Machine Learning Safety System for Treadmill Users

Ravishankar Natarajan, Xinrui Yu, Jafar Saniie
Embedded Computing and Signal Processing (ECASP) Research Laboratory (http://ecasp.ece.iit.edu)
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago IL, U.S.A.

***Abstract*—In this paper, we present an AI-based treadmill assistant that employs Computer vision and Deep learning algorithms to offer a comprehensive safety net for treadmill users. The system is designed and implemented with minimal components including an off-the-shelf camera, speaker, an IoT smart plug, and Jetson Nano (a single board GPU-based embedded computing device). To provide an extensive set of safety features such as face recognition, facial stress detection, object detection, and distance and aspect measurement to the treadmill user, the system uses CNN architectures based on ResNet, MobileNet SSD, and customized fully connected neural networks. This set of features monitored in real-time will play a crucial role in mitigating treadmill user falls and injuries.**

Keywords—Computer Vision, Deep Learning, CNN, MobileNet SSD, ResNet, Treadmill safety

I. Introduction

Treadmills are one of the most widely used fitness equipment that is quite common in homes and gymnasiums and is used by people in a wide range of age groups. But there are a number of safety issues with respect to the treadmill especially due to the presence of the moving belt. The moving belt increases the risk of falling and could potentially lead to severe injuries like broken bones, sprained joints, head injuries, and so on [1,3]. It is an even greater risk for children [2] as due to their young age would be less aware of the safety precautions to take while using the treadmill and thus more prone to injuries.

Most of the current solutions for treadmills in the market are mainly focused on improving the comfort level of the user by providing cushioning and shock absorbers [4-5]. But, this type of feature by no means protects the user from fall injuries. In most treadmills, the only safety feature against fall injuries is the treadmill safety key [6] which is a key attached to the front panel of a treadmill through a string at one end and connected to the user's clothing on the other end using a clip. The purpose of this key is to provide the user with an emergency stop feature as the treadmill immediately stops once the key is out of its place on the treadmill which would happen if the user falls during the workout. This is a useful but most underused safety feature [7] as it would work only if the user ensures that the key is clipped onto themselves before the start of the workout and it has been found that most users tend to neglect this step as they find it inconvenient thus compromising on safety. Furthermore, this solution only helps in mitigating the extent of the injuries in the event of a fall and does not help in avoiding a fall in the first place. Given these drawbacks of the existing solutions, we believe that a market niche exists in the area of treadmill safety related to fall injuries where a solution that not only aims at protecting the user in the event of a fall but also provides feedback to the user to avoid a fall in the first place, and further promotes widespread adoption by being more convenient yet affordable would be highly valued by the users and is bound to succeed.

Based on the above reasoning, we propose a machine learning treadmill safety system that essentially consists of the following features:

- Automatic facial recognition based user authorization and automatic switch-on of the treadmill
- Automated facial stress analysis and drift monitoring of the user (from the center of a treadmill) coupled with voice feedback to the user during the workout for fall prevention
- Automatic fall detection and switch-off of the treadmill to mitigate the severity of the injuries due to the fall

The automatic facial recognition feature ensures that only authorized users can switch on the treadmill and use it. It would, in a way, act as a child-lock system as younger children would not be able to turn on the treadmill on their own since the Facial recognition system would not allow them to do so. The automated facial stress analysis feature would constantly monitor the facial expression of the user for indication of overexertion as facial action has been found to have a strong correlation to the amount of exertion perceived by the user [8]. The drift monitoring feature coupled with voice feedback to the user ensures that the users maintain their position at the center of the treadmill thus reducing the risk of a fall. The automatic fall detection and switch-off prevent fall injuries in a similar manner to the treadmill safety key but does so in an automatic fashion without the requirement of a physical key. This feature also ensures that an unattended treadmill is always in the switched-off state and thus prevents injuries to young children and adults as well who might inadvertently step onto the treadmill without realizing that the belt is moving.

In this paper, we introduce a prototype as a proof of concept that provides all the features that are listed above. The system to realize the aforementioned features is shown in Figure 1. As shown in Figure 1, apart from the treadmill and the user using the treadmill, the system fundamentally consists of the following parts:

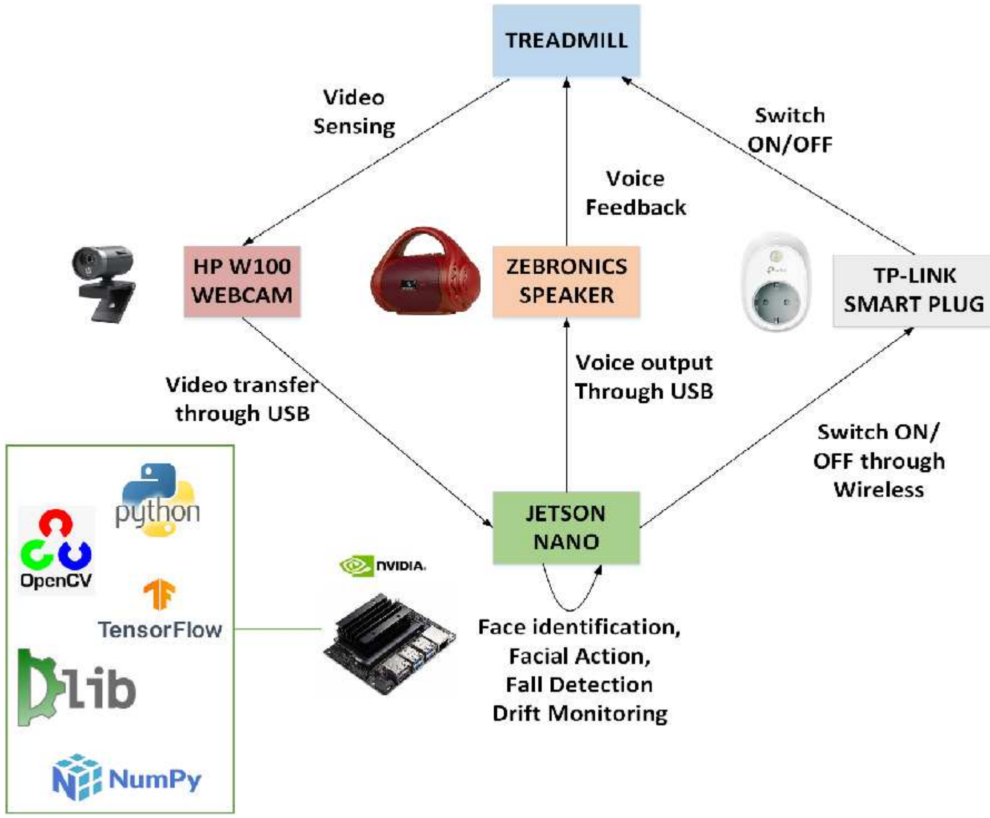

Figure 1. The Treadmill Safety System Diagram

- **Brain**: Central to the system is the Jetson Nano [9] which is used for all the computations. This is a single board computer containing both an ARM processor and the GPU with 128 CUDA cores. On the Jetson Nano, Python 3.0 based code is running that utilizes the DLIB library [10], Jetson-inference GitHub repository[11], OpenCV [12], and TensorFlow [13] libraries for face recognition, object detection, drift measurement, and facial stress detection. Most of these use pre-trained deep neural network models and custom-made fully connected neural network models as well.

- **Vision**: The HP W100 webcam which has a resolution of 640x480 pixels is used to stream video in real-time to the Jetson Nano through a USB interface. OpenCV libraries [14] running within the Jetson Nano are used to capture the video from the camera frame by frame and subject these frames to further processing.

- **Voice**: Speakers from Zebronics are used to enable the Jetson Nano to communicate with the Treadmill user. The speaker is connected to the Jetson Nano through the USB interface, and there is python code running within the Jetson Nano that utilizes the google text to Speech library[15] to provide the audio output to the speaker.

- **Smart Plug**: Further, the TP-link HS 100 Smart Plug is used to control the power to the Treadmill. It is connected to the Jetson Nano through Wi-Fi. Jetson Nano has software running [16] within it to wirelessly switch on/off the Smart Plug.

Section II furnishes more details on the system design of the treadmill safety system. Section III explains further in detail the underlying algorithms behind the features like automatic facial recognition, automated facial stress analysis, drift monitoring, and automatic fall detection. Section IV finally discusses the results of the project

II. System Design

A. The Treadmill Workout Session Lifetime Chart

To understand the role played by each of the parts of the treadmill safety system, it is imperative to grasp the overall system flow during a workout session. This system flow is depicted in the form of the treadmill workout session chart in Figure 2. The system flow as depicted in Figure 2 essentially consists of the following steps:

- **Webcam Turn ON**: The user steps onto the treadmill and turns on the webcam. The webcam senses the users and starts providing the video feed to the Jetson Nano.

- **Face Recognition and Authorization**: Jetson Nano runs the Face recognition application that utilizes the Dlib library to identify the user and check if the identified user is authorized to use the treadmill.

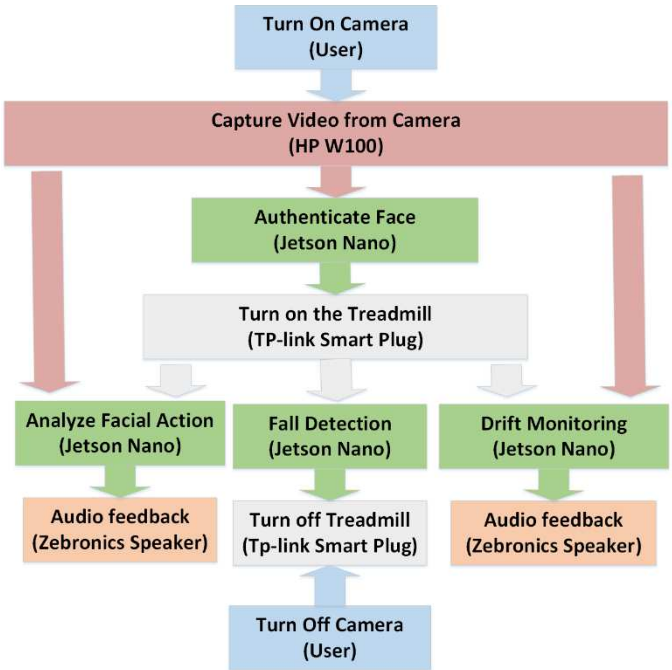

Figure 2. Treadmill Workout Session Chart

- **Treadmill Power ON**: Only if the user is authorized, will a welcome message be sent through the Zebronics speaker, and the power to the treadmill is switched on.
- **Video Frames Analysis**: The Jetson Nano internally starts various applications namely the drift detection application, the object detection application, and the facial stress detection application so that during the workout, it can constantly monitor the user through the video feed from the webcam and the video frames are subjected to further analysis using the various applications.
- **Feedback to The User**: Based on the analysis, suggestive feedback is provided to the treadmill user through the Zebronics speaker.
- **Treadmill Power Off**: When the user gets off the treadmill or experiences a fall, then the Jetson detects this event and shuts off the power to the treadmill.

III. FEATURE-WISE DEEP DIVE

A. Real-time Face Recognition

The real-time Face recognition process consists of multiple sub-steps namely face detection, facial feature extraction, and feature matching.

a. *Face Detection through Frontal Face Detector*

Face detection essentially refers to locating one or more faces in the image and marking them with a bounding box. Face detection is done using the frontal face detector algorithm of Dlib [17] that uses the Histogram of Oriented Gradients (HOGS) feature [18] combined with linear classifiers.

b. *Feature Extraction using Dlib*

In the feature extraction stage, the features of the face are extracted and then provided to the feature matching stage. The feature extraction step is carried out by a Dlib face recognition algorithm [19] that utilizes a ResNet network [20] as shown in Figure 3.

c. *Feature Matching through Euclidean Distance*

Post the feature extraction process, the feature matching process that is executed involves calculating the Euclidean distance [21] between the face descriptors of pre-trained images and the face descriptor that has been currently calculated in real-time. If the value of this Euclidean distance is less than 0.6, then it is considered to be a match. The entire process is described in Figure 3 below.

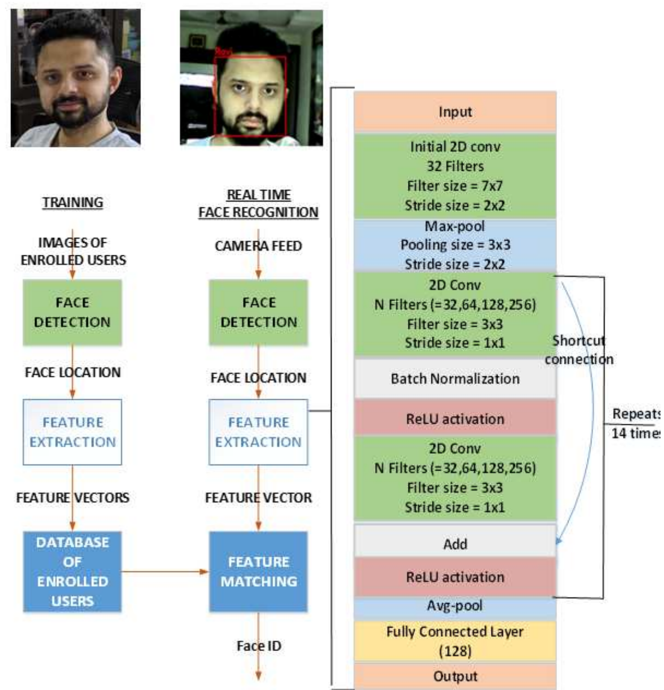

Figure 3. Real-time Face Recognition

B. Real-time Facial Stress Recognition

The facial stress recognition process essentially comprises two parts namely facial landmarks extraction using Dlib libraries [22] and inferencing of the facial stress using a fully connected neural network created using TensorFlow.

C. Real-time Drift Detection

There are two types of drift possible. One being backward or forward drift and the other being side-ways drift which is left or right drift.

a. Similarity of Triangle Principle

The drift whether backward drift or sideways drift is essentially determined by using the similarity of triangle principle which states that the ratio of the size of the object to the distance of the object from the camera is equal to the size of the image in the camera and the focal length of the camera.

b. Backward Drift Detection

The way similarity of triangle principle is used for backward/forward drift detection is illustrated in Figure 4.

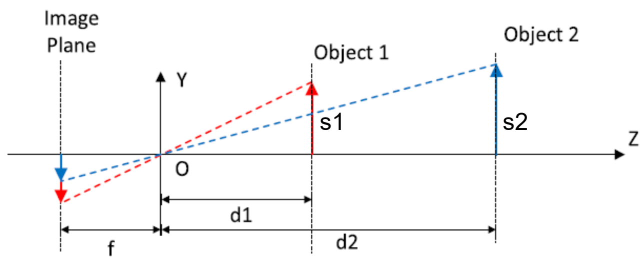

Figure 4. Backward Drift Detection

The drift is given by the following formula:

$$(d2 - d1) = s * f * (\frac{1}{s2} - \frac{1}{s1}) \quad (1)$$

In the case of drift detection of the treadmill users, s is the actual width of the user's face, and s1 and s2 are the widths of the user's face in pixels on the image plane corresponding to distance d1 and d2 respectively of the user from the camera.

c. Sideways Drift Detection.

Once the backward drift is found, the next step is to find the sideways drift. This is also based on the similarity of triangle principle. The method to find sideways drift is illustrated using Figure 5.

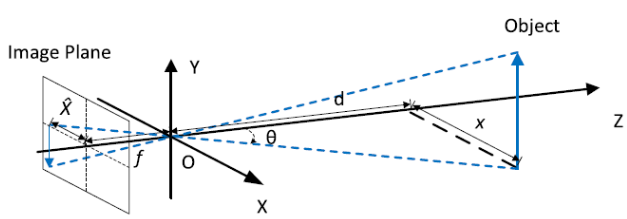

Figure 5. Sidewise Drift Detection

Here, x is the sideways drift of the object with respect to the center of the treadmill, and x' is the corresponding sideways drift that is seen by the image of the object. Also, d is the distance of the object from the camera, and f is the focal length of the camera. the sideways drift of the user can be easily calculated using the below formula:

$$X = \left(\frac{d}{f}\right) * x' \quad (2)$$

D. Real-time Object Detection for End of Workout Detection and Auto Switch off

MobileNet SSD [23] based Object detection machine learning algorithm is used to detect the end of a workout. An end of workout can be either because of the user alighting the treadmill after the completion of workout or due to the user losing balance and falling. The object detection algorithm detects both these events by constantly monitoring for the "person" object in front of the treadmill. In case the person is no longer detected either because the user has stopped working out and got off the treadmill or has experienced a fall, then the treadmill is automatically switched off.

IV. RESULTS

A. Face Recognition Results

The face recognition was trained and then tested on a live webcam feed and it was able to accurately recognize the face.

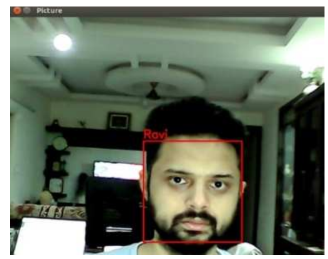

Figure 6. Face Recognition Results

B. Facial Landmarks Detection Results

The facial action was dependent on the ability to accurately collect the facial landmarks for a live webcam video.

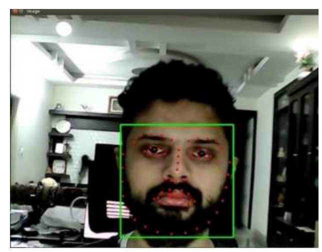

Figure 7. Facial Landmark Detection Results

C. Facial Stress Detection Results

Once the facial landmarks were accurately detected on a live webcam feed, the DNN-based classifier was deployed that was again able to accurately determine if the facial expression was neutral or stressed.

Figure 8. Neutral/Stressed Expression Detection Results

D. Drift Detection Results

Able to detect the backward as well as sideways drift with sufficient accuracy.

Figure 9. Initial position before drift

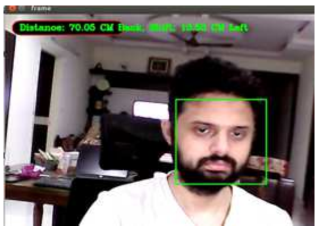

Figure 10. Final Position after backward and sideways drift

E. *Person detection results*

Person detection using MobileNet SSD was achieved as follows:

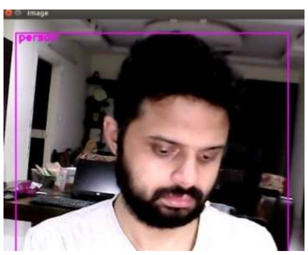

Figure 11. Person Detection

F. *The Treadmill Safety System Prototype*

The treadmill safety system prototype was constructed using the various hardware components like Jetson Nano Embedded computing system, HP W100 Webcam, Zebronics Speaker, TP-link smart plug, and treadmill, and by incorporating the required libraries and developed application within the Jetson Nano. All the functionalities and features of the treadmill safety system namely facial recognition and user authorization, "person" detection, drift detection, facial stress detection, and automatic turn on and turn off of the treadmill were all successfully tested in this realistic setting. A snapshot of the prototype, while it was being tested, is shown in Figure 12. The various components of the setup have been marked and labeled for the convenience of the viewer.

V. Conclusion

In this paper, we have presented a smart AI-based treadmill safety system that provides automatic real-time features like face recognition, facial expression detection, drift detection, and fall detection thus providing a comprehensive safety net to the treadmill user. This is done by employing Computer vision and Machine learning algorithms within Jetson Nano which is a GPU-based embedded computer. The overall system is achieved with minimal components that are available off-the-shelf like a camera, speaker, and an IoT smart plug in addition to Jetson Nano. We firmly believe that this extensive set of features will play a crucial role in mitigating treadmill user falls and injuries and it is therefore imperative that every treadmill be equipped with such a system.

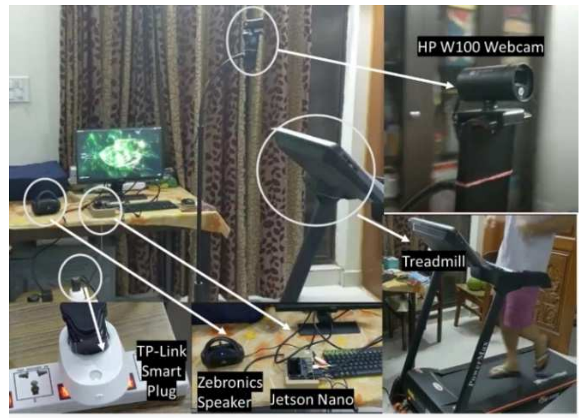

Figure 12. Treadmill Safety System Prototype

References

[1] BBC, Peloton: Child killed in 'tragic' treadmill accident. [Online]. Available: https://www.bbc.com/news/business-56451430 [Accessed: Feb. 10, 2022].

[2] Law Offices of Joel L. Katz LLC., Grandmother suffers brain injury after treadmill accident. [Online]. Available: https://www.joelkatzllc.com/blog/grandmother-suffers-brain-injury-after-treadmill-accident/ [Accessed: Feb. 10, 2022]

[3] Silver Lining, Treadmill Safety. [Online]. Available: https://www.thesilverlining.com/safety-tips/treadmill-safety [Accessed: Feb. 10, 2022]

[4] How do you absorb shocks from a Treadmill? [Online]. Available: https://iwassweet.com/how-do-you-absorb-shocks-from-a-treadmill/ [Accessed: Feb. 10, 2022]

[5] Treadmill talk, Treadmill Cushioning Explained. [Online]. Available: https://www.treadmilltalk.com/treadmill-cushioning.html [Accessed: Feb 10, 2022]

[6] Fun On Foot, What is a Treadmill Safety Key and How Do you Use it. [Online]. Available: https://www.funonfoot.com/treadmill-safety-key/#:~:text=A%20treadmill%20safety%20key%20is,is%20out%20of%20its%20place. [Accessed: Feb 10, 2022]

[7] Treadmill run, Are Treadmill Keys Universal? (No, Here's Why) [Online]. Available: https://www.treadmill.run/are-treadmill-keys-universal.html [Accessed: Feb 10, 2022]

[8] Timme S, Brand R (2020) Affect and exertion during incremental physical exercise: Examining changes using automated facial action analysis and experiential self-report. PLoS ONE 15(2): e0228739. https://doi.org/10.1371/journal.pone.0228739

[9] Nvidia Developer, [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit [Accessed: Feb 10, 2022]

[10] Dlib C++ Library, [Online]. Available: http://dlib.net/ [Accessed: Feb 10, 2022]

[11] Jetson-inference Github, [Online]. Available: https://github.com/dusty-nv/jetson-inference [Accessed: Feb 10, 2022]

[12] OpenCV. [Online]. Available: https://opencv.org/ [Accessed: Feb 10, 2022]

[13] TensorFlow [Online]. Available: https://www.tensorflow.org/ [Accessed: Feb 10, 2022]

[14] [OpenCV, Getting Started with Videos. [Online]. Available: https://docs.opencv.org/4.x/dd/d43/tutorial_py_video_display.html [Accessed: Feb 10, 2022]

[15] gTTS 2.2.3. [Online]. Available: https://pypi.org/project/gTTS/ [Accessed: Feb 10, 2022]

[16] Python-Kasa Github, [Online]. Available: https://github.com/python-kasa/python-kasa [Accessed: Feb 10, 2022]

[17] Dlib C++ Library, [Online]. Available: dlib.net/face_detector.py.html [Accessed: Feb 10, 2022]

[18] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, pages 886–893, June 2005.

[19] Dlib C++ Library, [Online]. Available: dlib C++ Library - dnn_face_recognition_ex.cpp [Accessed: Feb 10, 2022]

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385, 2015.

[21] PBarret, technical white paper, [Online]. Available: Microsoft Word - euclid.docx (pbarrett.net) [Accessed: Feb10, 2022]

[22] Dlib C++ Library, [Online]. Available: dlib C++ Library - face_landmark_detection_ex.cpp [Accessed: Feb 10, 2022]

[23] Howard, A. G., “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”, arXiv e-prints, 2017.