
VFP290K: A Large-Scale Benchmark Dataset for Vision-based Fallen Person Detection

Jaeju An^{1*}, Jeongho Kim^{2*}, Hanbeen Lee², Jinbeom Kim², Junhyung Kang²,
Minha Kim¹, Saebyeol Shin¹, Minha Kim², Donghee Hong¹, Simon S. Woo^{123†}

¹College of Computing and Informatics, Sungkyunkwan University, South Korea

²Department of Artificial Intelligence, Sungkyunkwan University, South Korea

³Department of Applied Data Science, Sungkyunkwan University, South Korea

{anjaeju, rlawjdghek, gksqls5707, kjinb1212, gogo0920, kimminha,
toquf930, sunshine01, hdh12345, swoo}@g.skku.edu

1 Metadata Information

Our metadata for the VFP290K dataset consists of 7 definitions, including light conditions, camera heights, background, location, view, scene, occlusion, and the number of frames:

Light condition ($M_{day,night}$). Light condition is one of the most important features impacting the fallen person detection performance, as the quality of the image is highly dependent on it. We consider two light conditions, M_{day} and M_{night} . In the VFP290K dataset, it is represented 0 for M_{day} and 1 for M_{night} .

Camera height ($M_{low,high}$). Camera height is directly related to model CCTV and indoor camera environments. It is not trivial to model all the possible height requirements for different CCTV devices, as there are many different types of CCTV with varying positions. To address this issue, we film the video into several camera heights with M_{low} and M_{high} , where M_{low} is a camera height about 1 ~ 3m and M_{high} represents higher than 3m.

Background ($M_{street,park,building}$). One of the main contributions of our approach is to have numerous backgrounds, from public areas to indoor environments. We divide the background into the following three sub-categories: street, park, and building. We assign 0, 1, 2 for the street, park, and building backgrounds, respectively.

Location ($M_{location}$) & **View** (M_{view}). We identify each location for three background categories. The total number of locations for each background is 13, 30, and 6 for a street, a park, and a building, respectively.

Also, we record each view in the location by listing the videos. The viewpoints' minimum, average, and maximum numbers are 1, 2.6, and 15, respectively. We try to differ the viewpoints in a specific area, such as a playground.

Scene (M_{scene}). Along with the three background categories, we also categorize the filming locations as scenes. We enumerate each scene by each location divided by each view, composing 131 different scenes in detail. We record each scene by varying the filming view dramatically in each location.

Occlusion ($M_{occlusion}$). We specify whether the video contains an occlusion case or not, assigning 0 for the video without the case and 1 for others.

*Equal Contribution

†Corresponding author

Number of Frames (M_{frame}). We specify the number of frames for each video to control the distribution of the benchmark.

We provide the metadata in the *CSV* file format, which has 178 rows and 9 columns. The metadata with our dataset can be downloaded in the following link: <https://sites.google.com/view/dash-vfp300k/>.

2 Consent Form

Vision-based Fallen Person (VFP) Detection Study

Thank you for your information about our study! Your participation can help us create a dataset to detect fallen persons more effectively. This study requires you to take a sequence of natural actions such as roaming, standing, walking, falling, etc., under specific scenarios. And your actions will be video-taped. Your participation will take less than 6 mins.

Eligibility. You must be aged 18 or older to participate.

Study procedure. You are invited to participate in a research study conducted by Professor Simon Woo at Sungkyunkwan University, South Korea. This form explains information about our study. You should ask questions about anything that is unclear to you (see Contact Information below). Your participation is voluntary. And your actions will be video-taped.

Potential Risks and Discomforts. There is minimal risk to you from feeling discomfort. You are asked not to take actions that discomfort you.

Alternatives to Participation. Your alternative is not to participate in this study; if you are an SKKU student, your grades will not be affected, whether or not you participate in this study.

Confidentiality. Sungkyunkwan University’s Human Subjects Protection Program (HSPP) reviews and monitors research studies to protect the rights and welfare of research subjects. We will protect your privacy in the following way:

1. Your personal contact information will only be stored on our server and will be deleted after the study.
2. You have a right at any time to request this data to be removed from our server by sending an email to the Principal Investigator at swoo@g.skku.edu.
When the results of the study are published or discussed in conferences, no identifiable information will be used. We will list our publications and publications of any researchers who use this data on our project page.

Potential Benefits to Participants and/or to Society. You may not directly benefit from your participation in this study; however, you may also learn how to detect different user activities. Researchers hope this new, promising approach to obtain new datasets that can greatly improve the existing fallen person detection system.

Participation and Withdrawal. Your participation is voluntary. Your refusal to participate will involve no penalty or loss of benefits to which you are otherwise entitled. You may withdraw your participation at any time and discontinue without penalty. You are not waiving any legal claims, rights, or remedies because of your participation in this research study.

Compensation. You will not be compensated for participation. Your participation is voluntary.

3 Occlusion Scenarios

We try to include rich occluded instances in terms of the fall situations. In our VFP290K dataset, there are realistic occlusion and overlap cases as following:

- Case 1: Fallen person occluded by general objects, such as chairs or cars.

- Case 2: Fallen person occluded by buildings.
- Case 3: Fallen person occluded by being out of frame.
- Case 4: Fallen person overlapped with people around the subject.
- Case 5: Fallen person overlapped with another fallen person.
- Case 6: Fallen person overlapped or occluded with the combination of the above situations.

We present visual examples for the above cases in Fig. 1.



Figure 1: Example images for occlusion cases we considered. The red and green bounding box indicates a normal person and a fallen person, respectively.

4 Annotation Instructions

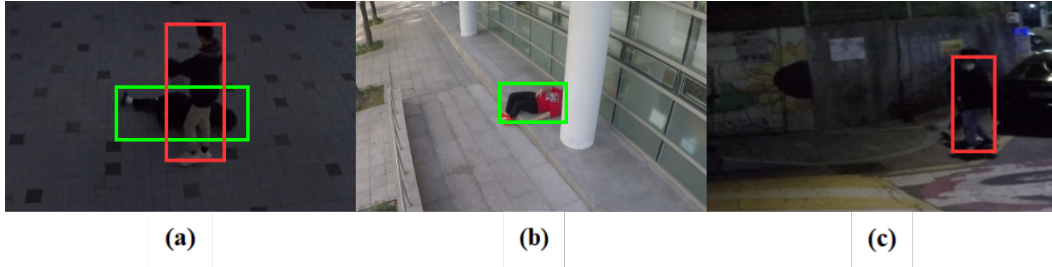


Figure 2: Example images for occlusion labeling rule and overlap labeling rule. In (a), the entire body of a fallen person is annotated in the green box, but note that only the visible part is annotated in (b). In (c), there are two overlapped people, but only a front person is labeled.

Please annotate the images assigned to you on the LabelImg program. Since there are many various situations, you have to familiarize yourself with the below rules and carefully label all the objects if they corresponded. After the verification process, we will give you edits on video by video if your works have some problems.

Unlike other detection datasets, we focus on a person. You have to assign “class 1” to the bounding box of a fallen person and “class 0” to that of a non-fallen person for a binary classification task.

Rule 1. Occlusion Labeling Rule. When the person is occluded but exposes any part of his or her body, annotate the entire body, referring to Fig. 2.(a). When the overall part of the person is occluded, annotate the only visible part of the body, referring to Fig. 2.(b).

Rule 2. Overlap Labeling Rule. In the case that the two bounding boxes from normal and fallen persons are overlapped entirely, and the behind box is in the front box, you do not have to annotate behind the box, referring to Fig. 2.(c).

5 Anonymization (De-identification) Process

We first anonymize all personally identifiable information such as license plates, the face of pedestrians. For anonymization, we crop the interested region and downsample it five times to apply mosaic followed by blurring. Then, we upsample the processed cropped image by five times and attach it to the original position.

For visualizing the candidate cases, we depicted some examples in Fig. 3. For license plates, we annotated all of them and conducted the de-identification process. For identifiable people, we annotate the face and conduct the process except for people who cannot be recognizable due to their wearing, such as masks or hats. Also, we count non-recognizable in two cases: one is an appearance from behind, the other is a tiny object which looks like a low-resolution image.



Figure 3: Example images for de-identification. (a) and (b) are the images after applying the de-identification process, as they are identifiable. However, (c) is the image that is not applied de-identification, as it cannot be recognizable.

6 Visualization Setting for t-SNE

For visualization, we employed an ImageNet-pretrained VGG-19 [14] network for embedding an image into a vector space and used t-Stochastic Neighbor Embedding (t-SNE) [15] for dimension reduction. We sample 2,000 images from ours as well as MultiCam [1] and the dataset proposed by Adhikari et al. [9]. The sampled images are then embedded into a feature vector of 1,000 lengths by VGG-19. The embedded feature vector is reduced to two dimensions by using t-SNE. In our dataset, we use the two pairs based on light condition and camera height. We did not use any modification on our dataset, such as normalization or resizing, to preserve the original characteristics of ours.

7 Baseline model & Configuration

Faster R-CNN [13]. Faster R-CNN is a two-stage detector. It proposes to use Region Proposal Networks (RPNs) to share convolution layers with object detection networks so that Faster R-CNN can achieve faster training based on an end-to-end structure than Fast R-CNN [5].

Cascade R-CNN [2]. Cascade R-CNN is a two-stage detector using several classifiers based on Faster R-CNN. The classifiers are employed progressively to conduct new classification tasks by receiving the bounding boxes generated by the previous classifier. They assume that the bounding boxes created at each step are more accurate than the previous and that the subsequent classifier optimizes performance by learning on a higher IoU than the prior one.

DetectoRS [10]. DetectoRS, a two-stage detector, introduces Recursive Feature Pyramid (RFP) and Switchable Atrous Convolution (SAC) to improve the detection performance.

Table 1: Detailed hyper-parameter settings and configurations of our experiments. Faster R-CNN and Cascade R-CNN are abbreviated as F R-CNN and C R-CNN, respectively. In the inference phase, we use the best epoch obtained from validation performance.

| Model | Optimizer | Learning Rate | Momentum | Weight Decay | Epoch | Schedules | | | |
|-----------|-----------|---------------|----------|--------------|-------|-----------------|--------|------------------|--------------|
| | | | | | | Policy | Warmup | Warmup Iteration | Warmup Ratio |
| F R-CNN | SGD | 0.02 | 0.9 | 0.0001 | 12 | step=[8, 11] | linear | 500 | 0.001 |
| C R-CNN | SGD | 0.02 | 0.9 | 0.0001 | 12 | step=[8, 11] | linear | 500 | 0.001 |
| DetectoRS | SGD | 0.02 | 0.9 | 0.0001 | 12 | step=[8, 11] | linear | 500 | 0.001 |
| RetinaNet | SGD | 0.01 | 0.9 | 0.0001 | 12 | step=[8, 11] | linear | 500 | 0.001 |
| YOLO3 | SGD | 0.001 | 0.9 | 0.0005 | 273 | step=[218, 246] | linear | 2000 | 0.1 |
| YOLO5 | SGD | 0.01 | 0.937 | 0.0005 | 100 | - | linear | 1000 | - |
| DETR | AdamW | 0.0001 | - | 0.0001 | 150 | step=[100] | - | - | - |

RFP enhances its image representation power through a recursive structure that looks at images twice or more. SAC performs convolution with different atrous rates for the same feature and combines the results with the switch function.

RetinaNet [7]. RetinaNet is a one-stage object detection model that handles class imbalance issues. They introduce focal loss, which uses a modulating term to the cross-entropy loss for learning on hard negative samples. RetinaNet is a unified network made up of a backbone and two task-specific subnetworks.

YOLOv3 [11]. YOLOv3 (You Only Look Once, Version 3) is a one-stage object detection model based on YOLO [12]. It works in real-time and identifies specific objects in videos, live feeds, or images. YOLOv3 predicts a bounding box with confidence using logistic regression, performing more accurately than previous versions.

YOLOv5 [6]. YOLOv5 (You Only Look Once, Version 5) is a one-stage object detection model following the basics of the existing YOLO [12]. Cross Stage Partial (CSP) network [16] and Path Aggregation Network (PANet) [8] are used for the backbone and neck network of the model. This model is implemented with a PyTorch framework and includes five different model sizes.

DETR [3]. DETR is the first to employ transformer architecture successfully on an object detection task. It learns a two-dimensional representation of an input image using a standard CNN backbone network. The model flattens the data and adds a positional encoding before passing it to a transformer encoder. Following that, a transformer decoder receives a fixed number of learned positional embeddings as input and attends to the encoder output. It cannot be classified one or two-stage model, as it does not use a bounding box anchor.

Hardware Configurations. We used Intel XEON Gold 6230 2.1GHz CPU with NVIDIA RTX 3090 24GB, CUDA v11.1, and cuDNN v7.6.5 for GPU usage.

Model Configurations. We conduct our experiments based on MMDetection toolbox [4] (for 6 models except YOLOv5) and official code³ for once. For the detailed configuration of each model, we use default settings that is summarized in Table 1.

8 Performance of Anomalous Event Detection

Table 2: Experimental results for anomalous behavior detection on the VFP290K benchmark dataset. We conduct this experiment with promising models on our dataset: Cascade R-CNN, and RetinaNet. One-class indicates training fallen people only while binary-class indicates training both fallen and non-fallen people. Furthermore, bold text indicates better performance.

| Model | Cascade R-CNN | | RetinaNet | |
|-------|---------------|--------------|-----------|--------------|
| | One-class | Binary-class | One-class | Binary-class |
| mAP | 73.8 | 75.1 | 72.9 | 75.0 |
| AP50 | 88.2 | 87.4 | 90.5 | 91.0 |
| AP75 | 79.8 | 81.1 | 78.7 | 81.1 |

³<https://github.com/ultralytics/yolov5>

Detecting fallen persons can be considered as a one-class problem in the form of anomalous event detection. In this case, we can define a fallen situation as an anomalous event. Thus, a detection model learns the features related to only fallen person. We conduct the anomalous event detection experiment to verify which method produces more desirable performance, training fallen only (one-class) or training both fallen and non-fallen (binary-class). The experimental setting is the same as the benchmark experiment of our VFP290K, and we use Cascade R-CNN and RetinaNet that show the best performance in the two-stage and one-stage models based on the benchmark experiment.

As presented in Table 2, Cascade R-CNN achieved 73.8, and RetinaNet achieved 72.9 mAP on one-class detection, respectively. It is slightly underperforming compared to the binary-class detection. Since there is a dependency between fallen and non-fallen persons, binary-class is more suitable for detecting fallen persons than one-class. This experiment showed that our dataset can also be used for anomalous event detection.

References

- [1] E. Auvinet, C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. Multiple cameras fall dataset. *DIRO-Université de Montréal, Tech. Rep*, 1350, 2010.
- [2] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.
- [3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.
- [4] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.
- [5] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
- [6] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, A. Hogan, lorenzomammama, yxNONG, AlexWang1900, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, F. Ingham, Frederik, Guilhen, Hatovix, J. Poznanski, J. Fang, L. Y. 于力, changyu98, M. Wang, N. Gupta, O. Akhtar, PetrDvoracek, and P. Rai. ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements, Oct. 2020. URL <https://doi.org/10.5281/zenodo.4154370>.
- [7] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
- [8] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018.
- [9] G. Mastorakis and D. Makris. Fall detection system using kinect’s infrared sensor. *Journal of Real-Time Image Processing*, 9(4):635–646, 2014.
- [10] S. Qiao, L.-C. Chen, and A. Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10213–10224, 2021.
- [11] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.
- [12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

- [13] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28: 91–99, 2015.
- [14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.
- [15] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
- [16] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh. Cspnet: A new backbone that can enhance learning capability of cnn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 390–391, 2020.