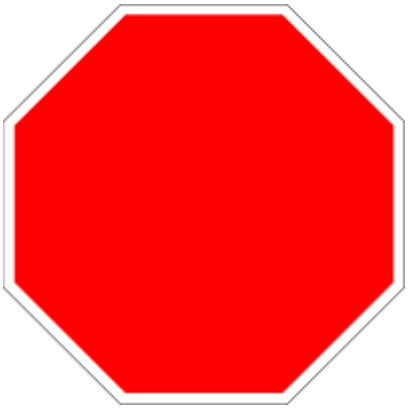
정지 표시 감지

이 강의에서는 하르 분류기(Haar Cascade)가 무엇인지, 어떻게 작동하는지, 물체 감지에 왜 중요한지 배우게 됩니다.
특히 Zumi에서는 정지 표시를 감지할 수 있습니다!

객체 식별

우리는 아래 표시을 정지 표시로 식별하는데 문제가 없는데, 이것을 어떻게 알게되었을까요?
컴퓨터가 우리처럼 표시를을 식별하는데는 어떤 어려움이 있을까요?

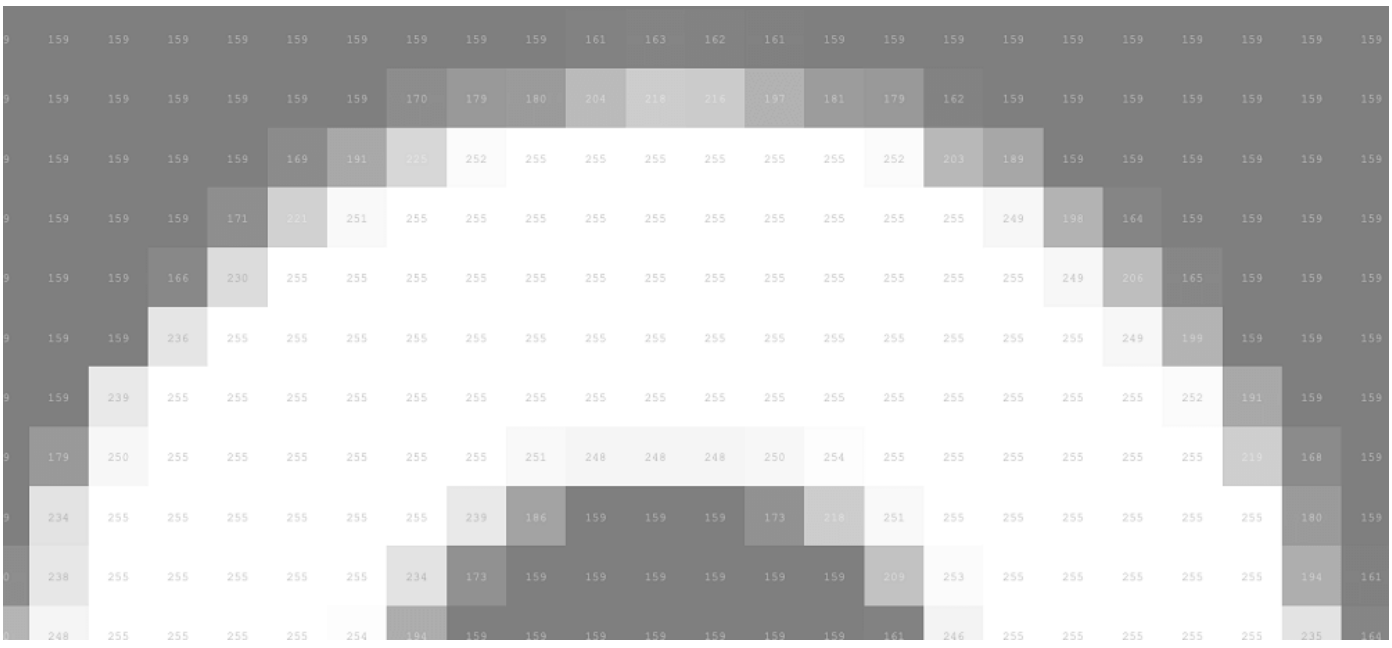
이것이 정지 신호임을 식별하는 가장 쉬운 방법은 "STOP"이라고 읽는 큰 글자입니다.
그러나 컴퓨터는 정지 신호를 감지하기 전에 영어를 읽고 이해하는 법을 배워야 합니다.
따라서, 긍정적 이미지라고 하는 정지 신호의 여러 사진과 정지 신호가 **없는** 이미지인 부정적 이미지를 입력하여 컴퓨터를 가르칩니다.
이는 인간이 학습하는 방식과 유사합니다. 이것은 어떤 기호일까요?



쉽죠? "STOP"이라는 단어를 제거하더라도 정지 신호가 되는 특정 기능인 빨간색 팔각형을 보고 선택할 수 있습니다.
컴퓨터도 유사한 방식으로 훈련됩니다.

이미지는 픽셀의 집합이며 컬러 이미지의 각 픽셀은 일반적으로 세 개의 숫자 집합으로 표현됩니다.
이제 해당 사진을 회색조로 변환한다고 상상해 보십시오.

회색조 이미지에서는 3개가 아닌 픽셀당 하나의 숫자만 필요합니다.
가장 낮은 값은 검은색을 나타내는 0이고 가장 높은 값은 흰색을 나타내는 255입니다. 그 사이의 값은 회색 음영입니다.
이미지에 충분히 가깝게 확대하면 개별 픽셀을 볼 수 있습니다.
아래 이미지에서 "STOP"의 "S" 상단을 보고 있습니다.



일단 컴퓨터가 긍정적인 이미지와 부정적인 이미지를 본 후에 정지 신호가 있는 이미지와 없는 이미지를 파악할 수 있습니다.
컴퓨터는 무엇을 보고 있는지 인식하지 못하지만 픽셀이 더 밝거나 더 어두운 특정 배열에서 픽셀 값 패턴을 볼 수 있습니다.

하르 분류기(Haar Cascade)

모델이 학습되면 모든 기능이 하르 분류기(Haar Cascade) 파일에 저장됩니다.
이러한 파일은 컴퓨터에서 정지 신호를 감지하는 데 사용할 수 있는 .XML(확장 가능한 마크업 언어)입니다.
그 안에는 탐지하도록 훈련된 것을 분류하기 위한 모든 정보가 포함된 수천 줄의 코드가 있습니다.
얼굴, 자동차, 거리 표지판, 미소 등을 감지하기 위해 온라인에서 사용할 수 있는 .XML 캐스케이드 파일이 있습니다.

하르 분류기(Haar Cascade) 훈련은 시간이 많이 걸립니다.
괜찮은 분류기를 만들려면 수천 개의 긍정적이고 부정적인 이미지와 약간의 컴퓨팅 성능이 필요합니다.

우리는 시간과 노력을 절약하기 위해 사전 훈련된 모델을 사용합니다.

코드

라이브러리 가져오기

정지 신호를 감지하는 데 필요한 코드는 비전 라이브러리에 있습니다. 카메라와 zumi 라이브러리도 필요합니다!

In [ ]:

```
from zumi.zumi import Zumi
from zumi.util.screen import Screen
from zumi.util.vision import Vision
from zumi.util.camera import Camera
import time

zumi = Zumi()
camera = Camera()
screen = Screen()
vision = Vision()
```

detect_stop_sign()

이 함수는 비전 라이브러리의 일부이며 정지 신호가 감지되면 True를 반환하고 감지되지 않으면 False를 반환합니다. 이미지 매개변수를 사용하므로 사진을 찍어 함수에 전달해야 합니다. 이 기능은 이미지를 회색조로 바꿉니다. 여기 예가 있습니다!

In [ ]:

```
camera.start_camera()  # 카메라 켜기
img = camera.capture()  # 사진을 찍어서 img 변수에 저장
print(vision.detect_stop_sign(img))  #정지 표시가 감지되었냐에 따라서True 거나 Fasle를 출력
camera.show_image(img)  # 찍은 사진을 보여주기
camera.close()  # 카메라 끄기
```

활동: 운전 및 정지

정지 신호를 감지하고 올바른 조치를 취하는 것은 자율 주행 자동차에서 매우 중요합니다.
이 활동에서는 정지 신호가 감지될 때까지 Zumi가 앞으로 운전하도록 코드를 작성합니다.
의사 코드는 어떻게 작성하면 좋을까요?

<의사코드>

사진을 찍기
정지 신호가 감지되었는지 확인
True인 경우 반복문을 중지하고 중단합니다.
그렇지 않으면 앞으로 나아갑니다
100번 반복

아래에 코드를 작성합니다.
스크롤을 내리면 해결 방법이 있습니다.

In [ ]:

```
zumi.reset_gyro()

camera.start_camera()

for i in range(100):

    # 사진 찍기
    # detect_stop_sign 호출
    # 만약 참이면 반복문을 끝냅니다.
    # 그렇지 않으면 forward_step을 호출합니다.

zumi.stop()
camera.close()
```

해결 방안: 운전 및 정지

아래 두 가지 가능한 해결 방법을 살펴보세요.

- for문과 while문을 사용하는 방법입니다. 차이를 확인해보세요.

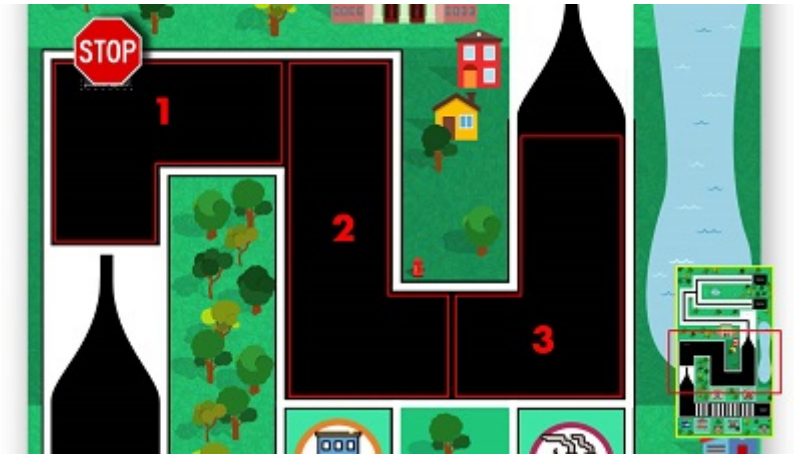
In [ ]:

```
zumi.reset_gyro()

camera.start_camera()

for i in range(100):

    img = camera.capture()  # 사진 찍기
    stop_sign_detect = vision.detect_stop_sign(img)  # detect_stop_sign 호출

    if stop_sign_detect:  # 만약 참이라면 참이면 반복문을 끝냅니다.
        print("정지 표시를 발견했습니다")
        break
    else:                 # 그렇지 않으면 forward_step을 호출
        zumi.forward_step(20,0)

zumi.stop()
camera.close()
```

In [ ]:

```
zumi.reset_gyro()

camera.start_camera()

stop_sign_detect = False # 조건을 False로 초기화

try:
    while not stop_sign_detect:

        img = camera.capture()  # 사진 찍기
        stop_sign_detect = vision.detect_stop_sign(img)  # detect_stop_sign 호출
        zumi.forward_step(20,0)

    print("정지 표시를 발견했습니다")
finally:
    print("완료")
    zumi.stop()
    camera.close()
```

대회 : 어떻게 사용되나요?



맵 안에는 정지표시가 놓여져 있습니다.
주미는 이 정지표시를 확인하고, 기다렸다가 정지표시가 사라지면 이동할 수 있습니다.
주미는 정지표시를 확인했다는 의미로 도레미 음계를 재생하게 됩니다.

예제: 기다린후 주행하기

이 예에서 Zumi는 정지 신호가 감지되는 동안 기다렸다가 정지 신호가 감지되지 않으면 앞으로 운전합니다.

In [ ]:

```
zumi.reset_gyro()

camera.start_camera()

stop_sign_detect = True # 조건을 False로 초기화

try:
    while stop_sign_detect == True: # stop_sign_detect 값이 참이라면 반복문을 실행하기
        img = camera.capture()
        stop_sign_detect = vision.detect_stop_sign(img)
        print("정지 표시가 사라지길 기다리는 중")

    print("정지 표시가 사라졌습니다")

finally:
    print("완료")
    zumi.forward(duration=3)
    camera.close()
```