Occlusion-Aware Skeleton Trajectory Representation for Abnormal Behavior Detection

Onur Temuroglu[1], Yasutomo Kawanishi[1], Daisuke Deguchi[1], Takatsugu Hirayama[1], Ichiro Ide[1], Hiroshi Murase[1], Mayuu Iwasaki[2], and Atsushi Tsukada[2]

[1] Nagoya University, Japan
onurt@murase.is.i.nagoya-u.ac.jp
[2] Sumitomo Electric Industries, Ltd., Japan

Abstract. Surveillance cameras are expected to play a large role in the development of ITS technologies. They can be used to detect abnormally behaving individuals which can then be reported to drivers nearby. There are multiple works that tackle the problem of abnormal behavior detection. However, most of these works make use of appearance features which have redundant information and are susceptible to noise. While there are also works that make use of pose skeleton representation, they do not consider well how to handle cases with occlusions, which can occur due to the simple reason of pedestrian orientation preventing some joints from appearing in the frame clearly. In this paper, we propose a skeleton trajectory representation that enables handling of occlusions. We also propose a framework for pedestrian abnormal behavior detection that uses the proposed representation and detect relatively hard-to-notice anomalies such as drunk walking. The experiments we conducted show that our method outperforms other representation methods.

Keywords: Pose skeleton · Anomaly detection · Surveillance cameras

1 Introduction

Pedestrian-vehicle accidents occur frequently, and deaths in these accidents are not uncommon. Drivers not reacting fast enough to the behavior of pedestrians is one of the reasons for these accidents, making the task of understanding pedestrian behavior important. Drunk pedestrians especially pose a danger to traffic due to their abnormal walking patterns and decreased ability to react to surrounding environment. By detecting abnormally behaving pedestrians and warning drivers to their presence, we consider that it is possible to reduce accidents. While in-vehicle cameras can be used for this purpose, blind spots will always exist due to their positioning. As such, surveillance cameras are expected to play an active role in detecting abnormally behaving pedestrians that might pose a danger to drivers nearby.

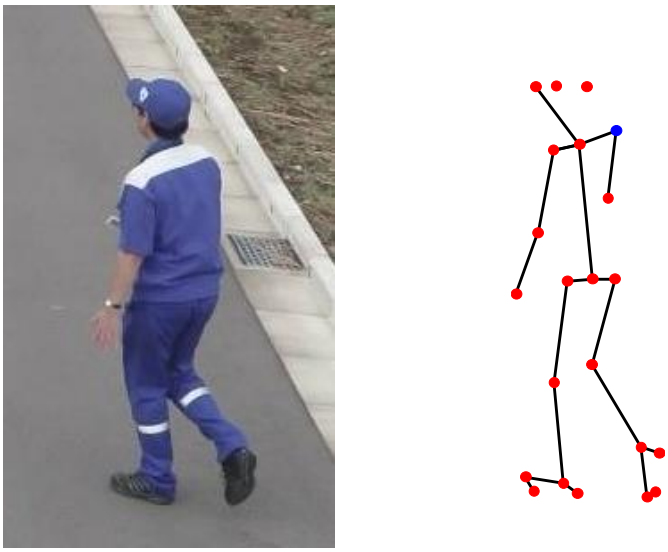
There are multiple works done on anomaly detection from a surveillance camera footage. However, most of these works focus on a different goal and make use of the pixel-based features directly [4, 5, 13]. These features are high-dimensional, and as such have large amounts of completely irrelevant information that could reduce the efficiency of models, or become harmful by acting as noise, masking relevant information [14]. These high-dimensional pixel-based features also open the way for variance that does not directly relate to behavior, such as different clothing or background, to become a problem.

To overcome the problems that might be caused by noise and variance, this paper proposes occlusion-aware 2D pose skeleton trajectory representation to detect abnormal behavior of a pedestrian in a surveillance camera footage. Pedestrian behavior is comprised of the actions pedestrians take, while the actions themselves can be deduced from the body skeleton trajectories of the pedestrians. Due to this, while pose skeletons are smaller in data size, they pack the same relevant information as appearance features and are easier to work with as they can be structured into a set of keypoint locations consistently, even revealing direction information.

While state-of-the-art pose estimation methods [2, 3, 8, 12] can achieve highly accurate results, surveillance camera images often include pedestrians that are occluded mainly due to their orientation and sometimes due to external elements, leading to missing keypoint information. An example of this can be seen in Figure 1. By implementing a distance metric that takes into consideration the effect of missing keypoint information, we should be able to achieve high accuracy behavior classification even for occluded data.

After acquiring the pose skeletons, we combine them into a sequence and use an AutoEncoder [6] network to encode and decode it. We classify the sequence into normal or abnormal by calculating the difference between the input and output. As AutoEncoder networks are not able to reconstruct data that vastly differ from those they were trained on, a large difference between input and output indicates an anomaly in the scope of the training data [11]. By using this method we can detect any kind of abnormal behavior without ever needing to define them, as we can just define the normal behavior instead. The contributions of this paper are as follows:

- We propose a skeleton trajectory representation that takes occluded keypoints into consideration and a distance metric for the said representation to achieve accurate classifications.
- We propose a framework for detecting abnormally behaving pedestrians from surveillance camera images, using the proposed skeleton trajectory representation. By doing so, we also eliminate the effects of variance such as clothing or background. Thanks to the processes we apply, our framework can detect abnormal behaviors even with relatively little posture change.
- We evaluate our method against other conventional methods using data taken in a real-like environment and show that our method achieves the highest accuracy in classifying normal and abnormal behaviors.



(a) Original image (b) Pose estimation result

Fig. 1. Example of an occluded pose estimation result. Note that the right hand and the eyes are not detected at all due to the pedestrian's orientation. The blue dot indicates the right shoulder.

The rest of this paper is organized as follows. In section 2, we give summaries of related work. In section 3, we describe our complete method in detail. In section 4, we evaluate our method against other methods. In section 5, we conclude our paper with a discussion on results and future work.

2 Related Work

Anomaly detection has always been a popular task in computer vision fields. There are multiple previous works challenging the anomaly detection problem for surveillance purposes, with different task settings and methodologies. Before deep learning methods became popular, Piciarelli et al. [9] used One-Class SVM to detect anomalous behavior by analyzing trajectories. There are also more recent works involving different deep learning methods. Hasan et al. [4] used AutoEncoder networks to extract features and detect abnormal events. Sultani et al. [13] employed a 3D convolutional network combined with fully connected layers to make the decision of whether a sequence is abnormal or not. Hinami et al. [5] used a generic convolutional network with environment specific classifiers to classify the type of abnormal action. However, all of these methods use appearance features directly, which as we mentioned, are susceptible to noise and variance such as different clothing and backgrounds. Bera et al. [1] used pedestrian tracking to learn the trajectories of the crowd and classifies trajectories deviating from the crowd trend as anomalies.

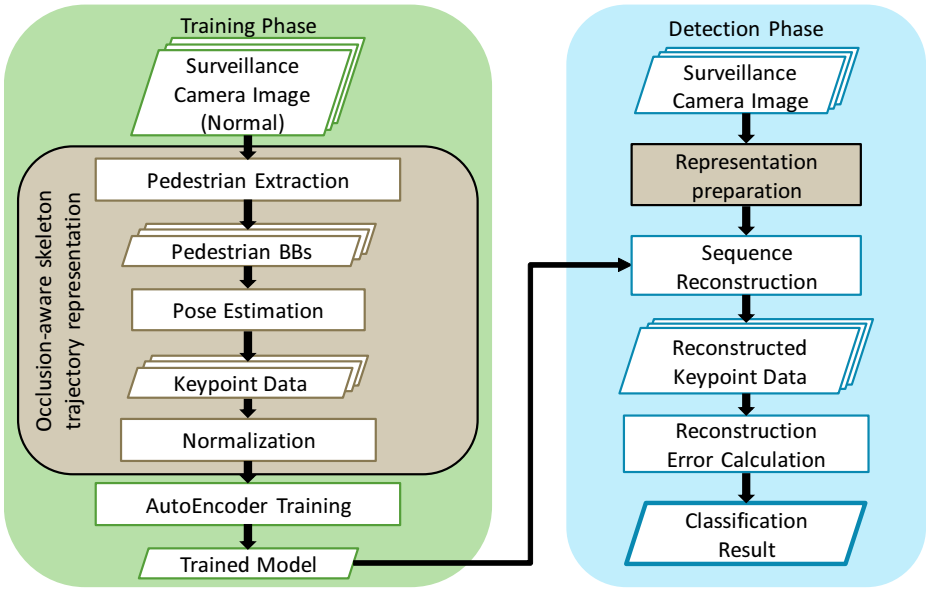


Fig. 2. Overall process flow of the proposed framework for abnormal behavior detection.

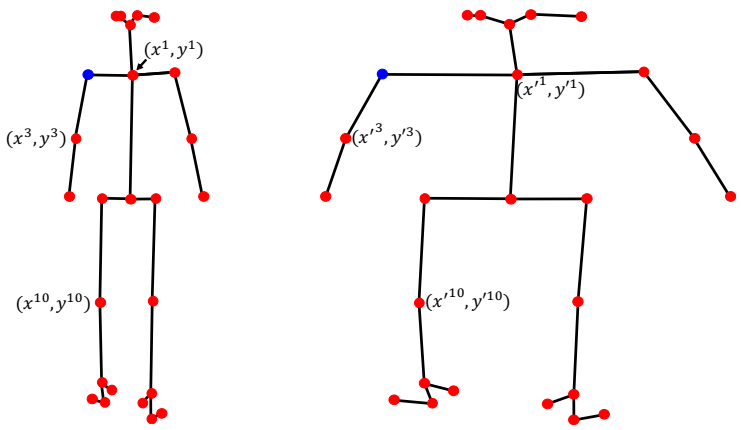
There are also some works using pose skeleton information for anomaly detection. An example of this is the work by Morais et al. [7], which proposes an RNN based method using pose skeletons. However, their method does not take into consideration the often occurring cases of occluded pose estimation results, which our paper addresses.

3 Abnormal Behavior Detection Making Use of Skeleton Representations

In this section, we describe our method for abnormal behavior detection in detail. This section mainly consists of two parts: Representing the skeleton information using the occlusion-aware skeleton trajectory representation and the details of implementation for our complete framework. Figure 2 shows the overall process flow for our framework.

3.1 Occlusion-aware skeleton trajectory representation

In our research, we define human skeleton as a set of 2D points $\{(x^j, y^j)\}_{j=1}^{J}$, where each element represents one of J body joints we use as shown in Figure 3. We use the term keypoints for these body joint points. Often, there are cases where at least one of these keypoints are occluded and are not estimated properly. However, as the networks that use skeletons to classify behavior take fixed



(a) Original scale (b) Normalized scale

Fig. 3. An example showing the structure of skeleton representation. The dots represent joint locations and the blue dot represents right shoulder.

shapes as input, these missing keypoints must also be represented in a way. If not represented properly, these missing keypoints will have an effect on both training and classification steps, leading to inaccurate results. In this paper, we use a homogeneous coordinate system-like representation with an additional dimension for occlusion flags. By making this distinction between detected and missing keypoints, we achieve a less noisy training set and open the way for a better loss function, explained in section 3.2.

We first need to apply pose estimation on an N-frame long pedestrian sequence we acquire using YOLO [10], as detailed in section 3.2. For pose estimation we use OpenPose [2]. For each frame, OpenPose generates a heatmap of each keypoint and part affinity fields, which it then uses to estimate the coordinates of keypoints together with a confidence value for each keypoint. A joint is considered missing or occluded if the values for it are zero. Using OpenPose, we obtain a skeleton trajectory $\mathcal{P} = \{P_1, ..., P_n, ..., P_N\}$, where each skeleton is represented as $P_n = \{(x_n^1, y_n^1, c_n^1), ..., (x_n^j, y_n^j, c_n^j), ..., (x_n^J, y_n^J, c_n^J)\}$. Here, x_n^j, y_n^j are 2D coordinates of the jth keypoint in the nth frame, and $c_n^j \in [0, 1]$ is the confidence level of the jth keypoint in the nth frame. We then normalize these values using Eqs. (1) and (2) and obtain a normalized skeleton P'_n:

$$
\begin{aligned}
x'^j_n &= \begin{cases} \frac{x_n^j - \min_i(x_n^i)}{\max_i(x_n^i) - \min_i(x_n^i)} & \text{if } c_n^j > 0 \\ 0 & \text{otherwise} \end{cases} \\
y'^j_n &= \begin{cases} \frac{y_n^j - \min_i(y_n^i)}{\max_i(y_n^i) - \min_i(y_n^i)} & \text{if } c_n^j > 0 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
\tag{1}
$$

$$c_n'^j = \begin{cases} 1, & c_n^j > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Eq. (2) is used here to make the difference between missing keypoints and detected keypoints more apparent. This normalization is applied to each element of $\mathcal{P}$ to obtain $\mathcal{P}'$.

3.2 Implementation details

In this section we explain the implementation and general process flow of the proposed framework, divided into multiple subsections.

Pedestrian detection. 2D pose estimation in images is a popular research topic with multiple implementations readily available. However, even when using a state-of-the-art technology, there are cases where it might estimate pedestrian poses incorrectly, especially in cases where a pedestrian is distant from the camera.

To overcome this problem, we extract regions with pedestrians which allows us to focus only on them for pose estimation. To do this, we apply pedestrian detection and N-frame tracking on the target footage, which outputs a pedestrian bounding box sequence $\mathcal{D} = \{d_1, ..., d_n, ..., d_N\}$ for each pedestrian, and then apply pose estimation on each detected pedestrian in each frame. Here, the result of each pedestrian detection consists of four values $d_n = \{x_n, y_n, w_n, h_n\}$ where x_n, y_n denotes the center point of detection in a frame, w_n the width, and h_n the height of the pedestrian. To ensure even the distant pedestrians, which are the major cases of low accuracy detections and offsets, are completely covered by the bounding box, we double the height and width to obtain bounding boxes with a size of $2h_o \times 2w_o$ for each detection. The end result is that we are able to extract more accurate pose skeleton information, allowing us to detect even abnormal behaviors that show relatively little change in posture such as drunk walking.

Conversion of detections into skeleton representations. We input the pedestrian bounding box sequence $\mathcal{D}$ and convert the sequence into skeleton trajectory representation with the method described in Section 3.1, acquiring $\mathcal{P}'$.

Training the AutoEncoder network. AutoEncoders are popular networks for feature extraction. They work by learning to represent the data they are trained on in lower dimensions through compression and decompression of the data. As such, in theory, they are not able to represent data that are not similar to the data they were trained on, and will have problems reconstructing them. This characteristic of AutoEncoders is exploited for anomaly detection. By only

training on normal data, we are able to separate normal and abnormal behaviors based on their reconstruction results.

We use the representation $\mathcal{P}'$ we generated in section 3.1 as the input for our AutoEncoder. The proposed AutoEncoder is a very simple network, comprised of only a few dense layers. This is because the pose skeletons are compact while having relevant information. The network consists of an encoder network E and a decoder network D which are used sequentially to acquire the reconstruction $\hat{\mathcal{P}}' = D(E(\mathcal{P}'))$. As we map the x, y values for missing keypoints to 0 and the network tries to reconstruct them, we use the custom loss function shown in Eq. (3) to optimize the parameters of the network

$$L(\hat{\mathcal{P}}', \mathcal{P}') = \frac{1}{2NJ} \sum_{n=1}^{N} \sum_{j=1}^{J} c_n'^j ((\hat{x}_n'^j - x_n'^j)^2 + (\hat{y}_n'^j - y_n'^j)^2), \tag{3}$$

where $(\hat{x}_n'^j, \hat{y}_n'^j)$ are the reconstruction result of the jth keypoint of the nth skeleton. Note that we introduce all the parameters used in Eq. (3) in section 3.1.

Abnormal behavior detection. During test time, we use the loss function as it is to calculate the reconstruction difference between our normalized test input $\mathcal{P}^*$ and its respective output $\hat{\mathcal{P}}^*$, and classify the input as abnormal if the difference is above a certain threshold τ as shown in Eq. (4). Here, τ is determined empirically as explained in section 4.4.

$$C(\mathcal{P}^*) = \begin{cases} \text{normal} & \text{if } L(\hat{\mathcal{P}}^*, \mathcal{P}^*) \leq \tau \\ \text{abnormal} & \text{otherwise} \end{cases} \tag{4}$$

4 Experiment

In this section, we describe the experiment we prepared to evaluate the proposed method in detail.

4.1 Drunk walking dataset

The data we used in our experiment were taken by ourselves and consists of multiple people walking in a normal fashion for the normal pattern, and acting out drunkenness while walking for the abnormal pattern. With the relatively little posture changes in the dataset, it is useful for showing the effectiveness of the proposed method in classifying abnormal behaviors with little posture changes. It contains 112 video clips of people walking in aforementioned patterns, taken at 10 frames per second. Out of these video clips, 56 are normal clips and 56 are abnormal clips. Figure 4 shows example sequences from the dataset. We process this data into 7,223 normal sequences and 10,629 abnormal sequences with a length of 30 frames each. The discrepancy in sequence counts is due to an abnormal clip being longer than a normal clip on average. Out of the 7,223

(a) Normal sequence

(b) Abnormal sequence

Fig. 4. Sample sequences from the dataset. The facial region is masked for the sake of anonymization in this paper.

normal sequences, we use 4,124 for training, 1,147 for validation, and 1,952 for testing. For the abnormal sequences, we randomly pick 1,000 sequences out of 10,629 sequences for testing. Note that we use the same exact sequences for preparing each of the comparison methods in section 4.2.

4.2 Comparison methods

In addition to the proposed method, we prepare multiple representations of our drunk walking dataset for use as comparison methods in our experiment, detailed as follows:

Comparative 1 (Keypoints without occlusion flags): This method is closest to our representation, featuring 2D coordinates of J keypoints for N frames where the nth frame is $P_n^e = \{(x_n^1, y_n^1), ..., (x_n^j, y_n^j), ..., (x_n^J, y_n^J)\}$.

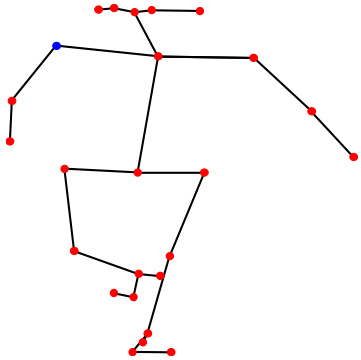
Comparative 2 (Heatmaps): A heatmap represents an image based output that shows the possible area for each keypoint at the same size as the input image, with higher pixel value being more probable. In this experiment we prepare sequences composed of images with a size of 96×96 pixels where each pixel has a value between 0 and 1.

Comparative 3 (Cropped pedestrian images): For this method, we crop the greyscale results of the pedestrian detection in a square shape and scale the images to the same size of 96×96 pixels. We then normalize the values of pixels between 0 and 1.

Figure 5 visualizes the three variations.

(a) Cropped pedestrian image

(b) Heatmap



(c) 25 keypoints (normalized)

Fig. 5. Different variations of data used for experiments. The blue dot indicates the right shoulder.

4.3 Network parameters

We prepare multiple AutoEncoder based networks for the variations we prepared. For heatmaps and cropped pedestrian image sequence, we use a Convolutional AutoEncoder network where the sequence of frames are given as channels while we use a standard AutoEncoder network comprised of dense layers for the other variations. However, we adjust the parameter count in accordance to the keypoint counts for each. For the AutoEncoder, we compose it with four encoding and four decoding layers, while reducing the number to three for the Convolutional AutoEncoder, as complex and powerful networks shrink the reconstruction gap between normal and abnormal patterns. We use sigmoid as the activation function on the last layer for all networks, and ReLU for all middle layers. Also, we use mean squared error as the loss function for all variations except for the proposed one.

Table 1. Results of methods by epoch count.

Epoch	Comparative 1 (Keypoints no flags)	Comparative 2 (Heatmap)	Comparative 3 (Image)	Proposed
20	0.838	0.889	0.577	**0.927**
100	0.856	0.855	0.551	**0.914**

4.4 Experimental setting

For the pedestrian detection and pose estimation methods, we applied YOLO [10] and OpenPose [2], respectively, and used all available keypoints ($J = 25$).

We opted to use balanced accuracy for the accuracy metric as shown in Eq. (5).

$$\text{Balanced Accuracy} = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP}+\text{FN}} + \frac{\text{TN}}{\text{TN}+\text{FP}}\right) \tag{5}$$

Here, TP, TN, FP, and FN represent the number of true positive, true negative, false positive, and false negative results, respectively. Using balanced accuracy instead of standard accuracy allows us to evaluate the performance more accurately when using unbalanced data.

We decided on the threshold by acquiring the reconstruction difference for all test data and calculating the balanced accuracy on 100 different thresholds. Here these thresholds are chosen as intervals between the mean reconstruction difference of normal data, and the abnormal data. We used the threshold with best results out of them for each of the methods. We trained the networks for 20 epochs and 100 epochs. The reason we evaluated at a low epoch count is because the aim is not to accurately reconstruct, but to have a large difference of reconstruction accuracy in normal patterns and abnormal patterns.

4.5 Results

The results are summarized in Table 1. We can see that methods that make use of pose skeleton representation vastly outperform the method using cropped pedestrian images. We can also see that the proposed method of pose skeleton representation and custom loss function outperforms the comparison methods. Figure 6 shows the reconstruction difference distribution of the data for the proposed method, and the relation of the thresholds with the balanced accuracy.

4.6 Performance on data with different levels of missing joints

We evaluated the accuracy of methods by grouping the experimental data into various levels of missing keypoint information. For the grouping, we use 25 keypoint data with a length of 30 frames for abnormal sequences and split them into three groups to evaluate them separately. We use all 10,629 abnormal sequences and separate them into three datasets as Easy (0–22), Moderate (23–86), and

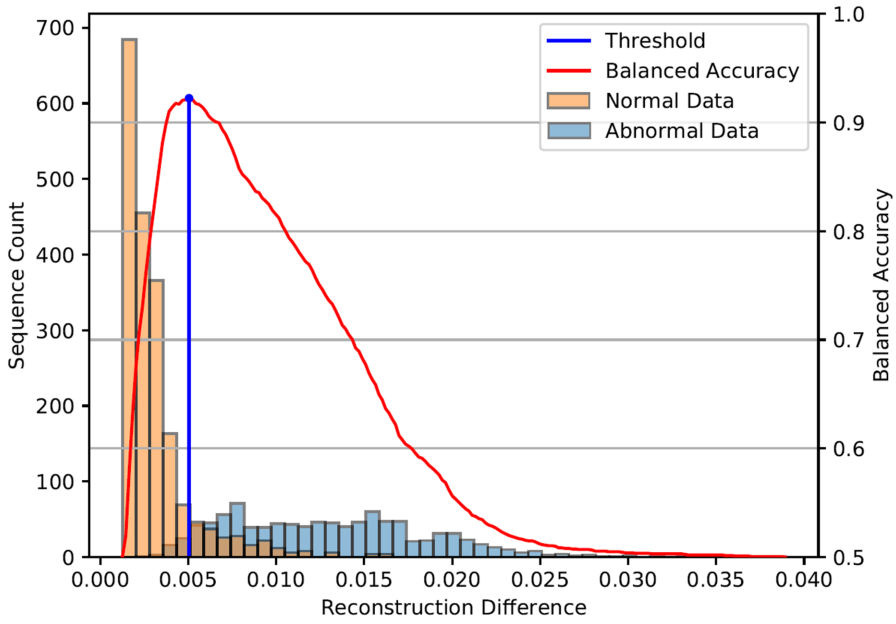


Fig. 6. Distribution of reconstruction difference of the data for the proposed method and the relation of threshold selection to the balanced accuracy.

Table 2. Accuracy of methods on grouped abnormal data.

Dataset	Comparative 1 (Keypoints no flags)	Comparative 2 (Heatmap)	Proposed
Easy (0–22)	0.710	0.911	**0.986**
Moderate (23–86)	0.958	0.988	**0.996**
Hard (87+)	0.850	0.865	**0.894**

Hard (87+) according to the number of joints missing out of 750 total joints in a sequence. This gives us 3,435, 3,611, and 3,583 sequences for Easy, Moderate, and Hard sequences, respectively. The graph showing sequence counts by missing keypoints is shown in Figure 7.

In this experiment, we evaluated three methods, namely the methods that use heatmaps, skeletons with no occlusion flags, and skeletons with occlusion flags (proposed). Table 2 shows the accuracy of classification for each method on the grouped abnormal data. We can see that the representation without occlusion flags has the lowest accuracy in all categories. This is because of the noise that occur in the training data, as a result of the representation. The network is trained on this noisy data, and adapted for it, which is why it has trouble trying to classify data in the easy category. We can see this effect diminishes for the moderate category, as it is what the network adapted to. The hard category has

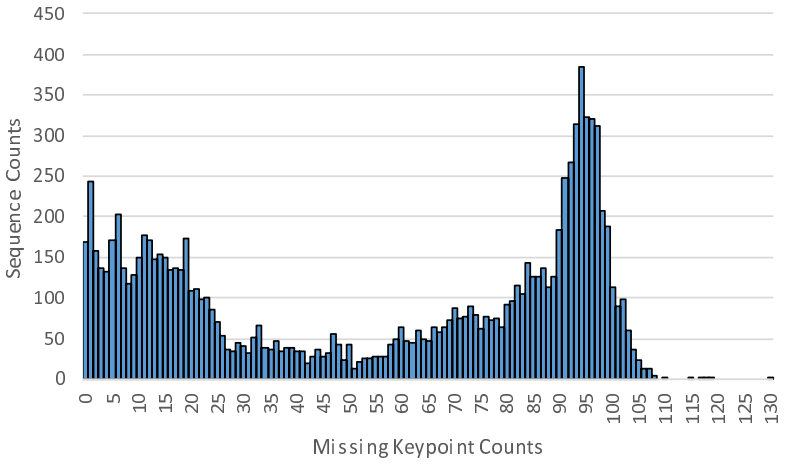


Fig. 7. Histograms of sequence counts by missing keypoints.

a lower accuracy compared to moderate, due to emerging patterns that are too extreme to be classified corretly by the network.

These changes in classification accuracy by categories are also observed on the heatmap representation, albeit to a lesser extent. This is because the heatmap representation removes the noise by removing the representation of missing keypoints.

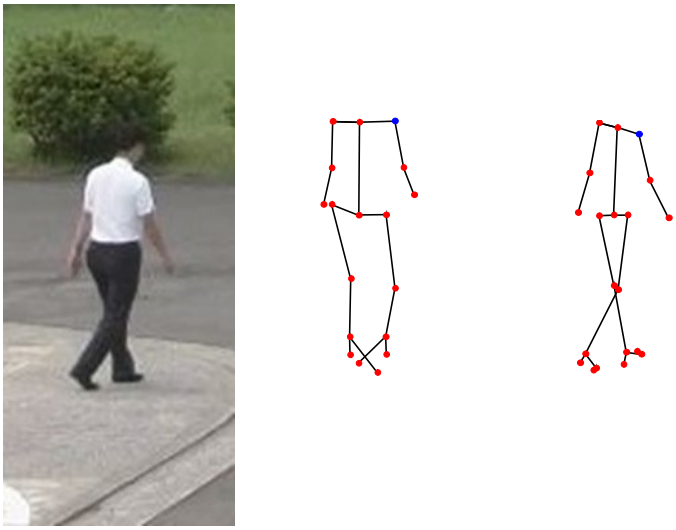
Finally , the proposed method is seen to have the highest accuracy, with less difference between categories, showing the effectiveness of the proposed method in representation of the skeleton information.

5 Conclusion

In this paper, we proposed a framework for abnormal behavior detection that makes use of skeleton representation. We also proposed a representation for pose skeletons alongside a custom loss function to be used in the framework, which takes into account missing keypoints that are seen often due to the nature of surveillance cameras. We included a pedestrian detection method as a step for constructing the representations, which played a role in more accurate estimations. An example of this can be seen in Figure 8.

We compared our method with multiple representation methods and showed the effectiveness of the proposed method. We also evaluated the robustness of the representations to missing keypoints in data, and showed that the proposed method outperforms others in most cases.

As future work, we are considering the incorporation of motion information as they are compact and highly representative of behavior.



(a) Detection result (b) Pose estimation on complete image (c) Pose estimation on detection result

Fig. 8. Comparison between using OpenPose on a complete image and on a pedestrian detection result. The pose estimation target pedestrian cropped from a larger image is shown in (a). Result of using OpenPose on the complete image is shown in (b) and the result of using OpenPose on the detection result is shown in (c). The blue dot indicates the right shoulder.

Acknowledgment

Parts of this research were supported by MEXT, Grants-in-Aid for Scientific Research.

References

1. Bera, A., Kim, S., Manocha, D.: Realtime anomaly detection using trajectory-level crowd behavior learning. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 50–57 (2016)
2. Cao, Z., Simon, T., Wei, S. E., Sheikh, Y.: Realtime multi-person 2D pose estimation using part affinity fields. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. pp. 7291–7299 (2017)
3. Fang, H. S., Xie, S., Tai, Y. W., Lu, C.: RMPE: Regional multi-person pose estimation. In: Proceedings of the 2017 IEEE International Conference on Computer Vision. pp. 2334–2343 (2017)
4. Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A. K., Davis, L. S.: Learning temporal regularity in video sequences. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. pp. 733–742 (2016)
5. Hinami, R., Mei, T., Satoh, S.:. Joint detection and recounting of abnormal events by learning deep generic knowledge. In: Proceedings of the 2017 IEEE International Conference on Computer Vision pp. 3619–3627 (2017)
6. Hinton, G. E., Salakhutdinov, R. R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
7. Morais, R., Le, V., Tran, T., Saha, B., Mansour, M., Venkatesh, S.: Learning regularity in skeleton trajectories for anomaly detection in videos. In: Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition. pp. 11996–12004 (2019)
8. Papandreou, G., Zhu, T., Chen, L. C., Gidaris, S., Tompson, J., Murphy, K. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In: Proceedings of the 2008 European Conference on Computer Vision. pp. 269–286 (2018)
9. Piciarelli, C., Micheloni, C., Foresti, G. L.: Trajectory-based anomalous event detection. IEEE Transactions on Circuits and Systems for Video Technology, **18**(11). pp. 1544–1554 (2008)
10. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788 (2016)
11. Sakurada, M. Yairi, T.: Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis. 4p. (2014)
12. Sekii, T.: Pose proposal networks. In: Proceedings of the 2018 European Conference on Computer Vision. pp. 342–357 (2018)
13. Sultani, W., Chen, C., Shah, M.: Real-world anomaly detection in surveillance videos. In: Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition. pp. 6479–6488 (2018)
14. Zimek, A., Schubert, E., Kriegel, H. P.: A survey on unsupervised outlier detection in highdimensional numerical data. Statistical Analysis and Data Mining: The ASA Data Science Journal **5**(5), 363–387 (2012)