**상세 설계 문서: 희귀 동물 탐지**

**프로젝트 제목:** 멸종 위기동물 탐지 및 추적 관찰 시스템  
**버전:** 1.0  
**날짜:** 24.12.27

**1. 개요**

이 문서는 observation 카메라와 자율 이동 로봇(AMR)의 tracking 카메라에서 얻은 데이터에서 희귀동물 탐지를 하기 위한 상세 설계를 다룹니다. 객체 탐지를 사용하는 데 필요한 아키텍처, 알고리즘, 데이터 처리 및 시스템 상호작용이 포함되어 있습니다.

**2. 시스템 아키텍처**

객체 탐지를 위해 온보드 하드웨어(카메라, Jetson-Orin 프로세서, 노트북 프로세서)와 소프트웨어(OpenCV, YOLO)를 활용합니다. Observation에 대한 처리는 노트북 내에서, Tracking에 대한 처리는 AMR 내에서 로컬로 수행됩니다.

**3. Observation 객체 탐지 서브시스템 설계**

**3.1 Observation 객체 탐지 요구사항**

* **객체 인식:** Observation 객체 탐지 시스템은 사전 학습된 데이터 관찰 대상이 카메라에 포착되면 인식해야 합니다.
* **위치 파악:** Observation 시스템은 관찰 대상을 포착하면 포착된 지점의 좌표를 저장해야 합니다.

**3.2 아키텍처**

1. **이미지 캡쳐 모듈:** USB 카메라를 사용하여 라이브 비디오를 캡처하고, 프레임을 객체 탐지 모듈로 전달합니다.
2. **객체 탐지 모듈:** 노트북에서 YOLO8를 사용하여 실시간 객체 탐지를 수행합니다.
3. **좌표 생성 모듈:** OpenCV 라이브러리를 사용하여 이미지에서 인식된 타깃의 좌표를 추정합니다.

**3.3 Observation 데이터 흐름**

1. **입력 데이터:** USB 카메라로부터 캡처된 라이브 비디오 프레임
2. **프로세스 흐름:**
   * **이미지 전처리:** USB 카메라에서 수집된 프레임을 YOLO 모델에 입력하기 위해 이미지 크기를 조정합니다.
   * **World 좌표계 생성:** 사전 입력된 비콘의 거리 데이터와 이미지 프레임의 픽셀 거리 정보를 통해 world 좌표계를 생성합니다.
   * **객체 탐지:** YOLO 모델을 사용하여 타깃 객체가 이미지 안에 있으면 객체를 식별합니다.
   * **타깃 좌표 생성:** 비콘과 타깃의 상대위치 정보를 계산해 타깃의 world 좌표를 계산합니다.
   * **로그 기록:** 타깃이 탐지된 좌표와 시간을 로그로 남깁니다.
3. **출력 데이터:** 추적할 타깃의 world 좌표, 로그.

**4. Tracking 객체 탐지 서브시스템 설계**

**4.1 Tracking 객체 탐지 요구사항**

* **객체 인식:** Tracking 객체 탐지 시스템은 사전 학습된 데이터를 통해 관찰 대상이 카메라에 포착되면 인식해야 합니다.
* **위치 파악:** Tracking 시스템은 관찰 대상을 포착하면 포착된 지점의 좌표를 저장해야 합니다.

**4.2 아키텍처**

1. **이미지 캡처 모듈:** USB 카메라를 사용하여 라이브 비디오를 캡처하고, 위치 추정 모듈로 전달합니다.
2. **위치 추정 모듈:** OpenCV 라이브러리를 사용하여 이미지에서 인식된 타깃의 좌표를 추정합니다.

**4.3 Tracking 객체 탐지 데이터 흐름**

1. **입력 데이터:** USB 카메라로부터 캡처된 라이브 비디오 프레임, AMR의 현재 위치
2. **프로세스 흐름:**
   * **이미지 전처리:** USB 카메라에서 수집된 프레임을 YOLO 모델에 입력하기 위해 이미지 크기를 조정합니다.
   * **객체 탐지:** YOLO 모델을사용하여 타깃 객체가 이미지 안에 있으면 객체를 식별합니다.
   * **타깃 위치 추정:** AMR과 타깃의 상대위치 정보를 계산해 타깃의 world 좌표를 계산합니다.
   * **데이터 기록:** 타깃을 탐지하는 동안 1초 간격으로 타깃 이미지를 저장하고 위치 정보 로그를 남깁니다.
3. **출력 데이터:** 타깃의 world 좌표, 타깃 이미지, 로그

**5. 구성 요소 설명**

**5.1 이미지 갭쳐 모듈**

* **사용 장치:** USB 카메라
* **입력:** 라이브 비디오 피드
* **출력:** 객체 탐지를 위한 프레임
* **기능:** 이미지를 지속적으로 캡처하여 객체 탐지 모듈 및 위치 추정 모듈에 전달

**5.2 객체 탐지 모듈**

* **사용 라이브러리:** YOLOv8
* **입력:** 비디오 프레임
* **출력:** 탐지 객체 클래스 번호, 바운딩 박스 정보
* **기능:** 목표 객체를 인식하고 프레임 내 바운딩 박스의 위치와 크기를 도출

**5.3 좌표 생성 모듈**

* **사용 라이브러리:** OpenCV
* **입력:** 비콘 위치 정보, 프레임 내 객체 위치 정보
* **출력:** world 좌표에서의 객체 좌표
* **기능:** 비콘의 좌표 정보와 프레임 내에서 탐지된 객체의 정보를 처리하여 객체의 world좌표 계산

**5.4 위치 추정 모듈**

* **사용 장치:** USB 카메라
* **입력:** 라이브 비디오 피드
* **출력:** 객체 탐지를 위한 프레임
* **기능:** 이미지를 지속적으로 캡처하여 위협 탐지 모듈에 분석을 위해 전달

**6. 데이터 저장 및 로그 기록**

* **Observation 로그:** 탐지된 동물의 위치, 탐지 시간 저장
* **Tracking 로그:** 동물 추적 시작, 종료, 추적 중 대상의 위치 정보 저장
* **이미지 로그:** 네비게이션 및 동물 탐지 로그는 AMR에 48시간 동안 보관되며, 멸종위기동물은 더 오랫동안 저장 가능

**7. 사용자 인터페이스 설계 (대시보드)**

* **실시간 모니터링:** 대시보드에 Observer 라이브 비디오 피드, AMR의 라이브 비디오 피드 표시
* **작동 상태 표시:** 대기중, 탐지중, 추적중으로 현재 작동 상태 표시
* **수동 제어:** 보안 담당자가 필요시 AMR을 수동으로 제어할 수 있도록 기능 제공

**8. 테스트 및 검증**

**8.1 네비게이션 테스트**

* **경로 계획 테스트:** 다양한 장애물 및 환경 레이아웃에서 경로 계획을 검증
* **장애물 회피 테스트:** 정적 및 동적 장애물에 대한 AMR 반응 테스트
* **SLAM 정확도 테스트:** 생성된 맵과 로봇 위치가 정확한지 확인

**8.2 동물 탐지 테스트**

* **객체 탐지 테스트:** YOLO가 미리 정의된 동물 객체를 감지하고 분류할 수 있는지 검증
* **지연 테스트:** 감지에서 알림까지의 지연 시간을 측정하여 실시간 반응을 보장
* **오탐 및 누락 테스트:** 실제 동물을 정확히 식별하고 오탐을 최소화하는 신뢰성 검증

**9. 오류 처리 및 복구**

1. **네비게이션 오류:** 위치 추정이 실패할 경우, AMR은 멈추고 SLAM을 통해 위치를 다시 설정 시도. 해결되지 않으면 모니터링 PC로 알림 전송.
2. **동물 탐지 오류:** 카메라나 객체 탐지 모델에 오류가 발생하면 PC 대시보드에 알림을 전송해 유지보수를 요청.
3. **네트워크 손실:** AMR은 자율 네비게이션을 계속하며, 연결이 복구될 때까지 로그를 로컬에 저장.

**10. 성능 요구사항**

* **네비게이션 지연:** 경로 재계산 및 장애물 회피 업데이트는 100ms 이내로 처리
* **동물 탐지 지연:** 객체 감지 및 알림 전송은 1초 이내로 처리
* **배터리 수명:** 최소 8시간 연속 운영 가능

**11. 부록**

* **소프트웨어 라이브러리:** ROS2, OpenCV, YOLO, 및 지원되는 Python 라이브러리
* **용어 사전:** SLAM, YOLO, 위치 추정, 경로 계획과 같은 주요 용어 정의
* **참고 문헌:** 설계에 사용된 ROS2, YOLO 및 기타 라이브러리 문서 링크