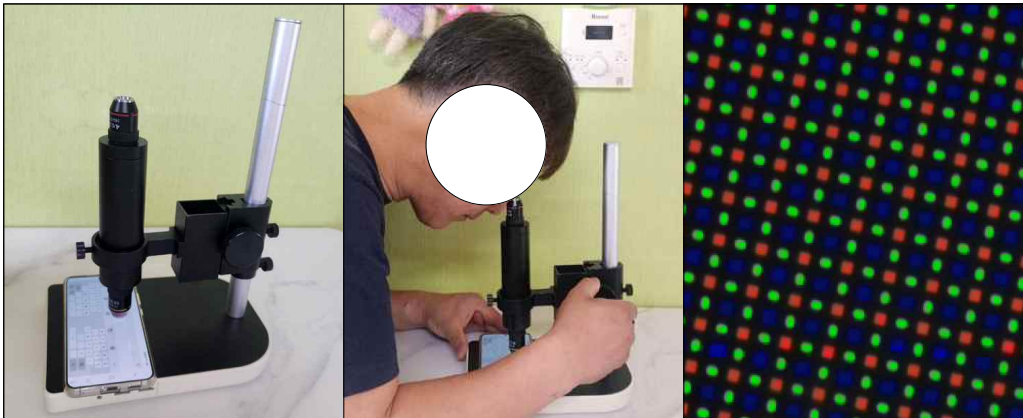


※ 유의할 점: 경통의 양끝이 다르게 생겼습니다. 10배 렌즈는 경통의 튀어나온 나사 부분에 체결하고, 4배 렌즈는 안쪽으로 나 있는 나사 부분에 체결합니다. 아래 그림에서 10배 렌즈가 결합된 부분을 자세히 보면 밖으로 돌출된 부분에 체결되어 있는 것을 볼 수 있습니다. 다음 단계에서 10배 렌즈 대신 카메라를 체결하기 쉽도록 하기 위해서입니다.



(4) 현미경 관찰하기

- 아직 현미경 조명을 설치하지 않았기 때문에 스마트폰을 이용하여 스마트폰의 픽셀 모양을 관찰해볼 수 있습니다.
- 관찰할 때는 10배 대물렌즈를 스마트폰 화면에서 밀착시킨 후 천천히 올리면서 관찰합니다. 픽셀이 보이면 조심스럽게 초점을 맞추도록 합니다.



(5) 라즈베리파이 디지털 현미경 만들기

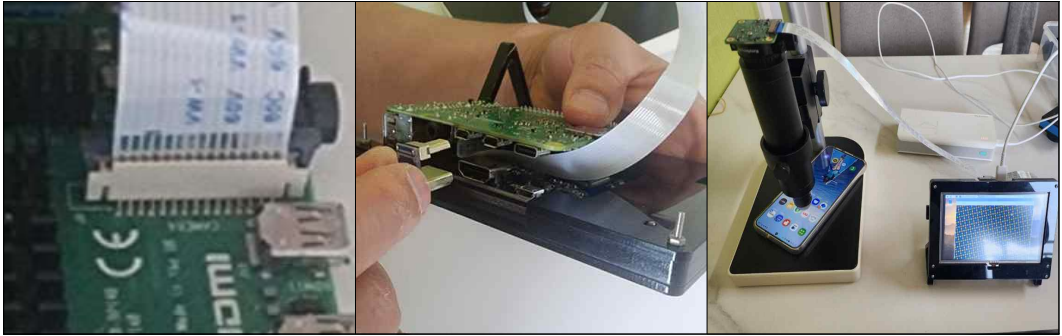
- 10배 렌즈의 채결을 풀고 라즈베리파이 HQ 카메라를 경통에 채결합니다.
- 카메라를 경통에 채결한 후 그림과 같이 HQ 카메라에 플렉스 케이블을 끼워줍니다.



※ 유의할 점: 플렉스 케이블을 끼울 때 방향에 주의합니다. 한쪽은 절연면이고 한쪽은 은도금이 되어 있습니다. 은도금이 되어 있는 쪽이 카메라 단자에 연결될 수 있도록 유의합니다.



- 케이블의 반대쪽은 라즈베리파이 카메라 슬롯에 끼워줍니다. 그림을 참고하여 은도금되어 있는 면이 그림과 같이 배치되도록 끼워줍니다.
- 케이블이 잘 끼워졌으면 라즈베리파이와 모니터를 결합시킵니다.



- 다음 명령어를 실행하여 카메라를 작동시키고 이미지에 초점을 맞추어 관찰합니다.

```

$ workon py3cv4
$ cd project/robot
$ python take_webcam_pictures.py -p 0

```

```

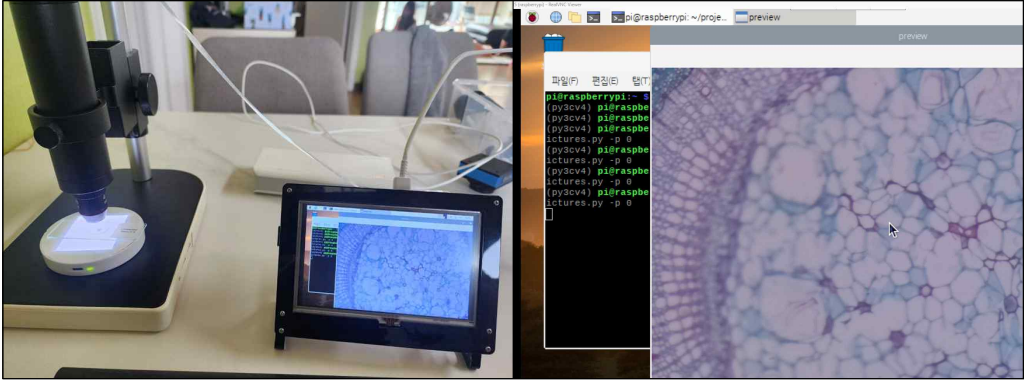
pi@raspberrypi:~ $ v
(py3cv4) pi@raspber
(py3cv4) pi@raspber
(py3cv4) pi@raspber
ictures.py -p 0
(py3cv4) pi@raspber
ictures.py -p 0
(py3cv4) pi@raspber
ictures.py -p 0

```

4. 제작한 현미경을 더 똑똑하게 만들어 볼까요?

(1) 현미경 조명 설치하기

- LED 조명을 현미경 스탠드에 설치하고 현미경용 샘플 이미지를 관찰해봅니다.



(2) 현미경 업그레이드해보기

- 직접 제작한 현미경을 사용해보면서 불편한 점이나 개선할 점에 대해 생각해봅니다. 여러분이 생각한 아이디어를 알려주세요.



여러분의 생각을 알려주세요.

6. SenseHAT을 활용한 스마트 암시야 현미경 확장

지금까지 우리는 라즈베리파이와 카메라 모듈을 이용하여 디지털 현미경을 제작하고, 인공지능 모델을 통해 세포를 분류하는 실습을 진행했습니다. 이번 장에서는 라즈베리파이의 확장 보드인 **SenseHAT**을 추가로 장착하여, 물리적인 렌즈 교체 없이 코딩만으로 명시야(Bright-field)와 암시야(Dark-field)를 자유롭게 전환할 수 있는 스마트 멀티모달 현미경으로 업그레이드해 보겠습니다.

(1) SenseHAT 알아보기

SenseHAT은 라즈베리파이 생태계에서 가장 인기 있는 하드웨어 확장 보드(HAT: Hardware Attached on Top) 중 하나입니다. 복잡한 배선이나 브레드보드 없이 라즈베리파이의 GPIO 핀에 꽂기만 하면 바로 사용할 수 있습니다.



그림 1. SenseHAT의 8x8 RGB LED 매트릭스

- **8x8 RGB LED 매트릭스:** 총 64개의 LED로 구성되어 있으며, 각 LED의 색상과 밝기를 개별적으로 제어할 수 있습니다. 현미경 실습에서는 이 LED 매트릭스가 **프로그래밍 가능한 스마트 조명(광원)** 역할을 합니다.
- **다양한 센서 내장:** 자이로스코프, 가속도계, 자력계, 온도, 기압, 습도 센서 및 5방향 조이스틱이 내장되어 있어 다양한 IoT 프로젝트에 활용할 수 있습니다.
- **Python 라이브러리 지원:** `sense_hat` 라이브러리를 통해 간단한 코드로 모든 기능을 제어할 수 있습니다.

(2) SenseHAT 하드웨어 장착하기

SenseHAT을 기존에 제작한 라즈베리파이 현미경 시스템에 장착해 보겠습니다.

- **1 전원 차단:** 하드웨어를 연결하기 전, 반드시 라즈베리파이의 전원을 끄고 전원 케이블을 분리합니다.
- **2 지지대 설치:** 제공된 4개의 지지대(스탠드오프)를 라즈베리파이 보드의 네 모서리 구멍에 나사로 고정합니다.
- **3 GPIO 결합:** SenseHAT의 커넥터를 라즈베리파이의 40핀 GPIO 헤더에 맞추어 수직으로 눌러 끼웁니다. 핀이 휘지 않도록 주의합니다.
- **4 고정 및 전원 인가:** SenseHAT 위에서 나사를 조여 단단히 고정시킨 후, 전원을 다시 연결합니다.

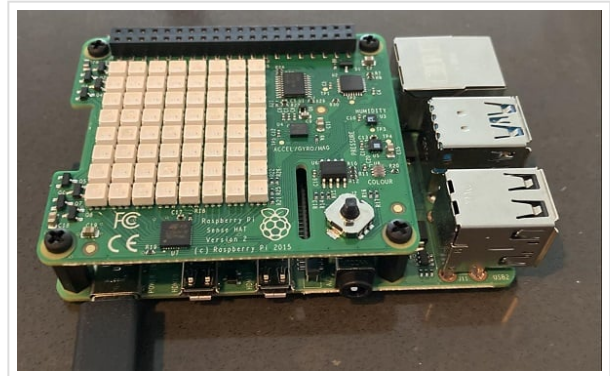


그림 2. 라즈베리파이에 SenseHAT이 결합된 모습

※ **현미경 배치 주의사항:** 현미경 스탠드에 라즈베리파이를 배치할 때, SenseHAT의 LED 매트릭스 부분이 대물렌즈 바로 아래(약 5cm 거리)에 오도록 위치를 조정해야 시료에 빛이 올바르게 전달됩니다.

(3) 소프트웨어 설치 확인

라즈베리파이 OS에는 `sense_hat` 라이브러리가 기본적으로 설치되어 있습니다. 설치 여부를 확인하거나 재설치가 필요한 경우 아래 명령어를 터미널에서 실행합니다.

```
$ sudo apt update
$ sudo apt install sense-hat
$ sudo reboot
```

설치가 완료되면, 아래의 간단한 테스트 코드를 실행하여 SenseHAT이 정상적으로 동작하는지 확인합니다.

```
from sense_hat import SenseHat

sense = SenseHat()
sense.show_message("Hello!", text_colour=[255, 255, 0])
sense.clear()
```

위 코드를 실행하면 LED 매트릭스에 노란색 글씨로 "Hello!"가 흘러가면 정상적으로 연결된 것입니다.

(4) 코딩으로 현미경 조명 제어하기

SenseHAT이 정상적으로 연결되었다면, Python 코드를 작성하여 LED 매트릭스를 제어해 봅시다. LED의 점등 위치에 따라 현미경의 관찰 모드가 달라집니다.

가. 명시야(Bright-field) 조명 만들기

명시야 현미경은 빛이 시료를 직접 통과하여 밝은 배경에 어두운 상이 맺히는 일반적인 관찰 방식입니다. 이를 위해 LED 매트릭스의 **중앙 4×4 영역**만 빛나게 합니다.

```

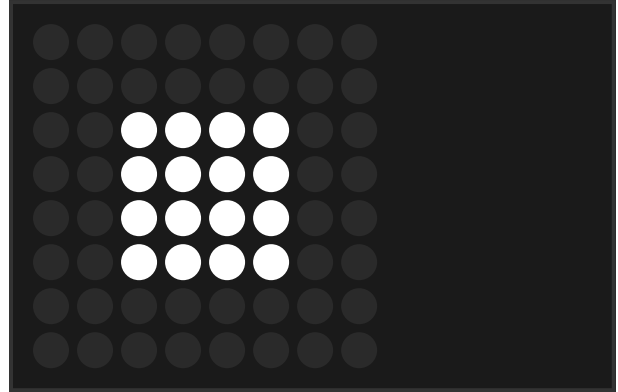
from sense_hat import SenseHat

sense = SenseHat()
sense.clear()

# 중앙 4x4 LED만 흰색으로 켜기
# (명시야 패턴)
for r in range(2, 6):
    for c in range(2, 6):
        sense.set_pixel(
            c, r, 255, 255, 255)

print("명시야 조명이 켜졌습니다.")

```



명시야 패턴: 중앙 4×4 점등

나. 암시야(Dark-field) 조명 만들기

암시야 현미경은 시료를 비스듬히 비추어 산란된 빛만 렌즈로 들어오게 합니다. 어두운 배경에 시료의 윤곽(세포벽 등)이 밝게 빛나 염색 없이도 투명한 시료 관찰이 가능합니다. LED 매트릭스의 **가장자리 테두리**만 빛나게 합니다.

```

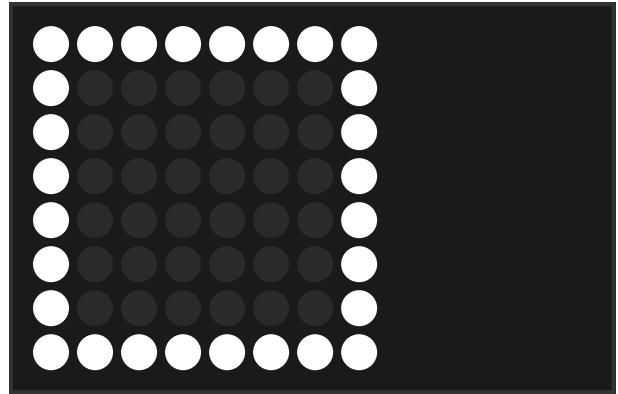
from sense_hat import SenseHat

sense = SenseHat()
sense.clear()

# 외곽 테두리 LED만 켜기
# (암시아 패턴)
for i in range(8):
    # 상단 및 하단 테두리
    sense.set_pixel(i, 0, 255,
255, 255)
    sense.set_pixel(i, 7, 255,
255, 255)
    # 좌측 및 우측 테두리
    sense.set_pixel(0, i, 255,
255, 255)
    sense.set_pixel(7, i, 255,
255, 255)

print("암시아 조명이 켜졌습니다.")

```



암시아 패턴: 외곽 테두리 점등

(5) 명시야 vs 암시야 이미지 촬영 및 비교

각 조명 코드를 실행한 상태에서 아래 명령어로 이미지를 촬영합니다.

```
# 명시야 이미지 촬영
$ libcamera-still -o bright_field.jpg

# 암시야 이미지 촬영
$ libcamera-still -o dark_field.jpg
```

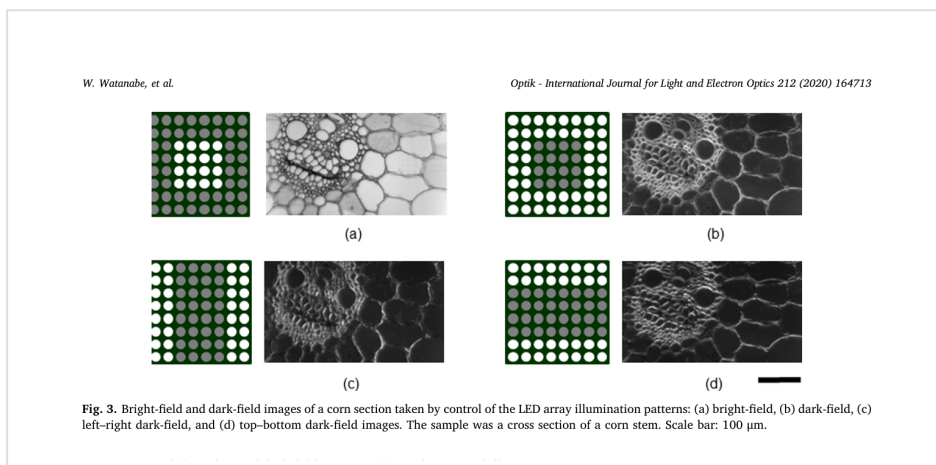


그림 3. 옥수수 줄기 단면의 명시야(좌)와 암시야(우) 이미지 비교
(출처: Watanabe et al., *Optik* 212, 2020)

명시야 이미지에서는 세포 전체의 구조가 밝은 배경에 나타나며, 암시야 이미지에서는 세포벽이 밝게 강조되어 투명한 시료도 선명하게 관찰됩니다.

(6) 심화: 방향성 암시야 패턴 실험

전체 테두리가 아닌 특정 방향의 LED만 켜면, 그 방향과 평행한 시료의 미세 구조가 더욱 강조됩니다. 아래 코드를 수정하여 좌-우 암시야와 상-하 암시야 이미지를 각각 촬영해 봅시다.

```

from sense_hat import SenseHat

sense = SenseHat()
sense.clear()

# 좌-우 암시야 패턴 (좌측 열과 우측 열만 점등)
for i in range(8):
    sense.set_pixel(0, i, 255, 255, 255) # 좌측 열
    sense.set_pixel(7, i, 255, 255, 255) # 우측 열

```

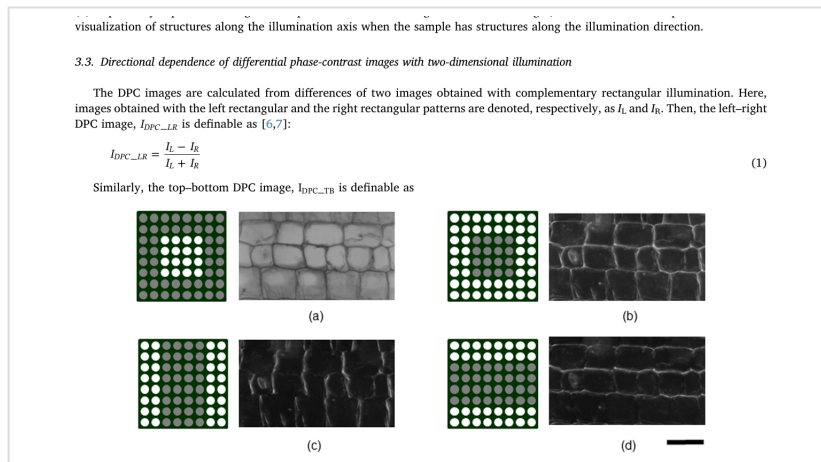


그림 4. 방향성 암시야 이미지: 좌-우 암시야(좌)와 상-하 암시야(우)
(출처: Watanabe et al., Optik 212, 2020)

좌-우 암시야 이미지에서는 수직 방향의 세포벽이, 상-하 암시야 이미지에서는 수평 방향의 세포벽이 각각 더 선명하게 나타납니다. 이처럼 조명 방향을 코딩으로 제어하면 물리적인 장치 변경 없이 다양한 관찰 모드를 구현할 수 있습니다.

(7) 현미경 업그레이드 아이디어

SenseHAT을 이용하면 조명 제어 외에도 다양한 확장이 가능합니다.

- **환경 모니터링:** SenseHAT의 온도·습도 센서를 활용하여 세포 관찰 환경을 실시간으로 기록할 수 있습니다.
- **원격 제어:** NNG(nanomsg next generation) 라이브러리를 활용하면 네트워크를 통해 외부에서 조명 패턴을 원격으로 변경하는 스마트 실험실을 구축할 수 있습니다.
- **AI와의 결합:** 명시야와 암시야 이미지를 모두 수집하여 인공지능 모델의 학습 데이터로 활용하면, 세포 분류 정확도를 더욱 높일 수 있습니다.

 **생각해보기:**

코딩으로 조명을 자유롭게 바꿀 수 있다면, 인공지능 모델이 세포를 더 잘 인식하도록 만들기 위해 어떤 조명 패턴을 사용하는 것이 가장 효과적일까요? 여러분의 아이디어를 자유롭게 나누어 봅시다.