J Inf Process Syst, Vol.21, No.5, pp.508~516, October 2025
https://doi.org/10.3745/JIPS.02.0227 ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Single-Object Fall Detection Using Pose Estimation 3D Coordinates via Generating 3D Object Coordinates from 2D Object Coordinates

Jinmo Yang[1], Kidu Kim[2], and R. Young Chul Kim[1,*]

Abstract

Object fall detection is one of the significant applications in pose estimation. Traditional approaches heavily rely on fully mature neural networks, which may be complex to implement and resource-intensive. In this paper, we suggest a simple approach to detect Human fall on a single object using a mathematical comparison mechanism from 3D pose estimation landmarks lifted from 2D landmarks, suitable for low-specification systems. Our research focuses on dimensional lifting in 2D to 3D pose estimation based on object tracking—to adapt mapping 2D toons to 3D toons. For future research, we aim to develop a lightweight neural network for enhanced performance from ensemble effects and complex action detection. We also plan on enhancing the algorithm for multi-person detection. At this moment, our research extends into 3D toon generation. Our method achieves an F1-score of 94.7% (accuracy of 96.7%) with 28.2 FPS in detecting falls from webcam footage in a controlled environment.

Keywords

Dimensional Lifting, Pose Estimation, Object Tracking Window

1. Introduction

Pose estimation is a widely used computer vision neural network technique to analyze movements in objects such as humans and animals by denoting and utilizing joints in the body as landmarks—pose estimation results [1]. From the arrangement of the landmarks, the actions or statuses of the object can be detected. The most conventional way to process the arrangement is to feed the landmarks into a deep-learning neural network capable of processing spatial and temporal data [2, 3] or to use the image frame as an input to a fully mature neural network [4]. However, developing such neural networks requires advanced knowledge of conventional convolutional layers for spatial processing, attention/LSTM-based layers for temporal processing, and a high-end system to run the network [5, 6]. This paper proposes a simple algorithm for single human fall detection without developing a fully mature neural network. Instead, relationships among the axial distributions of pose estimation landmarks are analyzed. This detection mechanism is done without further machine or deep learning, allowing the detection algorithm to run on low-level systems. We will present related works in Section 2, our approach to fall detection in

Manuscript received August 8, 2024; accepted January 27, 2025.

***Corresponding Author:** R. Young Chul Kim (bob@hongik.ac.kr)
[1] SE Lab., Hongik University, Seoul, Korea (yjmd2222@g.hongik.ac.kr, bob@hongik.ac.kr)
[2] Telecommunications Technology Association, Seongnam, Korea (kdkim@tta.or.kr)

Section 3, the experiment in Section 4, the comparison in Section 5, the application effort of our mechanism on 3D toon generation in Section 6, and the conclusion in Section 7.

2. Related Works

Pose estimation can be used to detect driving drowsiness. In the study by Yang et al. [7], pose estimation and a simple dense neural network consisting of fully connected layers were used to detect drowsy driving. An image frame from a lateral-view webcam was processed by a pose estimation model, producing an intermediate output of landmarks of the driver. Then, this set of landmarks entered the dense neural network, which classified the pose as normal or drowsy. This process was repeated at finite intervals, producing classification results and warning messages every 5 seconds. The detection mechanism relied on the assumption that the driver's pose in drowsiness was dangerous at any point and that a time window for consecutive poses would not have been necessary for the lateral view of the already downward, dozing head. If the frontal view of closing eyes was used instead—as in most traditional methods—the composite model would have required cumulative closure counts of the eyes, along with the changes in the model subcomponents for eye detection. This cumulative detection is necessary because single-frame detections will produce false positives from blinking the eyes in the normal driving state [8]. Although there may be tradeoffs, choosing a different set of data from a different perspective can help reduce model complexity and use of resources; the traditional way of detecting drowsy driving by eye closure requires constant measurements for a final prediction of sleeping eyes or normal eyes, but the lateral view requires only a single shot to predict the sleeping posture.

The performance of state-of-the-art computer vision models for human fall detection increases as they develop. Lee et al. [4] used a dynamic vision sensor camera to produce a new fall detection dataset with which a temporal network (DVS-TN) was trained. Wang and Deng [9] used a cascade of two random forest models—machine learning models—to train from the preprocessed features of landmarks obtained from BlazePose to detect human fall. Even on a budget laptop CPU, their work achieved a speed of 29.7 frame per second (FPS), outperforming many others in speed, albeit with a slightly lower accuracy of 89.99%. Achieving such scores required only a traditional machine learning model that used lower computational resources than major deep learning algorithms. This raises the question of whether a learning algorithm is necessary for human fall detection, pointing to research using a simple comparison algorithm to achieve the same goal.

A 3D production technology has been extensively researched for video or image generation. In the task Artificial Intelligence-based User Interactive Storytelling 3D Scene Authoring Technology Development supported by the Korea Creative Content Agency (KOCCA), the efforts to create reproducible 3D toons from natural language analysis are underway. These efforts include using Unified Modeling Language (UML) diagrams to model the scene, redefining case grammar to capture important context from sentences, and classifying themes using machine learning algorithms [10-12]. Mapping 2D toons to 3D toons is also being researched by mapping 2D objects to 3D counterparts in the database and converting 2D objects to 3D objects, where dimensional lifting is considered.

3. Simple Fall Detection Mechanism

Our object fall detection mechanism is divided into three parts: single-frame pose estimation, single-

frame fall detection, and multi-frame fall detection. First, a 3D pose estimation model produces landmarks of a person in an image frame. An OpenPose-based lightweight model was selected. This model predicts 2D landmarks from the image and subsequently 3D landmarks from locations of 2D counterparts in the image [13, 14], as shown in Fig. 1. Producing 3D landmarks from 2D landmarks is called lifting [15]. Ultimately, this OpenPose-based model produces an output of 3D landmarks of a person present in the single image frame.

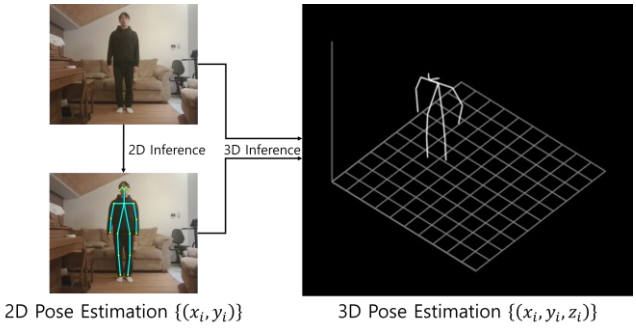

Fig. 1. The 3D pose estimation process with OpenPose-based model.

Secondly, the distributions of 3D axial directions are compared from the detected landmarks. Many statistics are used in comparing distributions, but the standard deviation is chosen as the relevant statistic. The standard deviation measures the extent of scattering in a set of values [16]; the scattering in width, depth, and height can be compared to determine if the records are more dispersed in one direction than in the others. The standard deviation is calculated as follows:

$$\sigma_p = \sqrt{\frac{\sum_k^n \left(p_k - \mu_p\right)^2}{n}}. \tag{1}$$

where p, n, and μ_p are the position in the direction of width x, depth y, or height z; the total number of landmarks; and the average position in p direction, respectively. If the coordinates are far apart from one another, the value of the standard deviation is large. As in inverse, if the coordinates are close together, the standard deviation value is small. Since x and y are coordinates of the horizontal directions, if the standard deviation of z, the height, is less than that of either of the horizontal directions, the person that the landmarks correspond to is considered to have fallen in this image frame, explained by the following equation:

$$\text{currentStatus} = \sigma_z < \sigma_x \text{ or } \sigma_z < \sigma_y \tag{2}$$

The value of currentStatus can be either 1, representing the state of fall, or 0, representing the state of normal.

Lastly, a series comprising currentStatuses of consecutive temporal sets of landmarks are aggregated. This is to ensure that the real state of fall must not change. In other words, there must be no false positives in single images, such as “temporarily getting close to the ground to pick up something that was dropped” being marked as real fall. For a window of $\Delta t = t_f - t_s$ seconds from start of t_s seconds to the final of t_f seconds, if the proportion of the series is greater than equal to the threshold, the person is in the cumulative state of fall, represented by the following equation:

$$\text{cumulStatus}(t_s, t_f) = \text{int}\left(\frac{\sum_{t=t_s}^{t_f} \text{currentStatus}(t)}{\#_{t=t_s}^{t_f} t} \geq \text{threshold}\right). \quad (3)$$

As with currentStatus, the values of cumulStatus are 1 for cumulative-fall and 0 for cumulative-normal.

And the overall process of our fall detection mechanism is shown in Fig. 2.

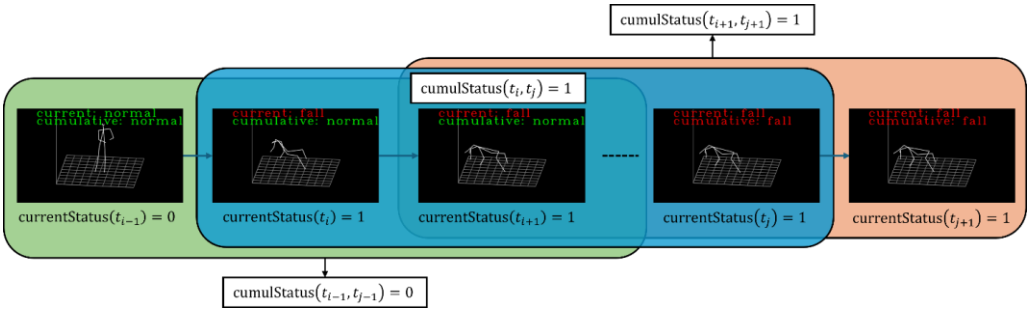

Fig. 2. The process of our fall detection mechanism.

4. Experiment

Our simple fall detection mechanism was tested on a system with the i7-13700HK, RTX4070, 32 GB of RAM, and a Full HD webcam in a light environment. The experiment procedure is as follows:

1. Set up the computer and webcam system.
2. Start the simple fall detection program with a window time of 10 seconds and a threshold of 0.8.
3. One human subject enters the webcam view. All limbs must remain inside the view.
4. The subject lays on the ground for 10 seconds.
5. Record the detection result.
6. Repeat Steps 4 and 5 with the subject laying on the ground for 5 seconds and standing for 5 seconds in series and standing continuously for 10 seconds.
7. Repeat Steps 4–6 10 times.

Following the procedure, the results are obtained and accumulated to produce a confusion matrix, shown in Fig. 3.

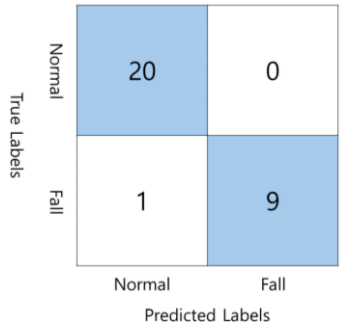

Fig. 3. The confusion matrix of fall detection.

As seen in Fig. 3, the system achieved 94.7% on the F1-score (96.7% on accuracy) with 28.2 FPS. This is possible because only a single person is always present in the webcam view, and all the body limbs are clearly visible. The only factor reducing the metric score was the incorrect detection of the limbs in pose estimation. This indicates that, for the task of detecting human falls, our mechanism will produce correct results at high confidence, given that the single-person pose estimation output quality is high.

5. Comparison

Here, we compare the performance of our algorithm against other notable works with machine- or deep-learning algorithms. Table 1 shows their accuracies [17-20].

Table 1. Comparison of accuracy among fall detection models

Method	Accuracy (%)
Chang et al. [17]	98.1
Wang and Deng [9]	89.9
Wu et al. [18]	99.8
Osigbesan et al. [19]	80.5
Zampino et al. [20]	92.4
Proposed method	96.7

In Table 1, the highest accuracy, at 99.8%, is reported by Wu et al. [18]. The model's internal structure is convolutional layers, which may result in high correctness performance at the cost of low computation performance.

The accuracy of our model is at the median of the models. Although the performance in correctness is mild, low-level systems can benefit from the simple calculations for detection against the heavy computations of artificial intelligence (AI) algorithms.

6. Semi-3D Toon Generation via 2D Object Tracking

Mapping natural-language-generated 2D toons to 3D toons is ongoing research in the KOCCA project [10-12]. An emerging effort is to accomplish this by applying pose estimation and dimensional lifting. A toon is a graphic art represented by a sequence of images to tell a story in which many entities, such as actors, objects, and the background, interact with one another [21]. One thing to consider for toon generation is to track all the entities in the cut. This is to identify the states of the entities being tracked, thereby ensuring smooth transitions from the previous cut to the next and allowing reusability of the states.

For tracking objects—specifically for the utilization of the kinematics of these objects in future 3D toon development—the CentroidKF tracker is chosen. Fig. 4 shows the schematics of the CentroidKF tracker's tracking process. The CentroidKF tracker assigns a unique ID to the object by applying the Kalman filter and using the intermediate results in the centroid tracker [22]. The Kalman filter estimates the next position of the object with its respective velocity and acceleration calculated from the previous

frames [23]. Then, the centroid tracker selects the minimum distance between the estimated position from the previous frame and the current position from the object's trajectory [22, 24].

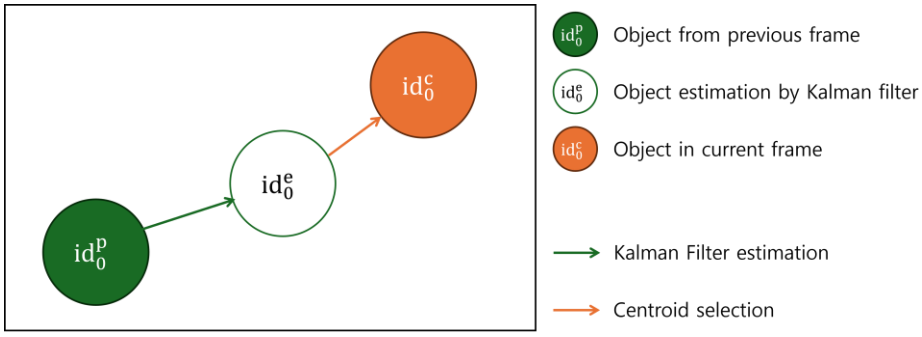

Fig. 4. The CentroidKF object tracker.

For the actual application of pose estimation, the sets of pose estimation landmarks are grouped and enclosed in so-called bounding boxes. Then these bounding boxes are processed by the CentroidKF tracker for ID assignment on the bounding boxes, allowing the landmarks to be tracked.

Fig. 5 shows the identification of a single person tracked by a bounding box enclosing the landmarks. Based on the results, three remarks can be made. First, as the Kalman filter estimates the future positions of the object from kinematics, the occlusions of the object can be mitigated. This can be helpful in detecting an object screened by other objects as well as the object limbs (parts) hidden behind the object's body. Secondly, in addition to tracking a 2D object, tracking in the 3D space may be more suitable for representing the movements of the 3D object. Lastly, when applied to fall detection, our simple fall detection mechanism can be modified to detect multiple sets of landmarks with different IDs.

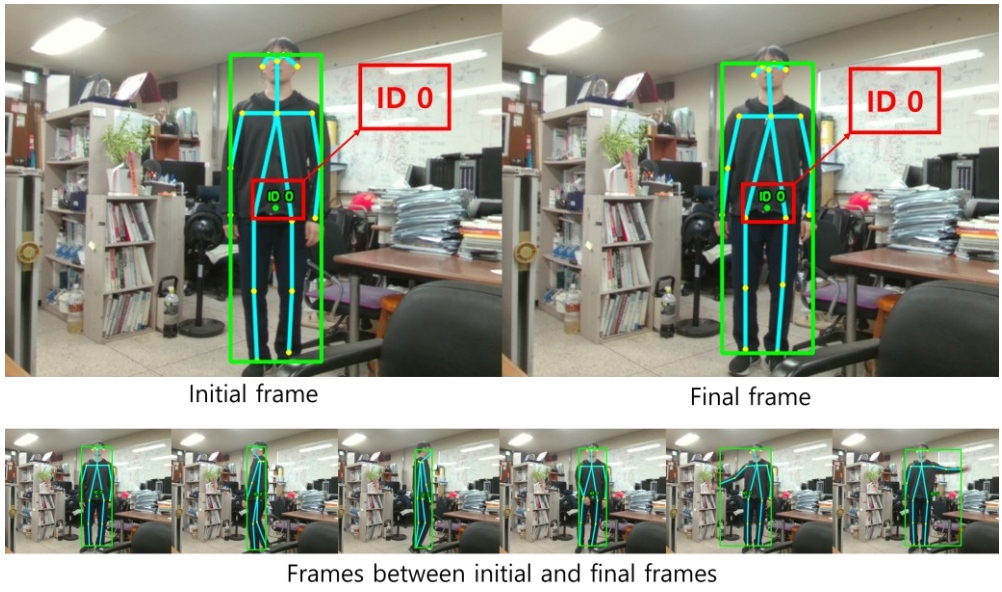

Fig. 5. Tracking the pose estimation landmarks.

7. Conclusion

Detecting actions or statuses from pose estimation has traditionally required a fully mature neural network capable of processing spatial and temporal data. In a classification task for which the set of pose estimation landmarks shows a direct pattern, a simple mathematical comparison can be used to detect such a pattern. We have presented a simple human fall detection mechanism to detect human falls from webcam footage and achieved an F1-score of 94.7% (accuracy of 96.7%) at 28.2 FPS. This can be useful if the system has low specifications for running—training or monitoring—a fully mature neural network or the output data quality of the pose estimation is high. However, if the actions show complex patterns in the spatial or temporal space or the intermediate pose estimation output quality is low, our approach may have degradation in performance. Nevertheless, this is preliminary work for detecting a broad spectrum of emergencies and generating 3D toons. For future work, we aim to design metrics for comparing models by performance in correctness and computation; develop a fully mature, lightweight neural network to compare with our algorithm; extend our mechanism from single-person detection to multi-person detection via object tracking; train the neural network for complex action detection; and lift objects from 2D to 3D for 3D toon generation.

Conflict of Interest

The authors declare that they have no competing interests.

Funding

This research was supported in part by Korea Creative Content Agency (KOCCA) grant funded by the Ministry of Culture, Sports and Tourism (MCST) in 2025 (Project Name: Artificial Intelligence-based User Interactive Storytelling 3D Scene Authoring Technology Development; Project No. RS-2023-00227917307820870502 01) and in part by 2025 Hongik University Innovation Support Program Fund.

References

[1] C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz, and M. Shah, “Deep learning-based human pose estimation: a survey,” *ACM Computing Surveys*, vol. 56, no. 1, article no. 11, 2023. https://doi.org/10.1145/3603618

[2] D. C. Luvizon, D. Picard, and H. Tabia, “2D/3D pose estimation and action recognition using multitask deep learning,” in *Proceedings of the IEEE Conference on Computer Vision And Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 5137-5146. https://doi.org/10.1109/CVPR.2018.00539

[3] H. J. Bae, G. J. Jang, Y. H. Kim, and J. P. Kim, “LSTM (long short-term memory)-based abnormal behavior recognition using AlphaPose,” *KIPS Transactions on Software and Data Engineering*, vol. 10, no. 5, pp. 187-194, 2021. https://doi.org/10.3745/KTSDE.2021.10.5.187

[4] H. Lee, J. Kim, D. Yang, and J. H. Kim, “Embedded real-time fall detection using deep learning for elderly care,” 2017 [Online]. Available: https://arxiv.org/abs/1711.11200.

[5] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A survey of convolutional neural networks: analysis,

applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999-7019, 2022. https://doi.org/10.1109/TNNLS.2021.3084827

[6] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "Spatio-temporal attention-based LSTM networks for 3D action recognition and detection," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3459-3471, 2018.b https://doi.org/10.1109/TIP.2018.2818328

[7] J. Yang, J. Kim, R. Y. C. Kim, and K. Kim, "Learning model for avoiding drowsy driving with MoveNet and dense neural network," *International Journal of Internet, Broadcasting and Communication*, vol. 15, no. 4, pp. 142-148, 2023. http://dx.doi.org/10.7236/IJIBC.2023.15.4.142

[8] T. Danisman, I. M. Bilasco, C. Djeraba, and N. Ihaddadene, "Drowsy driver detection system using eye blink patterns," in *Proceedings of 2010 International Conference on Machine and Web Intelligence*, Algiers, Algeria, 2010, pp. 230-233. https://doi.org/10.1109/ICMWI.2010.5648121

[9] Y. Wang and T. Deng, "Enhancing elderly care: efficient and reliable real-time fall detection algorithm," *Digital Health*, vol. 10, article no. 20552076241233690, 2024. https://doi.org/10.1177/20552076241233690

[10] J. Kim, J. Kong, H. D. Heo, S. H. Chun, and R. Y. C. Kim, "toon image generation of main characters in a comic from object diagram via natural language based requirement specifications," *The International Journal of Advanced Smart Convergence*, vol. 13, no. 1, pp. 85-91, 2024. http://doi.org/10.7236/IJASC.2024.13.1.85

[11] H. Kim, J. H. Kong, H. S. Son, and R. Y. C. Kim, "Best practice on automatic toon image creation from JSON file of message sequence diagram via natural language based requirement specifications," *The International Journal of Advanced Smart Convergence*, vol. 13, no. 1, pp. 99-107, 2024. http://doi.org/10.7236/IJASC.2024.13.1.99

[12] W. S. Jang, J. Kong, K. S. Yi, J. Kim, Y. J. Jin, H. Kim, K. Kim, and R. Y. C. Kim, Classifying themes of natural language-based sentences based on C3Tree model for 3D scene authoring," in *Proceedings of the 12th International Conference on Green and Human Information Technology (ICGHIT)*, Hanoi, Vietnam, 2024.

[13] D. Osokin, "Real-time 3D multi-person pose estimation demo," GitHub, 2021 [Online]. Available: https://github.com/Daniil-Osokin/lightweight-human-pose-estimation-3d-demo.pytorch.

[14] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt, "Single-shot multi-person 3D pose estimation from monocular RGB," in *Proceedings of 2018 International Conference on 3D Vision (3DV)*, Verona, Italy, 2018, pp. 120-130. https://doi.org/10.1109/3DV.2018.00024

[15] F. Zhou, J. Yin, and P. Li, "Lifting by image: leveraging image cues for accurate 3D human pose estimation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 7, pp. 7632-7640, 2024. https://doi.org/10.1609/aaai.v38i7.28596

[16] S. El Omda and S. R. Sergent, "Standard deviation," 2024 [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK574574/.

[17] W. J. Chang, C. H. Hsu, and L. B. Chen, "A pose estimation-based fall detection methodology using artificial intelligence edge computing," *IEEE Access*, vol. 9, pp. 129965-129976, 2021. https://doi.org/10.1109/ACCESS.2021.3113824

[18] L. Wu, C. Huang, L. Fei, S. Zhao, J. Zhao, Z. Cui, and Y. Xu, "Video-based fall detection using human pose and constrained generative adversarial network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 4, pp. 2179-2194, 2024. https://doi.org/10.1109/TCSVT.2023.3303258

[19] A. Osigbesan, S. Barrat, H. Singh, D. Xia, S. Singh, Y. Xing, W. Guo, and A. Tsourdos, "Vision-based fall detection in aircraft maintenance environment with pose estimation," in *Proceedings of 2022 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Bedford, UK, 2022, pp. 1-6. https://doi.org/10.1109/MFI55806.2022.9913877

[20] C. Zampino, F. Biancospino, M. Brienza, F. Laus, G. Di Stefano, R. Romano, A. Pennisi, V. Suriani, D. D. Bloisi, "Fall detection using NAO robot pose estimation in RoboCup SPL matches," in *Proceedings of the*

9th Italian Workshop on Artificial Intelligence and Robotics co-located with the 21th International Conference of the Italian Association for Artificial Intelligence (AIRO@AIxIA), Udine, Italy, 2022, pp. 88-95. https://iris.uniroma1.it/handle/11573/1673086

[21] H. Tobita, "Comic computing: creation and communication with comic," in *Proceedings of the 29th ACM International Conference on Design of Communication*, Pisa, Italy, 2011, pp. 91-98. https://doi.org/10.1145/2038476.2038494

[22] A. M. Deshpande, "Multi-object trackers in Python," GitHub, 2020 [Online]. Available: https://github.com/adipandas/multi-object-tracker

[23] G. F. Welch, "Kalman filter," in *Computer Vision*. Cham, Switzerland: Springer, 2021, pp. 721-723. https://doi.org/10.1007/978-3-030-63416-2_716

[24] F. Anderson, "Real time, video image centroid tracker," in *Proceedings of SPIE 1304: Acquisition, Tracking, and Pointing IV*. Bellingham, WA: International Society for Optics and Photonics, 1990. https://doi.org/10.1117/12.2322200

Jinmo Yang **https://orcid.org/0009-0005-0174-3185**

He received his B.S. degree in Physics from Korea University in 2019. Since March 2024, he is with the Department of Software Communication in Hongik University as a graduate student. His current research interests include natural language processing and software quality measurement, both based on AI approach.

Kidu Kim **https://orcid.org/0009-0003-7726-1970**

He received the Ph.D. degree in Electronic and Computer Engineering from Hongik University. He is currently working at Telecommunications Technology Association. His research interests are in the areas of test maturity model, software testing, AI software testing.

R. Young Chul Kim **https://orcid.org/0000-0002-2147-5713**

He received the B.S. degree in Computer Science from Hongik University, Korea in 1985, and the Ph.D. degree in Software Engineering from the department of Computer Science, Illinois Institute of Technology (IIT), USA in 2000. He is currently a professor in Hongik University. His research interests are in the areas of test maturity model, model based testing, metamodeling, software process model, software visualization, and validation for reinforcement learning software.