젯슨 나노를 이용한 온프레미스 주거 침입 탐지 시스템 구현

안정인*, 김윤*
*강원대학교 컴퓨터정보통신공학과
e-mail : ji5489@gmail.com, yooni@kangwon.ac.kr

Implementation of On-Premise Home Intrusion Detection System Using Jetson Nano

Jung-In An*, Yoon Kim*
*Dept of Computer Science and Engineering, Kangwon National University

요 약

최근의 상용 홈 보안 솔루션 서비스는 SaaS[5] 방식을 사용하며 대용량의 영상 데이터를 중앙 클라우드에서 처리하는 시스템으로 운영하고 있다. 이러한 시스템은 이용자의 개인 정보가 네트워크를 통해 유출될 수 있는 문제점을 가지고 있다. 본 논문은 Jetson Nano와 안드로이드 애플리케이션을 각각 서버-클라이언트로 구성한 온프레미스(On-Premise) 주거 침입 탐지 시스템을 제안한다. 제안 시스템은 클라우드 서비스를 이용하지 않아 개인 정보 처리의 외부 의존성을 제거하였다. 또한 서버-게이트웨이 구간의 통신 도청 취약점[2]을 해결하기 위하여 별도의 내부 네트워크를 구축하였다. 서버에서 탐지된 침입 이벤트는 RabbitMQ를 통해 안드로이드 애플리케이션으로 전송하여 사용자가 알림을 통해 주거 침입 이벤트와 실시간 영상을 확인할 수 있도록 구현하였다.

1. 서론

최근 IoT 기기를 기반으로 하는 스마트홈 구축 사례들이 증가함에 따라 각종 보안 이슈들이 발생하고 있다. 논문[1]에서는 스마트홈 이용자 15명 중 80% 이상이 보안 취약점과 개인 정보 침해 가능성에 대해 걱정하고 있음을 분석하였다. 실제 침해 사례들을 분석한 결과, 대부분의 홈 CCTV 침해사고가 네트워크 카메라와 게이트웨이 사이의 통신 구간 도청으로 이루어졌으며 이를 해결하는 방안으로 논문[2]에서는 사용자 인증과 통신 구간 암호화 방식을 제안하였다.

본 논문은 상기 내부 통신 구간 취약점[2]을 해결하기 위하여 온프레미스(On-Premise) 주거 침입 탐지 시스템을 제안한다. 시스템은 서버와 네트워크 카메라로만 이루어진 별도의 내부망 네트워크를 구축하여 취약점을 완화하였다. 서버에서 생성된 침입 이벤트를 외부 네트워크로 전달하기 위해 TLS[3] 보안 연결을 적용하였다. 실시간 영상을 활용하여 침입 여부를 판별하기 위해 저전력 딥러닝 연산 장치로 Jetson Nano를 이용하였다. 주거지 침입을 탐지하기 위해 NVIDIA Transfer Learning Toolkit과 NVIDIA 자체 데이터셋으로 미리 학습된 DetectNetV2_ResNet10 모델을 활용하였다. 실험 중 침입탐지 알고리즘에서 거주자가 침입자로 판별되어 알림이 울릴 수 있다는 점을 확인한 뒤, BLE(Bluetooth Low Energy) 비컨을 이용하여 주거자의 스마트폰 비컨을 탐색하고 이를 발견할 시 침입 이벤트를 발생하지 않도록 하여 문제점을 해결하였다. 서버에서 발생한 침입 이벤트는 RabbitMQ를 이용하여 안드로이드 애플리케이션으로 전송하고 사용자가 확인할 수 있도록 하였다.

2. 설계 및 구현

시스템의 전체 과정은 객체 탐지 단계와 후처리 단계로 구분하여 그림 1과 같이 설계하고, 이에 따라 전체 과정을 Jetson Nano에 구현하였다.

2.1. 객체 탐지 단계

서버 시작 시 사용자가 미리 설정한 CCTV 영상의 RTSP 주소에 연결하였다. Deepstream Framework는 RTSP 패킷 내 H.264 압축 영상을 하드웨어를 이용하여 디코딩하여 TensorRT 라이브러리에 전달하는 동작을 반복한다. TensorRT는 먼저 DetectNetV2_ResNet10 모델을 읽고 FP16 정확도 양자화[4]를 수행하여 메모리에 저장하고 추론한다. 이후 TensorRT 라이브러리에서 찾아낸 객체 정보들을 NvDCF 트래킹 모듈을 이용하여 같은 객체가 여러번 감지되지 않도록 객체 추적을 수행한 뒤 이어서 후처리 단계를 거쳤다.

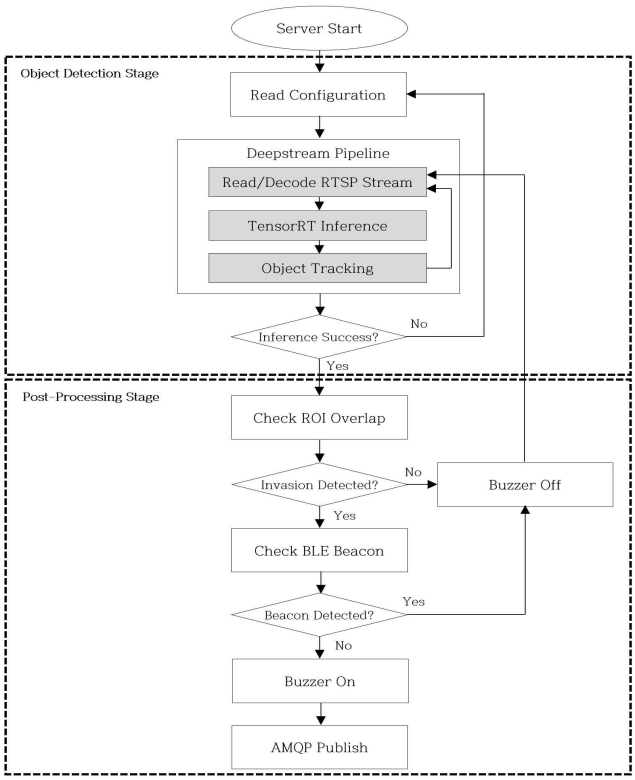
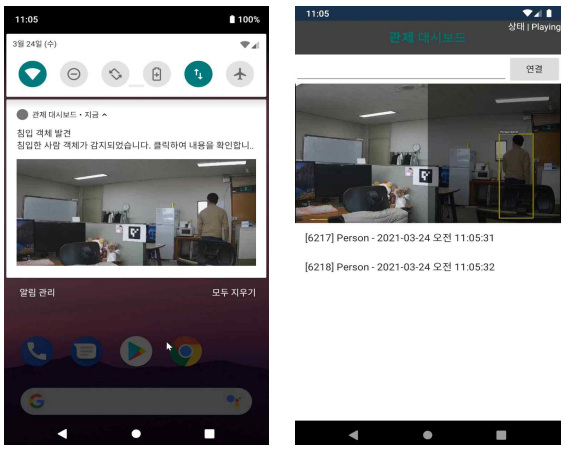


그림 1. 젯슨 나노를 이용한
온프레미스 주거 침입 탐지 시스템 구조



(가) (나)

그림 2. 안드로이드 애플리케이션 화면 구성
(가) 알림창 화면 (나) 애플리케이션 내부 화면

2.2. 후처리 단계

후처리 단계에서 시스템은 추적된 객체와 미리 설정된 관심 영역(ROI) 간의 겹치는 부분을 계산하고 겹친 부분이 객체 크기의 75%보다 작을 때에는 이벤트를 발생하지 않도록 하였다. 서버는 주거자의 스마트폰의 비컨을 탐색하여 주거자가 침입자와 동일인인지 확인하는 과정을 거친다. 스마트폰 비컨을 발견하지 못했을 경우 주거자가 아닌 것으로 판단하고 침입 이벤트를 발생한다. 이벤트 발생 시 객체 정보와 시간 정보를 RabbitMQ에 삽입하고 버저를 울린다. 안드로이드 애플리케이션은 RabbitMQ에 삽입된 정보를 주기적으로 갱신하여 객체 정보가 있는 경우 해당 내용을 큐에서 가져와 사용자에게 그림 2와 같이 보여준다. 사용자는 스마트폰의 실시간 알림과 버저를 통해 침입자에 대한 정보와 영상을 확인할 수 있다.

3. 실험 환경

본 논문의 시스템은 Jetson Nano 2GB 모델을 사용하여 구축하였으며 서버는 표 1에 해당하는 소프트웨어를 사용하여 구현하였다.

소프트웨어명	소프트웨어 버전
Deepstream SDK	5.0.1
TensorRT	7.1
CUDA	10.2
RabbitMQ	3.8.11
L4T(Linux4Tegra)	32.5.1
Docker	19.03.6
Ubuntu Linux	18.04.5 LTS

표 1. 소프트웨어 버전

실시간 영상 확인을 위해 네트워크 카메라 Dahua DH-IPC-8242FN을 사용하였으며 1920x1080 해상도 및 30FPS로 설정하여 실험하였다.

시스템 배포에는 Docker를 이용하였다. 이를 통해 개발 환경과 결과물을 이미지의 형태로 Jetson Nano에 배포하여 각종 환경변수 변화에 강인한 시스템을 구축하였다. 안드로이드 애플리케이션은 C#과 Xamarin을 이용하여 개발하였다.

4. 결론

본 논문에서는 NVIDIA사의 Jetson Nano와 미리 훈련된 모델을 이용한 침입 감지 시스템을 구현하였다. 시스템 설계 간 보안성을 높이기 위해 서버와 네트워크 카메라 간 내부망을 구축하였으며 RabbitMQ로 이벤트 내용을 암호화하여 전송하였다. 관심 영역을 이용한 침입 이벤트 발생과 이를 전송하는 시스템을 제안하고 이를 안드로이드 애플리케이션으로 구현하여 작동을 확인하였다.

참고문헌

[1] Eric Zeng, Shrirang Mare, Franziska Roesner, "End User Security and Privacy Concerns with Smart Homes" Proceedings of the Thirteenth USENIX Conference on Usable Privacy and Security, pp. 65-80, July. 2017.

[2] Myongyeal Lee, Jaepyo Park, "Analysis and Study on Invasion Threat and Security Measures for Smart Home Services in IoT Environment" The Journal of The Institute of Internet,

Broadcasting and Communication (IIBC), Vol. 16, No. 5, pp. 27-32, Oct. 2016.

[3] T. Dierks, E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2" Network Working Group, Aug. 2008.

[4] Hao Wu, Patrick Judd, Xiaojie Zhang, Mikhail Isaev, Paulius Micikevicius, "Integer Quantization for Deep Learning Inference: Principles and Empirical Evaluation" eprint arXiv:2004.09602, Apr. 2020.

[5] Michael Cusumano, "Cloud computing and SaaS as new computing platforms" Communications of the ACM, Vol. 53, No. 4, Apr. 2010.