Medicon Engineering Themes
Volume 8 Issue 5 May 2025
Article Type: Research Article
ISSN: 2834-7218

Risk Analysis for Fall Detection: Exploiting using GAIT, Part Affinity Field and Machine Learning

Jyoti Patil Devaji*, Rajeshwari Mattimani, Sushma Garawad and Suneeta VB

Electronics and Communication, KLE Technological University, Hubli, India

***Corresponding Author:** Jyoti Patil Devaji, Electronics and Communication, KLE Technological University, Hubli, India.

Received: March 05, 2025; **Published:** May 19, 2025

Abstract

According to our survey, the primary cause of injury or death for people around 60 years are from falls [1, 2]. It is estimated that around 35percent of elder get injured by falling every year, scientifically, it is proven that the Reason for falling is because of an imbalance of the center of gravity, i.e the CG (center). of gravity) of the person is unstable. It is necessary to find fast and effective way to find fall detection to help the elderly fall. This fall detection can also be used for patients in hospitals and in road traffic for drunk people. In this paper, we are going to detect the fall of a person using an open pose model. Basically, we are going to extract the key point/joints of a person and then we are going to analyze the gait of a person by taking the centroid as a key point for fall detection. This method is effectively able to detect falls with an accuracy of 90 percent.

Keywords: fall detection; open pose; center of gravity; key points

Introduction

A Gait is a pattern of limb movements made during locomotion i.e while walking or running. Human giat are the various ways in which a human can move naturally or by specialized training. Based on the movement of the limbs doctors can analyze the gait of a person and diagnose problems like neurological or musculoskeletal problems. 'Gait analysis' is a method for identifying abnormalities in the gait cycle, i.e. the way a person walks. A gait cycle can be defined into two phases first is swing phase and the second is stance phase. The stance phase begins when the foot first touches the ground through a 'heel-strike' and comprises of all activities till the same foot leaves the ground through a 'toe-off'. It involves 'Initial Double Stance, Initial Single Stance', 'Mid stance', 'Terminal Single Stance' and 'Final Double Stance' sub stages, all this activity covers 60 percent of the gait cycle. On the other hand, swing phase covers for remaining 40 percent of gait cycle occurs when the foot is not in contact with the ground and comprises 'Initial Swing', 'Mid Swing', 'Terminal Swing'[3].

A confidence score then via Part Affinity fields we associate the detected key points to form the skeleton of a person [4]. After extracting the key points we use the center of the hip joint as reference to detect the fall detection. We are using 3 main conditions to detect the person falling [8]. First, If the velocity of descent is greater than critical value. Second, we are going to check the angle between the ground and head of a person is less than 45 percentage. Third width to height of the body's external rectangular is greater than 1. If all these 3 conditions are satisfied then we are displaying the output as fall detected.

Related Work

Gait Analysis using Image Processing

In this model they have created an algorithm that takes a video file as input and outputs the step length and heel points for each frame. At first background subtraction is performed to isolate the person from the background for gait analysis. They have used K-nearest neighbors background subtraction method [9]. After background subtraction, the binary mask is used for contour tracking. The contour with five largest areas is used to create the rectangular bounding box. Later, Different Features are extracted like a) Feet location, the bounding box defines the toe of foot and heel of back foot b) Stride length, the difference in distance between the bounding box of consecutive frames gives the stride length c) Stride length, is the time required to completely achieve stride length once. d) Heal-strike, toe-off points, the temporal plot of width can be used for analyzing heel-strike and toe- off time points [3]. Fig. 1 shows the result of Gait Cycle.

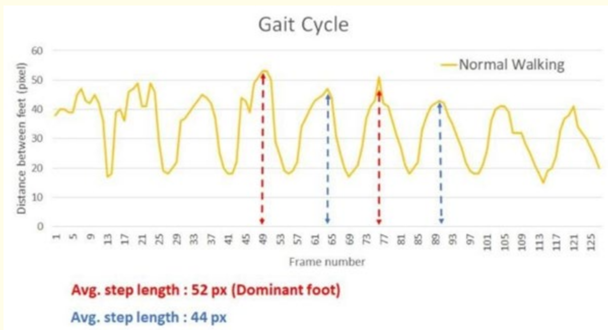

Figure 1: Gait pose.

The above model was experimented for gait analysis to detect abnormal gait by studying the crust and troughs of a heel strike. It was tested on subjects like a person dragging his leg while walking, a person falling down with a precursor of drifted apart gait. Apart from abnormality it is also used to detect types of gait like subject jogging, marching, detect dominant foot in normal gait [3].

OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields

In this paper they have discussed multi-person pose estimation. There are 2 ways to approach pose estimation module, first is top-down model, where we first identify the person by removing the background noise and drawing the bounding box for the subject and then identify the key points of the subject (person), but suffers from early commitment i.e if people in the image are close in proximity then there is no recourse to recovery. The second model is a bottom up approach [10, 11] which is robust and can identify key points of a person even in occlusion. In this model we first identify the key points and then match the points by bipartite matching. In this model they have used 10 layers of VGG16 model [12] for feature extraction and then the extracted features are passed to a convolutional Neural Network where we first predict the confidence score of the key points. After getting the confidence score we perform part affinity field for part association (PAF). PAF gives the 2D vector fields that maps the location and orientation of body parts in an image. After gathering the 2D vector field from PAF bipartite matching is done to associate body parts of the candidates to obtain full body pose. This model has been evaluated on three different dataset 1) MPII human multi-person data set 2) COCO key point challenge dataset. 3) foot dataset.

Fall Detection Based on Key Points of Human-Skelet Using Open Pose

In this paper, they have designed a model to detect fa detection by analyzing the key points of a person using open pose [8]. So the basic step was to extract the features of the human body, so in order to extract the features they have used an open pose model which determines the posture of the body. Once the postures are recorded they are subjected to 3 basic decision conditions 1) whether the velocity of descent at the center of hip joints is greater than the critical velocity, 2) whether the angle between the ground and the head of a person is less than 45 degree, 3) width to height ratio of the bounding box is greater than 1. If all these 3 basic conditions are satisfied then we display an output that human fall occurred or else if any one of the conditions gets failed then we check for any fall detection in the next frame. The formula used to calculate the speed of descent of a person with a hip joint as the center is

$$V = \frac{yt2 - yt1}{t} \qquad (1)$$

where v is the change in velocity, yt1 and yt2 is the change in y-coordinates at frame at time t1 and t2 and t is the difference in time at time t2 and t1. 0 = V < v' 1 = V >= V'.

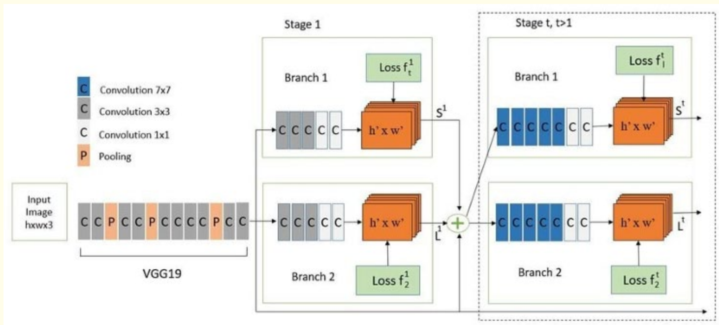

Figure 2: Architecture of open-pose.

where v' is the threshold falling speed of the hip joint center. If v >= v then we conclude then fall detection occurs. The formula to calculate angle between the ground and head of a person is:

$$arctan = \frac{Yh - Yl}{Xh - Xl} \qquad (2)$$

which gives the angle between head and leg.

Methodology

Open Pose

The Open pose is an open-source library that uses convolutional neural networks and supervised learning and based on caffe. Open pose can detect human body parts in a single image [13-21]. Open pose uses a bottom-up approach to detect the position of the hot spot map of the human body.

In the open pose the first step is to extract the features from the image. For the feature extraction we are going to use the first 10 layers VGG-19 [22]. The extracted features are then passed into two parallel branches of convolutional layers. The first branch predicts confidence maps of a person which identifies the set of 18-key points of the human. The second branch predicts Part Affinity Fields

(PAFs) which associates the different body parts of the human. As the final part, we take the output of Confidence Maps and Part Affinity Fields to obtain the pose of a person by applying the greedy bipartite matching algorithm.

At stage 1 Part Affinity Fields are predicted from the feature map of the base network. In subsequent stages it uses the output from the previous stage and the Features of image F to produce refined predictions

$[L^1 = (F)]$ (1)

$$[L^t = (F, Lt1), A2 <= t <= TP] \quad (2) \quad (3)$$

Where: L1 = Part Affinity field at stage 1.

1 = CNNs at Stage 1 t = CNNs at Stage t TP = Total number of PAF stages.

At stage 2 it uses the output of part affinity fields and predicts the confidence maps of the person.

$$STp = \mathrm{t}(F, LT)t, At = TP \quad (3)$$

$$St = t(F, LT, St1), ATP < t <= TP + TC \quad (4)$$

Where: t=CNNs at Stage t TC=total number of confidence map stages.

We apply the L2 loss function at the end of each stage to effectively predict the part affinity field at the first branch and confidence map at the second branch, as there could be some data inconsistency in a dataset. The loss function of Part affinity field branch at stage ti and loss function of the confidence map branch at stage Tk are:

$$f^t il = \sum c = 1 \sum pW(p).||Lti(p)L(p)||^2 2, \quad (5)$$

$$f^t sk = \sum j = 1 \sum pW(p).||Stk(p)S(p)||^2 2 \quad (6)$$

Where L∗c = ground truth PAFsS = ground truth of part confidence map W is a binary mask with ΣW (p) = 0 to prevent the extra loss when an annotation is missing at the pixelp* [23]. *The final equation would be* $\sum f = TpTP \sum + TC \sum f + t = \sum Tp + 1f$ (7) The final inference of confidence maps and PAFs is passed to the greedy algorithm for body part association]The first step is to extract the features from the image using the first 10 layers of VGG-19 model, which is the base network F.

At stage 1 Part Affinity Fields are predicted from the feature map of the base network. In subsequent stages it uses the output from the previous stage and the Features of image F to produce refined predictions

$[L^1 = (F)]$(1)

$$[L^t = (F, Lt1), A2 <= t <= TP](2) \quad (4)$$

Where: L1 = Part Affinity field at stage 1.

1 = CNNs at Stage 1 t = CNNs at Stage t TP = Total number of PAF stages.

At stage 2 it uses the output of part affinity fields and predicts the confidence maps of the person.

$STp = t(F, LT)t, At = TP$ (3)

$St = t(F, LT, St1), ATP < t <= TP + TC$ (4)

Where: t=CNNs at Stage t TC=total number of confidence map stages.

We apply the L2 loss function at the end of each stage to effectively predict the part affinity field at the first branch and confidence map at the second branch, as there could be some data inconsistency in a dataset. The loss function of Part affinity field branch at stage ti and loss function of the confidence map branch at stage Tk are:

$$f^{t}il = \Sigma\ c = 1\ \Sigma\ pW\ (p).||Lti(p)L(p)||^{2}2, \quad (5)$$

$$f^{t}sk = \Sigma\ j = 1\ \Sigma\ pW\ (p).||Stk(p)S(p)||^{2}2 \quad (6)$$

Where L∗c = ground truth P AFsS∗ = ground truth of part confidence map W is a binary mask with ΣW (p) = 0 to prevent the extra loss when annotation is missing at the pixelp [23]. *The final equation would be* $\Sigma f = TpTP\ \Sigma + TC\ \ \Sigma f + t = \Sigma\ Tp + 1f$ (7) The final inference of confidence maps and PAFs is passed to the greedy algorithm for body part association

Confidence Maps

We generate confidence maps $S{*}jk$ for each person at location p Belongs to R where k is the number of person. Xj, $kbelongtoR$ is the ground-truth position of body part j for each person k in the image.

$s * jkexp(\frac{-||p-xj,k||^2}{\sigma^2}$ (8) The above equation is a Gaussian curve which is an aggregation of the individual confidence maps with gradual changes where sigma controls the spread of the peak.

$$s * j(P) = maxSj, k * (\mathrm{P}) \quad (9)$$

Part Affinity Field

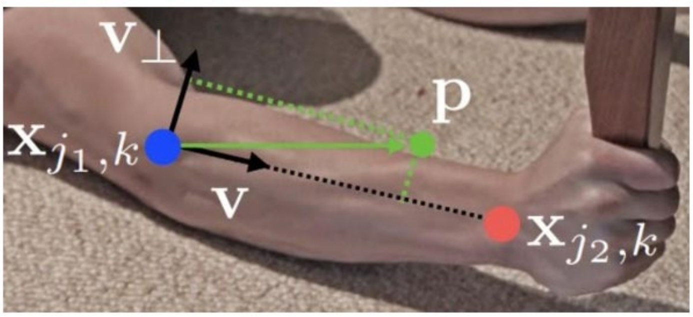

Figure 3

The part affinity field is used to map the body parts of a person to its body. Consider a arm shown in Fig. 3. Let Xj 2,k and Xj 2,k be the ground-truth location of body parts j1 and j2. If a point p lies on the arm then LC, would be a unit vector that points from j1 to j2, and for all other points, it would be zero-valued vector. While training, we define the ground-truth PAF, Lc, as $Lck * (p) = vi\ f\ Ponlimbc$, $k(10)$ *0 other wise* here v is the unit vector in the direction of the arm While testing, we perform line integral over the corresponding PAFs along the line segment of connecting body parts to measure the association between detected body parts.

$$v = \frac{Xj2,k-Xj1,k}{||Xj2,kXj1,k||2}(11)$$

$E = \Sigma\ u = 0,\ 1Lp(u)).\ \frac{dj2-dj1}{||dj2-dj1||2}du$ where, p(u) interpolates the position of the two body parts dj 1 and dj 2 $p(u) = (1 - u).dj2 + u.dj2$.

Fall detection

Once the postures are recorded from open pose they are subjected to 2 basic decision conditions 1) whether the velocity of descent at the center of hip joints is greater than the critical velocity, 2) whether the angle between the ground and head of a person is less than 45 degree. The procedure of implementation of our proposed approach is shown Fig 4. The Fig 5 gives the joint number of the body parts which are required to analyze gait for fall detection. Condition 1 (the speed of Descent of the body [8]): As the process of sudden fall leads to drastic change in center of gravity of a person in the vertical direction, we consider human hips as the center point and analyze the y coordinates of the hips for every five adjacent frames. The data points are 8(right hip), 11(left hip). The coordinates of the hips are s8(t) = (x8, y8) and S1(t) = (X11, Y11). Let y-coordinates of the hip joint at time *t1* be *yt1* = $\frac{Y8+\dot{y}11}{2}$ and y-coordinate at time t2 be Y t2 = $\frac{Y8+\dot{y}11}{2}$. So we can get the descent velocity of the hip joint by T = t2 - t1. V = $\frac{Yt2-Yt1}{T}$ When V is greater than critical velocity then we can say that fall is detected V>-= V' Fall detected V<V' safe (13) Where v is critical velocity.

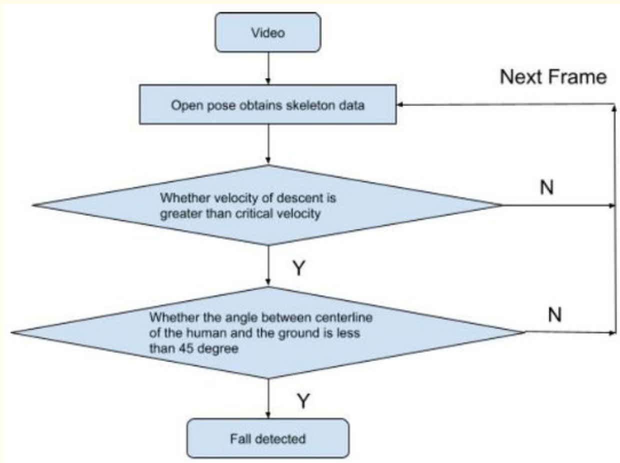

Figure 4: flow chart.

Condition 2 (the angle between the ground and head of a person [8]): While falling, the most common feature would be change in angle of the body with respect to ground. So, we calculate the fall with respect to body points head and ankle. When the angle between these 2 increases greater than 45 degree then we detect the fall. The data points are 0(head), 10(right ankle), 13(left ankle). The coordinates are s0(t) = $(x0, y0)$, s10 = $(x10, y10)$, s13 = $(x13, y13)$. SoS' = $\frac{s10+s13}{2}$, S'(t) = (X', Y') at time t, the angle between the head of a person and the ground is arctan = $\frac{Yh-Yl}{Xh-Xl}$ If the angle is less than 45 degree then fall is detected.

Points which are present in a human body are obtained as a part of final output as depicted in figure 8. On eliminating the background of the final output the key points still remain The is the output of the fall detection video. The model predicts the fall detection with the accuracy of 90 percent. This model has been tested on 20 videos and successfully predicts the fall detection with 90 percent accuracy.

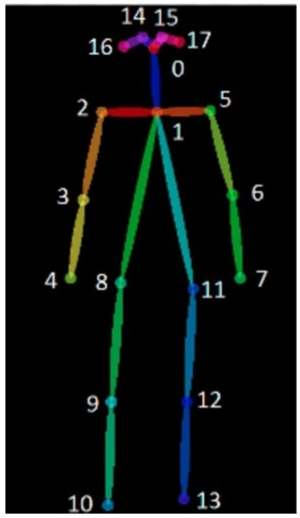

Figure 5: key points.

Conclusion

In this paper we have discussed methodology to detect fall detection using pose estimation. At first we are going to detect the key body parts of a human using an open pose open-source library. For detection we first extract the features of an image by using 10 layers of VGG16. Later we are going to utilize the extracted features to identify confidence scores to detect the key features and Part Affinity fields to associate the key points. After getting the human pose skeleton we are going to analyze the gait of a person for a fall. So, to identify fall we are checking 2 basic conditions.

First fall of descent and second is angle of a human with respect to ground. If both the conditions are satisfied then we are going to display an output as fall detected. As we don't have any datasets of fall, we have tested our model on a few videos and compared with different algorithms like Droghini et al. [24] which detects the fall by capturing sound waves. Shahzad et al. [25] which uses sensors in smartphones. Kepski et al. [26] uses microwave doppler sensors. Quadros et al. [27], the threshold method and machine learning are used to identify falls. Open Pose [28, 29] can be used to identify the images based fall detection as it is fast and convenient.

S. No	*Average Accuracy in percentage*	
	BODY Parts	*Accuracy*
1	0	0.81
2	1	0.88
3	2	0.95
4	3	0.87
5	4	0.92
6	5	0.87
7	6	0.71
8	7	0.84
9	8	0.83

10	9	0.75
11	10	0.86
12	11	0.75
13	12	0.82
14	13	0.87
15	14	0.89
16	15	0.93
17	16	0.86
17	17	0.71

Table 1: Accuracy of Each Key Point.

S. No	*Algorithms*	*Accuracy*
1	Mel-Frequency Cepstral Coefficient + SVM	Mel-Frequency Cepstral Coefficient + SVM
2	Threshold + SVM	91.7-97.8
3	Microwave Doppler + Markov model	95
4	Threshold + Madgwick's decomposition	91.1
5	openPose + LSTM	88.7
6	openPose + two thresholds	90

Table 2: Comparison of other Fall Detection Algorithms with Ours.

Figure 6: recognized image.

Citation: Jyoti Patil Devaji., et al. "Risk Analysis for Fall Detection: Exploiting using GAIT, Part Affinity Field and Machine Learning". Medicon Engineering Themes 8.5 (2025): 16-25.

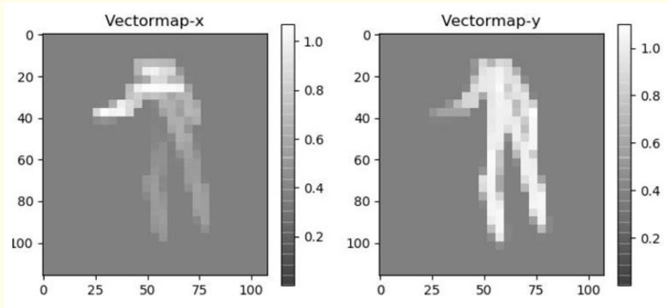

Figure 7

Figure 8: Fall detected images.

Acknowledgment

The work is supported by KLE Technological University Hubballi-580024.

References

1. Taylor L Christiansen., et al. “Patient activation related to fall prevention: a multisite study”. Jt Comm J Qual Patient Saf 46.3 (2020): 129-135.
2. US Preventive Services Task Force; David C Grossman., et al. “Interventions to prevent falls in community-dwelling older adults: Us preventive services task force recommendation statement”. JAMA 319.16 (2018): 1696-1704.
3. S Kulkarni and S Gupta. “Gait analysis using image processing”. 1 (2020): 1-5.
4. Z Cao., et al. “Openpose: Realtime multi-person 2d pose estimation using part affinity fields”. IEEE Transactions on Pattern Analysis and Machine Intelligence 43.1 (2021): 172-186.
5. Weiming. “Fall detection using pose estimation”.
6. KDJ Newell and A Yang. “Stacked hourglass networks for human pose estimation”. In Proceedings of the Computer Vision—14th European Conference, Amsterdam, The Netherlands (2016): 483-499.
7. E Pishchulin., et al. “A deeper, stronger2016, and faster multi-person pose estimation model”. Proceedings of the European Conference on Computer Vision, Munich, German (2018): 34-50.

8. W Chen., et al. “Fall detection based on key points of human-skeleton using openpose”. (2020): 1-17.
9. Mario Nieto-Hidalgo., et al “Gait analysis using computer vision based on cloud platform and mobile device”. 1 (2018): 10.
10. Eldar Insafutdinov., et al “Deepcut: Joint subset partition and labeling for multi person pose estimation”. 1.1 (2016): 10.
11. Bjoern Andres., et al “Deepercut: A deeper, stronger, and faster multi-person pose estimation model”. 1 (2016): 10.
12. K Simonyan and A Zisserman. “Very deep convolutional networks for largescale image recognition”. ICLR (2015).
13. Y Fu., et al “Pose-normalized image generation for person reidentification”. ECCV (2018).
14. DRA Bansal, S Ma and Y Sheikh. “Recycle-gan: Unsupervised video retargeting”. ECCV (2018).
15. TZC Chan, S Ginosar and AA Efros. “Everybody dance now”. ECCV (2018).
16. MM Veloso., et al “Teaching robots to predict human motion”. IROS (2018).
17. IOP Panteleris and A Argyros. “Using a single rgb frame for real time 3d hand pose estimation in the wild”. IEEE WACV (2018): 436-445.
18. TSH Joo and CY Sheikh. “Total capture: A 3d deformation model for tracking faces, hands, and bodies”. CVPR (2018).
19. C Theobalt., et al “Vnect: Real-time 3d human pose estimation with a single rgb camera”. ACM TOG (2017).
20. G Bradski. “The opencv library”. ACM TOG (2000).
21. Z Simon., et al “Realtime multi-person 2d pose estimation using part anity fields”. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA (2017): 7291-7299.
22. AZ Karen Simonyan. “Very deep convolutional networks for largescale image recognition”. 6 (2018): 10.
23. Y Sheikh. “Convolutional pose machines”. CVPR (2016).
24. Anna Esposito., et al “Multidisciplinary approaches to neural computing”. (2018): 97-107.
25. A Shahzad and K Kim. “Falldroid: An automated smart-phone-based fall detection system using multiple kernel learning”. IEEE Transactions on Industrial Informatics 15.1 (2019): 35-44.
26. K Shiba, T Kaburagi and Y. Kurihara. “Fall detection utilizing frequency distribution trajectory by microwave doppler sensor”. IEEE Sensors Journal 17.22 (2017): 7561-7568.
27. T de Quadros, AE Lazzaretti and FK Schneider. “A movement decomposition and machine learning-based fall detection system using wrist wearable device”. IEEE Sensors Journal 18.12 (2018): 5082- 5089.
28. S Kang., et al. “Human-skeleton based fall-detection method using LSTM for manufacturing industries”. 18.1 (2019): 1-4.
29. Qingzhen Xu., et al. “Fall prediction based on key points of human bones”. 540.22 (2015): 540.
30. Dehzangi O., Taherisadr M and ChangalVala R. “IMU-based gait recognition using convolutional neural networks and multi-sensor fusion”. Sensors 12 (2017): 2735.

Volume 8 Issue 5 May 2025