

<https://doi.org/10.7236/IIBC.2019.19.2.161>

IIBC 2019-2-22

## 지능형 행동인식 기술을 이용한 실시간 동영상 감시 시스템 개발

### Development of Real-time Video Surveillance System Using the Intelligent Behavior Recognition Technique

장재영\*, 홍성문\*\*, 손다미\*\*, 유호진\*\*, 안형우\*\*

Jae-Young Chang\*, Sung-Mun Hong\*\*, Dami Son\*\*, Hojin Yoo\*\*, Hyoung-Woo Ahn\*\*

**요약** 최근에 빠르게 확산되고 있는 CCTV와 같은 영상기기들은 거의 모든 공공기관, 기업, 가정 등에서 비정상적인 상황을 감시하고 대처하기 위한 수단으로 활용되고 있다. 그러나 대부분의 경우 이상상황에 대한 인식은 모니터링하고 있는 사람에 의해 수동적으로 이루어지고 있어 즉각적인 대처가 미흡하며 사후 분석용으로만 활용되고 있다. 본 논문에서는 최신 딥러닝 기술과 실시간 전송기술을 활용하여 이벤트 발생시 스마트폰으로 이상 상황을 동영상과 함께 실시간으로 전송하는 동영상 감시 시스템의 개발 결과를 제시한다. 개발된 시스템은 오픈포즈 라이브러리를 이용하여 실시간으로 동영상으로 부터 인간 객체를 스켈레톤으로 모델링한 후, 딥러닝 기술을 이용하여 인간의 행동을 자동으로 인식하도록 구현하였다. 이를 위해 Caffe 프레임워크를 개발된 오픈포즈 라이브러리를 다크넷 기반으로 재구축하여 실시간 처리 능력을 대폭 향상 시켰으며, 실험을 통해 성능을 검증하였다. 본 논문에서 소개할 시스템은 정확하고 빠른 행동인식 성능과 확장성을 갖추고 있어 다양한 용도의 동영상 감시 시스템에 활용될 수 있을 것으로 기대된다.

**Abstract** Recently, video equipments such as CCTV, which is spreading rapidly, is being used as a means to monitor and cope with abnormal situations in almost governments, companies, and households. However, in most cases, since recognizing the abnormal situation is carried out by the monitoring person, the immediate response is difficult and is used only for post-analysis. In this paper, we present the results of the development of video surveillance system that automatically recognizing the abnormal situations and sending such events to the smartphone immediately using the latest deep learning technology. The proposed system extracts skeletons from the human objects in real time using Openpose library and then recognizes the human behaviors automatically using deep learning technology. To this end, we reconstruct Openpose library, which developed in the Caffe framework, on Darknet framework to improve real-time processing. We also verified the performance improvement through experiments. The system to be introduced in this paper has accurate and fast behavioral recognition performance and scalability, so it is expected that it can be used for video surveillance systems for various applications.

**Key Words** : Video Surveillance System, Behavior Recognition, Openpose, Deep Learning

## 1. 서론

동영상 감시 시스템(video surveillance system)은 범

죄, 화재, 사고 등을 실시간으로 감시하고 예방하기 위한 시스템으로 최근 CCTV의 광범위한 보급으로 공공기관, 기업, 가정 등 다양한 분야에서 활용되고 있다. 과거의 동

\*정회원, 한성대학교 컴퓨터공학부

\*\*학생회원, 한성대학교 컴퓨터공학부

접수일자 2019년 3월 12일, 수정완료 2019년 4월 3일

게재확정일자 2019년 4월 5일

Received: 12 March, 2019 / Revised: 3 April, 2019 /

Accepted: 5 April, 2019

\*Corresponding Author: jychang@hansung.ac.kr

Dept. of Computer Engineering, Hansung University, Korea

영상 감시 시스템은 카메라와 연결된 녹화장치에 단순히 기록하는 수준에 머물렀으나 점차 기술의 진화로 인터넷을 이용하여 원격으로 저장하거나 감시자가 실시간으로 모니터링하는 수준으로 발전하였다. 하지만 다수의 감시 영상을 24시간 모니터링하는 것은 시간과 비용 측면에서 매우 비효율적이다. 예를 들어 사회적 약자가 머무르는 병원, 요양원, 유치원 등에서 발생한 폭력이나 사고는 더욱 즉각적인 대처가 필요하지만 인력과 예산의 부족으로 실시간 대처가 거의 이루어지지 않고 있다. 이 경우 사건이 발생한 이후 사후 분석용으로 활용할 수밖에 없는 한계가 있다. 조사에 따르면 기존 CCTV 시스템은 관제 요원이 감시를 하더라도 모니터링 시작 후 22분이 지난 후에 약 95%의 감시 오류가 발생한다는 실험 결과가 있다<sup>[1]</sup>. 이에 따라 기존의 CCTV를 일일이 사람이 관찰하는 방법은 관제 능력을 현저히 떨어트린다는 것을 알 수 있다. 따라서 범죄나 사고 같은 이상(abnormal) 행동을 자동으로 인식하고, 이러한 상황을 실시간으로 전달하는 지능형 영상 감시 시스템의 필요성이 점차 높아지고 있다.

기존의 자동 행동인식(behavior recognition) 기술은 컴퓨터 비전(vision) 분야에서 주로 연구되었다<sup>[2,3]</sup>. 예를 들어 동영상에서 움직임이 발생할 경우 객체의 특징 추출하고 벡터화하는 알고리즘을 활용하여, 벡터들의 변화가 사전에 정해진 행동인지를 판별하는 방식을 사용하였다. 하지만 이러한 기존의 방식은 객체나 행동의 다양한 변화에 대처하지 못하여 정확성이 떨어지는 문제를 안고 있다. 따라서 상용 감시 시스템에서도 단순히 객체를 탐지하거나 추적하는 정도의 기능만을 제공하고 있고, 객체를 인식하고 그 객체의 행동양상까지 인식하는 수준의 제품을 출시하지는 못하고 있다. 하지만 최근 들어 딥러닝(deep learning)으로 대표되는 인공지능 기술의 비약적인 발전으로 객체를 탐지하고 학습을 통해 자동으로 행동을 인식하는 기술이 현실화되고 있다<sup>[4, 5]</sup>.

본 논문에서는 딥러닝을 활용하여 동영상에서 인간을 인식하고, 행동을 실시간으로 판별하는 동영상 감시 시스템의 개발결과를 제시한다. 제안된 시스템은 단순히 행동의 주체인 인간을 탐지만 하는 것뿐만 아니라 딥러닝을 기반으로 인간이 어떤 행동을 하고 있는지 실시간으로 감지할 수 있도록 구현하였다. 또한 감지 후에 모바일 애플리케이션으로 구현된 클라이언트에게 실시간으로 알림(notification) 서비스를 제공하는 기능을 포함한다. 인간을 인식하기 위한 방법으로 동영상 데이터로 부

터 인간객체를 스켈레톤(skeleton)으로 모델링해주는 오픈포즈(Openpose)<sup>[6]</sup> 라이브러리를 활용하였다. 오픈포즈는 2017년 카네기멜론(Carnegie Mellon) 대학에서 개발한 오픈소스(open source)로, OpenCV와 Caffe 프레임워크를 기반으로 개발되었다. 이러한 스켈레톤 모델을 이용하여 딥러닝 알고리즘을 통해 다양한 인간에 대한 행동들을 학습한 후 폭력 등의 행동을 자동으로 인식하도록 구현하였다. 특히 동영상 감시 시스템은 실시간으로 행동을 인식하는 것이 중요하므로 기존의 Caffe 프레임워크<sup>[7]</sup>에서 구현된 오픈포즈의 딥러닝 네트워크 모델을 보다 빠른 다크넷(Darknet)<sup>[8]</sup> 기반으로 재구축하여 실시간 감지가 가능하도록 구현하였다. 또한 자동인식이 필요한 인간행동 종류에 따라 지속적인 실시간 탐지와 정보 분석이 가능하도록 설계하였으며 그에 따른 경고나 알림 서비스를 통해 실시간 관제가 가능하도록 개발하였다. 따라서 방화, 도난 등의 행동뿐만 아니라 노약자와 같은 지속적인 관찰이 필요한 상황과 같은 다양한 용도로 확장 시킬 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 전반적인 동영상 감시 시스템의 구조를 제시한다. 3장에서는 본 시스템의 핵심인 행동인식 과정을 설명하고, 4장에서는 스마트폰으로 이벤트 발생 정보를 전송하는 시스템의 구조와 과정을 설명한다. 마지막으로 5장에서는 결론을 맺는다.

## II. 동영상 감시 시스템의 구조

본 논문에서 제안하는 전반적인 동영상 감시 시스템의 구조는 그림 1과 같다. 우선 CCTV나 IP 카메라와 연결된 RTSP 서버는 다크넷 기반으로 구현된 행동인식 서버에 실시간으로 동영상을 전송한다. 촬영된 동영상은 스마트폰과 같은 사용자의 애플리케이션으로도 실시간 전송하여 모니터링할 수 있다. 행동인식 서버에서는 동영상에서 인간객체를 인식하고 스켈레톤을 추출한다. 추출된 스켈레톤 정보는 딥러닝을 통해 사전에 학습된 이상행동인지를 판별한다. 이때 딥러닝 엔진에 입력되는 벡터화된 스켈레톤 정보는 사전에 정해진 프레임 수만큼의 정보를 모두 입력하여 특정 시간동안에 변화된 벡터들로 이상 행동 유무를 판단한다. 행동인식 서버에서 사전에 학습된 이상행동이 감지되면 감지된 시점의 동영상

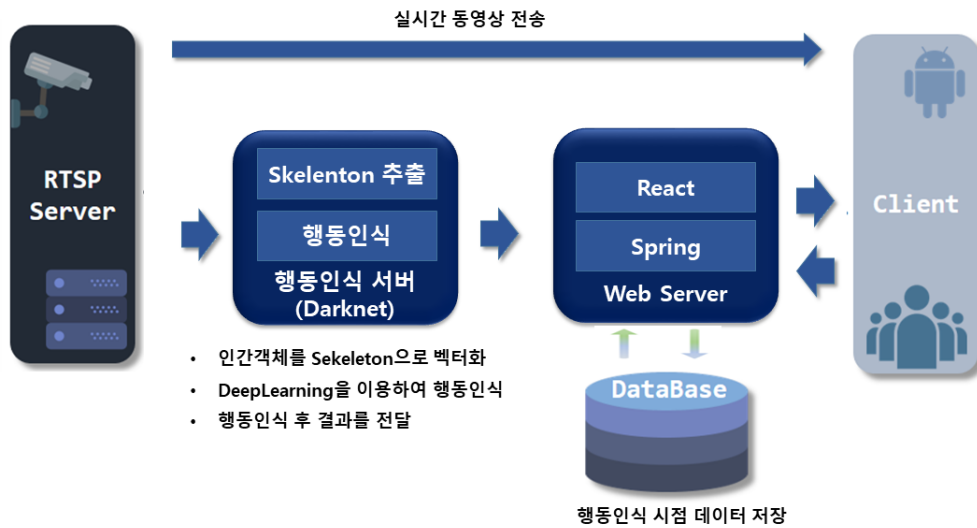


그림 1. 동영상 감시 시스템의 전반적인 구조  
 Fig. 1. Overall Architecture of Real-time Video Surveillance System

클립(clip)과 촬영일, 장소 등의 메타 데이터를 웹서버로 전송한다. 이때 웹서버는 동영상을 포함한 관련 데이터를 데이터베이스에 저장한다. 이와 동시에 클라이언트 쪽으로도 상황이 발생하였음을 알려주며, 클라이언트는 백그라운드에서 알림 기능을 이용하여 사용자가 인지할 수 있게 해준다.

사용자가 동영상을 다운로드하기를 원할 때 스프링(Spring)을 통해 파일의 데이터들을 받을 수 있다. 웹서버에서 스프링은 후위(back-end) 처리를 하고 전단부(front-end)는 React<sup>[9]</sup>를 사용하여 웹뷰(web view)를 표현한다. 사용자가 단말기에서 동영상 리스트를 요청하면 스프링 서버에서는 데이터베이스에 접근하여 저장되어 있는 모든 동영상을 검색하여 사용자에게 전달한다. 이를 통해 사용자는 이상행동에 대한 동영상들을 선택적으로 다운로드 받아 확인할 수 있다. 결과적으로 사용자 단말기에서는 RTSP 서버로부터 직접 받은 실시간 동영상 데이터와 행동인식 서버와 웹서버를 거쳐 받은 폭력 발생 시점의 데이터를 확인할 수 있다.

### III. 딥러닝을 이용한 행동인식

그림 1의 행동인식 서버는 실시간 동영상을 입력으로 받고 입력 비디오에서 특징(feature) 추출 및 전처리

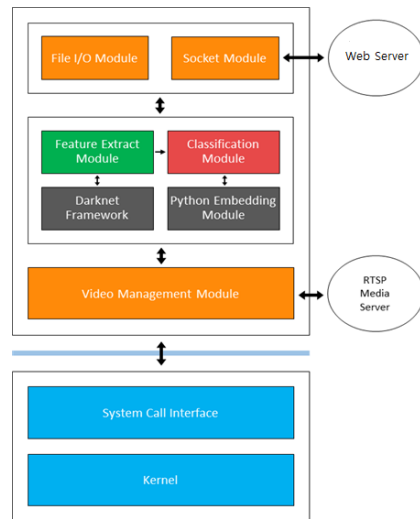


그림 2. 행동인식 서버 구조  
 Fig. 2. Behavior Recognition Server Architecture

(preprocessing)를 진행한 후 어떤 행동인지 분류(classification)한다. 분류된 행동인식 결과와 해당 동영상은 웹 서버로 전송된다. 행동인식 서버의 구체적인 구조는 그림 2와 같다. 이 그림에서 비디오 관리 모듈(Video Management Module)은 RTSP 미디어 서버로부터 전송된 동영상을 OpenCV를 이용하여 프레임(frame) 단위로 분할한다. 분할된 데이터는 다크넷을 기

반으로 구축된 특징추출 모듈(Feature Extraction Module)에서 인간객체를 자동으로 탐색하고 스켈레톤 정보를 추출한다. 스켈레톤 추출은 다크넷 기반으로 재구축된 오픈포즈를 이용하였다. 오픈포즈는 동영상으로부터 인간 객체를 인식하고 스켈레톤으로 모델링하도록 학습된 딥러닝 네트워크로 다단계의 CNN(Convolution Neural Network)<sup>[10]</sup> 구조를 갖는다. 스켈레톤 모델로 생성된 특징들은 행동인식 분류모델을 통해 어떠한 행동인지를 판별하고 그 결과를 웹 서버로 전송한다.

스켈레톤 데이터는 그림 3과 같이 사람의 양 귀와 눈, 코, 팔꿈치, 팔목, 허리, 무릎, 다리 등 18개의 포인트를 분류하는 keypoints\_pose\_18 분류 모델을 사용하였다<sup>[6]</sup>. 앞서 설명한 바와 같이 스켈레톤 추출에 사용된 오픈포즈 라이브러리는 본래 Caffe를 기반으로 개발되었으나 성능 향상을 위해 다크넷 프레임워크 기반으로 재구축하였다. 다크넷은 딥러닝 개발을 위한 프레임워크로 규모가 작고 성능이 매우 뛰어난 프레임워크로 알려져 있다.

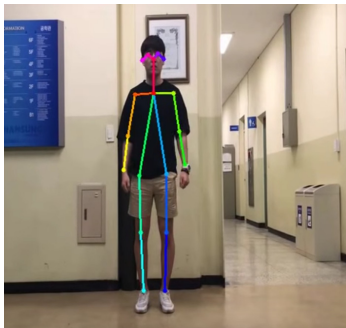


그림 3. 오픈포즈의 keypoints\_pose\_18 모델을 사용한 스켈레톤 추출 결과

Fig. 3. A Skeleton Extraction Example using keypoints\_pose\_18 Model of Openpose

본 시스템의 개발과정에 실험한 결과 기존의 Caffe 기반 오픈포즈 라이브러리를 통해 스켈레톤 데이터를 추출할 경우 평균 초당 11 프레임 밖에 추출할 수 없었다. 여기에 딥러닝을 통해 행동을 분류하는 모델까지 같이 동작할 경우 초당 6 프레임도 기대하기 힘든 수준이었다. 이러한 성능의 오픈포즈를 빠르게 동작하기 위해 다크넷을 환경에서 재구축하였다. 이와 같이 기존 오픈포즈의 107개의 딥러닝 네트워크 레이어(layer)를 다크넷에서 재구성한 후 특징을 추출하여 기존 평균 초당 11프레임에서 약 초당 40 프레임까지 성능을 향상시킬 수 있었다.

이 실험은 인텔 코어i7 7700k CPU와 48G RAM, 그리고 Titan XP GPU 환경에서 실행되었다.

스켈레톤 형태로 추출된 특징에 대해서 행동인식을 위한 분류는 파이썬(Python) 기반의 텐서플로우(Tensorflow) 환경에서 개발되었다. 학습 및 분류는 다음 과정을 따른다. 그림 4와 같이 오픈포즈를 통해 각 프레임마다 추출된 스켈레톤에 대해서 16개 프레임을 하나의 입력 데이터셋(dataset)으로 정의하였다. 초당 8개의 프레임이 사용되었으므로 16개 프레임은 2초 동안의 영상에 해당한다. 이렇게 생성된 16개의 프레임 정보를 차례로 큐(queue)에 삽입한다. 큐는 FIFO 구조이므로 첫째로 입력된 스켈레톤 정보가 동영상의 첫 프레임이고 가장 나중에 들어온 정보는 마지막 프레임이다. 오픈포즈로부터 지속적으로 스켈레톤 정보가 생성되면 그 정보는 매번 큐에 삽입하고 그때마다 큐에 저장된 16개의 프레임들이 하나의 데이터셋을 형성한다. 이와 같이 매 프레임마다 윈도우 형태로 학습과 분류를 위한 데이터를 생성하였다. 결과적으로 큐에 들어가 있는 데이터 셋은 시간에 흐름에 따른 인간의 행동 변화를 표현한다.

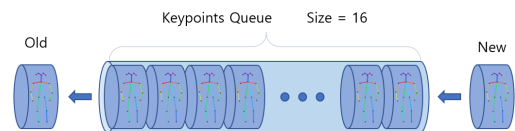


그림 4. 행동인식을 위한 학습 및 분류 데이터셋 생성 과정  
Fig. 4. Generation Process of Learning and Classification Datasets for Behavior Recognition

행동인식을 위한 분류 모델로는 SVM(Support Vector Machine), DNN(Deep Neural Network), RNN(Recurrent Neural Network)을 선정하였고 분류의 정확성을 평가하였다. DNN은 4개의 숨은층(hidden layer)으로 설계하였으며, 각 층(layer)는 256 ~ 512개의 노드(node)로 구성하였다. 출력은 13개의 클래스를 분류할 수 있도록 설계하였다. 클래스의 종류는 표 1과 같다. RNN은 시퀀스 길이(sequence length)를 16으로 하고 10개의 숨은층을 갖는 LSTM(Long Short-Term Memory)<sup>[11]</sup>을 사용하였다. 학습과 테스트를 위해서 클래스당 1,200개의 데이터를 인위적으로 생성하였으며, 학습과 테스트를 위해 데이터를 각각 7:3의 비율로 나누어 실험하였다.

표 1. 실험에 사용된 행동 클래스  
 Table 1. Behavior Classes in Experiments

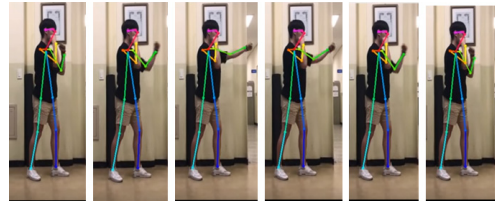
행동 클래스	설명
left-walk	왼쪽방향 걷기
right-walk	오른쪽 방향 걷기
hold	서있기
clap	손뼉치기
jump	제자리 뛰기
left_right_punch	왼쪽방향 오른손 펀치
left_left_punch	왼쪽방향 왼손 펀치
right_right_punch	오른쪽방향 오른손 펀치
right_left_punch	오른쪽방향 왼손 펀치
left_right-kick	왼쪽방향 오른발 차기
left_left_kick	왼쪽방향 왼발 차기
right_right_kick	오른쪽방향 오른발 차기
right_left_kick	오른쪽방향 왼발 차기

실험 결과는 표 2과 같다. 이 표에서 보는바와 같이 SVM에 비해서 DNN과 RNN이 높은 정확도를 보였다. RNN이 가장 높은 정확도를 보였지만 DNN에 비해서 차이는 크지 않았다. 실험환경이나 각 방법의 신경망 구조, 파라미터(parameter) 등의 변화에 따라 달라질 수 있으나 대체적으로 두 모델의 차이점은 크지 않은 것으로 판단된다. 본 실험은 각 분류별로 행동이 명확한 데이터들(걷기, 가만히 있기 등)을 이용하여 진행하였다. 하지만 동영상 데이터는 실시간으로 변화한다. 따라서 두 가지 행동이 겹치는 상황(예를 들어 걷는 행동 직후 점프를 하는 장면)에서의 분류는 취약할 수밖에 없다. 본 시스템의 구현과정에서 수행한 실험에서는 이러한 상황에서 DNN이 RNN보다 약간 우수한 정확도를 보였다. 하지만 정해진 행동을 인식하고 알림을 보내는 본 시스템의 개발 목적에는 큰 영향이 없어 보다 구체적인 실험과 분석은 실시하지 않았다. 만약 이러한 문제가 중요한 이슈가 될 경우에는 교차검증(cross validation)이나 앙상블(ensemble)과 같은 정확도를 높이기 위한 다양한 방법이 시도되어야 할 것으로 보인다.

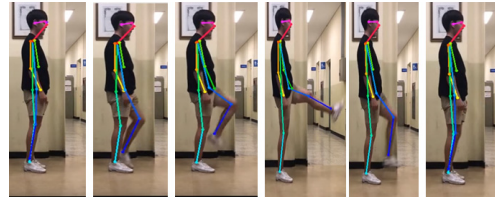
표 2. 행동인식 정확도 실험 결과  
 Table 2. Experimental Results of Behavior Recognition Accuracy

분류모델	분류 정확도(%)
SVM	92.39
DNN	94.14
RNN	94.19

그림 5는 학습과 테스트에 사용된 동영상에서 주요 동작들에 대한 각 장면들의 예를 보여준다. 그림 5(a)는 펀치 동작을 나타내며 그림 5(b)는 발차기 동작이다. 이 그림에서 보는 바와 같이 각 행동에 대해서 오픈포즈가 정확한 스키텔론을 실시간으로 추출하고 있음을 알 수 있다.



(a) 펀치 동작



(b) 발차기 동작

그림 5. 행동인식 학습에 사용된 주요 동작과 추출된 스키텔론 정보

Fig. 5. Sample Actions and Extracted Skeletons

## IV. 웹서버와 클라이언트 구조

### 1. 스프링 서버와 React

이상행동으로 판명된 동영상과 관련정보를 클라이언트에게 전송하기 위한 웹서버 구조는 그림 6과 같다. 스프링 서버는 RESTFUL API와 소켓으로 통신하는 구조로 되어있으며 동영상 목록, 감지 알림, 동영상 다운로드 기능을 제공한다. 먼저 동영상 목록 제공기능은 클라이언트로부터 해당 요청이 오면 데이터베이스로 부터 동영상 리스트를 생성하고 전달한다. 감지 알림은 행동인식 서버에서 이상행동이 감지되면 소켓을 통해 해당 동영상 정보를 보내준다. 해당 동영상 정보를 받으면 데이터베이스에 그 정보를 기록한다. 이후 클라이언트에게 이상행동을 감지했다는 푸쉬(push)를 요청을 한다. 마지막으로 동영상 다운로드 기능은 클라이언트에서 요청이 오면 저장소에 있는 동영상들 중에 해당 동영상을 탐색하고 소켓을 통해 동영상 파일을 클라이언트에게 전송한다.

React는 자바스크립트 라이브러리로 사용자 인터페이스를 만들기 위해 사용되었다. React는 Node.js를 사용하여 개발되며 뷰(view)는 React 프레임워크를 사용하여 사용자에게 제공된다. 동영상 목록은 Axios 모듈을 사용하여 스프링 서버에게 요청을 보내고 결과를 전달 받는다.

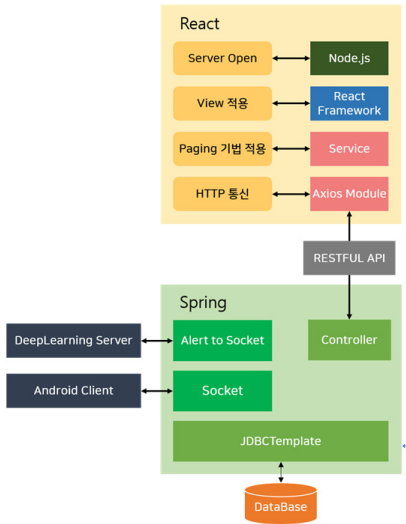


그림 6. 웹서버 구조  
Fig. 6. Web Server Architecture

## 2. 클라이언트 구조

사용자에게 서비스되는 클라이언트는 안드로이드 환경에서 개발되었다. 전반적인 구조는 그림 7과 같다. 사용자 인터페이스는 액티비티(activity)가 계속 생기거나 소멸되는 방식보다 프래그먼트(fragment)를 이용함으로써 사용자 인터페이스 스레드(thread)가 처리 할 작업을 감소시키는 방식으로 개발되었다. 각 프래그먼트들은 TabLayout으로 관리되며, 사용자가 보다 사용하기 쉽게 해당 프래그먼트를 이동시킬 수 있는 구조로 개발하였다. 우선 프래그먼트 파일에서는 React에서 뷰를 제공받아 웹뷰 형태로 보여준다. RtsplayView에서는 FFmpeg와 JNI 라이브러리를 이용하여 RTSP 서버로부터 전달된 데이터를 이용하게 된다. NDKAdapter는 안드로이드가 JNI 라이브러리를 참조 할 수 있게 해주는 역할을 한다. 이를 토대로 RTSP 서버에서 넘어온 데이터들을 FFmpeg를 통하여 인코딩/디코딩을 하게 되고, 서피스뷰(SurfaceView)를 통해 사용자에게 실시간으로 동영상

제공한다.

안드로이드 서비스에서는 스프링으로부터 넘어올 데이터들을 받는 소켓과 스트림들을 준비한다. 소켓은 '이상행동 알림용'과 '동영상 다운로드'를 위한 두 가지 용도에 따라 분리하여 개발하였다. 이 작업들은 애플리케이션이 종료되거나 슬립(sleep)상태여도, 백그라운드에서 계속 작동하며, 사용자가 언제든 알림을 받을 수 있게 된다.

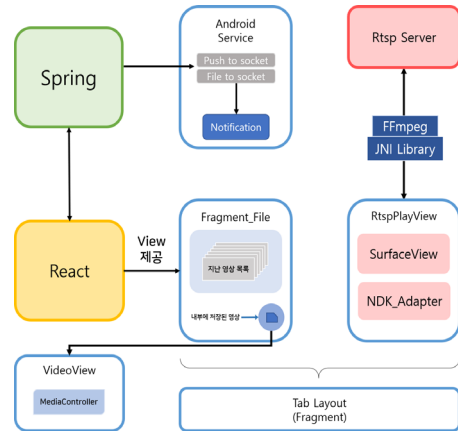


그림 7. 클라이언트 구조  
Fig. 7. Client Architecture

## V. 결론

본 논문에서는 동영상 소스로 부터 실시간으로 인간의 이상행동을 자동으로 인식하고 사용자 단말기에 알림 서비스를 제공하는 동영상 감시 시스템의 개발 결과를 소개하였다. 기존의 Caffe 기반의 오픈포즈에서는 성능 문제로 유실되는 프레임들이 다수 발생하였으나 다크넷 기반으로 딥러닝 네트워크를 재구축하여 성능을 획기적으로 향상시켰다. 뿐만 아니라 행동인식에서는 DNN과 RNN 등의 딥러닝 기술을 이용하여 90%가 넘는 높은 인식률을 보였다.

본 시스템은 학습시키는 이상행동의 종류에 따라 지속적인 실시간 탐지와 정보 분석이 가능하며, 그에 따른 경고나 알림 서비스를 통해 실시간 관계 능력을 향상시킬 수 있다. 대표적으로 CCTV를 이용하여 폭행 등의 이상행동을 감지하는데 활용될 수 있으며, 방화, 도난 등의 범죄행위 감지에 폭넓게 응용될 수 있다. 또한 환자, 노약



자, 유아 등 지속적 관찰이 필요하나 사람에 의해 계속적인 모니터링이 불가능한 경우에도 활용될 수 있다.

다만 본 시스템의 동영상에서 하나의 인간 객체에 대한 행동인식에 대해서만 학습하였으며 두 명 이상의 인간객체에 행동 인식은 아직 개발과 테스트가 이루어지지 않은 상태이다. 예를 들어 폭력 상황의 경우 두 명 이상의 객체에 대한 행동 인식이 동시에 이루어져야 가능하다. 이를 위해서는 한명이 아닌 여러 명에 대한 스키텔론을 동시에 학습해야하므로 행동인식을 위한 딥러닝 모델을 개선할 필요가 있다. 향후에는 이러한 문제의 해결에 중점을 두고 연구를 지속할 계획이다.

## References

- [1] National Institute of Justice(NIJ), Buyer Beware Vol.10/11, Ministry of Justice, USA, 2002.
- [2] X. Ji and H. Liu, Advances in view-invariant human motion analysis: A review, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 40(1) (2010), pp. 13-24.  
DOI: 10.1109/TSMCC.2009.2027608
- [3] J. Aggarwal and Q. Cai, Human motion analysis, Computer Vision and Image Understanding, 73(3) (1999), pp. 428-440.  
DOI: <https://doi.org/10.1006/cviu.1998.0744>
- [4] Z. Yin, C. Quangi, and Z. Yujin. Deep learning and its new progress in object and behavior recognition, Journal of image and graphics 19(2) (2014), pp. 175-184.
- [5] Q. Li, Z. Qiu, T. Yao, T. Mei, Y. Rui, and J. Luo, Action recognition by learning deep multi-granular spatio-temporal video representation. In Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, (2016), pp. 159-166.  
DOI: 10.1145/2911996.2912001
- [6] Z. Cao, T. Simon, S. Wei, and Y. Sheikh, Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.  
DOI: 10.1109/CVPR.2017.143
- [7] Y. Jia, et al, Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia, (2014), pp. 675-678.  
DOI: 10.1145/2647868.2654889
- [8] J. Redmon, Darknet: Open Source Neural Networks, Available in <http://pjreddie.com/darknet/>.
- [9] A. Banks and E. Porcello, Learning React: Functional Web Development with React and Redux, O'Reilly Media, Inc., (2017).
- [10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, (2012), pp. 1097-1105.  
DOI: 10.1145/3065386
- [11] S. Hochreiter and J. Schmidhuber, Long short-term memory, Neural Computation. 9 (8), (1997), PP. 1735 - 1780.  
DOI: <https://doi.org/10.1162/neco.1997.9.8.1735>
- [12] J. Kim and P. Rhee, Image Recognition based on Adaptive Deep Learning, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC) 18(1), (2018), pp.113-117

## 저자 소개

### 장재영(정회원)



- 1992년 : 서울대학교 계산통계학과 (이학사)
- 1994년 : 서울대학교 계산통계학과 (이학석사)
- 1999년 : 서울대학교 계산통계학과 (이학박사)
- 2000년~현재 : 한성대학교 컴퓨터공학부 교수

• 관심분야 : 데이터베이스, 데이터마이닝

홍 성 문(학생회원)



- 2013년 ~: 한성대학교 컴퓨터공학부
- 관심분야: 빅데이터분석, 데이터마  
이닝

손 다 미(학생회원)



- 2014년 ~: 한성대학교 컴퓨터공학부
- 관심분야: 빅데이터분석, 데이터마  
이닝

유 호 진(학생회원)



- 2014년 ~: 한성대학교 컴퓨터공학부
- 관심분야: 빅데이터분석, 데이터마  
이닝

안 형 우(학생회원)



- 2013년 ~: 한성대학교 컴퓨터공학부
- 관심분야: 빅데이터분석, 데이터마  
이닝

※ 본 연구는 한성대학교 교내학술연구비 지원과제임.