Combination of Optical Flow and Rank Pooling for Vision Based Fall Detection System

Vian Sebastian Bromokusumo
Department of Computer Science and Electronics
Universitas Gadjah Mada
Yogyakarta, Indonesia
viansebastianbromokusumo@mail.ugm.ac.id

I. Introduction

Falls pose a significant health risk, especially to the elderly, leading to serious injuries such as fractures, head trauma, and even loss of consciousness. The consequences include decreased mobility, loss of independence, and an increased fear of falling again. The World Health Organization [1] identifies falls as the second leading cause of accidental injury deaths globally, highlighting the vulnerability of the elderly due to age-related factors..

Computer vision has emerged as a promising solution to enhance elderly safety. Leveraging advanced image processing and Convolutional Neural Networks (CNNs), these systems can automatically detect falls in real-time, providing immediate alerts to caregivers. This technology ensures timely intervention, reducing the risk of serious injury and prolonged periods without assistance [2].

Traditional video analysis methods for fall detection are burdened by high computational demands and often suffer from accuracy issues, leading to false positives or missed falls. The need for significant processing power and continuous monitoring makes these approaches less feasible for real-time applications [2].

Recent advancements, such as rank pooling, optical flow, and object tracking, address these challenges by capturing temporal information, measuring motion between frames, and enhancing detection accuracy. These techniques enable the development of lightweight, efficient vision-based fall detection systems, providing reliable and timely alerts without high computational costs [3].

This work proposes a method to accurately detect fall through video input with the aim of maximum efficiency and lighten computational burden using sophisticated preprocessing techniques.

II. Method

The method proposed in this paper is taken to lighten the computational burden in video analysis. [3] introduced the use of optical flow, rank pooling, and further energy calculations to achieve what they call 'Enhanced Optical Dynamic Flow'. The approach of using rank pooling in action recognition is also supported by [2]. For this paper, the preprocessing method will implement frame extraction, optical flow calculation, and rank pooling. It is then fed to a transfer learning model, consisting of the EfficientNetV2 architecture [4] and custom convolution, maximum pooling, and output layers. It is then evaluated with the accuracy, precision, recall, and F1-score metric. The overall flow of the work in this paper can be seen in Fig. 1.0 below.

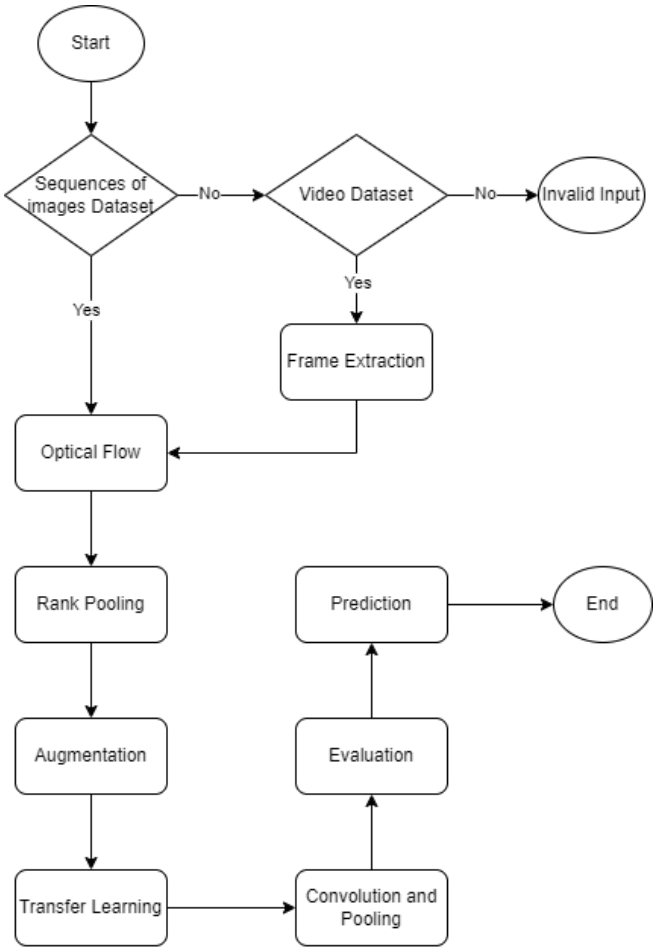

Fig 1.0: project flow visualization

A. Data Acquisition

First, the dataset is obtained from two main sources, the Multiple Cameras Dataset [5] (Multicam) and the UR Fall Detection Dataset [6] (URFD).

The Multicam dataset is composed of 24 scenarios, with 23 falling scenarios, and one daily activity scenario. Each scenario consists of eight videos, coming from eight different angles in a controlled environment. The videos are recorded in 120 fps, as it is slightly sped up. Based on the dataset documentation [5], the angles and environment setting can be seen in Fig. 2.0 below..

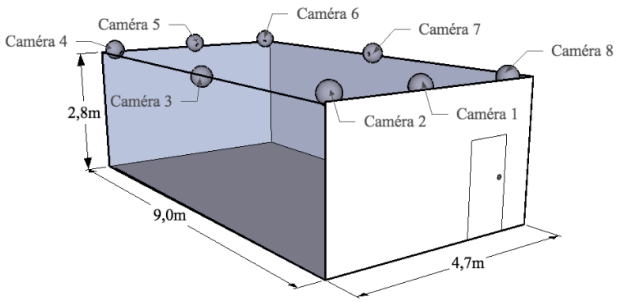

Fig. 2.0: environment setting

In the case of the URFD dataset, it is composed of 70 sequences of images, 30 being fall sequences and 40 being daily activities sequences. This dataset is recorded in 25 fps in video form, but the video itself is not provided in the source.

B. Data Preprocessing

The preprocessing methods done for the data used are frame extraction, optical flow calculation, rank pooling, and image augmentation.

Optical flow [7] is a pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene. In short, it calculates the vector field that represents the movement of each pixel from one frame to the next in a sequence of images, aiding in object tracking, motion estimation and analysis, and more. Rank Pool [2], is a pooling method that captures the temporal progression features within a video and outputs a single vector or image that represents the dynamics of the sequences. To simplify, it can be said that rank pool summarizes a sequence of images, capturing the temporal features, into a single vector or image.

[3] proposed the use of Enhanced Optical Dynamic Flow images, where the optical flow images are rank pooled, and further energy calculations are performed. In this paper, the method used will be limited to the rank pooled optical flow images. This method is used to highlight the use of optical flow, as well as to lighten the computational burden.

The output final dataset after performing the mentioned preprocessing method above are 46 daily activity and 53 fall optical flow rank pooled images. This is one of the disadvantages of the method as the dataset needed ideally would require a very large amount of videos or sequences of images. To combat this, image augmentation is used. It is very important to note that vertical flip is not enabled, as a 'fall' is defined by the increase in negative y-axis. The source code for the processing using histogram equalization and Gaussian blur can be seen as follows.

C. Modeling

Transfer Learning [8] is the use of pre-trained models in building a model. This aids in generalization improvement, significant increase in model performance, reduced training time, as well as reduction in computational burden. This is because pre-trained models are usually trained in very large datasets and have gone through optimizations to perform well in various usages.

EfficientNetV2S [9][10] is a 55-layer neural network architecture that is generated by a combination of scaling (width, depth, resolution) and neural architecture search. The development of the EfficientNetV2 is focused on speed and parameter optimization from the EfficientNet family by using Fused-MBConv layers which combine depthwise convolutions and pointwise convolutions, improving memory cost and efficiency. The EfficientNetV2 comes in three sizes, EfficientNetV2S (55-layers, focused on efficiency), EfficientNetV2M (85-layers, focused on balance), and EfficientNetV2L (115-layers, focused on acc\uracy). The base architecture of EfficientNetV2 can be seen in Fig. 3.0 below.

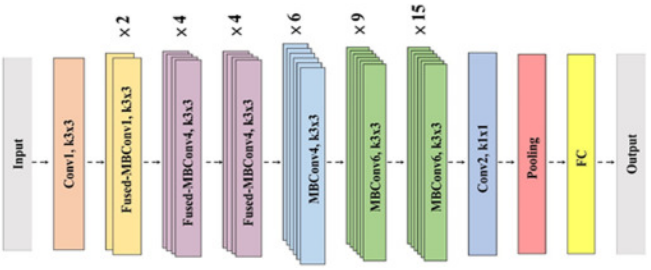

Fig. 3.0: base architecture of EfficientNetV2

Using the EfficientNetV2S as the pre-trained model on the ImageNet dataset, more layers are added to customize the model in detecting fall tasks. The extra layers added are Conv2D, GlobalAveragePool, MaxPool, BatchNormalization, Dropout, and Dense layers.

Finally, the model is compiled with the Adam optimizer, categorical cross entropy loss function, and accuracy metric

D. Prediction

The final prediction made in this paper takes a video as an input. Following training pipeline flow, first the frames are extracted. Then the optical flow is computed, and the optical flow images are rank pooled. This will be the input for the prediction, and the prediction result will be overlaid in every frame, then it is reconstructed back to video. The final output will be a video with the prediction result (Fall or Not Fall) printed on the video. The result video is linked.

E. Evaluation

The evaluation metrics used in this paper are Accuracy, Precision, Recall, and F1-Score. Although in the context of video analysis, Sensitivity and Specificity are also known metrics, they are not used in the evaluation as they can be inferred and calculated when knowing the Precision and Recall.

III. Results and Discussions

First, we can observe the training process in Fig. 4.0 below.

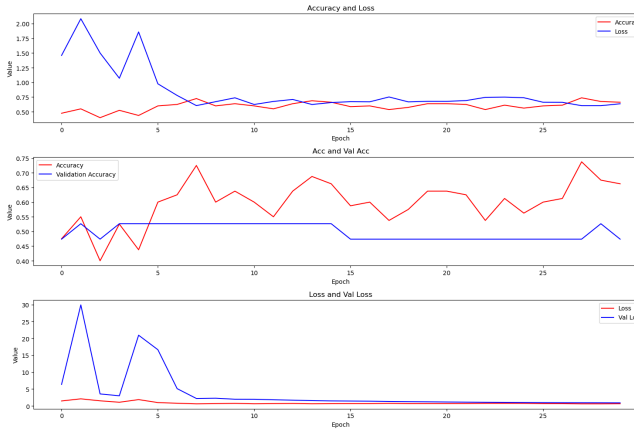

Fig. 4.0: training history metrics

It can be seen from the visualization that from the beginning, the training accuracy increases but the validation accuracy is stagnant, then decreases, with the highest recorded accuracy to be 0.82 and validation accuracy to be 0.52. This is a strong indication of overfitting. Potential reasons for them are as follows.

Considering that the final dataset used in for the model is 99 optical flow rank pooled images (46 ADL and 53 Fall), it can be concluded that the dataset is insufficient. In addition to that, various preprocessing methods can be applied further such as noise removal, segmentation, and many more. Another potential reason for this would be the nature of EfficientNetV2 [9][11], as well as improper use of transfer learning.

EfficientNetV2 is an architecture that relies on scaling, progressive learning, Squeeze-and-Excitation Blocks, and Fused-MBConv. Emphasizing on the progressive learning, EfficientNetV2 gradually increases input resolution as the training goes on. When the image input is noisy and underprocessed, unsatisfactory results may occur. In the case of transfer learning misuse, the pretrained weights used is the ImageNet, and the input is a rank pooled optical flow image. The big difference between ImageNet dataset and input may not transfer well, hence the cause of the underperformed performance.

We can observe the metrics results from table 1.0 below.

	Precision	Recall	F1	Support
RP_ADL	0	0	0	9
RP_FALL	0.53	1.00	0.69	10
Macro avg	0.26	0.50	0.34	19
Weighted avg	0.28	0.53	0.36	19
Accuracy	0.53			

Table 1.0: evaluation metrics.

Based on the table above, we can infer that the model barely predicts or misses all predictions towards class RP_ADL. This is supported by the corresponding 0 value between both precision and recall in this class. In addition to that, the precision for RP_FALL class is 0.53 and recall is 1.00. This means that the model correctly identifies all classes of RP_FALL, while out of all the predictions, only 53% were actually RP_FALL. The overall accuracy, macro average, and weighted average were also poor, and based on the results, we can infer that the model is biased to the RP_FALL class.

A. Conclusion

Based on the evaluations done, it can be concluded that the model is biased towards the RP_FALL class. The reason for this is due to an imbalanced dataset between RP_ADL and RP_FALL. Although it may not seem much, with a mere difference of 6 images. it must be noted that this is the rank pooled image coming from the optical flow images. Based on the output and debugging during the optical flow calculations, there are 8,647 frame pairs calculated for the RP_FALL, but an overpowering 14,376 frame pairs for the RP_ADL. This resulted in an imbalance in the amount of features present in the final rank pooled image of each class. While the model needs further processing to predict the RP_ADL class, the RP_FALL is easily calculated. Combined with the nature of EfficientNetV2 in gradual resolution increase, the input becomes inconsistent, therefore the results are as such.

It is worth noting that using this approach has significant advantages as well. From the original total of 266 videos or folders of sequences of images, it was reduced to only 99 images. This approach indeed increases efficiency and decreases computational power substantially. The recorded training time was only three minutes, and in the case of video analysis is incredibly fast.

For future improvements, some notes must be taken especially when taking this approach. Firstly, ensure that the dataset has a balanced amount of frames. This would create a balanced dataset, decreasing bias possibilities. Second, use more data. This method can ‘summarize’ a very large dataset into few images, and traditional deep learning needs a lot of data. Third, perform deeper preprocessing techniques to segment and further clean image data by reducing noise, etc. Lastly, perform deeper hyperparameter tuning to find the optimal model for this specific dataset and task.

In conclusion, this approach highlights video preprocessing that significantly increases efficiency and reduces computational requirements to produce accurate results. Considering the strengths and flaws, this approach is best applied when there is abundant video data in terms of quantity and/or size.

References

[1] World Health Organization. (2018). Falls. Retrieved from https://www.who.int/news-room/fact-sheets/detail/falls

[2] Fernando, B., Gavves, E., Oramas, J., Ghodrati, A., & Tuytelaars, T. (2017). Rank Pooling for Action Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4), 773-787.

[3] Chhetri, S., Alsadoon, A., Al‐Dala'in, T., Prasad, P. W. C., Rashid, T. A., & Maag, A. (2020). Deep learning for vision‐based fall detection system: Enhanced optical dynamic flow. *Computational Intelligence*. https://doi.org/10.1111/coin.12428

[4] Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller Models and Faster Training. arXiv preprint arXiv:2104.00298. https://arxiv.org/abs/2104.00298

[5] Auvinet, E., Rougier, C., Meunier, J., St-Arnaud, A., & Rousseau, J. (2010). *Multiple Cameras Fall Dataset*. Technical report 1350, DIRO - Université de Montréal..

[6] Kwolek, B., & Kepski, M. (2014). *Human fall detection on embedded platform using depth maps and wireless accelerometer*. Computer Methods and Programs in Biomedicine, 117(3), 489-501. https://doi.org/10.1016/j.cmpb.2014.09.005

[7] Barron, J. L., Fleet, D. J., & Beauchemin, S. S. (1994). Performance of Optical Flow Techniques. International Journal of Computer Vision, 12(1), 43-77.

[8] Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. Journal of Big Data, 3(1), 9. doi:10.1186/s40537-016-0043-6.

[9] Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller Models and Faster Training. arXiv preprint arXiv:2104.00298

[10] EfficientNetV2. (n.d.). Retrieved from https://keras.io/api/applications/efficientnet_v2/#efficientnetv2s-function

[11] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv preprint arXiv:1905.11946.

Output Video Link

https://drive.google.com/drive/folders/1Qwoif90Krefc1ahaG-TbIvCUhBQPYx5y?usp=sharing

Notebook Link

https://drive.google.com/file/d/1QkAqKqLJ-VwkL7eO3MBq4rwN44uP1oGu/view?usp=sharing

(make sure to choose ‘Open with Google Collaboratory’)