Fast and Accurate Fall Detection and Warning System Using Image Processing Technology

Thang Nguyen Dang
Dept. of Electrical and Electronics
Campus in Ho Chi Minh City University of Transport and Communications
Ho Chi Minh City, Viet Nam
5851031041@st.utc2.edu.vn

Tan Kim Le
Dept. of Electrical and Electronics
Campus in Ho Chi Minh City University of Transport and Communications
Ho Chi Minh City, Viet Nam
5851031038@st.utc2.edu.vn

Thai Phan Hong
Dept. of Electrical and Electronics
Campus in Ho Chi Minh City University of Transport and Communications
Ho Chi Minh City, Viet Nam
585104C047@st.utc2.edu.vn

Van Binh Nguyen
School of Electrical Engineering
International University
Ho Chi Minh City 700000, Viet Nam
Vietnam National University
Ho Chi Minh City 700000, Viet Nam
nvbinh@hcmiu.edu.vn

***Abstract*— Accidental falls can cause serious injuries, which can lead to serious medical problems, especially for construction and factory workers. This paper proposes a study on a fall detection system based on computer vision. This system is applied to help detect people falling in harsh working environment such as dust, loud noise, few people working. From the recorded video streams, the data is processed to recognize a person falling, lying motionless. Algorithms for tracking people are implemented on a compact, easy-to-install embedded system. Experimental results show that the system ensures safety and can provide emergency assistance to people who have fallen within the view of the camera.**

Keywords—Fall detection, image processing, convolutional neural network

I. INTRODUCTION

Today, most manufacturing plants are operated by machines to optimize production capacity and minimize human error. But there are still some places where people have to be present to perform necessary jobs that cannot be replaced by machines. These places still need people to supervise and operate part of the production process. Places with harsh working conditions such as low-light environments, loud noise and dust often cause accidents which are difficult to recognize and give first aid in time. According to studies and reports, every year millions of people fall. In the US, the death rate from falls increased by 30% from 2007 to 2016 for adults. In fact, falls are very serious and expensive and can lead to broken bones and head injuries. Each year, the total medical costs of falls are up to more than $50 billion [1].

Fall detection can be done automatically thanks to today's advanced technology. There are devices that detect a decrease in acceleration and its direction, while some use a gyroscope to determine the position of the person's body in order to recognize the person who has fallen. But the biggest limitation of this technology is that these devices are not comfortable to wear and often have to use batteries with limited usage time.

In order to improve the efficiency of use and overcome the limitations of current devices, this study develops a system that helps to recognize a fall as soon as it occurs in order to warn of timely emergency assistance and reduce the risk of falling. The use of intelligent camera image processing technology is a fast, accurate and convenient drop detection solution. The camera can be installed in a suitable place, has good visibility, withstands harsh environments. The cameras are connected to an embedded computer, performing real-time processing and analysis. The results of the identification analysis are used to alert security and safety, as well as provide information to the appropriate emergency services [2-4].

The principle and operation of the appropriate network types for this study are presented in Section II. Section 3 describes the implementation steps of data collection, object recognition, and typical results under different conditions.

II. IMAGE PROCESSING TECHNOLOGY

Image processing is a method that uses computers to process digital images through an algorithm. As a result, one can get an enhanced image quality or can extract some useful information. Image processing techniques include four main parts: image quality enhancement processing, photo recognition, image compression, and photo query. The application of image processing is found in many aspects of life, such as in traffic, security, healthcare, games, etc... The core technology of image recognition and processing is the use of artificial neuron network to learn and recognize previously provided information. There are many different types of neural networks, and each type is suitable for different problems. In this study, convolutional neural network (CNN) is focused and used to develop the proposed system (Fig. 1).

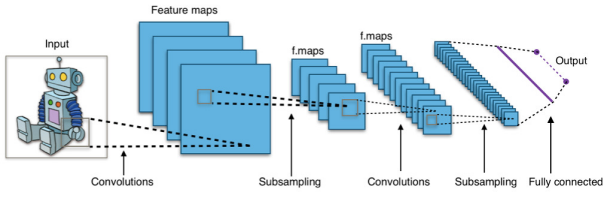

Fig. 1. Typical CNN architecture [5]

A convolutional neural network consists of an input and an output layer, as well as many hidden layers. The hidden layers of a CNN usually consist of a series of convolutional layers that convolve with a multiplication or other dot

product. They are called hidden layers because their inputs and outputs are masked by the final convolution and activation function. The activation function is usually a RELU layer, and is then followed by additional convolutions such as pooling layers, fully connected layers, and normalized layers.

Although the layers are colloquially known as convolution, this is by convention only. Mathematically, it is technically a sliding dot product or cross-correlation. This has implications for indices in matrices, where it affects how the weights are determined at a particular index point. The operation of different types of CNNs is described in the following sections.

A. *Region with Convolutional Neural Network (R-CNN)*

This is a method by which a selective search algorithm is used to extract 2000 regions from an image (region proposals). Therefore, instead of trying to classify a large number of regions, one can work with 2000 regions (Fig. 2). These 2000 candidate region proposals are warped into a square and fed into the CNN to generate a 4096-dimensional feature vector as output. CNN acts as a feature extraction tool and the output dense layer consists of features extracted from the image and the extracted features are fed into an support vector machine (SVM) to classify the presence of the object within that candidate region proposal.

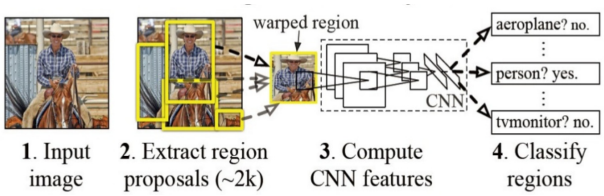

Fig. 2. Regions with CNN features [5]

In addition to predicting the presence of an object in the area proposals, the algorithm also predicts four offset values to increase the accuracy of the bounding box. For example, given an region proposal, the algorithm predicts the presence of a person, but that person's face in that region proposal may have been cut in half. Therefore, the offset values help to adjust the bounding box of the region proposal to fully receive the face.

B. *Faster R-CNN*

In the Faster R-CNN setting, detection occurs in two phases (Fig. 3). In the first phase, known as the region proposal network (RPN), the image is processed using a feature extractor and features at some selected intermediate levels are used to predict class-agnostic box proposals. The loss function for this first stage takes the form of Equation (1) using a grid of anchor points arranged in space, scale, and aspect ratio. In the second stage, these (typically 300) box proposals are used to crop features from the same intermediate feature map, which are then fed to the rest of the feature extractor in order to predict a class and refined class-specific box for each proposal.

Loss function is defined as [6]:

$$L(p_i,t_i)=\frac{1}{N_{cls}}\sum_i L_{cls}\left(p_i,p_i^*\right)+\lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i,t_i^*) \quad (1)$$

where i being the index of an anchor in a mini-batch, p_i being the predicted probability of anchor i^{th}. The ground-truth label is 1 if the anchor is positive, and is 0 if the anchor is negative. t_i is a vector representing the 4 parameterized coordinates of the predicted bounding box, t_i^* is that of the ground-truth box associated with a positive anchor. The classification loss L_{cls} is log loss over two classes (object vs. not object)

For the regression loss, one uses the robust loss function (smooth L_1) defined in Equation (2).

$$Loss(x,y)=\sum\begin{pmatrix}0.5*(x_i-y_i)^2, if\ |x_i-y_i|<1\\ |x_i-y_i|-0.5, otherwise\end{pmatrix} \quad (2)$$

The loss function for this second stage box classifier also takes the form of Equation (1) using the proposals generated from the RPN as anchors. Notably, one does not crop the proposals directly from the image and reruns the crops through the feature extractor, which would be computed in duplicate. However, there is a part of the computation that must be run once per region, and thus the running time depends on the number of regions proposed by the RPN.

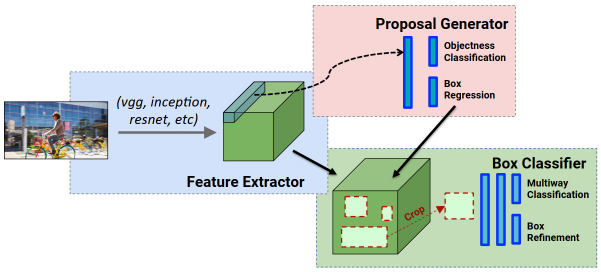

Fig. 3. Faster R-CNN is a single, unified network for object detection. [6]

The reason "Fast R-CNN" is faster than R-CNN because the 2000 region proposals do not have to be fed to the convolutional neural network every time. Instead, the convolution operation is done only once per image and a feature map is generated from it.

C. *Region Proposal Networks (RPN)*

The Region Proposal Network (RPN) takes an image (of any size) as input, and the output is a set of rectangular object proposals, each with an objectness score. To generate region proposals, slide a small network over the convolutional feature map output by the last shared convolutional layer. This small network takes an n × n spatial window of the convolutional feature map as an input. Each sliding window is mapped to a lower dimensional feature. This feature is injected into two fully sibling connected layers - a box regression layer (reg) and a box classification layer (cls). We use n = 3 in this paper, noting that the effective receptive field on the input image is large (171 and 228 pixels for ZF and VGG, respectively). This subnet is illustrated at a single position in Fig. 4. Note that because the subnet operates in a sliding window fashion, the

fully connected layers are shared across all spatial locations. This architecture is naturally implemented with one n × n convolutional layer followed by two sibling 1 × 1 convolutional layers (for reg and cls, respectively).

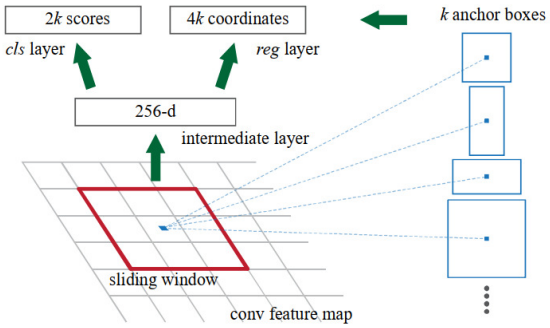

Fig. 4. Region Proposal Networks

At each sliding window position, one simultaneously predict multiple regional proposals, where the number of maximum possible proposals for each location is denoted by *k*. So the reg layer has 4k outputs encoding the coordinates of *k* boxes and the cls layer outputs 2k scores that estimate probability of object or not object for each proposal. The *K* proposals are parameterized relative to k reference boxes, which are called anchors. An anchor is centered at the mentioned sliding window and associated with scale and aspect ratio. By default we use 3 scales and 3 aspect ratios, yielding k = 9 anchors at each slide position. For a convolutional feature map of size W × H (typically ~2,400), there are W × H × k anchors in total.

For training RPNs, we assign a binary class label (whether an object or not) to each anchor. A positive labels is assigned to two types of anchors: (i) anchor/anchors with the highest Intersection over Union (IoU) overlap with ground truth box, or (ii) anchors with IoU overlap higher than 0.7 with any ground truth box. Note that a single ground truth box can assign a positive label to multiple anchors. Usually the second condition is sufficient to identify positive samples; but the first condition is still applied because in rare cases the second condition may not find a positive sample. We assign a negative label to a non-positive anchor if its IoU ratio is lower than 0.3 for all ground truth boxes. Anchor neither positive nor negative does not contribute to the training objective.

III. Experimental Results

A. *Fall Detection Classifier Using TensorFlow*

Before taking any steps for fall detection using digital image processing, a sufficiently large dataset needs to be collected first. Collected data needs to be in the range of 1000 images of falls (the larger the dataset, the better the results). These images are actually collected in the field or need to be created under specific conditions.

In this paper, about 1200 images of more than 30 fall cases were gathered in different environments. Half of the images were received in good light condition, the subject was blended with backgrounds, and half were in low light condition, increasing the difficulty of recognition (Fig. 5).

For the training sections, this machine learning method requires these photos to be divided into two main folders, Test Dataset and Training Dataset. With 20% of the images in the Test Dataset folder and the remaining 80% of them in the Training Dataset folder.

Fig. 5. Selected typical training data

The steps below are required to set up the training environment and perform the identification.

- Install Anaconda, CUDA and cuDNN: in this study we used the latest version of Anaconda. This Anaconda is a powerful tool that contains all the necessary libraries for a machine learning or deep learning project. We used Anaconda to install TensorFlow version 1.15.0, along with CUDA 10.0.130 and cuDNN 7.6.5. This first step could be difficult because library version mismatch leads to a lot of problems in training time. But this is the most important step in creating a quality identity system.
- Set up the Anaconda Virtual Environment and Object Detection folder structure: the entire TensorFlow object detection repository is available at https://github.com/tensorflow/models. R-CNN and Faster R-CNN model for training is then downloaded.
- Gathering and labeling images
- Generating training data
- Creating label maps and configuring training
- Training
- Exporting the inference graph
- Testing and using the trained object detection classifier.

TABLE I. Models Comparision

Model	Evaluation Factor		
	Training Time (h)	*Speed (ms)*	*Accuracy %*
R-CNN	7	51	81
Faster R-CNN	5	58	84

After deploying two models R-CNN and Faster R-CNN, the training results show that the system works smoothly. According to Table 1, it takes at least 7 hours to train the R-CNN model, while the time is 5 hours when training the Faster R-CNN. The test accuracy is up to more than 84%.

B. Experiments on Rapsberry Pi4

In this paper, the fall detection application is run on a small embedded system Raspberry Pi 4. This device has good compatibility with existing libraries. In addition, simple installation, ease of use, compactness are advantages suitable for various installation conditions. A camera is directly connected to the embedded system with a resolution of 1280 x 720 pixels.

As can be seen in the table of results (Table 2), the system achieves a fairly high detection rate for Faster R-CNN under good conditions, up to 86%. To get better results, it is necessary to upgrade the training dataset. In low light condition, the ratio is reduced but still very reliable. In this case, an extra light for the camera or using an infrared camera can be an easy and quick solution.

TABLE II. RESULTS OF DETECTING IN DIFFERENT CONDITIONS

Network	**Conditions**		
	Good light Conditions	***Low Light Conditions***	***Vibration Conditions***
R-CNN	41/50 (82%)	38/50 (76%)	35/50 (70%)
Faster R-CNN	43/50 (86%)	40/50 (80%)	37/50 (74%)

In addition, in the condition of slight vibration, the results show that the detection rate is also acceptable with 70% and 74% for R-CNN and Faster R-CNN cases, respectively. In this case, the oscillation dataset needs to be upgraded to improve the recognition quality. Some typical actual identification results are listed in Fig. 6.

On the other hand, taking advantage of the proposed embedded system, the detection function can incorporate an environment-aware (or pre-installed) function. In addition, when wearing uniforms or specialized protective gear, better quality is achieved due to easy separation of the object from the environment.

Another function of the system is to warn fall cases with accident assistance service. When an accident is detected, the system generates an alarm through the control of the horn and indicator lights, and sends alarm information to the emergency phone through popular channels such as messaging and calling services.

IV. CONCLUSIONS

This study has successfully built a fall detection system using a video feed from a camera and implemented Faster R-CNN recognition with the support of TensorFlow. Image processing techniques and deep learning methods are applied in an advanced way to effectively distinguish falling activities in real time. This research can help reduce the effects of work-related accidents and could be extended to other areas such as transportation, health care, preschool observation, etc.

Fig. 6. Selected typical fall object identification results

REFERENCES

[1] B. Elizabeth and K. Ramakrishna, “Deaths from Falls Among Persons Aged ≥65 Years − United States, 2007 - 2016,” Morbidity and Mortality Weekly Report, CDC, 2018.

[2] P. Vallabh, R. Malekian, N. Ye and D. C. Bogatinoska, "Fall detection using machine learning algorithms," 24th International Conf. on Software, Telecommunications and Computer Networks (SoftCOM), pp. 1-9, 2016.

[3] M. Kreković et al., "A method for real-time detection of human fall from video," 2012 Proceedings of the 35th International Convention MIPRO, pp. 1709-1712, 2012.

[4] K. Singh, A. Rajput and S. Sharma, “Human Fall Detection Using Machine Learning Methods: A Survey,” International Journal of Mathematical, Engineering and Management Sciences, 2019.

[5] J. Huang, V. Rathod, C. Chen and M. Zhu, “Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2017.

[6] S. Ren, K. He, R. Girshick and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” IEEE Trans. on Pattern Analysis and Machine Intelligence. No. 39, 2015.