선진국방연구
Journal of Advances in Military Studies https://doi.org/10.37944/jams.v3i3.80
2020, Vol. 3, No. 3, 113-132.

Application of OpenPose and deep learning for intelligent surveillance reconnaissance system*

Kyujung Choi** · Suyeong Oh*** · Chaebong Sohn****

≪Abstract≫

In this study, defense surveillance reconnaissance systems were implemented through deep learning networks such as OpenPose and deep neural networks (DNN), convolutional neural networks (CNN), and long short-term memory (LSTM). This study proposes a target recognition method which differs from the existing surveillance reconnaissance systems. This method consists in distinguishing between ordinary people and targets by classifying motions in the images being filmed. Thus, the skeleton data of the target in the image are extracted using OpenPose. Then, keypoints included in the extracted skeleton data are entered into DNN,

* This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2016-0-00288) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

** (First Author) Kwangwoon University, Department of Electronics and Communications Engineering, Master student, kakm12@naver.com

*** (Co-Author) Kwangwoon University, Department of Electronics and Communications Engineering, Master student

**** (Corresponding Author) Kwangwoon University, Department of Electronics and Communications Engineering, Professor, cbsohn@kw.ac.kr

CNN, and LSTM to classify the motion. The classified motions are selected as motions learned in the military, such as overall security. When the system classifies motions and recognizes targets, it identifies them on the map and tracks them. The tracking algorithm calculates the movement direction of the target by calculating the change in the values of keypoints extracted through OpenPose by frames. Finally, it uses the depth information obtained from the camera to display targets on the map based on the camera location. All these computations are based on the use of the skeleton data rather than the entire image, thus reducing the overall computation.

Keywords : OpenPose, keypoints, deep neural networks, convolutional neural networks, long short-term memory

Ⅰ. 서론

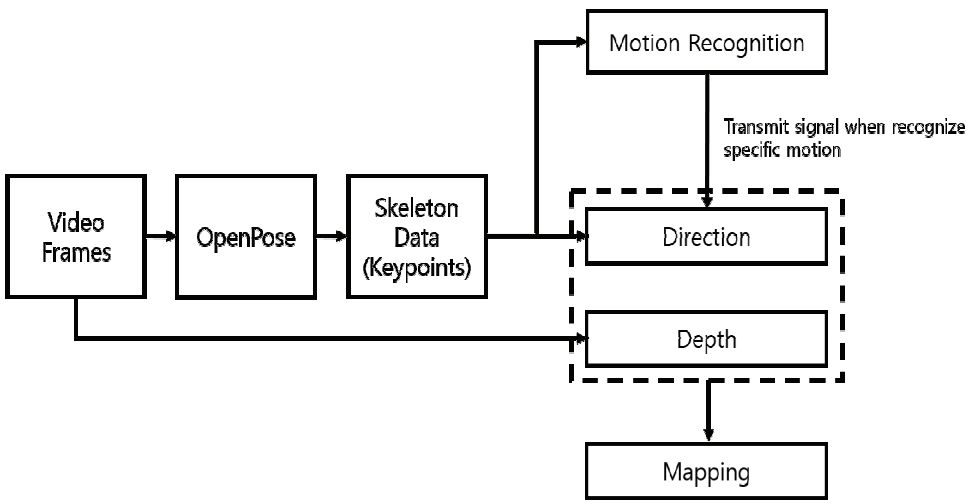
기존 군 감시 정찰 시스템은 센서와 같은 장비를 활용하거나 사람이 직접적으로 관찰하는 방식으로 구축되어 있다. 그러나 이와 같은 시스템은 추가적인 비용이 들어가거나 인력이 투입되기 때문에 비효율적이고 불안정하다는 문제점이 있다. 실제 2020년에도 총 11건의 군 부대 침입 사건이 발생하여 지속되는 문제점을 보완하기 위해 인공지능 네트워크 도입의 필요성이 제기된다. 이에 본 연구의 목적은 기존의 감시 정찰 시스템에 모션 분류 네트워크를 도입을 통해 효율화 방안을 제안하는 데 있다.

Figure 1은 본 연구의 제안 방식과 기존 시스템을 비교한 그림으로 OpenPose와 후방에 DNN, CNN, LSTM 등을 추가하며, 시스템에 target 추적 알고리즘을 추가로 구현하여 시스템의 다기능성을 확보하고자 한다. 특히, 추적 알고리즘은 시스템의 연산량을 줄이고자 target의 이동을 프레임별 target 이미지의 변화가 아닌 OpenPose에서 나오는 결과를 활용하여 계산하는 접근을 적용한다.

<Figure 1> Comparison of the current system (left side) with our suggested system (right side) in monitoring system

한편, 인공지능 네트워크를 통해 사람을 분류하는 기본적인 방법은 사람 전체의 이미지를 통해 분류하는 것이다. 하지만, 본 연구에서는 일반인과 target을 단순 이미지가 아닌 사람의 모션을 통해 분류하고자 하였다. 이미지 자체를 학습하는 object detection을 사용하게 되면 본 연구에서의 경우 target과 일반인을 나누는 기준이 불분명하게 되어 일반인을 target으로 인식하거나 다수의 사람을 인식할 경우 반응속도가 느려지는 등의 문제가 발생하여 그 한계가 분명하였다. 이를 해결하고자 현재 광범위한 분야에서 응용되고 있는 인공지능 네트워크 중 모션을 다루는 네트워크를 활용하여 target을 인식하고자 하였다. 모션을 다루는 네트워크는 OpenPose, DensePose, MaskR-CNN, Alpha-Pose 등 다수가 있었다. 본 연구는 카메라 자체의 depth 촬영기능을 활용하여 반드시 사람 keypoints의 3D 좌표를 추정할 필요가 없었다. 따라서 다수의 네트워크 중 2D 좌표를 추정하는 OpenPose를 사용하였으며 모션을 분류하는 네트워크는 DNN, CNN, LSTM 계열의

딥러닝 네트워크를 통해 비교하였다. 이를 통해 평균 87% 정도의 정확도로 모션을 분류하고 target을 인식 및 추적하는 시스템을 구축하였다. Figure 2는 본 연구에서 제안하는 시스템의 세부 구조이다.



<Figure 2> System flowchart

Ⅱ. 관련 문헌 검토

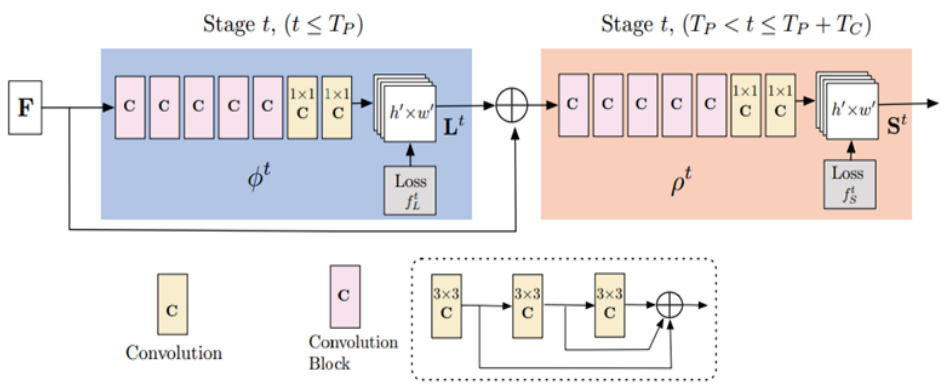
모션을 분류하는 시스템은 크게 하향식 시스템과 상향식 시스템으로 나뉜다. 하향식 시스템은 사람을 먼저 찾고 그 사람의 모션을 분류하는 방식을 말한다. 이는 모션 분류단계에서 사람 한명 한명에 대한 모션을 분류한다는 것을 의미한다. 이 경우에 모션을 분류해야 할 사람의 수가 적을 때는 문제가 없지만, 다수의 사람이 있는 경우 모든 사람의 모션을 분류해야 하므로 연산량이 많아지고 시스템이 느려지는 단점이 있다.

반면, OpenPose와 같은 상향식 시스템은 사람을 먼저 인식하는 것이 아니라 사람의 얼굴이나 관절, 손과 같은 특정 부위(keypoints)를 먼저 인식하고 이를 연결하여 사람의 skeleton 데이터를 출력한다. 본 연구에서는 이처럼 출력된 skeleton 데이터를 통해 사람의 모션을 분류하게 된다. 결과적으로 시스템은 사람 전체 이미지가 아닌 사람의 keypoints 데이터 값만을 사용하기 때문에 다수 사람들의 모션을 분류하는 데 있어 연산량이 감소하여 하향식 시스템보다 효율적이다. 이와 같은 모션 분류의 네트워크가 발전하면서 2D 좌표만을 추정하던 OpenPose에서, RGB 이미지 상의 인간의 모든 픽셀을 3D 표면에 맵핑하는 DensePose(Alp Güler, Neverova, & Kokkinos, 2018), 더 나아가 3D 모델과의 매칭을 통해 인간의 자세를 더 유연하게 나타내는 Total Capture 등 다양한 연구

가 진행되고 있으며, 본 연구는 촬영카메라의 자체적인 depth 측정기능을 활용하고자 OpenPose를 활용하였다.

Ⅲ. 연구방법

3.1 OpenPose



<Figure 3> OpenPose Networks architecture and process of extracting skeleton data

(source: https://www.geeksforgeeks.org/openpose-human-pose-estimation-method/)

OpenPose(Cao et al., 2019)가 사람의 skeleton 데이터를 추출하는 과정은 다음과 같다. 먼저, 입력 데이터를 30fps로 생성하여 멀티 stage의 CNN을 통해 fine-tune 하여 임의의 keypoints 데이터 F를 생성한다, 여기서 멀티 stage CNN은 VGG-19의 앞 단 10개 layer를 의미한다. 다음으로 데이터 F를 PAF(Part Affinity Fields), 그림3의 파란색 stage에 입력하여 임의의 keypoints 간의 관계성을 계산하고 가장 큰 값을 가지는 관계의 keypoints를 연결하도록 학습을 반복하여 사람 한 명에 대한 불완전한 skeleton 데이터를 생성한다. 그런 후, 해당 결과와 앞선 데이터 F를 concatenated 하여 Confidence Map(keypoints)을 학습하는 Figure 3의 베이지색 stage에 입력한다. 해당 stage에서 정확한 keypoints를 찾도록 학습을 반복한 결과 Figure 4와 같은 완전한 skeleton 데이터를 생성하게 된다.

<Figure 4> Output of OpenPose by motions

Figure 4와 같이 OpenPose를 통해 출력되는 skeleton 데이터를 각각 DNN, CNN, LSTM 계열의 딥러닝 네트워크에 입력하여 모션을 분류하고 그 정확도를 비교한다.

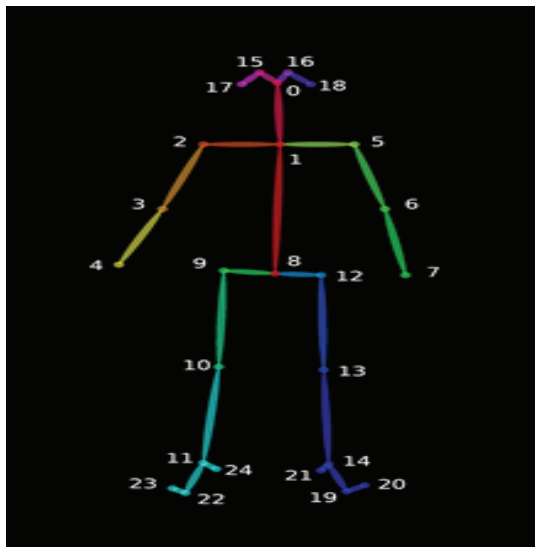
3.2 모션 분류

3.2.1 DNN (Deep Neural Networks)

OpenPose에서 생성되는 skeleton 데이터는 사람의 keypoints를 연결하며 생성된다. OpenPose는 이와 같은 keypoints를 0부터 24까지의 고유번호를 붙여주며 각 고유번호에 위치정보를 부여한다 (Figure 5). keypoints 위치정보는 OpenPose의 경우 x, y 2D 값을 가지게 된다. 본 연구에서는 25개의 keypoints에 대해 총 50개의 좌표값을 추출한다. Table 1은 울타리 모션 중 59번째 프레임에서 출력되는 keypoints의 좌표값을 보여준다. DNN는 각 모션의 프레임마다 50개의 데이터 값을 그대로 입력하여 모션을 분류한다.

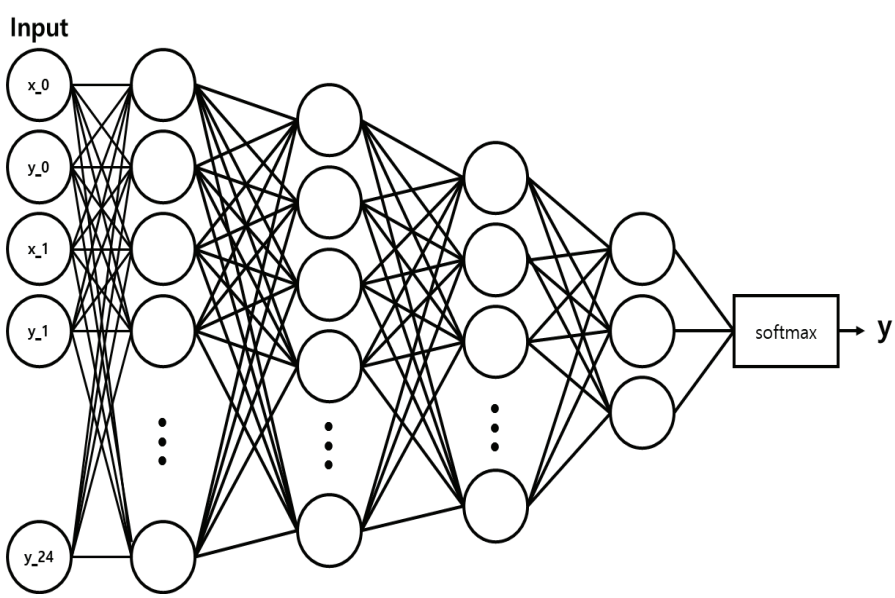
<Table 1> Keypoints values of 59 frame of fence motion

# keypoints	keypoints_x	keypoints_y
0	234.148	119.324
1	234.086	138.869
2	215.855	141.413
	. . .	
23	0	0
24	0	0



<Figure 5> Keypoints number extracted through OpenPose

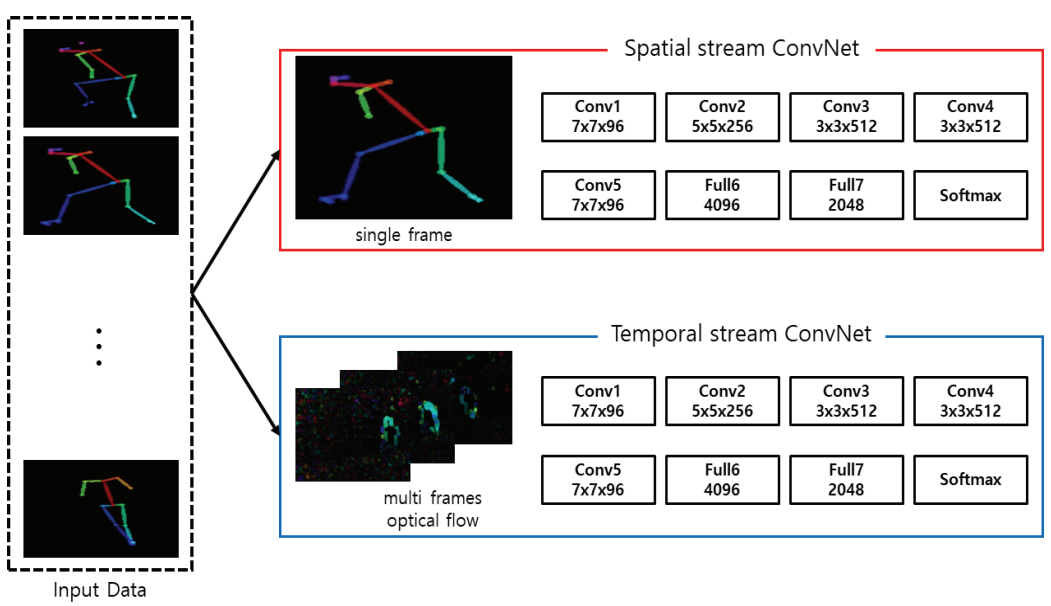
구현된 DNN은 3개의 hidden layer로 구현되어있다. 입력되는 50개의 데이터 값은 활성화 함수 ReLu를 통해 3개의 layer와 output layer를 거쳐 마지막 softmax를 통해 입력되는 데이터가 어떤 모션인지를 분류하게 된다. 본 연구에서 구현된 DNN 구조는 Figure 6과 같다.



<Figure 6> DNN(Deep Neural Networks) architecture of motion classification

3.2.2 CNN (Convolutional Neural Networks)

사람의 모션과 같은 연속적인 시계열 데이터를 CNN을 통해 분류하기 위해 본 연구에서는 두 개의 stream을 사용하였다(Simonyan & Zisserman, 2014). 각 stream마다 공간적인 요소와 시간적인 요소를 포함하고 있다. 공간적인 요소를 포함하는 stream에서는 영상 내 object에 대한 이미지 정보를 포함하기 때문에 단일 프레임을 입력하게 되고, 시간적인 요소가 포함된 stream에서는 object에 대한 움직임에 대한 정보를 포함하기 때문에 optical flow를 통한 데이터가 입력되게 된다. Figure 7은 이를 도식화한 그림이다. 각 stream을 통해 출력되는 데이터를 L2로 정규화된 softmax의 값을 feature로 사용하여 SVM을 학습시켜 모션을 분류한다(Karpathy et al., 2014). CNN은 DNN과 달리 Figure 5와 같이 skeleton 데이터를 입력하여 모션을 분류하였다. 본 연구의 OpenPose를 통해 연속적으로 출력되는 skeleton의 첫 번째 프레임을 공간 stream에 입력하여 모션에 대한 정보를 optical flow를 통해 출력되는 데이터를 시간 stream에 입력하여 연속적인 모션에 대한 정보를 추출하고 SVM을 통해 모션을 분류한다.



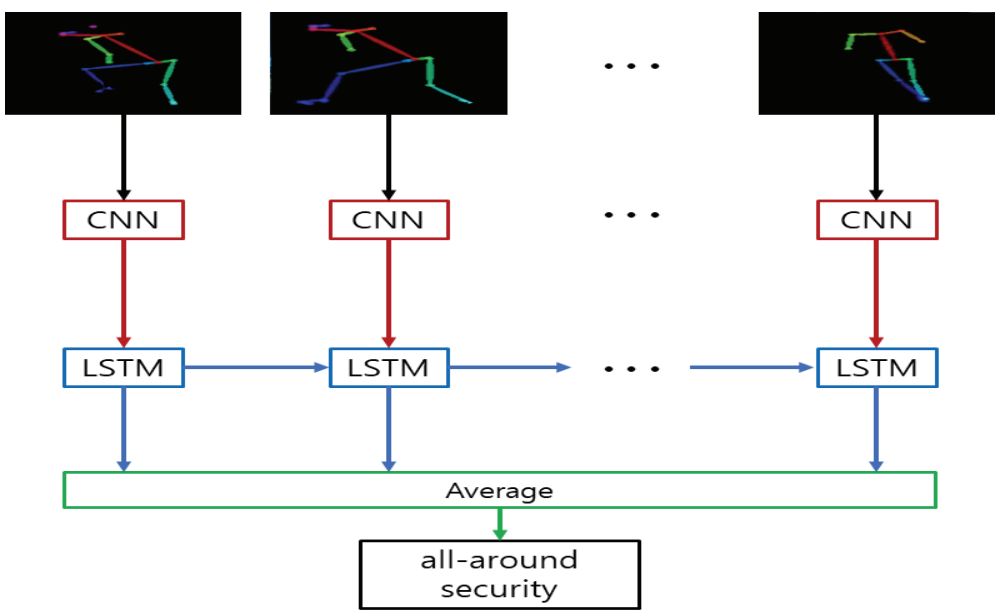
<Figure 7> The architecture of two stream CNN

3.2.3 LSTM (Long Short-Term Memory)

본 연구는 차원이 큰 데이터 처리를 위해 LSTM에 CNN을 결합한 모델 LRCN과 ConvLSTM을 통해 모션을 분류하며, LSTM에서 역시 CNN과 같은 데이터를 입력하여 연속적인 모션을 식별하다.

3.2.3.1 LRCN (Long Recurrent Convolution Networks)

LRCN에서는 연속적인 프레임을 계층적인 CNN을 통해 각 프레임의 feature를 추출하고 이를 LSTM의 입력데이터로 사용하게 된다(Donahue et al., 2015). 다음의 그림 8은 본 연구에서 구현된 LRCN의 구조를 보여준다. 먼저, CNN을 통해 일관성이 없을 수 있는 연속적인 skeleton 데이터에서 feature 값이 생성되고 어느 정도의 연관성이 생긴다. 또한, 이미지와 같이 차원이 큰 데이터가 입력되더라도 feature 맵만을 기억하면 되기 때문에 메모리 문제를 해결할 수 있다. 이와 같은 계층적인 CNN의 feature 값을 LSTM에 순차적으로 입력하고 그 결과값을 평균화하여 모션을 분류한다.



<Figure 8> The architecture of LRCN (Long Recurrent Convolution Networks)

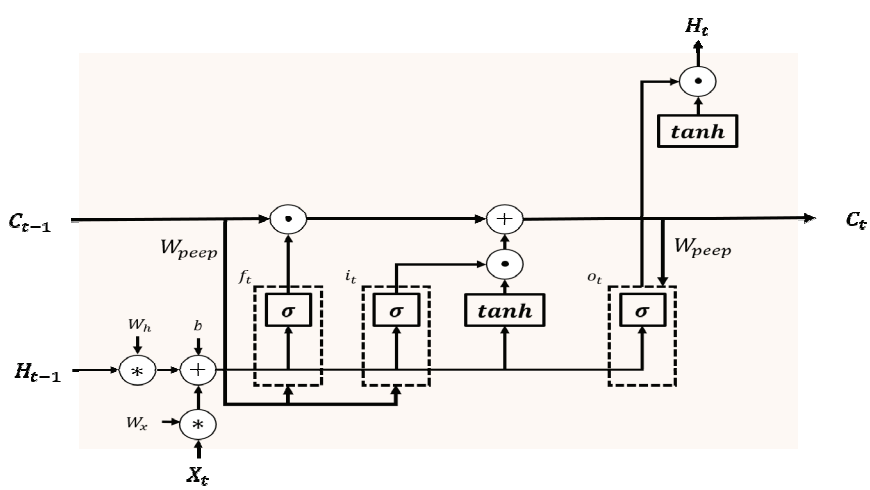
3.2.3.2 ConvLSTM

ConvLSTM(Simonyan & Zisserman, 2014)의 구현은 LRCN과 다른 방식으로 convolution의 내용을 추가하였다. 바로 LSTM 내부 식 자체에 convolution 연산을 대입하는 것이다. 단, 본 연구는 더 많은 시계열 데이터의 맥락(context)을 인식할 수 있도록 FC-LSTM(Fully-Connected Long Short-Term Memory) 모델을 기반으로 연구를 진행했으며 식(1)은 기반이 되는 FC-LSTM 내부 식을 보여주고 있다. 식에서 cell의 입력, 출력, state가 모두 1차원 벡터를 의미한다.

$$
\begin{aligned}
f_t &= \sigma(W_{x_f} x_t + W_{h_f} h_{t-1} + W_{c_f} \bigcirc c_{t-1} + b_f) \\
i_t &= \sigma(W_{x_i} x_t + W_{h_i} h_{t-1} + W_{c_i} \bigcirc c_{t-1} + b_{h_i}) \\
o_t &= \sigma(W_{x_o} x_t + W_{h_o} h_{t-1} + W_{c_o} \bigcirc c_t + b_o) \\
c_t &= f_t \bigcirc c_{t-1} + i_t \bigcirc \tanh(W_{x_c} x_t + W_{h_c} h_{t-1} + b_c) \\
h_t &= o_t \bigcirc \tanh(c_t)
\end{aligned} \tag{1}
$$

다음의 식(2)는 convolution 연산을 적용한 ConvLSTM의 내부 식을 보여준다. 기존의 식과 비교하게 되면 두 가지의 차이를 확인할 수 있다. 먼저, 식(2)에서 입력, 출력, state가 모두 3차원의 텐서의 형태이므로 기존의 곱하기 연산이 행렬 곱의 형태를 가진다는 점과 행렬 곱○ 대신 convolution 연산 ⊙이 대입되어 있다는 점이다. 이는 convolution의 연산을 통한 3차원 데이터의 채널 정보를 보존할 수 있다는 장점과 weight의 공유하여 그 수가 상대적으로 감소하는 장점을 의미한다. 이를 근거로 하여 ConvLSTM을 구현하였으며 Figure 9는 그 구조를 보여주고 있다. 결과적으로, LRCN과는 달리 연속되는 skeleton 데이터를 직접 입력받고 convolution 연산을 통해 내부적으로 처리하여 모션을 분류한다.

$$
\begin{aligned}
f_t &= \sigma(W_{x_f} * X_t + W_{h_f} * H_{t-1} + W_{c_f} \odot C_{t-1} + b_f) \\
i_t &= \sigma(W_{x_i} * X_t + W_{h_i} * H_{t-1} + W_{c_i} \odot C_{t-1} + b_{h_i}) \\
o_t &= \sigma(W_{x_o} * X_t + W_{h_o} * H_{t-1} + W_{c_o} \odot C_t + b_o) \\
C_t &= f_t \odot C_{t-1} + i_t \odot \tanh(W_{x_c} * X_t + W_{h_c} * H_{t-1} + b_c) \\
H_t &= o_t \odot \tanh(C_t)
\end{aligned} \tag{2}
$$



<Figure 9> The architecture of ConvLSTM

3.3 추적 알고리즘

본 연구에서는 target의 방향을 계산하기 위해 keypoints의 x좌표만을 활용하였다. y좌표의 경우 고도를 의미하여 target이 비행 혹은 어딘가를 올라가는 모션을 하지 않는 이상 사용되지 않기 때문에 활용하지 않았다. 또한, 카메라 각도 상 출력되지 못하는 keypoints에서는 0값이 출력이 되는 경우가 있어 특정 keypoints의 x 좌표값을 평균화하였다. 이렇게 평균화 값과 depth 값을 프레임별로 변화를 계산해서 target이 영상에서 벗어나도 어디로 이동하였는지 이동방향에 대한 정보를 계산한다.

$$X_f(n) = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$Target\text{의 이동 방향} = (X_{f+1}(n) - X_f(n), Depth_{f+1} - Depth_f) \tag{3}$$

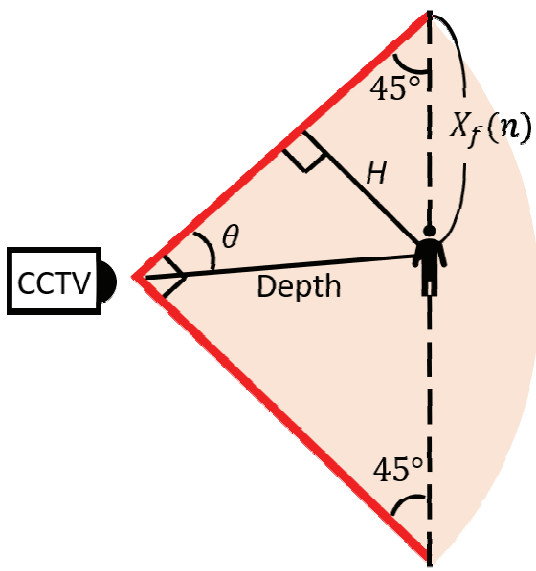
$$f = frame$$
$$n = keypoints$$

3.4 지도 표시

Target을 지도에 표시하기 위해 카메라의 depth 촬영기능과 카메라 촬영 각도가 90도라는 특성을 활용하여 극 좌표계를 만들어 지도에 표시하였다. 본 연구에서는 depth 촬영기능을 통해서 카메라와 target과의 depth를 추출할 수 있었다. depth값은 촬영되는 영상에서 픽셀마다 추출되었으며, 640×480의 해상도의 영상을 사용하였으므로 하나의 프레임마다 640x480개의 depth 값이 존재한다. 이 중 target의 depth 값을 추출하기 위해 keypoints의 좌표값을 target에 해당하는 픽셀을 좌표 값으로 사용하였다. 이때. keypoints 좌표들이 소수점 값이기 때문에 반올림한 좌표 주변 depth 값 8개와 자기 자신 depth 값까지 총 9개 depth 값을 평균화하여 depth를 계산하였다.

최종적으로 target의 위치를 정확하게 도출하기 위해 depth와 x좌표를 통해 새로운 좌표계를 도입하였다. 일반 영상 데이터에서는 그 값이 데카르트 좌표계 및 depth 정보로 이루어져 있지만, 이 값을 depth와 keypoints 좌표를 기반으로 극 좌표계로 변형하였다. 결과적으로 depth와 keypoints의 변화를 통해 target이 울타리를 넘는 것인지와 같은 구체적인 이동방향에 대한 추적이 가능해진다. 그 과정은 다음과 같다. 먼저, target을 지나도록 선을 그어 이등변 삼각형을 만든다. 여기서 다시 촬영 각도 축에 90도가 되도록 선을 연결하면 Figure 10과 같은 그림이 그려진다. 여기서 (3)에서 계산된 x좌표 값 계산과 (4)의 삼각함수계산을 통해 Θ를 계산하였다.

$$\sin(45^\circ) = \frac{H}{X_f(n)} \rightarrow H = X_f(n)\sin(45^\circ)$$
$$\sin(\theta) = \frac{H}{depth} \rightarrow \theta = \sin^{-1}(\frac{H}{depth}) \quad (4)$$



<Figure 10> Process of computing Θ focused on CCTV

Ⅳ. 실험

4.1 데이터 셋

본 연구는 target을 지도에 표시하기 위해 물체와의 depth까지 측정 가능한 IntelRealSenseD435 카메라를 사용하였다. 촬영 영상은 메모리 문제로 인하여 640x480의 해상도로 가지며 30fps로 dataset을 제작하였다. 선정한 모션은 무기소지, 사주경계, 울타리 모션으로 선정하였다. 각 모션에 대한 영상은 8개씩 각기 다른 각도로 촬영되어 모션 하나당 대략 2000프레임의 데이터가 추출되었다. 그중 6개의 영상을 학습 데이터로 나머지 2개를 테스트 데이터로 사용하였다.

4.2 결과

4.2.1 OpenPose

앞서 선정한 모션으로 제작한 dataset을 OpenPose에 입력하여 skeleton 데이터와 keypoints의 위치 값을 생성하였다. Figure 11은 OpenPose를 통해 출력된 keypoints와 이를 연결한 skeleton 데이터이며, 출력된 keypoints 중 고유번호 1번의 x, y 2개 좌표값이 기록된다(Table 2).

<Table 2> Keypoints values (number 1) of security motion by frame (coordinate value)

pose_x1	pose_y1	class	frame
589.098	201.483	security	1
581.342	198.863	security	2
576.069	194.993	security	3
578.65	192.409	security	4
577.395	192.332	security	5
563.039	189.736	security	6
	.		
	.		
	.		
629.513	192.366	security	180

<Figure 11> Skeleton data for security motion through OpenPose

4.2.2 모션 분류 및 target 인식

본 연구에서 모션 분류를 위해 구현한 정확도 계산은 다음의 기본 정확도식으로 계산되었다.

$$Accuracy(\text{정확도}) = \frac{TP + TN}{TP + FN + FP + TN}$$

$True Positive(TP)$: 실제 $True$인 정답을 $True$라고 예측(정답)
$False Positive(FP)$: 실제 $False$인 정답을 $True$라고 예측(오답)
$False negative(FN)$: 실제 $True$인 정답을 $False$라고 예측(오답)
$True negative(TN)$: 실제 $False$인 정답을 $False$라고 예측(정답)

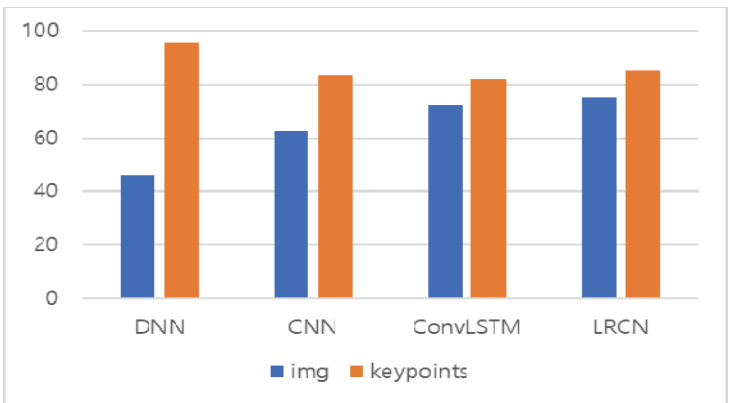
Table 3은 선정한 모션 분류에 대한 정확도를 기록하였다. 본 연구에서 제안하는 OpenPose에서 출력된 keypoints를 통해 모션을 분류하는 네트워크와 기존에 이미지를 활용하여 모션을 분류하는 네트워크와의 정확도를 비교하였으며 정확도 계산식은 동일한 식을 사용하였다. DNN, CNN, ConvLSTM, LRCN을 사용하여 모션을 분류하고 정확도를 비교하였다. 전체적으로 이미지로 학습한 경우가 keypoints로 학습한 경우보다 정확도가 낮은 것을 확인할 수 있다. 이는 다각도에서 촬영된 영향으로 2D keypoints의 경우 각도에 상관없이 일정하므로 영향을 덜 받았다고 추측할 수 있다. 네트워크별 모션 분류 정확도는 다음과 같다(Figure 12-14).

<Table 3> Accuracy of motion classification with image and keypoints (%)

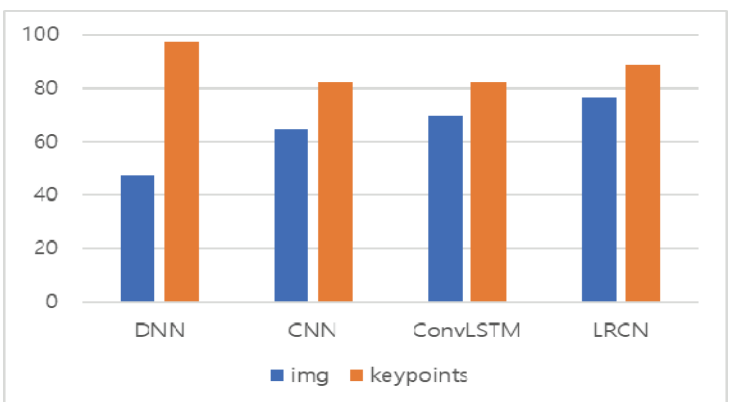
Network	img	keypoints
DNN	47.4	96.5
CNN	63.5	82.3
ConvLSTM	71.1	82.5
LRCN	75.2	87.1

4.2.3 추적 및 지도표시

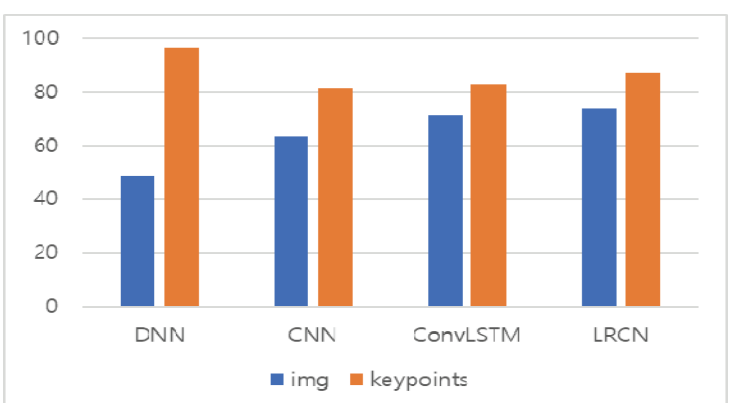
Table 4는 640x480의 데이터 그림 15의 전체 이미지에서 target의 머리에 해당하는 일부 이미지의 depth 값을 기록하였다. depth 값의 단위는 m이며 평균화해서 depth 값을 활용하였다. 예컨대 keypoints의 좌표값이 (375.7, 179.1)이라면, 해당 keypoints의 depth 값은 (375~377,178~180)의 평균값 6.46m로 계산할 수 있다.



<Figure 12> Accuracy of weapons motion by network



<Figure 13> Accuracy of security motion by network



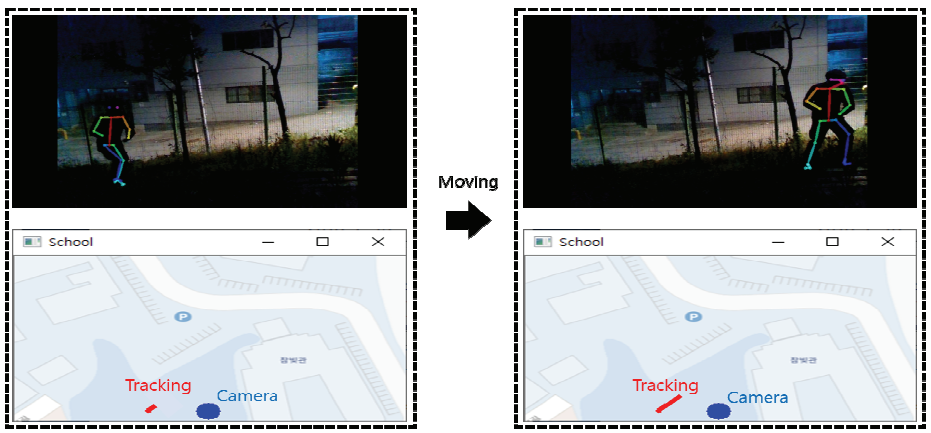
<Figure 14> Accuracy of fence motion by network

<Figure 15> Depth image of fence motion through IntelRealSenseD435

<Table 4> Depth value of target's head (m)

Keypoints related to target's head (width: 9, length: 6)									
	371	372	373	374	375	376	377	378	379
177	0	6.992	6.852	6.718	6.464	6.229	6.011	5.807	5.616
178	7.213	7.137	6.921	6.784	6.588	6.344	6.118	5.907	5.71
179	7.289	7.213	7.064	6.921	6.652	6.464	6.229	6.011	5.807
180	7.368	7.213	7.137	6.992	6.784	6.588	6.344	6.118	5.958
181	7.448	7.289	7.289	7.064	6.852	6.588	6.404	6.173	5.958
182	7.613	7.368	7.368	7.137	6.992	6.718	6.526	6.286	6.118

최종적으로 이와 같이 추출한 depth 값과 계산한 Θ를 통해 Figure 16과 같은 결과를 얻을 수 있었다. Target이 Figure 16의 왼쪽 그림과 같이 이동했을 때 Table 3과 같이 depth와 Θ가 계산되고 카메라 위치를 중심으로 하여 target 이동을 추적하여 경로를 나타낸다(right aspect in Figure 16). 빨간색 선은 target의 이동경로를 파란색 점은 카메라를 의미한다.



<Figure 16> Tracking result after system recognizes target

<Table 5> Depth and Θ values of movement of target

	depth(m)	Θ(°)
Frame 1	7.27	4.34
Frame 2	6.81	18.23
Frame 3	6.36	24.58

V. 결론 및 논의

본 연구에서는 사람의 모션을 분류하는 것으로 target을 인식하고 추적할 수 있는 감시 정찰 시스템을 제안하였다. OpenPose를 통해 추출되는 keypoints와 target의 이미지를 통해 모션을 분류한 결과를 비교하였을 때, keypoints를 통한 분류 정확도가 DNN을 제외하고 약 14% 정도로 높았으며 DNN의 약 50% 정도로 가장 높은 향상을 확인할 수 있었다. 하지만, 이는 분류해야 될 모션의 수가 적었기 때문으로 여겨진다. DNN과 CNN 및 LSTM의 모션 분류 정확도의 차이는 분류해야 될 모션의 수가 증가할수록 DNN의 overfitting 현상으로 인하여 역전될 것으로 보이며 앞으로의 연구에서는 이를 보완하는 방향으로 진행되어야 할 것이다. 또한, 본 연구의 추적 알고리즘에서 target의 x좌표만을 활용하여 이동경로 및 위치 결과를 추적하였지만, y좌표까지 활용하게 되면 감시 영역이 하늘로까지 넓어져 그 실용성이 확장될 것이다. 이와 같은 점들이 보완되어 지속적인 연구를 하게 된다면 보다 실용적이고 효율적인 군 감시 정찰 시스템을 구축할 수 있을 것이다. 그러므로 본 연구를 활용하여 기존의 군 감시 정찰 시스템을 보다 효율적으로 개선할 수 있으며, 모션 분류를 적용하여 유연하고 신속하게 대처할 수 있는 시스템을 구축할 수 있을 것이다.

Acknowledgements

We would like to thank Editage (www.editage.co.kr) for English language editing.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Reference

Alp Güler, R., Neverova, N., & Kokkinos, I. (2018). Densepose: Dense human pose estimation in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7297-7306). https://arxiv.org/abs/1802.00434

Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S.-E., & Sheikh, Y. A. (2019). OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1 - 1. https://doi.org/10.1109/tpami.2019.2929257

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2625-2634.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition,* 1725-1732. https://ieeexplore.ieee.org/document/6909619

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. *Advances in Neural Information Processing Systems, 27*, 568-576. https://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos.pdf

원 고 접 수 일 2020년 11월 09일
원 고 수 정 일 2020년 12월 22일
게 재 확 정 일 2020년 12월 24일

국문초록

https://doi.org/10.37944/jams.v3i3.80

지능형 감시 정찰 시스템 구축을 위한 OpenPose와 Deep Learning 기술 적용방안 연구*

최규정** · 오수영*** · 손채봉****

본 연구에서는 국방 감시 정찰 시스템을 OpenPose와 DNN(Deep Neural Networks), CNN(Convolutional Neural Networks), LSTM(Long Short-Term Memory)과 같은 딥러닝 네트워크를 통해 구현하였다. 본 연구의 시스템은 기존의 감시 정찰 시스템과는 다른 방식의 거동수상자(target) 인식 방법을 제안하고 있으며, 제안하는 방법은 촬영되는 영상에서 사람들의 모션을 분류함으로써 일반인과 거동수상자를 구분하는 것이다. 이를 위해 OpenPose를 통해 영상 내의 대상의 skeleton 데이터를 추출한다. 이때, 추출되는 skeleton 데이터에 포함되는 keypoints를 DNN, CNN, LSTM에 입력하여 모션을 분류하게 된다. 분류되는 모션들은 사주경계와 같이 군에서 배울 수 있는 모션으로 선정하였다. 시스템이 모션을 분류하여 거동수상자를 인식하게 되면 지도에 이를 표시하고 추적을 한다. 추적 알고리즘에서는 프레임별로 OpenPose를 통해 추출된 keypoints 값의 변화를 계산하여 거동수상자의 이동방향을 계산하고 카메라에서 얻은 depth 정보를 활용하여 카메라 위치를 기반으로 거동수상자를 지도에 표시하도록 한다. 이와 같은 모든 연산은 전체 이미지가 아닌 skeleton 데이터를 활용하였기 때문에 전체적인 연산량을 감소시킬 수 있게 된다.

주제어 : 오픈포즈, 키포인트, 심층 신경망, 합성곱 신경망, 장단기 기억 신경망

* 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터지원사업의 연구결과로 수행되었음 (IITP-2020-2016-0-00288).

** (제1저자) 광운대학교 전자통신공학과, 석사과정, kakm12@naver.com

*** (공동저자) 광운대학교 전자통신공학과, 석사과정

**** (교신저자) 광운대학교 전자통신공학과, 교수, cbsohn@kw.ac.kr