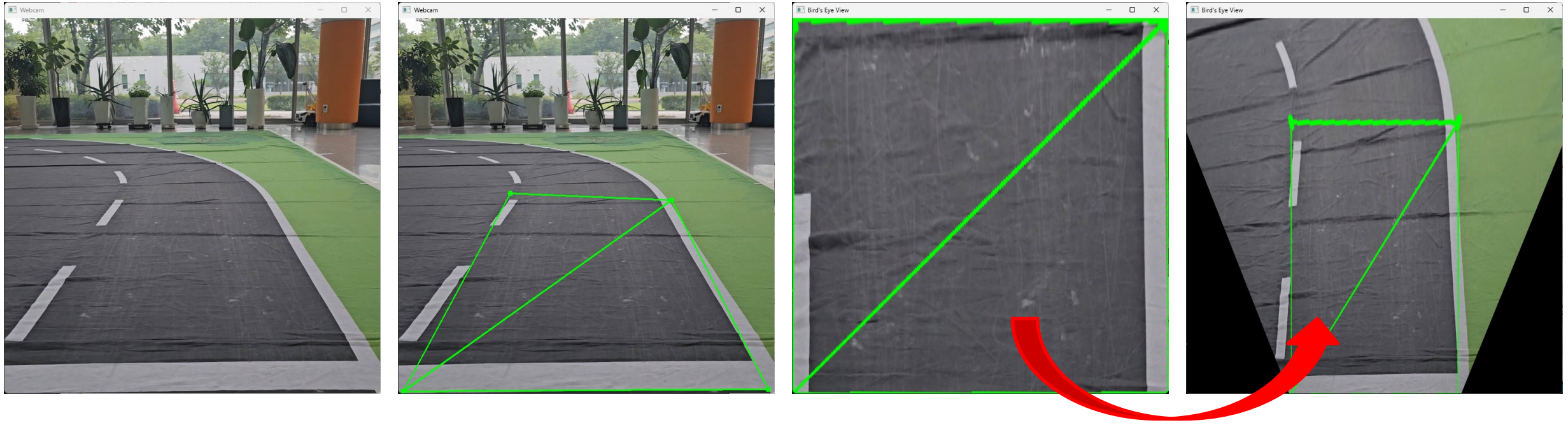
성균관대학교
SUNG KYUN KWAN UNIVERSITY(SKKU)

# Future Vehicle Education Workshop
# - Advanced Exercises (Assignments)

# Advanced Ex 1

- Bird’s-Eye-View application
  - Running with default settings, outputs like the 3rd picture→ Secure the view as shown in the 4th picture
  - Modify only the get_bird_eye_view() function in the start_lib.py file (1 line of settings).



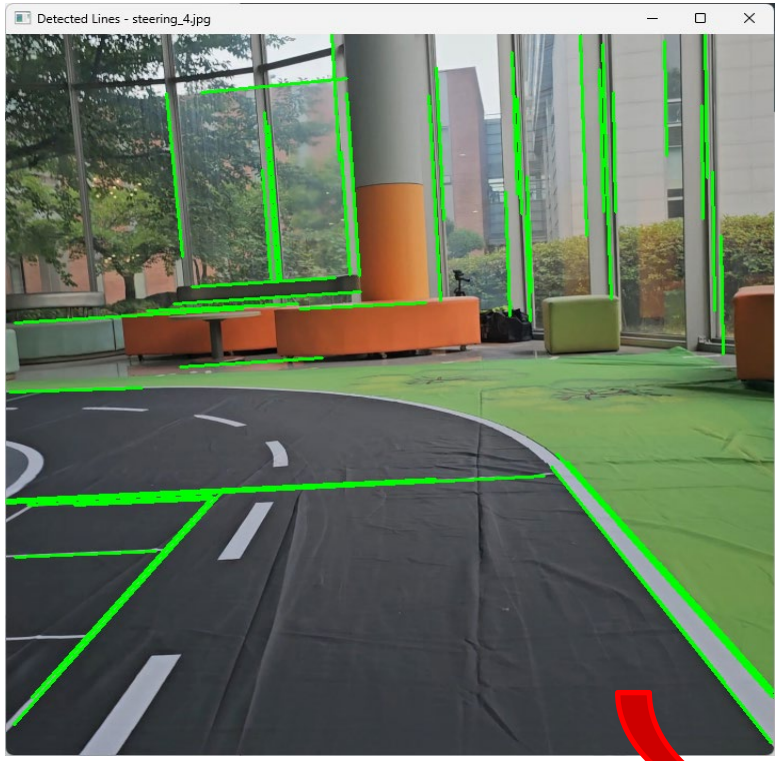
# Advanced Ex 1

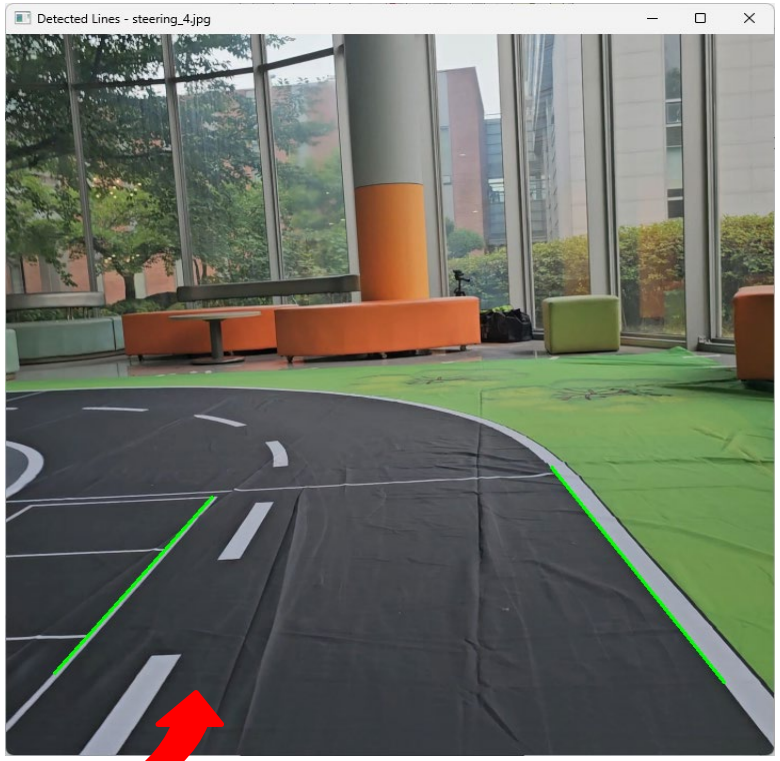
- Bird's-Eye-View application
  - Running with default settings, outputs like the 3rd picture→ Secure the view as shown in the 4th picture
  - Modify only the get_bird_eye_view() function in the start_lib.py file (1 line of settings).

```python
def get_bird_eye_view(image, output_size, points):
    height, width = output_size[1], output_size[0]
    scaled_points = [(int(p[0] * width / image.shape[1]), int(p[1] * height / image.shape[0])) for p in points]

    src_points = np.float32([scaled_points[0], scaled_points[1], scaled_points[3], scaled_points[2]])
    dst_points = np.float32([[0, height], [width, height], [0, 0], [width, 0]])

    matrix = cv2.getPerspectiveTransform(src_points, dst_points)
    bird_eye_view = cv2.warpPerspective(image, matrix, output_size)
    return bird_eye_view
```

# Advanced Ex 2

- Hough Transform application
  - Running with default settings, output like the 1st photo→ Denoise like the 2nd photo
  - Add only the hough_transform() function to the straight_lib.py file (refer to the code provided in the basic example, 7 lines).



# Advanced Ex 2

- Hough Transform application
  - Running with default settings, output like the 1st photo→ Denoise like the 2nd photo
  - Add only the hough_transform() function to the straight_lib.py file (refer to the code provided in the basic example, 7 lines).

```python
def hough_transform(image, polygon):
    resized_image = cv2.resize(image, (image.shape[1] // 2, image.shape[0] // 2))
    gray = cv2.cvtColor(resized_image, cv2.COLOR_BGR2GRAY)
    blurred = cv2.GaussianBlur(gray, (5, 5), 0)
    edges = cv2.Canny(blurred, 50, 150)

    # Hough 변환을 사용하여 직선을 검출
    lines = cv2.HoughLinesP(edges, 1, np.pi / 180, threshold=100, minLineLength=100, maxLineGap=10)

    # 검출된 직선을 이미지에 그림
    if lines is not None:
        for line in lines:
            x1, y1, x2, y2 = line[0]
            cv2.line(resized_image, (x1, y1), (x2, y2), (0, 255, 0), 2)

    return resized_image
```

# Advanced Ex 3

- Applying HSV color space conversion
  - Traffic Light Color Recognition Examples → Modify the color range, check the operation, and apply

# Advanced Ex 4

- Reduced speed on S-shaped course
  - Lane analysis, motor control code provided → Steering and speed calculation, then motor control connection

# Advanced Ex 5

- Understanding steering angle when reverse parking
  - Providing slope calculation and motor control code → Applied to reverse parking using parallel parking lines