

졸업논문

(2D 영상 데이터 내) 다중객체 검출과 RNN
모델을 활용한 공공시설 이용자 이상행동 감지
모델 개발

Multi-object detection (in 2D image data) and abnormal
behavior detection model of public facilities using RNN model

2023년

세종과학예술영재학교

정보과학

이 소 민 (李 昭 旼, Lee, So Min)
3109

허 지 성 (許 智 盛, Hur, Ji Seong)
3115

(2D 영상 데이터 내) 다중객체 검출과 RNN 모델을 활용한 공공시설 이용자 이상행동 감지 모델 개발

Multi-object detection (in 2D image data) and abnormal behavior detection model of public facilities using RNN model

이 논문을 세종과학예술영재학교 졸업논문 심사를 위해 제출함

2022년 7월 8일

정보과학

이 소 민 (李 昭 旼, Lee, So Min)

3109

허 지 성 (許 智 盛, Hur, Jiseong)

3115

심사위원 김 민 호(인)

심사위원 최 윤 호(인)

지도교사 문 광 식(인)

(2D 영상 데이터 내) 다중객체 검출과 RNN
모델을 활용한 공공시설 이용자 이상행동 감지
모델 개발

Multi-object detection (in 2D image data) and abnormal
behavior detection model of public facilities using RNN model

초록

현재 다중 이용시설 이용자의 이상행동 감지는 사람에 의해 이루어지고 있으며 이는 정확도의 감소와 시간적, 금전적 낭비의 원인이 된다. 본 연구에서는 이상행동의 감지를 위하여 2D 영상 데이터에서의 다중객체 파라미터 추출을 진행하였으며 다중 이용시설 이용자의 폭행, 실신, 파손 등의 이상행동 감지를 적은 시간과 높은 정확도로 판별하는 것을 목적으로 한다.

영상의 다중객체 파라미터 추출을 사용하여 객체들의 특성을 파악하고, 이를 시계열 변수로 사용하는 인공지능을 활용하여 이상행동을 효과적으로 분류하는 모델을 도출해내도록 하였다. YOLOv3의 object tracking을 통하여 얻은 시계열 데이터를 Recurrent Neural Network (RNN)을 사용하여 분석한 뒤 파라미터 조절을 통해 최종 정확도 76%의 모델을 도출해내었다.

본 연구는 공항에서의 이상행동 데이터를 기반으로 하고 있지만, 이를 공공장소나 인구 밀집 지역, 보안 시스템 등에 사용할 수 있어 활용도가 매우 넓으며, 통상적으로 사용되던 3D 영상 데이터를 처리하기 위한 Convolutional Neural Network(CNN)이 아닌 YOLO와 RNN을 합성한 모델을 사용하는 새로운 접근 방식에 의의를 가진다.

목 차

I. 서론	1
II. 이론적 배경	1
2.1 YOLOv3를 사용한 객체 탐지	1
2.2 영상탐지에서의 파라미터 조절	2
2.3 Times Series Classification을 활용한 모델	2
2.4 일반적인 이상행동 탐지	2
III. 연구 방법 및 절차	3
3.1 가상환경 설정 및 Object Detection 모델 실행	3
3.1.1 가상환경 설정	3
3.1.2 Object Detection 모델 실행	3
3.2 데이터 활용을 위한 Object Detection 모델의 수정	4
3.2.1 학습 데이터	4
3.2.2 이미지의 데이터 도출	5
3.2.3 영상의 프레임별 데이터 저장	6
3.3 RNN을 활용한 시계열 데이터의 분류	7
3.3.1 데이터의 전처리	7
3.3.2 모델 설정	7
3.3.3 최종 모델의 변수 설정	8
3.3.4 RNN 모델 학습값 변화	8
IV. 연구 결과	9
V. 결론	10
VI. 참고문헌	12

표 목 차

표 1 학습에 사용된 ‘공항에서의 이상행동 데이터셋’의 영상 캡처 이미지	5
표 2 YOLOv3의 ouput 파라미터 라벨링 및 분류	5
표 3 판별할 이상행동의 종류와 라벨링	7
표 4 YOLOv3 input으로 사용한 학습 데이터 형식	7
표 5 각 이상행동에 특화된 파라미터 분리	8
표 6 폭행 - 파라미터 조합에 따른 정답률과 오답률	9
표 7 실신 - 파라미터 조합에 따른 정답률과 오답률	9
표 8 파손 - 파라미터 조합에 따른 정답률과 오답률	9
표 9 최종 모델의 파라미터 조합 및 라벨링	9

그림 목 차

그림 1 ‘공항에서의 이상행동 데이터셋’ 영상의 YOLOv3 output 화면 p4	4
그림 2 시계열 데이터의 처리 과정	7

I. 서론

현재의 공항 보안 시스템은 여러 대의 CCTV 영상을 대형 스크린에 띄워 24시간 주시하며 특이점이 발견될 경우 해당 CCTV를 확대해서 집중 감시하는 구조이다. 국내의 주요 공항인 인천 공항에는 약 9000여대의 CCTV가 설치되어있으며(2019년 기준) 공항 상황실과 세관, 법무부 출입 관리실에서 개별적으로 모니터링을 진행한다. 공항뿐만 아니라 지하철역, 공원 등 의 다중이용시설도 마찬가지이다. 이러한 시스템은 지속적인 감시를 필요로 하기 때문에 인력 낭비가 발생하고 이상행동을 하는 사람을 놓치는 사건이 발생할 수 있다. 이를 해결하기 위하여 효율적인 모니터링을 위한 추가적인 시스템 개발이 필요하며, 본 연구에서는 그 대안으로 딥러닝을 이용한 이상행동 감지 알고리즘을 통한 영상 인식 인공지능 개발을 제안한다. 소리 등 특정 신호에 반응하여 동작을 수행하는 CCTV는 이미 시중에 보급된 바 있지만, 특정 이상행동을 인식하고 분류하여 송출하는 CCTV는 아직 시도 단계에 그치고 있다. 이상행동을 스스로 분류하고 의심 행동을 보고하는 시스템이 갖추어진다면 기존보다 빠른 이상행동의 대처와 공공장소 치안 유지의 자동화가 진행될 수 있을 것이다.

이를 위하여 다중객체 감지를 위한 Object detection 기술과 시계열 데이터 처리를 위한 RNN 모델을 활용하여 공항에서의 이상행동을 감지하고 분류하는 시스템을 제작하고자 한다. 본 연구는 공항에서의 이상행동 데이터 셋을 기반으로 진행되지만, 파라미터 설정에 따른 추적 정확도를 비교한다는 점에서 다중이용시설 치안 유지 및 저비용으로 객체 다중 추적이 가능한다는 점에서 의의와 활용도가 있다.

II. 이론적 배경

2.1 YOLOv3를 사용한 객체 탐지

YOLOv3 모델은 여러 종류의 클래스를 학습에 이용하는 상황에서 유리하며 비교적 처리 속도가 짧은 편에 속한다. 따라서 실시간으로 여러 객체의 이상행동을 감지해야 하는 본 연구에 적합하다고 판단하여 YOLO v3 모델을 사용하기로 정하였다. YOLO v3 모델은 기본적으로 학습된 모델에 추가로 학습을 진행하여 모델의 성능을 강화할 수 있으나, 정지 상태에서 사람의 행동만을 주시한다는 점에서 YOLO v3 모델을 그대로 사용하여도 문제 없을 것이라 판단했다.

밀집 환경에서 다중 객체 검출을 효과적으로 수행하기 위해서는 여러 조작이 필요하다. Simple Online and Realtime Tracking (SORT)의 경우 다중 객체 추적 알고리즘으로 YOLO v3와 같은 별개의 객체 검출 시스템과 함께 사용된다. SORT는 뛰어난 객체 성능 추적을 보여주지만, CCTV 관제소가 설치되는 환경과 24시간 항시 모니터링하며 분석을 진행 해야 함을 고려하였을 때 본 연구에 적합하지 않다고 판단했다. 따라서 별개의 객체 검출 시스템 없이, 독자적인 연산과 파라미터 조정을 통해 다중 객체 검출을 수행하고자 한다.

2.2 영상탐지에서의 파라미터 조절

파라미터(매개변수)는 여러 프로그램에 입력값으로 제공되는 데이터 중 하나를 가리키기 위해 사용된다. CCTV를 활용해 동적 객체의 위치를 추적하는 아래 연구에서는 “추적된 각 객체에 대한 사각형은 향후, 시각화 정합을 위한 객체의 위치정보로 활용된다. 위 연구에서는 추적된 객체가 사람인 경우, Sankaranara -yanan(2008)의 방안을 토대로 사각형 하단의 중심 점을 대표 위치로 활용하였으며, 차량 등의 경우 객체의 형태를 간략화한 사각형의 높이, 너비를 고려하여 하단으로부터 1/4 지점의 중심점을 대표 위치로 채택하여 활용하였다.”라는 방식으로 객체의 파라미터를 적절하게 설정하였다. 본 연구에서도 위와 같은 방법으로 파라미터를 적절하게 설정하여 학습을 진행하고자 한다.

2.3 Times Series Classification을 활용한 모델

각 프레임에서 추출한 정보는 사건의 흐름이 반영되어 있지 않거나, 매우 적게 반영되어 있다. 이상행동의 특징상 일정 시간 이상의 관찰을 요구하기에 데이터셋에 시간의 흐름을 담을 수 있는 방법을 선택하였다. 아래 연구는 주가 예측에 필요한 적정 일수를 찾기 위하여, 데이터를 시간순으로 묶어서 실험을 진행하였다.

전체 실험은 10개 종목에 대하여, 각 종목당 29개의 주가 예측 요소들의 조합을 구성하였으며, 주가 예측에 필요한 적정 선행주가 일수를 찾아보기 위하여 1일($t-1$), 2일($t-2, t-1$), 3일($t-3, t-2, t-1$), 4일($t-4, t-3, t-2, t-1$), 5일($t-5, t-4, t-3, t-2, t-1$), 10일($t-10, \dots, t-1$), 20일($t-20, \dots, t-1$), 30일($t-30, \dots, t-1$), 40일($t-40, \dots, t-1$), 50일($t-50, \dots, t-1$)의 선행 주가를 가지고 총 2900개의 실험이 행해졌다.”

해당 기술은 본 연구에서 데이터에 시간을 표현하기 위한 방법으로 활용될 수 있을 것이다. 특정 수로 프레임을 묶어서 학습을 진행시키고, 적절한 중첩을 통하여 더욱 효과적인 이상행동 분석이 가능할 것이다.

2.4 일반적인 이상행동 탐지

인간의 행동을 탐지하는 주제는 인공지능 분야에서 중요하게 다루어지고 있다. 일반적으로 SITP, 그라디언트 히스토그램등이 사용되어 왔으며 최근 인공지능 기술의 발전으로 딥러닝을 활용한 이상행동 탐지 기술이 주목받고 있다. 비디오 데이터 분석에는 AlexNet과 같은 CNN 신경망이 적용된 모델이 주로 사용되는데, 프레임과 프레임 간의 상관관계가 반영되지 못해 좋은 결과를 얻지 못한다. 따라서 아래 연구에서는 기존 CNN 모델의 단점을 보완 및 개선하여 인간 행동 인식에 특화시키는 방법을 제안하였다.

우선 CNN에 모델에 여러 주의 모듈(attention module)을 쌓아 잔류 주의 네트워크(residual attention network)를 구축한다. 이후 각각의 주의 모듈에 시간 및 공간 데이터를 개별적으로 처리하기 위해 채널 주의(channel attention)와 공간 주의(spatial attention)모듈을 포함 시켜 구축하였다.

해당 방법은 시공간 정보 캡처 용량이 높아 학습 후 우수한 정확도를 보여준다는 장점이 있

다. 그러나 다수의 카메라에서 입력된 정보를 소수의 처리 시스템으로 판단해야 하는 폐쇄 회로의 특징상 시간적, 비용적 문제가 발생한다. 따라서 본 연구에서는 해당 모델을 직접적으로 사용하지 않고, 시간 및 공간 데이터를 개별적으로 처리해야 한다는 것을 알고 활용한다.

III. 연구과정 및 절차

3.1 가상환경 설정 및 Object Detection 모델 실행

3.1.1 가상환경 설정

연구 초기에는 visual studio 기반으로 실행되는 Darknet을 활용해 Object detecting을 시도했다. 위 방식에서는 CMAKE, NDIVIA CUDA, NDIVIA cuDNN 등 여러 보조 프로그램과 주변 환경설정을 요구한다. 설치 절차는 다음과 같다.

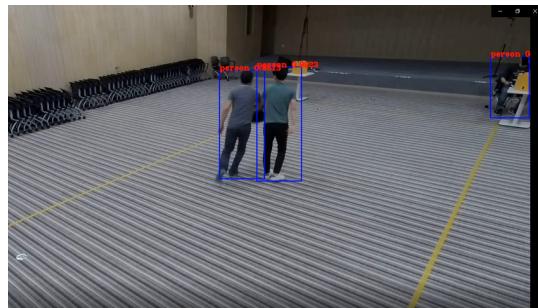
1. python 3.7.7, GIt 2.26.2, CMAKE 3.17.2, Visual Studio Installation 2019를 설치 한다.
2. 사용중인 그래픽카드 모델(RTX 2080 super)의 드라이버를 설치 및 업데이트를 한다.
3. 그래픽 카드에 알맞은 CUDA 및 cuDNN를 NDIVIA 사이트에서 설치한다.
4. OpenCV 4.1.0 설치하고 파일 경로를 설정해준다.
5. CMAKE를 사용해 OpenCV를 재구성한다.
6. Visual Studio 2019를 사용해 OpenCV 파일을 Build한다.
7. git clone 명령어를 활용해서 Git hub의 darknet 파일을 설치하고, 필요한 파일을 할당한다.
8. Visual Studio로 Darknet를 실행하고 코드를 작업 환경에 알맞게 변경한다.
9. Yolo v4와 Darknet을 Visual Studio의 버전을 바꿔가며 컴파일한다.
10. Test image와 video를 통해 작동을 테스트한다.

Anaconda를 활용해 tf_gpu를 설치 및 activate 했고, 해당 가상 환경에 Yolo v3를 설치하였다. 이후 pip를 활용해 OpenCV를 설치하니 일련의 과정 없이 간단하게 Yolo v3를 구동할 수 있는 환경을 조성할 수 있었다.

3.1.2 Object Detection 모델 실행

연구 초기에는 github에서 다운로드받은 기존의 YOLOv3 모델을 실행하였다. 이미지의 경우에는 감지된 물체의 종류와 정확도, 좌표를 출력하였다. 또한 원본 이미지에 위와 같은 내용을 물체에 표시한 직사각형 우측 상단에 표기하여 새로운 이미지로 저장하였다.

영상의 경우에는 프레임별로 영상을 분석해 각 프레임에 있는 물체의 종류와 정확도를 물체에 표시한 직사각형 우측 상단에 표기하여 새로운 동영상으로 저장하였다. 이때 프레임별로 감지된 물체에 대한 정보의 출력은 별도로 나타나지 않았다.



[그림 1] ‘공항에서의 이상행동 데이터셋’ 영상의 YOLOv3 output 화면

3.2 데이터 활용을 위한 Object Detection 모델의 수정

3.2.1 학습 데이터

AI HUB를 통해 ‘공항에서의 이상행동 데이터셋’의 다운로드 허가를 받아 이를 학습 데이터로 사용하였다. 데이터셋은 10초-5분 길이의 mp4 파일과 이상행동에 대한 정보가 담긴 xml 파일로 이루어져 있다. 각 이상행동에 대한 데이터셋의 개수가 학습에 충분하고, 실제 공항에서의 상황과 같이 이상행동에 참여하지 않는 사람들도 영상 내에 지속적으로 등장하며, 정상행동 데이터셋의 비중의 이상행동 데이터셋의 비중보다 크기 때문에 연구에 적합하다고 판단하여 사용하였다.

본 연구에서는 총 실신 348개, 파손 319개, 폭행 338개, 정상 2109개의 동영상 파일을 사용하였다. 학습 셋과 테스트 셋의 비율은 8:2로 지정해서 학습을 진행하였다. (학습을 위한 동영상 개수 : 실신 278개, 파손 255개, 폭행 270개, 정상 1687개)



[표 1] 학습에 사용된 ‘공항에서의 이상행동 데이터셋’의 영상 캡처 이미지

3.2.2 이미지의 데이터 도출

이미지의 데이터 저장을 위하여 pandas와 numpy 모듈 설치가 필요하다. 설정해둔 anaconda의 가상 환경에서 conda install openpyxl을 실행한다. 위에서 pandas, numpy 모듈을 import 한 뒤 이를 각각 pd, np 라는 이름으로 사용하였다. 한 행당 프레임에서 감지된 물체 하나를 배정하고, 이후 RNN에 입력할 데이터를 csv 파일로 저장한다. 이때 detection을 사용하여 도출해낸 값은 총 14개로, RNN 학습을 위하여 하나의 대푯값으로 나타낼 수 있는 특징들을 추출하였다. 실신, 파손, 폭행, 정상행동을 구분하기 위하여 유의미한 값을 가지는 특징들은 아래의 표의 항목들과 같다.

번호	변수명 라벨링	추출해낼 데이터
1	per_num	한 프레임당 사람의 수
2	obj_num	한 프레임당 물체의 수
3	per_num_c	전 프레임 대비 사람의 수 변화
4	obj_num_c	전 프레임 대비 물체의 수 변화
5	per_n_d	프레임 내에서 가장 거리가 가까운 사람들 사이의 거리
6	per_f_d	프레임 내에서 가장 거리가 먼 사람들 사이의 거리
7	per_obj_n_d	프레임 내에서 가장 거리가 가까운 사람과 물체 사이의 거리
8	per_obj_f_d	프레임 내에서 가장 거리가 먼 사람과 물체 사이의 거리
9	per_s_h	프레임 내에서 가장 작게 측정된 사람의 높이 (detection box 기준)
10	per_b_w	프레임 내에서 가장 크게 측정된 사람의 너비 (detection box 기준)
11	average_acc	프레임 내에서 측정된 모든 물체의 평균 정확도
12	nesting_size	프레임 내에서 가장 많이 중첩된 면적의 크기
13	per_nesting_size	프레임 내에서 가장 많이 중첩된 면적의 크기 / 사람의 면적
14	per_s_size	프레임 내에서 가장 작은 사람의 면적

[표 2] YOLOv3의 output 파라미터 라벨링 및 분류

위와 같이 변수를 설정한 이유는 아래와 같다.

- 1 - 2 : 각 프레임을 나타내는 주요변수
- 3 - 4 : 사람 및 물체의 새로운 등장 및 퇴장을 나타내기 위한 변수
- 5 - 6 : 폭행 등 이상행동이 발생했을 때 사람들의 분포도를 확인하기 위한 변수
- 7 : 도구의 사용 빈도가 높은 이상행동을 특정하기 위한 변수
- 8 : train set에서 촬영용 소품이 인식되어 해당 부분을 제거하기 위해 설정한 변수.
- 9 - 10 : 폭행 피해자는 폭행 과정에서 몸을 움츠리는 경향이 있으며 높이와 너비가 모두 낮은 형태를 취함. 실신한 사람은 바닥에 쓰러진 상태로 머무르기에 너비가 길고 높이가 낮은 형태를 취함
- 11 : 다른 파라미터 값들의 신뢰도를 측정하는 함수로 사용
- 12 : e 변수보다 값이 크게 나타나므로 사람이 근접하여 겹쳐지는 상황에서 효과적으로 사용 가능
- 13 : 거리에 따라 겹쳐진 면적의 크기가 다르다는 점을 보완하기 위해서 설정한 변수
- 14 : 9 - 10를 보완하기 위한 변수, 사람이 움츠리는 상황을 인지하기 쉬울 것으로 예상

따라서 detection box가 표시되는 동영상 파일을 output으로 가지는 기존의 YOLOv3 모델을 위에서 설정한 1-14번 변수를 output으로 가지는 모델로 수정하는 과정을 거쳤다. 이를 위해 사람과 물체를 구별하여 list에 이들의 detection box 정보와 함께 저장하였다. 그 과정에서 단일 객체 내에서 도출해낼 수 있는 변수들을 list에 저장하였다(사람이 가지는 최대 너비, 최소 높이, 사람이 가지는 면적 등). 하나의 프레임에 대한 detection이 모두 끝나고 나면, list에 저장되어 있는 값들을 통해 여러 객체를 비교하여 도출해낼 수 있는 변수들을 계산하였다(사람과 사람 사이 최소/최대 거리, 물체와 사람 사이 최소/최대 거리, 프레임 내에서 가장 많이 중첩된 면적의 크기 등). 같은 방식으로 프레임 내에서 측정된 모든 물체의 평균 정확도 등의 변수도 list에 저장한다.

3.2.3 영상의 프레임별 데이터 저장

변수는 모두 ‘parameters’이라는 리스트(14개의 리스트의 묶음)에서 관리하였다. 2.1에서 도출해낸 detection box에 대한 변수를 2.2에서 1-14번 변수로 가공한 뒤 영상이 끝날 때 까지 각 parameter를 14개의 리스트로 나누어 저장하였다.

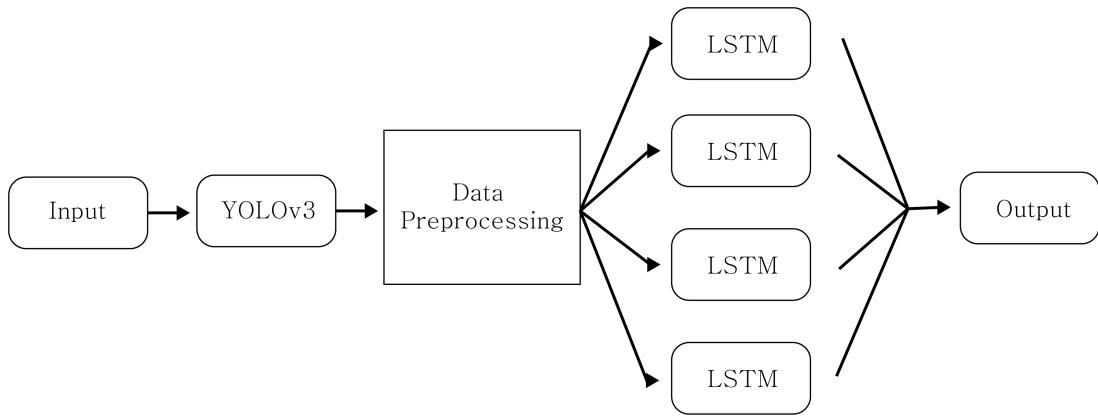
이미지의 데이터 저장을 위하여 pandas와 numpy 모듈 설치가 필요하다. 설정해둔 anaconda의 가상 환경에서 영상의 프레임별 데이터 출력을 위하여 2.1의 변수 10개를 리스트로 저장하는 기능을 기존모델의 detect_video.py에 추가하였다. 이때 csv 파일에 저장하기 위하여 divide라는 변수를 설정하여 divide개의 변수를 한 묶음으로 가지는 리스트를 생성하였다. 30fps의 영상을 사용했으므로 divide를 80으로 지정하여 model이 160 frame(영상길이 5초)를 하나의 묶음으로 보고 학습하도록 했다. 하나의 파라미터에 대한 값들의 나열을 a_i 라고 할 때, 리스트의 분할 방식은 아래와 같다.

$$[a_1, a_2, a_3, \dots] \rightarrow [[a_1, \dots, a_{80}, a_{81}, \dots, a_{160}], [a_{81}, \dots, a_{160}, a_{161}, \dots, a_{240}], \dots, [a_{80n+1}, \dots, a_{80(n+1)}, a_{80(n+1)+1}, \dots, a_{80(n+2)}]]$$

또한 기존의 영상 데이터에서는 영상이 끝나면 모델 또한 자동으로 종료되는 구조였으나, 추후의 Time Series Classification 학습을 위하여 파일 내부의 동영상을 연속적으로 재생하여 도출해낸 변수를 모두 저장하도록 모델을 수정하였다. 영상 데이터셋의 모든 프레임에 대한 감지가 종료된 시점에서 parameters를 추출하여 14개의 csv 파일로 저장하였다.

3.3 RNN을 활용한 시계열 데이터의 분류

앞서 3.2장에서 도출해낸 데이터는 시계열성을 띠고 있는 데이터이다. 따라서 이의 특성을 보존하여 학습시킬 수 있는 RNN 모델을 사용하여 이를 학습시켰다. 또한 RNN 모델에 투입되는 변수의 종류와 activation function과 dense 층의 갯수 등을 조절하여 최적의 정답률을 이끌어 내었다. 전체적인 흐름은 아래 그림과 같다.



[그림 2] 모델의 작동 모습

3.3.1 데이터의 전처리

object detection을 통해 추출한 물체의 정보를 RNN model이 읽어올 수 있도록 전처리 과정을 진행했다. 수정한 YOLOv3 모델을 동작시키면 앞서 설명한 9개의 파라미터가 영상의 프레임별로 저장되어 csv 파일로 내보내진다. 이후 csv 파일을 직접 조작하여 파라미터 별로 묶어서 저장하고 실수형으로 자료형을 변환시켰다. 해당 파일은 xtrainset (parameter number) / xtestset 으로 명명했다. 이후 각 이상행동의 종류에 따라서 아래 라벨링된 값을 xtrainset의 행 개수만큼 가지는 ytrainset (parameter number) / ytestset을 생성하였다.

행동 종류	정상	파손	실신	폭행
Label	0	1	2	3

[표 3] 판별할 이상행동의 종류과 라벨링

최종적으로 만들어진 학습 데이터의 크기는 다음과 같다.

xtrain_set	(5199, 320, 14)	xtest_set	(1448, 320, 14)
ytrain_set	(5199,1)	ytest_set	(1448, 1)

[표 4] YOLOv3 input으로 사용한 학습 데이터 형식

3.3.2 모델 설정

초기의 모델 설정은 14개의 변수를 모두 포함한 상태로 설정한다.

- 1) Time step : 시계열 데이터의 특성상 데이터를 얼마만큼 잘라서 모델을 제작해야 하는지 고려해야 한다. Time step을 i로 설정하게 되면, t-i부터 t까지의 데이터를 가지고 다음 값을 예측하게 된다. 전처리 과정에서 divide = 80으로 설정하여 320개의 데이터를 묶었으므로, timestep = 320으로 지정하였다.
- 2) LSTM & Dense 레이어: LSTM과 Dense 레이어를 각각 한 개, 두 개씩 생성하였다. 레이어의 갯수는 조정 가능한 값으로 복잡한 학습일 경우 그 개수를 증가시키는데, train data의 양을 고려했을 때 2개가 적합하다고 판단되었다.

- 3) Activation function: 딥 러닝 네트워크에서는 값들을 노드에 전달할 때 비선형 함수를 통과시켜 전달한다. 주로 ‘sigmoid Function’, ‘tanh’, ‘relu’ 가 사용되며, 해당 모델에서는 tanh를 사용하였다. LSTM의 특성상 이전 step의 정보가 이번 학습에 사용되는데, relu함수와 같은 경우에는 출력이 발산할 수 있다. 따라서 기울기가 0과 1사이를 유지하는 tanh 함수를 사용하여 기울기의 역전파가 더욱 잘 이루어지도록 하였다.
- 4) Dropout : overfitting을 방지하기 위해 전체 노드중 일부의 연결을 끊는다. 해당 학습에서는 0.5로 두어 50%의 노드들이 랜덤하게 훈련되도록 하였다.
- 5) repeats: 가중치를 달리하여 학습을 n번 시킨다. n=10으로 설정하여 어떤 조건에서 평균적으로 학습이 잘 이루어지는지를 확인하였다.

3.3.3 최종 모델의 변수 설정

RNN의 input 변수 종류를 정하기 위하여 변수를 4개의 군으로 분리하였다. 각각 폭행, 유기, 실신에 특화된 군으로 나누어 각 군에 대하여 최적의 정답률을 보이는 변수의 조합을 선발해낸다. 이후에 각 군의 선발된 변수들을 하나의 모델의 input 값으로 설정하여 최종 모델을 만들어내도록 한다.

공통, 폭행, 유기, 실신의 4가지 군으로 변수들을 아래와 같이 분리하였다.

번호	추출해 낼 데이터	분리된 군
1	한 프레임당 사람의 수	공통
2	한 프레임당 물체의 수	공통
3	전 프레임 대비 사람의 수 변화	공통
4	전 프레임 대비 물체의 수 변화	공통
5	프레임 내에서 가장 거리가 가까운 사람들 사이의 거리	폭행
6	프레임 내에서 가장 거리가 먼 사람들 사이의 거리	폭행
7	프레임 내에서 가장 거리가 가까운 사람과 물체 사이의 거리	유기
8	프레임 내에서 가장 거리가 먼 사람과 물체 사이의 거리	유기
9	프레임 내에서 가장 작게 측정된 사람의 높이	폭행, 실신
10	프레임 내에서 가장 크게 측정된 사람의 너비	폭행, 실신
11	프레임 내에서 측정된 모든 물체의 평균 정확도	공통
12	프레임 내에서 가장 많이 중첩된 면적의 크기	폭행, 유기
13	프레임 내에서 가장 많이 중첩된 사람의 면적의 크기	폭행
14	프레임 내에서 가장 작은 사람의 면적	폭행, 실신

[표 5] 각 이상행동에 특화된 파라미터 분리

다음과 같이 각 행동에 특화된 변수들을 군으로 설정한 뒤, 하나의 군에서 여러 변수를 설정하고 조합하여 기존 모델의 입력값으로 설정하였다.

3.3.4 RNN 모델 학습값 변화

1) 기존 모델

위의 최종 변수들을 입력값으로 사용한 모델은 평균 73.066%의 정답률을 보인다.

2) Activation function을 ‘tanh’로 사용

일반적인 신경망과 RNN의 아키텍처를 비교할 시에 RNN에서는 은닉층에서 활성화 함수로 ReLU 대신 tanh를 사용하는 경향이 있다. 따라서 기존 모델의 활성화 함수를 tanh로 사용하였다.

IV. 연구 결과

아래는 3절에서 분류한 군을 바탕으로 조합한 변수들 중 해당 이상행동에 대하여 가장 높은 정답률을 보였던 상위 3개의 조합들의 정답률을 표로 나타낸 것이다.

폭행			
번호	파라미터 조합	폭행 데이터셋 정답률	이외 이상행동 오답률
폭행 - 1	1,2,3,4,5,6,9,10,11,13	75%	12%
폭행 - 2	1,2,3,4,5,6,9,10,11,12,13	64%	19%
폭행 - 3	1,2,3,4,5,6,9,10,11,12,13,14	55%	24%

[표 6] 폭행 - 파라미터 조합에 따른 정답률과 오답률

실신			
번호	파라미터 조합	실신 데이터셋 정답률	이외 이상행동 오답률
실신 - 1	1,2,3,4,10,11	79%	21%
실신 - 2	1,2,9,10,11,14	69%	26%
실신 - 3	1,2,3,4,9,10,11	78%	10%

[표 7] 실신 - 파라미터 조합에 따른 정답률과 오답률

파손			
번호	파라미터 조합	파손 데이터셋 정답률	이외 이상행동 오답률
파손 - 1	1,2,3,4,11	66%	13%
파손 - 2	1,2,3,4,7,11,12	72%	13%
파손 - 3	1,2,3,4,7,8,11,12	70%	18%

[표 8] 파손 - 파라미터 조합에 따른 정답률과 오답률

따라서 최종 RNN 모델에 입력되는 변수의 조합은 아래의 표와 같다.

번호	변수명 라벨링	추출해낼 데이터
1	per_num	한 프레임당 사람의 수
2	obj_num	한 프레임당 물체의 수
3	per_num_c	전 프레임 대비 사람의 수 변화
4	obj_num_c	전 프레임 대비 물체의 수 변화
5	per_n_d	프레임 내에서 가장 거리가 가까운 사람들 사이의 거리
6	per_f_d	프레임 내에서 가장 거리가 먼 사람들 사이의 거리
7	per_obj_n_d	프레임 내에서 가장 거리가 가까운 사람과 물체 사이의 거리
8	per_obj_f_d	프레임 내에서 가장 거리가 먼 사람과 물체 사이의 거리
9	per_s_h	프레임 내에서 가장 크게 측정된 사람의 높이 (detection box 기준)
10	per_b_w	프레임 내에서 가장 크게 측정된 사람의 너비 (detection box 기준)
11	average_acc	프레임 내에서 측정된 모든 물체의 평균 정확도
12	nesting_size	프레임 내에서 가장 많이 중첩된 면적의 크기
13	per_nesting_size	프레임 내에서 가장 많이 중첩된 면적의 크기 / 사람의 면적
14	per_s_size	프레임 내에서 가장 작은 사람의 면적

[표 9] 최종 모델의 파라미터 조합 및 라벨링

여러 변수의 조합으로 time series classification을 진행한 결과, 변수의 개수와 정답률은 비례하지 않는다는 사실을 관찰할 수 있다. 이는 임의로 설정된 변수가 RNN 모델에 개입함으로써 각 이상행동에 대한 정답률을 낮추며, 나아가 이를 다른 이상행동이라고 판단할 가능성 또한 높아졌기 때문이다. 이는 특히 12번과 14번 변수에서 뚜렷하게 관찰할 수 있다. 12번 변수는 본래 폭행과 유기에 특화된 변수라고 판단하였으나, 다른 영상 데이터 셋에서도 겹쳐 있는 물체가 등장하는 등 이상행동과 관계없이 파라미터 수치가 크게 나오는 경우가 지속적으로 발생하였다. 따라서 12번 변수가 포함된 경우, 정상행동 영상을 이상행동 영상이라고 판단하는 비율이 급증하였다. 14번 변수 또한 3차원 데이터를 2차원인 영상으로 가공하는 과정에서 사람의 면적이 실제와는 다르게 나타나는 경우가 많았으며 사람의 구분이 불가능한 YOLOv3 모델 특성상 이상행동에 직접적으로 관여하지 않는 면적이 저장되는 경우가 많았다. 이는 해당 이상행동에 대한 정답률은 낮추고, 다른 이상행동에 대한 오답률을 높였다.

RNN에 입력되는 최종 변수들을 설정한 이후 RNN의 기본 파라미터를 모델에 적합하게 수정하는 과정을 거쳐 최종 모델의 정답률은 73.066%에서 76.450%로 증가하게 되었다. 이는 은닉층에서 사용되는 activation function을 tanh 함수로 변경함으로써 단계별 nomalizing을 성공적으로 수행되었기 때문이다.

V. 결론

본 연구는 다중 이용시설에서의 이상행동 영상을 폭행, 파손, 실신, 정상 4가지로 분류하는 것을 목표로 진행하였다. YOLOv4를 사용하여 각 객체 detection box의 정보를 RNN 학습에 필요한 변수로 변환시켜 저장한 뒤 적절한 변수들의 조합을 RNN의 입력값으로 설정하였다. 이후 RNN 모델의 파라미터를 변경한 뒤 최종적으로 이상행동을 분류할 수 있는 모델을 만들 어내었다.

변수의 조합과 RNN의 학습 파라미터를 정답률과 대조하여 최종 모델을 도출해낸 결과, 정답률은 평균 76%임을 알 수 있다. 4가지 경우에 대해서 76%의 정답률을 보인 것으로 보아 모델이 연속적인 파라미터 추출을 통해 이상 행동간의 연관성을 파악하고, 예측을 수행했음을

확인할 수 있다. RNN 모델의 특성상 가중치에 따라서 모델의 성능이 달라질 수 있기에 적절한 가중치를 가지는 모델을 찾는 것이 우선으로 보이며, Test set의 비율을 늘리는 것으로 정답률을 현재보다 더 개선할 수 있을 것으로 보인다. 또한 현재는 이상 행동의 종류가 4가지로 제한되어 있지만, 학습 데이터를 폭넓게 가져온다면 더 많은 이상행동을 분석할 수 있을 것이다. 약 5초 정도의 짧은 영상에서 추출한 정보만으로 이상행동을 감지하는 본 연구의 특성상 실시간 감지에 적합하며, 공항 외에도 실시간 감지를 요하는 폭넓은 분야에 활용할 수 있을 것으로 보인다.

본 연구에서는 이상행동 동영상 데이터를 분석하기 위하여 기존에 사용되던 3D CNN이 아닌 YOLO와 RNN을 결합하여 사용함으로써 중간 과정을 확인하고 학습률을 높이기 위한 변수의 선택과 RNN kernel size 등을 조절하기 용이하게 하였다. 변수들을 각 이상행동에 따라 군으로 분리하고, 입력되는 변수들의 조합에 따라 달라지는 학습률을 관찰하며 기존에 사용되던 블랙박스 모델이 아닌 분리된 모델을 사용하는 방식의 이점을 활용하는 방향으로 연구를 진행함으로써 영상 분석 모델을 새로운 방식으로 접근할 수 있게 되었다. 이는 이상행동 분석뿐만 아닌 여러 파라미터를 입력값으로 사용하는 영상 분석 분야의 연구에 폭넓게 활용될 수 있을 것으로 기대된다.

또한 기존의 이상행동 감지 연구로 이루어졌던 여러 선행연구에서는 3D CNN 분석을 통하여 이상행동인지의 여부만 판단하였다면, 본 연구는 이상행동의 종류를 나누어 총 4가지의 상태로 분류할 수 있도록 하였다. 이를 통하여 실제 상황에 적용하기 비교적 적합하며, 다양한 이상행동이 이루어질 수 있는 다중이용시설에서 이상행동을 큰 틀로 분류한 상태로 정보를 제공함으로써 빠른 조치가 이루어질 수 있도록 한다.

학습에 사용된 데이터셋은 공항에서의 이상행동 상황을 가정하여 촬영한 것으로, 실제로 공항을 비롯한 다중 이용시설에서의 영상에 모델을 적용해보지 못하였다. 영상 내에 이상행동에 관여하는 사람이 2-3명, 이상행동에 관여하지 않는 사람들이 3-4명으로 제한되어 있는점을 고려할 때, 실제 상황에서의 우발적이고 혼잡한 상황에서 정답률이 떨어질 가능성이 있다는 점이 본 연구의 제한점이다. 추후 공항을 포함한 다중 이용시설에서의 실제 데이터셋을 확보한다면 이를 연구에 반영하여 더욱 발전시킬 수 있을 것으로 보인다. 또한 이상행동에 사용되는 물체와 이상행동의 분류에 대한 데이터 셋을 업데이트하면 감지할 수 있는 이상행동의 폭이 넓어지고 정확도가 상승할 것이라 예상된다.

References

김선덕. "YOLO 알고리즘을 이용한 선박 기관실에서의 화재 검출에 관한 연구." 国内석사학위논문 木浦海
洋大學校, 2020. 전라남도

박상진, 조국, 임준혁, 김민찬, Park Sang-Jin, Cho Kuk, Im Junhyuck, and Kim Minchan. "CCTV 영
상을 활용한 동적 객체의 위치 추적 및 시각화 방안." 지적과 국토정보 51.1 (2021): 53-65.

우윤희, 최권택, 이정근.(2019).실시간 응용을 위한 혼합형 다중 객체 추적 시스템의 구현.한국정보기술학
회논문지,17(11),1-8.

한국컴퓨터정보학회, Jan. 2020, pp. 55-61, doi:10.9708/JKSCI.2020.25.01.055.

한태동. (2021). LSTM을 이용한 주가 예측: 기술 지표, 거시 경제 지표, 시장 심리의 조합을 중심으로. 융
복합지식학회논문지, 9(4), 192 page

Kim, Jee-Hyun, and Young-Im Cho. "A New Residual Attention Network Based on Attention
Models for Human Action Recognition in Video." Journal of the Korea Society of
Computer and Information, vol. 25, no.

Abstract

Currently, abnormal behavior detection of public facilities is carried out by humans, which causes reduction in accuracy and waste of time and money. In this study, we extracted multiple object parameters from 2D image data to detect abnormal behavior. The purpose of this research is to determine abnormal behavior detection (attack, fainting, breakage) at public facilities with low time and high accuracy.

The characteristics of objects were identified by using multi-object parameter extraction of images, and a model that effectively classifies abnormal behavior was derived by using RNN, which has a time series variable. Time series data obtained through object tracking of YOLOv3 were analyzed by RNN, and a model with a final accuracy of 76% was derived through parameter adjustment.

Although this study is based on abnormal behavior data at airports, it is highly utilized so it can be used in public places, densely populated areas, and security systems. Also, it is significant as a new approach since it uses a model that synthesizes YOLOv3 and RNN rather than using the commonly used 3D CNN.

Keywords : abnormal behavior, public facilities, Time series classification, parameter adjustment, multiple object parameters, YOLOv3, RNN

감사의 글

먼저, 본 연구를 1년동안 성심성의껏 지도해주신 정보과학과 지도교사 문광식 선생님께 깊은 존경과 감사를 드립니다. 해당 연구에 대한 사전지식과 경험이 전무한 상태로 연구를 시작한 저희를 항상 올바른 방향으로 이끌어 주시고, 연구가 꾸준히 진척될 수 있도록 지도해주셔서 논문을 성공적으로 마무리할 수 있게 되었습니다.

또한, 개발 환경의 초기 설정에 도움을 주신 서지원 교수님과 이후의 모델 설계 방향성을 제시해주신 김민호 교수님께도 감사의 말씀을 드립니다. 마지막으로 심사를 맡아주시며 논문의 완성도를 높일 수 있도록 피드백을 제공해주신 최윤호 교수님, 김민호 교수님께도 감사드리며 논문을 완성할 수 있도록 다양한 방면에서 지도해주신 모든 선생님들께도 감사의 말씀을 올립니다.

이소민

아직 연구활동에 미숙한 저희를 잘 지도해주신 문광식 지도교사 선생님께 감사의 말씀을 드립니다. 처음 해보는 정보과학 연구활동이 낯설고, 어려움의 연속이였지만 적극적으로 도움을 주시고 항상 응원해주신 덕분에 연구활동을 끝마칠 수 있었습니다. 선생님께서 해주신 조언 역시 연구활동의 초심 및 본질을 잊지 않게 하는것에 많은 도움이 되었습니다. 전문적인 조언으로 초기 연구의 방향성을 잡아주신 서지원 교수님, 학교에 직접 방문하여 연구 세팅에 많은 도움을 주신 김민호 교수님께도 감사의 말씀을 전합니다. 그리고 항상 옆에서 많은 조언과 도움을 준 전준서 친구에게도 고마움을 표합니다. 1년간 함께 연구활동을 진행하며 여러 어려움을 헤쳐나가기 위해 노력한 이소민 친구에게도 감사의 말을 전합니다. 본 연구에서 얻은 귀중한 경험을 토대로 훌륭하게 성장하여 저도 타인에게 힘이 될 수 있는 존재가 될 수 있도록 하겠습니다. 감사합니다.

허지성

논문 작성의 기여도

▶ 이소민

연번	위 치	기여내용
1	서론 1p	적절한 자료 조사와 선행연구 학습을 통해 본 연구가 지향하는 목표를 파악하고, 이를 서론에 적절히 녹여냄.
2	연구 방법 및 절차 4p-5p	YOLOv3 모델의 작동 원리를 공부하고 여러 파이썬 라이브러리와 모듈을 활용하여 설정한 파라미터를 도출할 수 있도록 모델을 적절히 수정하였음. python에서 출력된 raw data가 전처리 과정을 거쳐 학습에 이용될 수 있는 .csv 형식으로 변환시키는 것에 기여함.
3	연구 방법 및 절차 7p-8p	각 이상 행동별 파라미터를 분석 및 연구해 설정하고 각 파라미터별 정답률을 수치적으로 시각화하여 모델의 성능을 향상시키는 것에 기여함. 또한 정답률의 변화와 그 원인이 되는 파라미터를 분석함으로써 각 이상행동의 탐지와 파라미터 사이의 상관관계를 도출해냄.
4	연구 결과 및 결론 9p-11p	연구를 통해 얻어낸 결과를 적절히 정리하여 연구 결과를 알림. 향후 연구의 방향성을 제시하고 본 연구와 다른 연구와의 차별성, 가지는 의의를 적절하게 언급함.

▶ 허지성

연번	위 치	기여내용
1	서론 및 이론적 배경 1-2p	뉴스 기사를 통해 공항에서의 폐쇄회로 감시 체계가 가지는 문제점을 파악하고, 선행 연구와 사전 학습을 통해 연구의 방향성을 잡았으며 이에 따라 서론 일부와 선행 연구 부분을 작성함.
2	연구 방법 및 절차 2-4p	본 연구에 적합한 딥러닝 모델을 탐색하고 최종적으로 YOLOv3를 선정하여 학습 환경을 구축함. YOLOv3 모델의 작동원리를 공부하고 앞서 설정한 파라미터를 도출할 수 있게 모델을 적절히 수정하여 학습에 필요한 데이터 셋을 구축하는데 기여함.
3	연구 방법 및 절차 6p-9p	선행 연구로 학습한 내용에 기반하여 파라미터를 설정하고, 시계열 데이터의 성질이 들어날 수 있도록 적절히 전처리 하였음. 또한 colab을 활용해 LSTM 기반의 딥러닝 환경을 구축하였고 이후 파라미터와 각 학습 필터별 특징점을 연구하였음. 이후 해당 값과 조건을 적절히 조절하며 보다 뛰어난 정확도를 가지는 딥러닝 모델을 제작하였음.

2022 . 7 . 8

지도교사 확인 : 문광식 (사인)