Quest Journals
Journal of Software Engineering and Simulation
Volume 9 ~ Issue 5 (2023) pp: 05-12
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

A CNN-Based Model for Fall Detection Using Inexpensive Security Cameras

Xiyu Chen New Hampton School

Abstract— *As the aging population becomes increasingly common, falls among the elderly are becoming a significant issue. Effective fall detection and monitoring systems use AI on existing and widespread affordable in-home security camera systems. In this paper, a Convolutional Neural Network (CNN) based model proposes to achieve this using a training dataset from inexpensive security cameras. The dataset used for this purpose comprises 192 video files from 24 scenarios taken by affordable wide-angle IP cameras monitoring the entire room. The videos are preprocessed into structural and image data respectively, with five features selected and extracted through methods including motion history image, gaussian mixture model, machine learning based pose-estimation algorithms, and traditional computer vision methods. The efficacy of conventional machine learning models is compared to the CNN model devised for fall detection. The CNN approach achieves a 93.09% F1 score on the test set, outperforming the best-performing conventional machine learning model, random forest, which achieved an F1 score of 76.88%.*

Keywords—*Deep learning, Supervised learning, Convolutional neural network (CNN), Fall detection, video surveillance*

Received 25 Apr., 2023; Revised 03 May, 2023; Accepted 05 May, 2023 © The author(s) 2023. Published with open access at ***www.questjournals.org***

I. INTRODUCTION

The aging human population is a significant issue in many countries. According to the WHO, the proportion of people over 60 years in the world's overall population will double from 12% to 20% between 2015 and 2050 [1]. As people age, they are more likely to have impairments in their physical and cognitive abilities that can increase their risk of falling, which may include decreased muscle strength, balance ability, and vision or hearing loss [2]. About 32 million cases of elderly people falling are reported annually, and more than 32000 people have lost their lives from falling [3]. Unreported cases of falling are a cause for worry since they go untreated, leading to an increased potential for more severe health issues. It is increasingly crucial to detect falls automatically to provide preventative steps or early treatments.

Many fall detection systems use body sensors or fixed multi-camera systems [4]. For example, Rodrigues presents a comparison of machine learning models based on data from inertial sensors to realize real-time fall detection. Though models such as KNN deliver a high accuracy with the inertial sensors, the installation of the sensors do not fit the everyday use of the elderly population [5]. Similarly, Usmani proposed a study on various types of wearable sensors and multiple machine learning algorithms. The study achieved a 98% accuracy using Inertial Measurement Unit (IMU) applied on people's waists [6]. Among other examples of work, Shu utilized multi-camera setups to employ a machine learning model on a single-board computer. They analyzed the fall pattern with machine learning and pre-defined rules. This model applies to an approximate distance of 60 meters from the camera, higher than most commonly used camera-based systems [7] .

While applying body sensors or multi-camera setups leads to better accuracy, they are also less affordable and less accessible for many families. Also, sensor-based approaches only work when older people wear them, which might cause some inconvenience or reluctance due to their physical and mental ineptitude attributed to old age. Despite their accuracy and reliability, these methods seem infeasible due to their price and inconvenience. Therefore, a system based on inexpensive security cameras is proposed in this paper. The dataset is chosen for its accessibility: RGB videos shot with wide-angled and high-position surveillance cameras [8].

Machine learning algorithms based on computer vision, including decision trees, support vector machines, and random forests, can be applied to perform fall detection based on structured data with various extracted features. They are a reliable fall detection tool using low-cost CCTV cameras. They can also assist in all-day fall detection that sends out warnings to intervene before or when any fall is detected.

II. RELATED WORK

Many prior papers combine computer vision and conventional machine learning. Some examples of work with these combinations include Sukma Wahyuni's system based on Motion History Image (MHI) and Approximated Ellipse. Ellipse orientation and ratio are calculated from the MHI to detect falling [9]. Other examples of work include Rougier's system analyzing the shape deformation through video sequences. They track the silhouette and input it into a Gaussian Mixture Model (GMM) [10]. These provide the premise for combining Unsupervised learning algorithms such as K-means and Gaussian Mixture Model (GMM) with conventional computer vision methods to detect falls. GMM-based background extraction combines people's aspect ratios to provide fast fall detection.

Beyond those machine learning models, deep learning models also show immense prediction reliability, albeit at the cost of intricate and complex structures. The interconnected neurons in the human brain inspire deep learning models. This artificial structure helps to find a nonlinear relationship between inputs and outputs by adjusting weights through iterations. Examples of deep learning models include the perceptron, the simplest model often used for logic gates; recurrent neural networks (RNNs), a model that makes use of a previous stage; and Long Short-Term Memory Networks (LSTMs), a model that features a repeating module to store many previous stages. Convolutional neural networks (CNN) are compelling deep learning architecture. It features convolution and pooling layers, which contributes to the ultimate goal of feature extraction. Using a backpropagation algorithm, the model learns to adapt complicated features from low to high levels. To extract features, CNN uses a *kernel* filter to adapt to slight changes in image shape and position. This nature largely suits CNN to computer vision tasks [11].

Experts have done prior work on CNN-based fall detection. Chhetri has proposed a model to detect falling using the Enhanced Dynamic Optical Flow technique, which improves the accuracy and robustness of vision-based fall detection under dynamic lighting conditions [12]. Deep learning models can also be combined to detect falling; Salimi demonstrated a model using Fast Pose Estimation method based on LSTM and CNN. The model achieved about 97% accuracy with a relatively low computational demand [13].

The rest of this paper is organized as follows: Section III illustrates the data collection methodology, preprocessing, and model training. Section IV presents the accuracy of the models and results. At last, section V expands the current work into potential future applications.

III. METHODOLOGY

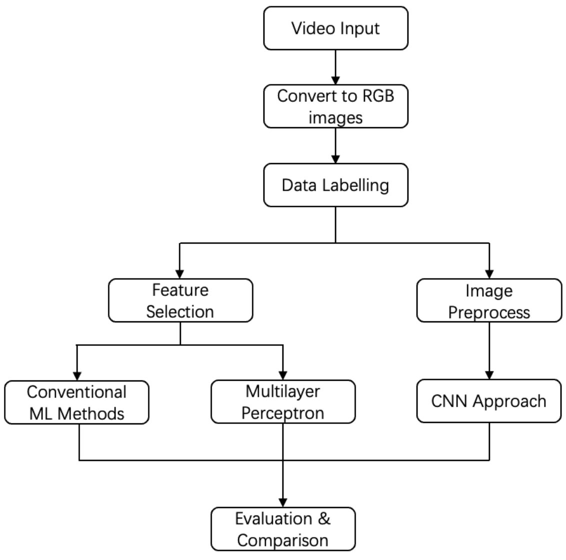

Figure. 1. General Methodology

Figure. 1. Presents the general structure of the research approach in this paper. A video dataset is first chosen based on data from affordable and single RGB camera settings. Then, the videos are converted into RGB images frame by frame and labeled following the dataset's technical report [8]. For conventional machine learning models and multilayer perceptron, five features are selected and stored in a structured format for input to the models next. The models above are trained on this tabular data. For the CNN model, the raw images are first preprocessed before the CNN model structure is developes on the resulting image data. The accuracy of these models are compared and evaluated. These steps are underlined in the subsection below.

A. Data Collection

The dataset chosen is collected by researchers including E. Auvinet and C. Rougier. The videos are taken in a lab-simulated environment where various falls are recorded using inexpensive CCTV cameras. There are 24 subsets, including about 192 videos. Wide angle surveillance cameras are installed high on the ceiling, which might distort the objects. The videos are accelerated to 120 frames per second [8].

The angle of the camera affects the bias produced by each model. For instance, if the angles are fixed in every case, the models might learn from insignificant features and noise to perform overfitting. Conventional computer vision models may work in certain angles significantly better than some other angles. Therefore, in each set, cameras from 8 angles in a room each provide one short clip from when the individual enters the room to their leaving. All video clips are treated as individual fallings without correlating with other angle videos in one single scene. Different forms of falling also affect the detection process.

The dataset covers nine possible cases: walking or standing, falling, lying on the ground, crouching, moving down, moving up, sitting, lying on a sofa, and moving horizontally. The 9 cases are prelabeled with the dataset and used as ground truth for this research [8]. After several tests on the model, cases when people are lying on the ground are chosen as positive samples, while cases when people are standing or walking are negative samples. Those two classes of data are input into the models. In early tests, all the data is treated as inputs in 9 classes or combined into 2 classes, but both data fail to converge the model. The model also seemed to overfit the training data even after many regularization techniques, such as L1, L2, and data argumentation. The speculation for the failure with 9 classes is that the image number for each class is largely imbalanced. Most of the time, people in the camera are standing, walking, or lying on the ground. If they are all combined into 2 categories, the pattern in the data becomes much less evident, making the model hard to converge.

Besides choosing from 9 classes of images, not every image is used in training. In some early experiments, using all the images to train models led to severely overfitting models, which could be attributed to the lack of data diversity. Therefore, one of ten images are chosen, representing that period of time.

B. Software Tools

Data preprocessing was achieved by using OpenCV and NumPy in Python. Feature extraction is performed using the OpenCV, pandas, and NumPy libraries. Conventional machine learning models are applied with scikit-learn library. Computer vision methods are performed with OpenCV. For example, Motion History Image is performed by functions like cv2.absdiff, cv2.threshold, and cv2.updateMotionHistory. Finally, the novel CNN model is implemented using sklearn and tensorflow.

C. Feature Selection

TABLE 1. 5 Selected Features and Their generation Method

Feature	Example	Generation method
Human aspect ratio	1.5898	Gaussian Mixture Model
Body angle (degrees)	63.625	Pose Estimation
Moving pixel percentage	1.7752	Motion History Image
Feet (left) movement (pixels)	5.099	Pose Estimation
Head movement (pixels)	8.544	Pose Estimation

Table 1 lists the features selected and approaches used for their generation to prevent any potential corner cases and eventually make the models more robust. Each approach is optimized to minimize the effect of noise caused by the inexpensive camera settings. Before the feature engineering part, the brightness and contrast of each image are slightly increased to better apply the methods listed. Each of the inputs is normalized before the models take it as input. The process of extracting these features is outlined in the following paragraphs.

The human aspect ratio is calculated through the gaussian mixture model to extract the general area involving the person of interest. The gaussian mixture model is an unsupervised machine learning model that classifies data clusters into a few classes. Many gaussians are used, which include a center, a width, and a possibility. This model is applied through OpenCV to do background subtraction. Then, the general shape of the person is determined to calculate the height and width, eventually providing the human aspect ratio. To avoid noise in the dataset significantly affecting the aspect ratio, the final data eliminates outliers by smoothing the data. Since each image is continuous, this step does not influence the data quality.

Motion History Image (MHI) calculates the moving pixel percentage, which presents the general path movement of an image by stacking the absolute difference between the current frame and the last one in videos. The absolute difference from the past 20 frames occurs as a less intense layer in the corresponding MHI image for an image, while the change in the current image is the most intense and bright. The output is monochrome, where the black area is unchanged, the white area is the current change, and the grey area is the change in the past 20 frames. Figure 2 presents some comparisons of the original images and the MHI made from the original images.

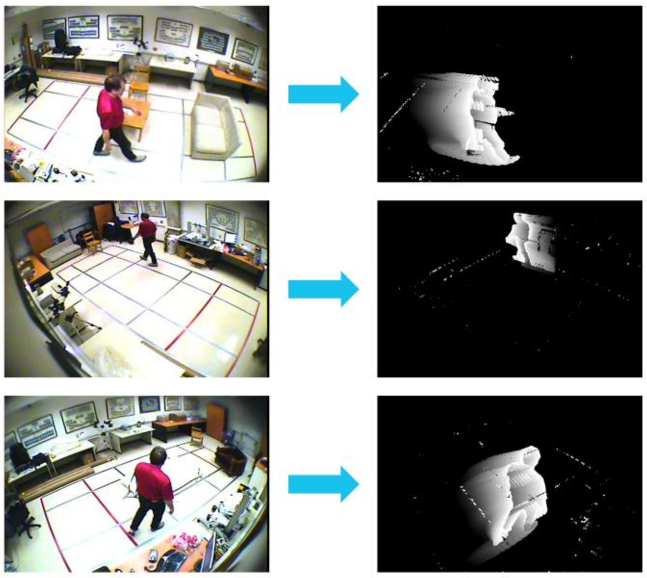

Fig. 2. Examples for the effect of MHI

From mathematical perspective, the formular of calculating the MHI is demonstrated below [14].

$$H_M(x,y,t) = \begin{cases} T & if\ \varphi_m(x,y,t) = 1 \\ H_M(x,y,t-1) - 1 & else \end{cases}$$

Note:

x = The change of position of pixels in horizontal direction
y = The change of position of pixels in vertical direction
$H_M(x,y,t)$ = Motion History Image (MHI)
$\varphi_m(x,y,t)$ = Motion update function (comparing frames)

EQUATION 1. MHI CALCULATION

The feature moving pixel percentage is calculated by finding the number of grey pixels divided by all pixels, indicating the general movement scale in a few frames (twenty frames in this case). This feature could prevent the misclassification of a person sitting down to a fall by determining its approximate velocity.
The feet and head movement are calculated similarly by applying mediapipe models to do pose estimation. The movement is calculated by finding the distance between the coordinate of left feet (x_1 and y_1) or head (x_2 and y_2) in the last frame and the coordinate in the current frame.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

EQUATION 2. POSE ESTIMATION

The angle of the person is also based on those two coordinates. This feature could prevent returning fallings in cases such as a person crouching or sitting down. The angle of the person perpendicular to the ground can be calculated by the geometry methods:

$$\tan^{-1}\theta = \frac{(y_2 - y_1)}{(x_2 - x_1)}$$

EQUATION 3. HUMAN ANGLE ESTIMATION

D. Conventional ML Models

As a baseline of comparison to the novel CNN approach, three conventional machine learning models were trained on the dataset. These models are chosen because of their popularity. They also vary in complexity, representing models not based on deep learning. They are applied to respond to a classification problem, which divides the cases of falling (positive case) and not falling (negative case). The following models were trained:

- SVM (Support Vector Machine): a machine learning algorithm that finds a hyperplane to divide the dataset into two classes. The hyperplane is found by finding the greatest possible margin between it and any data point in the dataset. [15]

- Decision Trees: a machine learning model that recursively evaluates features in each node to split the data. The predicted category is the mode of the categories on that leaf node after division from many children nodes. [16]
- Random Forest: an ensemble of many largely unrelated decision trees; the low correlation between those decision trees reduces individual errors, making the entire decision tree more accurate and stable. The prediction is eventually made by taking the result that most decision trees agree on. [15]

Each model is trained on the structured file consisting of the five extracted features. The hyperparameters are tuned with a grid search algorithm, which is an algorithm that searches through a manually assigned space of hyperparameters.

E. Multilayer Perceptron Model

As another comparison to the CNN approach, multi-perceptron, another deep learning model, is trained based on the pre-extracted features. Similar to CNN, it uses backpropagation, which learns to minimize the value of a cost function through iterations in each layer. While being a feedforward neural network, it only contains fully connected layers, not containing convolution layers and pooling layers like a CNN does.
A multilayer perceptron model is chosen here because of its simple structure difference from CNN. This difference makes it effective as a baseline model.

F. CNN Approach

A novel CNN model for fall detection is applied. Unlike the machine learning models above, CNN takes grid data as inputs, which in this case, are images. Therefore, the dataset is put into the CNN model in image format. On the one hand, it maximizes the available information input into the model, while on the other, it largely deaccelerates the training process and increases the risk of noise influence. Before the training, the raw image data is normalized and resized to perform a shorter training time.
Fig.3 depicts the architecture of the neural network

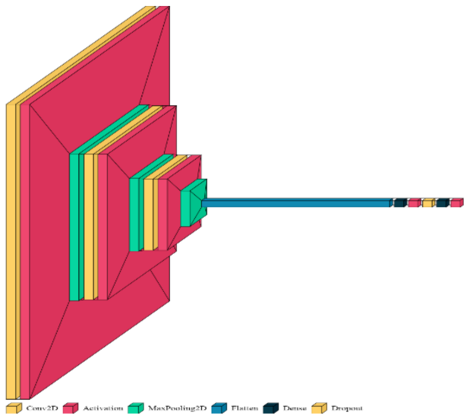

Fig. 3. Diagram of the architecture of the CNN model

The model started with three convolutional layers, which apply different numbers of 2 * 2 kernels to the input image. Each of these three convolutional layers and their activation layers is followed by a two-dimensional pooling layer which down-samples the input by taking the maximum value over a 2 * 2 window. Following this, a flattened layer converts the two-dimensional input into a one-dimensional vector that goes into the dense layers. The three dense layers (fully connected layers) feature different numbers of neurons and apply linear transformations to the input. Following the dense layers is the dropout layer, which randomly sets 50% of the input data into 0 to prevent overfitting. As a regularization method, this dropout layer prevents the model from overfitting the data. Eventually, an output layer gives a prediction of the probability of the falling happening.

The binary cross entropy loss function by Keras was used to realize this. The final output layer used sigmoid as its activation function. The training metrics were accuracy and F1 score. An early stopping callback was also used to prevent overfitting.

An optimization of RMSprop (Root Mean Square Propagation) was applied to train the model. The step size for each weight was adjusted to achieve better learning efficiency. In order to prevent the learning rate from becoming too small, RMSprop used the moving average of the squared gradient to scale the learning rate instead of just the gradient.

The model first encountered significant overfitting issues. To address this issue, a simpler model structure is trained with relatively fewer neurons and layers. Data augmentation is also applied to training data to increase the robustness of the model. This allows 0.2 sheer range and 0.2 zoom range, while a horizontal flip is allowed. This is intended to reduce the lack of diversity in the dataset, as mentioned in earlier sessions.

The F1 scores and accuracies from this CNN model were compared with conventional machine learning models introduced earlier in this paper.

G. F1 Scores

The models are each evaluated and compared with the F1 scores and accuracy. Accuracy is a metric that describes how many times a model made a correct prediction in the entire dataset. While accuracy has been used as the only metric for years, it fails to remain valid if the dataset is not class balanced. For instance, if only ten percent data is marked positive in the dataset, the accuracy would be high even if the model predicted everything positive.

Therefore, the application of the F1 score is introduced, for a more complete overview of the performance of models. F1 scores are the harmonic mean of precision and recall, which is presented in Equation 4 below:

$$F_1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

EQUATION 4. F1 SCORE

Unlike geometric or arithmetic means, harmonic means are more affected by any extreme values. A harmonic mean is taken here to keep a balance between precision and recall since predicting a non-existing fall and not predicting a fall both cause trouble in real life. Precision generally means the number of positive data among all data predicted positive, while recall means the predicted positive cases in all input positive cases. If they are interpreted in the F1 score, Equation 4 reduces to:

$$F_1 = \frac{true\ positive}{true\ positive + \frac{1}{2}(false\ positive + false\ negative)}$$

EQUATION 5. ADJUSTED F1 SCORE

The F1 scores are combined with model accuracy to give a thorough overview of the models' performance. The evaluation and comparisons that follow are based on both of these metrics.

IV. RESULTS

The F1 scores obtained are presented in Table 2 and Fig. 4.

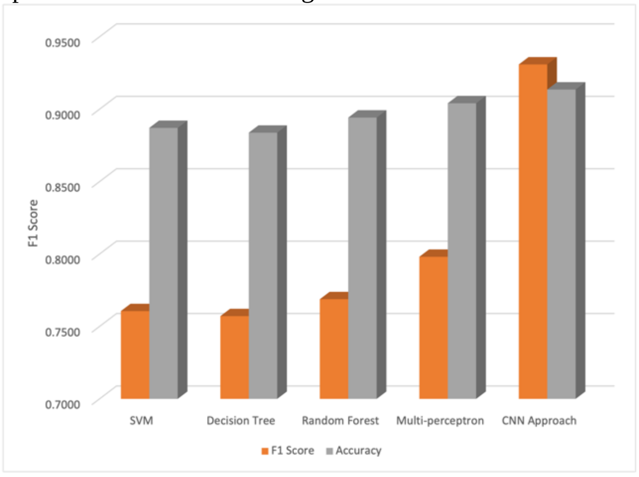

Fig. 4. F1 Scores and Accuracies

A. Conventional Machine Learning Models

A summary of the performance of each of the three conventional machine learning models is shown in Table 2: the SVM obtained a 0.8871 accuracy, the decision tree obtained 0.8838, and the random forest achieved 0.8942. The difference becomes closer when looking at the F1 scores. The F1 score SVM obtained is 0.7606, 0.7571 for the decision tree model and 0.7688 for the random forest model.

Conventional machine learning models performed relatively reliably after the image data was taken as structural data. Each model varies in complexity and mechanism, as introduced earlier in the paper. Though the differences are highlighted in the accuracy values, it is not clearly underlined in the models' F1 scores. SVM performs at 0.7606, slightly higher than a more easy and interpretable structure like the decision tree, which obtained a 0.7571 F1 score. Also, as an ensemble of decision trees, the random forest model consistently performs the best on both on F1 score and accuracy, with a 0.7688 F1 score and 0.8942 accuracies, proven to be more robust than the other two on the current unbalanced dataset. However, the F1 scores of all three of them are still evidently lower than the accuracy, which means either low precision or low recall. This suggests that those three conventional machine learning models are good at identifying the majority class correctly but didn't perform as well at identifying the minority class, which is the falling cases. This brings a risk that people are not detected by the model when they are fell on the ground.

Table 2. F1 Scores and Accuracies of Each Machine Learning Model

Evaluation Method	Conventional ML Models			Multi-perceptron	CNN Approach
	SVM	Decision Tree	Random Forest		
F1 Score	0.7606	0.7571	0.7688	0.7981	0.9309
Accuracy	0.8871	0.8838	0.8942	0.9041	0.9136

B. Multilayer Perceptron

An additional deep learning model is the multilayer perceptron, which is trained on the pre-extracted features. Consisted of three inter-connected layers and corresponding dropout layers, the model performed 0.9041 accuracy and 0.7981 f-1 score. The accuracy is slightly better than all three models above. The bigger difference is at the F1 score, which outperforms the other three by an obvious gap, showing a better ability to generalize well on both majority and minority classes.

C. CNN Classification

After testing with different model structures and data augmentations, the F1 score and accuracy were taken from the model performance of a test set.

The results are as follows: the accuracy obtained is 0.9136, and the F1 score acquired is 0.9309. The f1 score dramatically outperforms the conventional machine learning models and multilayer perceptron, while the accuracy also outperforms all other models by a smaller difference.

The model performed similar accuracy and F1 score on the training set, validation set, and test set, indicating that the model is fitting the data well. It suggests that the model is able to generalize well to unseen data. The F1 score of the model is evidently better than any other model tested, suggesting a good performance of the model on an unbalanced dataset like this. Though some oscillations were observed on the learning curve, the model eventually converged to an acceptable point. The consistent performance on the test dataset also suggests that the model is not overfitting or underfitting.

D. Discussion of Results

The CNN model devised consistently outperforms other models in the classification of falls in both accuracy and F1 scores. This indicates its ability to accurately identify positive instances while also minimizing the number of false positive predictions.

The evident performance difference could be attributed to the advantages of CNNs. CNNs are able to process large amounts of data efficiently, which is important to this case because of the huge size of the dataset. Additionally, CNNs are known for their ability to capture complex patterns and features in the data that may be difficult for traditional models to detect. In the case of fall detection, various camera angles made the data hard to be predicted by conventional machine learning algorithms, especially with the noise caused by the inexpensive IP cameras.

Unlike multilayer perceptron, which only has fully connected layers, CNNs are typically not fully connected. In contrast, CNNs always use a sparse connectivity pattern, which means that neurons in one layer are only connected to a subset of the neurons in the next layer. This is achieved by convolutional layers, as each filter is small-sized and covers a small region of the input data. Therefore, convolutional layers are able to learn local patterns in the images and ignore minor distortions and position changes by applying filters to the input images. Furthermore, the combination of successive conventional layers leads to a gradually more abstract representation of the data layer to layer.

V. CONCLUSION

In this paper, video data from inexpensive IP cameras are used to train conventional machine learning models, a multilayer perceptron, and a CNN model. The CNN model largely outperforms other models with higher accuracy and F1 score.

There are many potential applications of this model. For example, an affordable and reliable system consisting of single cameras at home to monitor the falling risks of family members. With the model deployed on a server, the system can send warnings to other members' mobile devices. The system can be updated on the server if the setting changes; otherwise, the system just works without major disruptions to everyday life. This system is especially useful in families with elder members who have a higher risk of falling.

In addition, the systematic comparison of machine learning models on a single inexpensive camera-based fall detection system can serve as a foundation for future research. The effectiveness of CNN models under the single inexpensive RGB camera setting is demonstrated in this comparison. This finding allows researchers to build more robust and accurate neural networks on single-camera fall detection at an affordable cost.

Some potential for future research includes using a more diverse dataset, which involves scenes in different lighting and occlusion conditions. Pose estimation could be integrated into the system, making the model more robust and more accurate under multiple-person scenarios. Transfer learning is also a possible solution to build a more robust model on the current dataset.

This research provides a systematic comparison of machine learning models under single camera dataset. The source of the dataset makes a wider application possible by successfully combining it with neural networks. It illustrates the relatively higher efficacy of neural networks, making them valuable to positively impact fall detection and intervention tasks.

REFERENCES

[1]. "Ageing and health." https://www.who.int/news-room/fact-sheets/detail/ageing-and-health (accessed Jan. 03, 2023).

[2]. P. Kannus, H. Sievänen, M. Palvanen, T. Järvinen, and J. Parkkari, "Prevention of falls and consequent injuries in elderly people," *The Lancet*, vol. 366, no. 9500, pp. 1885–1893, Nov. 2005, doi: 10.1016/S0140-6736(05)67604-0.

[3]. CDC, "Keep on Your Feet," *Centers for Disease Control and Prevention*, Dec. 16, 2020. https://www.cdc.gov/injury/features/older-adult-falls/index.html (accessed Jan. 03, 2023).

[4]. Z. Zhang, C. Conly, and V. Athitsos, "A survey on vision-based fall detection," in *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, Corfu Greece, Jul. 2015, pp. 1–7. doi: 10.1145/2769493.2769540.

[5]. T. B. Rodrigues *et al.*, "Fall Detection System by Machine Learning Framework for Public Health," *Procedia Computer Science*, vol. 141, pp. 358–365, 2018, doi: 10.1016/j.procs.2018.10.189.

[6]. S. Usmani, A. Saboor, M. Haris, M. A. Khan, and H. Park, "Latest Research Trends in Fall Detection and Prevention Using Machine Learning: A Systematic Review," *Sensors*, vol. 21, no. 15, p. 5134, Jul. 2021, doi: 10.3390/s21155134.

[7]. F. Shu and J. Shu, "An eight-camera fall detection system using human fall pattern recognition via machine learning by a low-cost android box," *Sci Rep*, vol. 11, no. 1, p. 2471, Jan. 2021, doi: 10.1038/s41598-021-81115-9.

[8]. E. Auvinet, C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Multiple cameras fall data set".

[9]. E. Sukma Wahyuni, M. Brado Frasetyo, and H. Setiawan, "Combination of Motion History Image and Approximated Ellipse Method for Human Fall Detection System," *International journal of simulation: systems, science & technology*, Jan. 2019, doi: 10.5013/IJSSST.a.19.03.13.

[10]. C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Robust Video Surveillance for Fall Detection Based on Human Shape Deformation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 5, pp. 611–622, May 2011, doi: 10.1109/TCSVT.2011.2129370.

[11]. Md. M. Islam *et al.*, "Deep Learning Based Systems Developed for Fall Detection: A Review," *IEEE Access*, vol. 8, pp. 166117–166137, 2020, doi: 10.1109/ACCESS.2020.3021943.

[12]. S. Chhetri, A. Alsadoon, T. Al-Dala'in, P. W. C. Prasad, T. A. Rashid, and A. Maag, "Deep learning for vision-based fall detection system: Enhanced optical dynamic flow," *Computational Intelligence*, vol. 37, no. 1, pp. 578–595, Feb. 2021, doi: 10.1111/coin.12428.

[13]. M. Salimi, J. J. M. Machado, and J. M. R. S. Tavares, "Using Deep Neural Networks for Human Fall Detection Based on Pose Estimation," *Sensors*, vol. 22, no. 12, p. 4544, Jun. 2022, doi: 10.3390/s22124544.

[14]. E. Chen, S. Zhang, and C. Liang, "Action Recognition Using Motion History Image and Static History Image-based Local Binary Patterns," *IJMUE*, vol. 12, no. 1, pp. 203–214, Jan. 2017, doi: 10.14257/ijmue.2017.12.1.17.

[15]. E. Y. Boateng, J. Otoo, and D. A. Abaye, "Basic Tenets of Classification Algorithms K-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review," *JDAIP*, vol. 08, no. 04, pp. 341–357, 2020, doi: 10.4236/jdaip.2020.84020.

[16]. S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques".