**종합설계 프로젝트 수행 보고서**

|  |  |
| --- | --- |
| **프로젝트명** | **실시간 총기 위협 감지를 위한 CCTV 감시 시스템** |
| **팀번호** | **S1-7** |
| **문서제목** | **수행계획서( )**  **2차발표 중간보고서( O )**  **3차발표 중간보고서( )**  **최종결과보고서( )** |

**2022.03.07**

**팀원 : 정수민 (팀장)**

**배성은 (팀원)**

**지도교수 : 나보균**

**문서 수정 내역**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **작성일** | **대표작성자** | **버전** | **수정내용** |  |
| 2022.01.03 | 정수민(팀장) | 1.0 | 수행계획서 | 최초작성 |
| 2022.02.27 | 정수민(팀장) | 2.0 | 2차발표자료 | 설계서추가 |

**문서 구성**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **진행단계** | **프로젝트**  **계획서 발표** | **중간발표1**  **(3월)** | **중간발표2**  **(5월)** | **학기말발표**  **(6월)** | **최종발표**  **(10월)** |
| **기본양식** | 계획서 양식 | 계획서 양식 | 계획서 양식 | 계획서 양식 | 계획서 양식 |
| **포함되는**  **내용** | Ⅰ. 서론  (1~6)  Ⅱ. 본론  (1~3)  참고자료 | Ⅰ. 서론  (1~6)  Ⅱ. 본론  (1~4)  참고자료 | Ⅰ. 서론  (1~6)  Ⅱ. 본론  (1~5)  참고자료 | Ⅰ. 서론  (1~6)  Ⅱ. 본론  (1~7)  참고자료 | I  II  III |

**이 문서는 한국산업기술대학교 컴퓨터공학부의 “종합설계”교과목에서 프로젝트 “실시간 총기 위협 감지를 위한 CCTV 감시 시스템”을 수행하는**

**S1-7팀(정수민, 배성은)이 작성한 것으로 사용하기 위해서는 팀원들의 허락이 필요합니다.**

**목 차**

**Ⅰ. 서론**

1. 작품선정 배경 및 필요성

2. 기존 연구/기술동향 분석

3. 개발 목표

4. 팀 역할 분담

5. 개발 일정

6. 개발 환경

**Ⅱ. 본론**

1. 개발 내용

2. 문제 및 해결방안

3. 시험 시나리오

4. 상세 설계

5. Prototype 구현

6. 시험 / 테스트 결과

7. Coding & DEMO

**Ⅲ. 결론**

1. 연구 결과

2. 작품제작 소요재료 목록

참고자료

I 서론

1 작품선정 배경 및 필요성

① 작품 선정 배경

현재 미국은 총기 소유가 합법인 국가로 Covid-19 사태 이전인 2019년

기준 총기 사고 피살자 수 15,208명, 미성년(0-17세) 총기 부상사〮망자수 37760명으로 총기 관련 범죄에 굉장히 취약한 상황임. 공공시설 및 주택에서의 CCTV 설치율이 계속 증가하고 있지만 땅이 워낙 넓은 탓에 인적이 드문 길거리나 야간에는 여전히 총기 관련 범죄에 취약함.

최근 미국의 CCTV 수요가 계속해서 증가하고 있으나, 미국 내 CCTV 수입 제품 점유율에서 중국이 44%를 차지할 정도로 중국에 대한 의존도가 큼.

② 작품의 필요성

* 유동인구가 많은 곳에서 총기 사고가 일어날 경우, 목격자가 신고하면

경찰 이 출동하여 상황을 정리할 수 있지만, 목격자가 없거나 신고하기 어려운 상황에 처하면 큰 인명 피해가 발생하거나 사건 사후 처리가 늦어질 수 있음.

* 기존에도 situation detection 기술로 특정 상황을 예방/신고 하는 시스템

은 있었으나 대부분 서버에서 모든 처리를 하는 서버 의존적인 시스템임.

* Gun threat detection은 최근에 활발히 연구되고 있지만 아직 application

이 많지 않으며 군 등에서 활용할 수 있음에도 불구하고 총기 소유가 불법인 우리나라의 특성상 국내 연구가 거의 없음.

2 기존 연구/기술동향 분석

|  |  |
| --- | --- |
| 논문 제목 | 저자 |
| Automatic Handgun Detection Alarm in Videos Using Deep Learning | Roberto Olmos  Siham Tabik  Francisco Herrera |
| Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning | Alberto Castillo  S. Tabik  F. Perez |
| Real-time gun detection in CCTV: An open problem | Jose L. Salazar Gonzalez  Carlos Zaccaro  Juan A. Alvarez-Garcia |

3 개발 목표

* Object detection model을 학습시켜 gun object detection을 수행
* RTSP 프로토콜을 사용하여 low-delay video streaming 구현
* DVR 서버에 detection result 전송
* PTZ Controllable Camera 등의 하드웨어를 원격지에서 제어

4 팀 역할 분담

**정수민**

시스템 모듈 설계

하드웨어(카메라, 마이크, 조명 등) 컨트롤 기능 구현

Yolo, fast-RCNN 등의 모델을 사용하여 object detection 성능 측정 및 학습

Gstreamer 등을 이용한 네트워킹 기능 구현

시스템 통합

**배성은**

Object detection 관련 사전조사

Opencv를 이용한 Video Capture 구현

Video Encoding 기능 구현

시스템 문서화

5 개발 일정

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 추진계획 | 12월 | 1월 | 2월 | 3월 | 4월 | 5월 | 6월 | 7-9월 |
| 조사 및 학습 |  |  |  |  |  |  |  |  |
| 시스템 설계 |  |  |  |  |  |  |  |  |
| 구현 |  |  |  |  |  |  |  |  |
| 테스트 및 데모 |  |  |  |  |  |  |  |  |
| 문서화 및 발표 |  |  |  |  |  |  |  |  |
| 최종보고서 작성 |  |  |  |  |  |  |  |  |

6 개발 환경

* **Hardware**: Jetson Nano Development Kit
* **OS**: Linux (Ubuntu, Arch)
* **Langauage**: Python 3
* **Frameworks**: Opencv, numpy, matplotlib, Glib,

GStreamer, Darknet

* **Developmenrt Methodology**: Agile

II 본론

1. 개발 내용
   1. Object detection
   2. Video Streaming
   3. Networking
   4. Hardware control
2. 문제 및 해결방안

**본 항목은 문제 발생/해결 시마다 지속적으로 업데이트 한다.**

**2.1** 기존에 yolov3-tiny detection model을 사용하였으나 detection performance 및 fps performance가 요구 수준에 미치지 못하였음. 추가적인 조사 결과 yolov4-tiny 모델이 있다는 것을 발견하고 성능 측정 결과yolov4-tiny가 두 측면에서 yolov3-tiny 모델보다 우세하다 판단하여 모델을 변경함.

**2.2** video streaming에서 need-data event 발생 시 framebuffer가 비어있는 등의 문제로 데이터를 전송하지 않으면 파이프라인이 블로킹되므로, 만약 framebuffer가 비어있다면 더미 프레임을 전송함.

**2.3** Laptop environment 에서 테스트시 opencv 및 matplotlib 기반의 렌더러가 정상적으로 작동하지 않으므로 opengl 혹은 qt 등의 렌더링 엔진으로의 변경이 필요함.

**2.4** 프로그램 장시간 가동 시 발열로 인해 방열판이 있음에도 시스템 온도가 80도까지 증가함. 시스템 안정성 확보를 위해 시스템 보드에 쿨러 장착이 필요함.

**2.5** 기존에 detection result 전송을 위해 python의 dill 라이브러리로 serialize하여 전송했으나, python-python 통신밖에 불가능한 dill의 단점으로 인해 DetectionResult 클래스 인스턴스를 json 형식으로 변환하여 serialize 후 전송하는 방법으로 변경함.

**2.6** Video Capture Thread에서 frame을 저장할 때 timestamp를 찍는데, 이를 그대로 사용하여 streaming 할 경우 전송 프로토콜 상에서 timestamp sync 문제가 발생하여, Video Streamer가 독자적인 timestamp를 다시 버퍼에 담아 전송하는 방식으로 변경함.

1. 시험 시나리오
2. 상세 설계
   1. 시스템 구성도

**Core classes**

테이블이(가) 표시된 사진

자동 생성된 설명

* 1. 시스템 모듈 상세 설계