Fall Alert: A Novel Approach To Fall Detection At Night

Kiran Kamble, Himanshu Donadkar, Ritik Poshattiwar and Abhinavram Ananth
Department of Computer Science and Engineering,
Walchand College of Engineering, Sangli, India
kirankamble5065@gmail.com,
hdhimanshu123@gmail.com
ritik.poshattiwar@gmail.com
ableabhinav@gmail.com

1. Abstract

In this paper we present a novel approach to detect irrecoverable fall of a person especially during night by leveraging CNN based object recognition and image processing techniques. We use the YOLO (You Only Look Once) object detection algorithm trained on the COCO dataset, to detect person from an input video stream, and find whether the person has succumbed to an irrecoverable fall by testing for fall over multiple frames. We begin by introducing the core idea behind our work and its importance followed by a brief review of previously done work in this regard. We then present various approaches to solve our objective using various possible solutions and discuss its results, followed by a solution that we implemented which gave appreciable results. The goal is to implement a solution to detect irrecoverable fall of security/army personnel at night, thus helping enhance security.

Keywords: Fall Detection, YOLO-COCO, Deep Learning, Image processing

2. Introduction

Computer Vision and Image Processing techniques have proven to give great solutions to various problems in different sectors like Retail, Security, Healthcare, Banking and Industry Automation. Problems related to Activities of Daily Living (ADL) have come to prominence that can be solved using Computer Vision and Image processing. By tracking various ADL movements like sitting, walking and falling, we can create solutions for various use-cases. One such use-case is Fall Detection of person in various environments. Such a use-case is very beneficial for various cohorts, and one such being Army and Security forces. Armed forces are at continuous vigil at places of importance such as military bases as well as secret safe-houses and hideouts. At such places, especially during night times, when activity in the surrounding as well as visibility of the environment is relatively low, enemy can strike the personnel in silence and stealth. Hence, there is a need to monitor whether the armed forces guarding the place, have encountered an irrecoverable fall from such strikes of the enemy, and take subsequent measures. Our objective is hence to detect and alert irrecoverable fall of armed forces/security personnel at night. Additionally, keeping the above discussion in mind, it is hence essential to continue advancing in the field of computer vision and image processing so as to find novel and accurate solutions to problems which were earlier not solvable using traditional methodologies like the ones stated above.

3. Literature Review

There has been some amount of research and work done in this regard as of now. Earlier machine learning practices were being used which mostly used input data from Microsoft's Kinect sensor. Other type of methodology involved detection of fall by fitness band worn by the person. As only the people wearing band can be checked for fall, we didn't focus on this methodology.

3.1 Classification of Human Fall from Activities of Daily Life using Joint Measurements[1]

This Paper uses Kinect for capturing depth images at a frame rate of 30 FPS, with Microsoft SDK v1.7. The methodology applied in this system uses the floor plane equation and the joint coordinates from the skeleton data generated by the Kinect runtime. These data are then used to compute the velocity & acceleration of the body, the distance between the head to floor plane and the position of the other joints to identify an unintentional fall movement from other activities of daily life. It uses the Kinect infrared sensor.

In this paper a human fall detection system using depth images generated by the Kinect infrared sensor. The experimental results show that the algorithm used on the system can accurately distinguish fall movements from other daily activities with an average accuracy of 94.43%.

3.2 Human Fall Detection from Depth Images Using Position and Velocity of Subject[2]
The fall detection algorithm in the proposed system uses depth information from Microsoft Kinect Sensor to compute velocity and position of the subject. These data from each consecutive frame are used to compute the velocity of the body to identify any abnormal activity continuously within the view of the Sensor.

3.3 Development of Human Fall Detection System using Joint Height, Joint Velocity and Joint Position from Depth Maps[3]
In this paper, fall detection is accomplished using both the velocity of head, the distance from head to floor and position of other joints. The system will confirm a fall if the subject remains on the floor without any movement for 5 seconds and then a fall alarm will be generated. In such cases the changes of distance from head to floor possess similar pattern except the time it takes. They used joint coordinates which were extracted from every consecutive frame to calculate the velocity.
Their experimental results showed that the algorithm used on the system can accurately distinguish fall movements from other daily activities with an average accuracy of 96.55%. Their system was also able to gain a sensitivity of 100% with a specificity of 95%. Their proposed system was able to distinguish all fall movements.

3.4 To summarize
With a review of previously done works, we wanted to build a system which won't require any special kind of sensors and expensive hardware. Rather it should detect fall from input given by a Night Vision CCTV camera / IR camera, which are generally used for surveillance purposes.

4. Our Implementation

4.1 Initial Approach
Our initial approach involved training a custom model to detect a person falling or not. For this we tried to find datasets of people falling, walking, etc. in order to train model on custom object. But we didn't get any proper dataset to work on. Hence, we created our own dataset which consisted of images of persons lying and walking in different postures and positions. Then we created our own model using Faster RCNN and YOLO. But the results we obtained were unsatisfactory. This happened because we could not bring much diversity in environment of the dataset. Also, custom object training models are fairly less accurate than the pretrained models. Therefore, we decided to use a pretrained model followed by some image processing to detect fall.

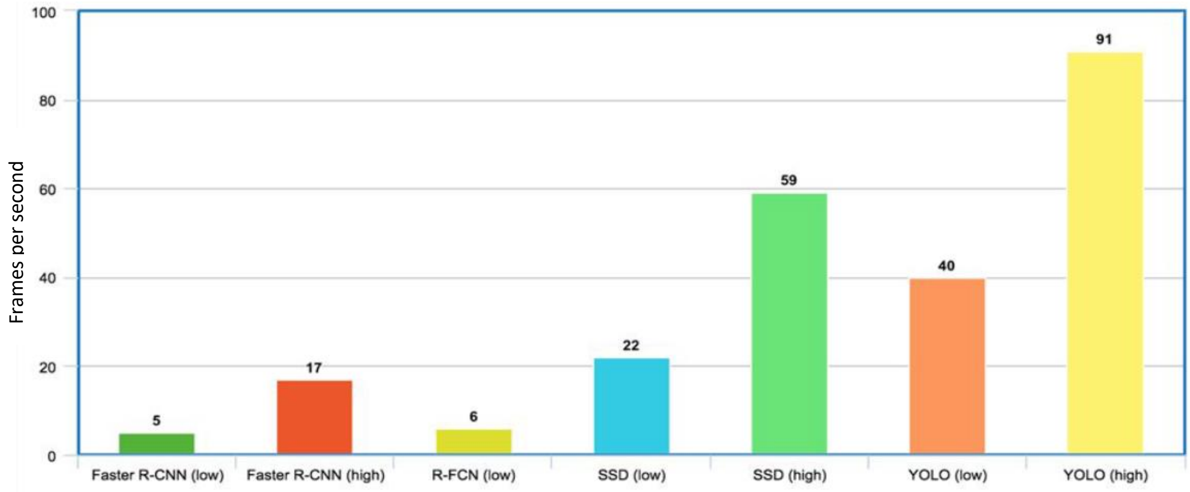


Figure 1: Comparison of various models

4.2 YOLO Model:
When it comes to deep learning-based object detection, there are three primary object detectors we'll encounter that are - RCNN family, Single Shot Detector (SSDs), YOLO. In the first R-CNN which is two-stage Detector uses selective search algorithm to propose candidate bounding boxes that could contain objects. These regions were then passed into a CNN for classification, ultimately leading to one of the first deep learning-based object detectors. The next version of R-CNN became a true end-to-end deep learning object detector by removing the Selective Search requirement and instead relying on a Region Proposal Network (RPN) that is (1) fully convolutional and (2) can predict the object bounding boxes and objectness scores (i.e., a score quantifying how

likely it is a region of an image may contain an image). The outputs of the RPNs are then passed into the R-CNN component for final classification and labeling. While R-CNNs tend to very accurate, the biggest problem with the R-CNN family of networks is their speed — they were incredibly slow, obtaining only 5 FPS on a GPU. To help increase the speed of deep learning-based object detectors, both Single Shot Detectors (SSDs) and YOLO use a one-stage detector strategy. These algorithms treat object detection as a regression problem, taking a given input image and simultaneously learning bounding box coordinates and corresponding class label probabilities. In general, single-stage detectors tend to be less accurate than two-stage detectors but are significantly faster so for Live stream Application it is very useful than R-CNN family. Taking a given input image and simultaneously learning bounding box coordinates and corresponding class label probabilities obtaining 45 FPS on a GPU.

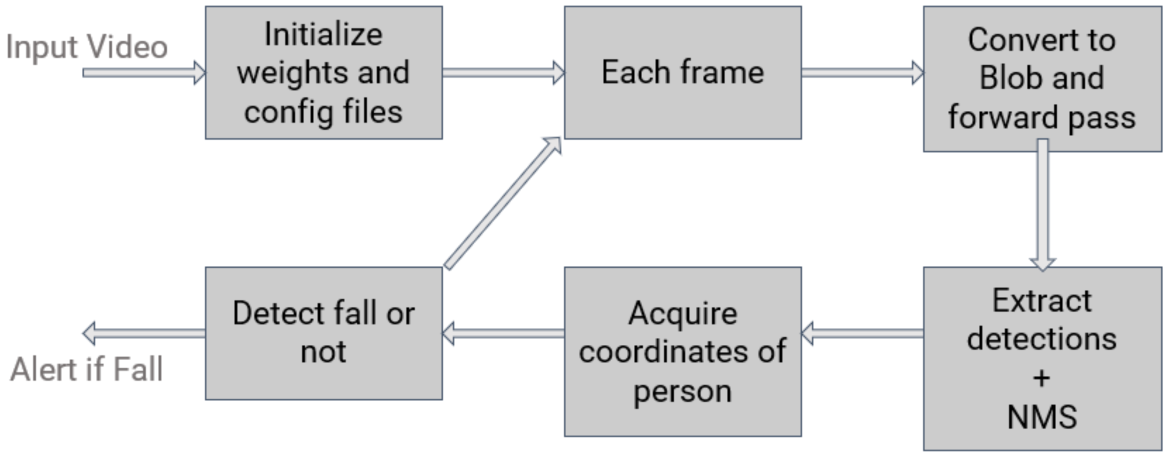


Figure 2: A Representative View of Our Model to Detect Fall.

4.3 Current Implementation

The above representation shows the workflow of our current implementation. Input for the system is a continuous video stream and output is a detection: whether fall or not, as well as a subsequent email message. The weights and config files of the model are first initialized. A video stream is input and for each frame the following procedure follows:

- Convert the frame into a Blob and pass it through the classifier to update weights of each output node.
- For each output in the output layer:
    - Check each detection in the current output node (for various classes)
    - Update the boxes, confidences and classIDs lists for the corresponding detections.
- Apply Non-Maxima Suppression to only keep one detection per object detected in the image.
- Only consider objects that are human and use them for further processing.
- Extract the coordinates of their bounding boxes and calculate width and height.
- If width is greater that height for a significant number of frames, detect as fall and alert the concerned authority via Email.

5. Result

Using the above-mentioned approach, we were able to achieve the following results for the two classes - Fall and No Fall. Accuracy: 93.16%, Recall: 1.0, Precision: 0.98

n=7223	Predicted: No	Predicted: Yes
Actual: No	4027	0
Actual: Yes	63	2702

Fig 3: Confusion Matrix

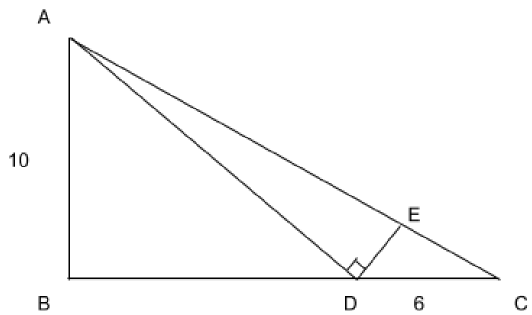
Fig 4: Person detected

Fig.5: Fall Alert

5.1 Possible Bottlenecks

5.1.1 Orientation of the camera

One of the possible bottlenecks we found out was based on the orientation of the camera. On our testing we found out that if a person falls directly in the line of sight of the camera, our system fails to detect the fall.
Assuming:
The width of a person is 3ft.
The height of the person on an average is 6ft.
Height at which the camera is present is 10ft.
The distance of the body from the camera must be about 12ft for a successful detection. Hence, we need to place the camera appropriately.
The following depicts our calculation for the same:



Scenario: Person is fallen directly in the line of sight of the camera.
Assuming width of person, DE = 3ft
And height of person, CD = 6ft

Let AB be elevation of camera = 10ft
Let D be the point where the person is situated.
Let ED be the height of the bounding box detected.
For a successful detection, the height detected must be less than 3ft for a fall, hence we need to find a position for the person (BD) such that ED is about 3ft.
Consider, BD (Distance of person from camera) = 12ft (Say)

AD = (100 + 144)1/2 =15.62ft
∠ACB = tan-1 (AB/BC) = tan-1 (10/18)=29o
∠ADB = tan-1 (AB/BD) = tan-1 (10/12)=40o
∠CDE = 180o -∠ADB - 90o = 90o - 40o =50o
∠CAD = 180o -∠ADC -∠ACD = 180o - 90o - 50o - 29o = 11o
tan(∠EAD) = ED/AD = ED/15.62
ED = 15.62 * tan(11o) = 3.03ft

Hence BD has to be about 12ft if camera is at a height of 10ft. In other words, The ratio of the height of camera to that of the distance of the person to the camera is 5:6.
Another way to solve this bottleneck is to use multiple cameras perpendicular to each other.

5.1.2 Detection of the person

Although YOLO-COCO detects objects with good accuracies, it is possible at times that the person itself is not detected. This usually happens very rarely; it may pose a bottleneck.

6. Conclusion and Future Scope

With our novel approach we were able to get some very impressive results using minimal hardware. Though we had to make some assumptions (minimum distance of the person requirement) we could detect fall of security personnel accurately if the said assumptions were met. We also discussed some possible bottlenecks and some solutions to the said problems.
Some improvements as future scope could be considered as follows:

- Adapt the Solution for Old Age Homes
- Multithreading
- Improve fall detection accuracy using better image processing techniques.

With the implementation discussed thus far, it is evident that Image Processing combined with object detection could prove fruitful for many applications, one of which has been discussed above.

7. References

[1] Y. Nizam, M. N. H. Mohd, and M. M. A. Jamil, “Classification of human fall from activities of daily life using joint measurements,” *J. Telecommun. Electron. Comput. Eng.*, vol. 8, no. 4, pp. 145–149, 2016.
[2] Y. Nizam, M. N. H. Mohd, and M. M. A. Jamil, “Human Fall Detection from Depth Images using Position and Velocity of Subject,” *Procedia Comput. Sci.*, vol. 105, no. March, pp. 131–137, 2017.
[3] Y. Nizam, M. N. H. Mohd, R. Tomari, and M. M. A. Jamil, “Development of human fall detection system using joint height, joint velocity and joint position from depth maps,” *J. Telecommun. Electron. Comput. Eng.*, vol. 8, no. 6, pp. 125–131, 2016.