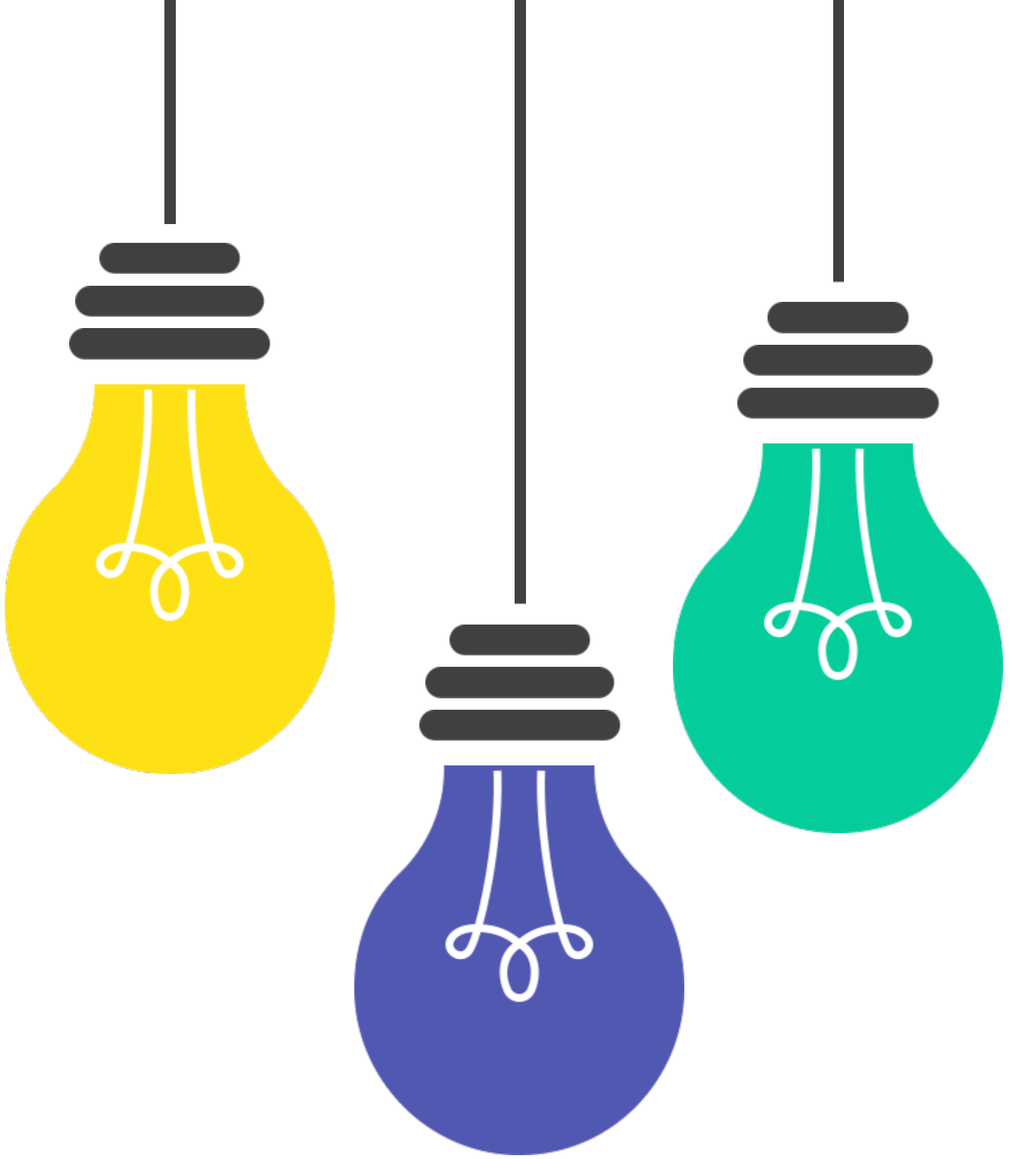
MINI PROJECT

Object Detection과 Mono Depth 를 활용한 장애물 감지 및 회피



03조 김정문, 임민우

Content 01

목표

Content 02

과정

Content 03

결과

Content 04

고찰 및 개선점

Content 01 목표

01 **object detecting을 활용한 장애물 인식**

02 **장애물 인지 후 mono depth를 활용한 거리 측정**

03 **장애물과의 거리에 따른 카메라 회전**

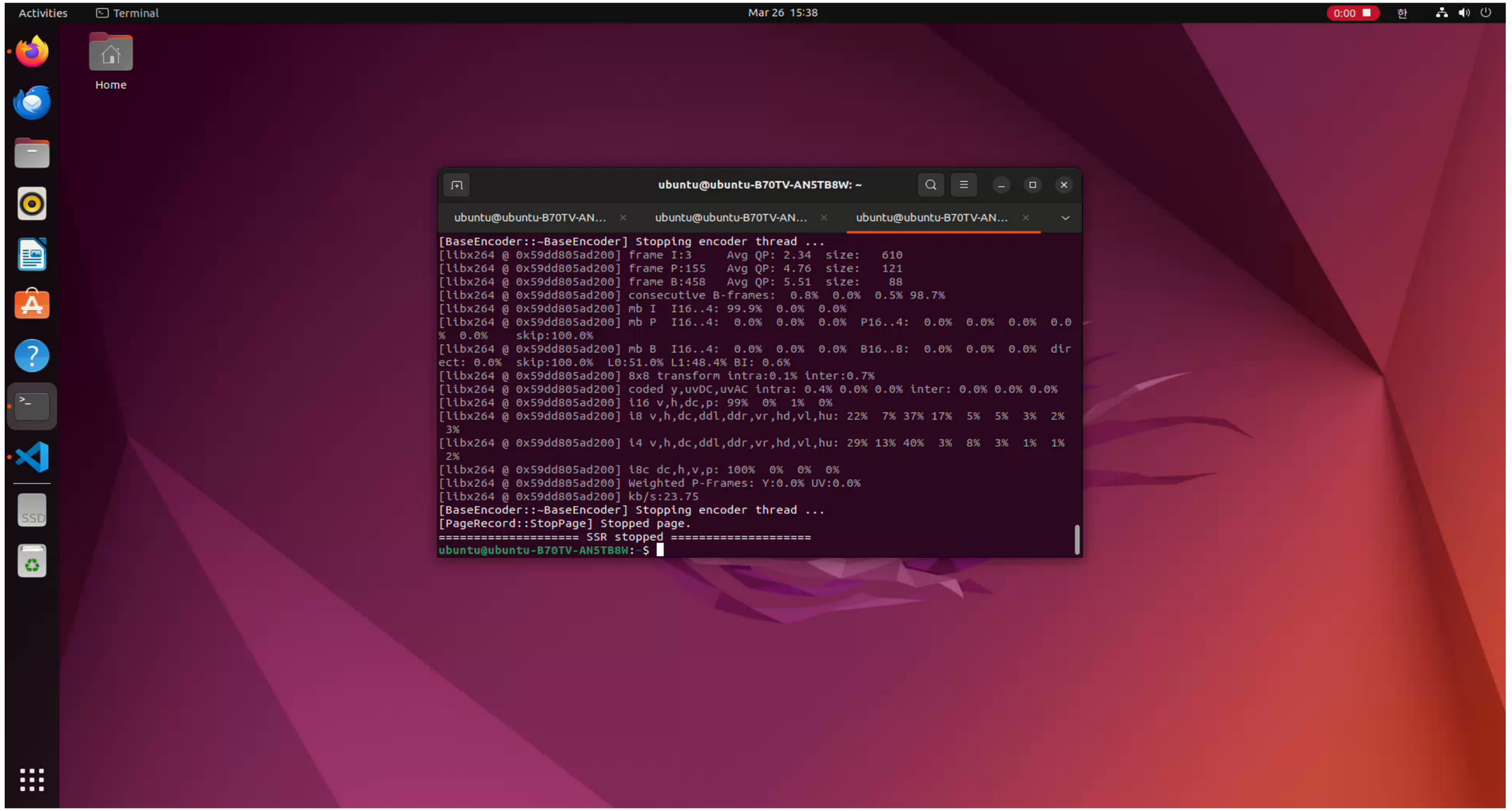
Content 02 과정(1)

```python
def draw_boxes(frame, boxes):
    i = 0
    for label, score, box in boxes:
        # Choose color for the label.
        color = tuple(map(int, colors[label]))
        # Draw a box.
        x2 = box[0] + box[2]
        y2 = box[1] + box[3]
        rgb_image = cv2.rectangle(img=frame, pt1=box[:2], pt2=(x2, y2), color=color, thickness=3)
        print(box[:2][0])
        a_image = rgb_image[box[:2][1]:y2, box[:2][0]:x2]
        cv2.imwrite(f'./image/{label}.jpg', a_image)
        i+=1
        # Draw a label name inside the box.
        cv2.putText(
            img=frame,
            text=f"{classes[label]} {score:.2f}",
            org=(box[0] + 10, box[1] + 30),
            fontFace=cv2.FONT_HERSHEY_COMPLEX,
            fontScale=frame.shape[1] / 1000,
            color=color,
            thickness=1,
            lineType=cv2.LINE_AA,
        )    return frame
```

Content 02 과정(2)

```
IMAGE_FILE = "./image/1.jpg"
image = load_image(path=IMAGE_FILE)#print(image.shape())
try:
    resized_image = cv2.resize(src=image, dsize=(network_image_height, network_image_width))
    input_image = np.expand_dims(np.transpose(resized_image, (2, 0, 1)), 0)
    result = compiled_model([input_image])[output_key]
    result_image = convert_result_to_image(result=result)
    result_image = cv2.resize(result_image, image.shape[:2][::-1])
except Exception as e:
    print(str(e))
```

Content 03 결과영상



Content 04 고찰 및 개선점

object detection
mono depth와의 연동 어려움

- 영상이 아닌 이미지로 detecting한 데이터를 mono depth로 넘겨줌

mono depth

거리값 표시에 대한 실시간 출력 어려움

개선점

object detecting한 영상을 실시간으로 mono depth로 넘겨주어 장애물을 인지하고 거리값을 측정.
최종적으로 장애물이 없는 방향으로 카메라 화면을 회전.

Q & A

Thank You