EasyChair Preprint

№ 11799

Human Actions Recognition System Based on Neural Networks

Juan Brito and Rigoberto Salomón

January 18, 2024

Human actions recognition system based on Neural Networks

XXXXXXXXXXX

[1] Princeton University, Princeton NJ 08544, USA
[2] Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany
lncs@springer.com
http://www.springer.com/gp/computer-science/lncs
[3] ABC Institute, Rupert-Karls-University Heidelberg, Heidelberg, Germany
{abc,lncs}@uni-heidelberg.de

Abstract. The recognition of human activities in videos is a relevant area of study due to its real-life applications, such as surveillance, security, healthcare, human-machine interaction, and monitoring. This research compares two recognition approaches, called simple and hybrid, using two specific data sets. The first set includes three classes: yoga, exercise, and dance; the second is a sample from the Kinetics-700 set, with five activities, four of which are violent. Both sets present low variability between classes and high variability within classes. To reduce computational costs, a pre-trained CNN model and simple techniques for reducing computational resources are used. The hybrid approach uses an additional model with three variants: GRU, LSTM, or BiLSTM. Even though all the models presented similar results, the simple approach, using a pre-trained architecture and a reconstructed top-head, proved to be the most effective, reaching an accuracy of 94%, while the hybrid approach using LSTM layers obtained 90%. The model demonstrated an adequate classification of violent activities, which could serve as a basis for developing a surveillance and security systems.

Keywords: Pretrained CNN · LSTM · GRU · BiLSTM.

1 Introduction

The field of computer vision and machine learning has made significant strides in predicting and recognizing human actions, with applications in sectors like security and intelligent rehabilitation. The abundance of online video data has played a crucial role in refining video classification accuracy (14). Action recognition, a complex process, involves two main tasks: action representation and action classification. The former converts video input into a machine-understandable representation, while the latter uses this representation to infer the performed action (23). Artificial Neural Networks (ANNs) have been instrumental in recognizing complex patterns within data, contributing to the successful prediction and classification of human activities (16; 7; 12; 15). This study uses the TensorFlow library to develop ANN models for differentiating various human actions,

considering factors like accuracy, computational cost, and classifiable activities. Addressing video classification challenges has led to the development of scalable models (5; 2; 1). The research aims to evaluate two video classification approaches based on pre-trained 2D convolutional neural network and RNN architectures for identifying human actions in datasets with large inter-class and small intra-class variability.

2 Related Work

Video classification, a branch of computer vision, involves devising algorithms for the automatic categorization of videos based on visual content (8). It's a challenging task due to video data's high dimensionality and temporal nature, requiring advanced feature extraction methods and modelling of temporal dynamics. This field has gained interest owing to applications in video surveillance, sports analytics, and education, among others. The actions in videos for the machines are nothing more than a set of pixels so computers; therefore the first thing is to look for an adequate representation and then infer the action to be performed. These two problems constructing an adequate representation of the video as input and inferring from this representation what action it is performing (8). The process of transforming video data into a feature-encapsulating vector or a sequence of vectors is termed action representation (10; 9; 13). Subsequently, action classification deals with understanding the content of the video (18; 11). Multiple techniques, particularly in deep learning, successfully integrate these two components. Action recognition can be broadly categorized into Shallow Approaches, Deep Architectures as depicted in Figure 1 (8).

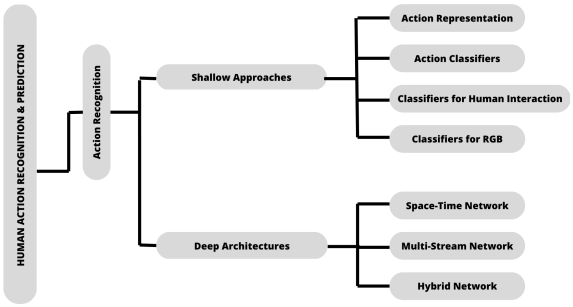


Fig. 1. General overview of the different human action recognition and prediction approaches.

3 Deep Learning Approaches

Deep learning methods address limitations of traditional features, modeling complex relationships and learning powerful features useful in video classification tasks (14). 3D CNN models, despite training challenges, offer effective spatiotemporal feature learning and excel in action and object recognition (17; 21). Multi-stream networks use multiple CNNs to model appearance and motion information, integrating spatial fusion functions and residual connections to overcome interaction deficiencies (20; 3; 4). Hybrid networks, merging CNNs and RNNs, capture spatial, temporal, and long-range dependencies, and apply enhancements like spatiotemporal graph convolution and attention models to further improve learning of structural and temporal information (24; 6; 22; 19). Nonetheless, they grapple with challenges like variations, cluttered background, camera motion, and uneven predictability.

4 Methodology

4.1 Movement Mix Dataset

Is a custom collection of high-definition TikTok and YouTube videos, with various resolutions, sizes, and a 30fps average frame rate. It features 172 training/validation and 29 test videos across three categories: dance, exercise, and yoga. The categories span multiple styles, environments, and exercises, leading to significant overlap and complexity in classification due to variations in body positions.

4.2 Danger Kinetics Dataset

A subsample of the large Kinetics-700 video collection, features 650,000 clips across 700 human action classes. It consists of YouTube videos of human-object and human-human interactions. Five classes - Punching Bag, Punching Person (boxing), Slapping, Throwing knife, and Walking through the snow - were selected for analysis of violent activities useful for surveillance. The classes encompass both similar and dissimilar actions, with the last one being non-violent.

4.3 Metrics

The study evaluates video classification using Accuracy, Precision, Specificity, and Sensitivity. Accuracy gauges overall correctness. Precision and Sensitivity assess correct identification of positive cases, while Specificity measures correct identification of negative cases.

4.4 Proposed Solution

This study aims to devise an efficient video classification model, focusing on neural network-based methods. Special emphasis is placed on minimizing computational demand.

1. Utilizing pre-trained 2D CNNs on individual frames.
2. Combining pre-trained 2D CNNs with GRUs.
3. Integrating pre-trained 2D CNNs with LSTMs.
4. Employing pre-trained 2D CNNs along with BiLSTMs.

4.5 Data Preprocessing

Both datasets utilized in the proposed models underwent identical preprocessing. A maximum of 220 frames per video was established, with each frame vectorized and labeled. To reduce noise, the first and last ten frames were removed, the frame rate was reduced to a third, and videos with potential mislabels or excessive noise were eliminated. All videos were resized to 224x224 without cropping and shuffled to ensure balanced distribution. The datasets were then partitioned into training (80%) and validation sets (20%).

1. For the first dataset: 172 videos for training, divided into exercise (68), dance (53), and yoga (52) classes, and 29 for testing, split into dance (8), exercise (10), and yoga (11) classes.
2. For the second dataset: 250 training videos, with 50 from each class, and approximately 65 for testing, with around 13 videos from each class.

4.6 Simple Approach

The simple approach used a two-part methodology: selection of a pre-trained convolutional neural network (CNN) model, and the reconfiguration of this model to suit video classification. The pre-trained model chosen was EfficientNetB0, selected for its excellent performance in various image classification tasks, and its efficiency owing to its optimized depth, width, and resolution. EfficientNetB0's architecture combines convolutional layers, depthwise separable convolutions, and squeeze-and-excitation blocks, enabling efficient feature extraction.

Top Head of Model For the simple approach, EfficientNetB0 was used for feature extraction, and a custom classifier was built as the top head of the model. The top head consists of six layers: Global Average Pooling, Batch Normalization, two Dropout layers, and two Dense layers. The model was trained using both datasets with ReLU and softmax activation functions, Stochastic Gradient Descent optimization, and a batch size of 16 for a maximum of 50 epochs. Early stopping was implemented to prevent overfitting, and performance metrics including accuracy, recall, precision, and specificity were used for evaluation. In the training and validation process, the model was trained twice, once for each

dataset, with the final dense layer's size varying. It is important to note some aspects of the training process, which are summarized in Table 1. The model comprises 4,219,304 total parameters. Out of these, 167,173 are trainable, and 4,052,131 are non-trainable. These parameters represent the model's complexity and the amount of data required for effective training. The model's evaluation process involves computing the mean of the probability vectors obtained from the trained model, owing to the use of a 2DCNN model and the aim of video classification. Figure 2 illustrates the process, showing the main steps in model building and the evaluation process.

Table 1. Summary of the Training and Validation Process

Aspect	**Details**
Activation functions	ReLU in the dense layer and softmax in the output layer
Dropout layers	Two dropout layers with a dropout rate of 0.4
Epochs	50
Optimization algorithm	Stochastic Gradient Descent (SGD) with a learning rate of 1e-4
Batch size	16
Early stopping	Monitors validation accuracy with a patience of 10 epochs

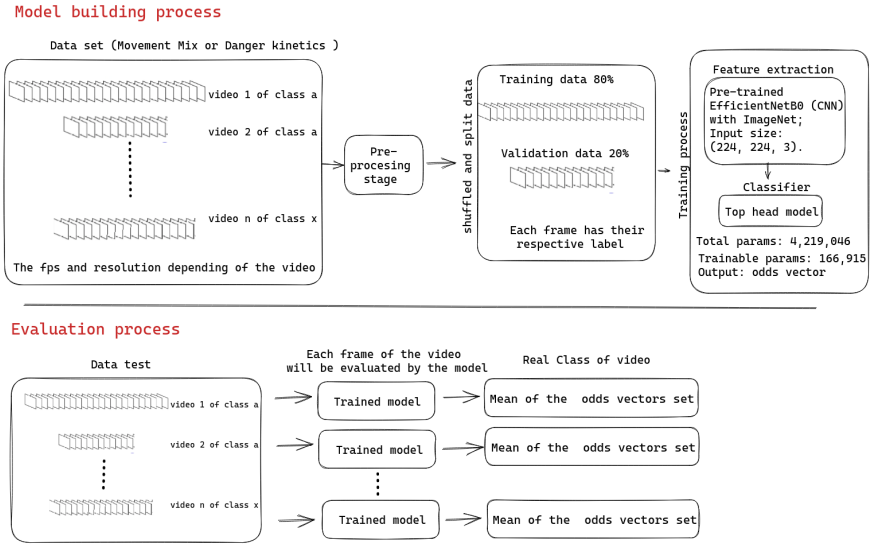


Fig. 2. Pipeline of the simple approach, encompassing both the model construction and the evaluation process.

4.7 Hybrid Approach

The hybrid approach consists of three stages. First, the pre-trained Efficient-NetB0 model is used for feature extraction, similar to the simple method. In

the second stage, each video is converted into a set of feature vectors representing frames. Finally, in the third stage, various Recurrent Neural Network (RNN) models, including Gated Recurrent Units (GRUs), Long Short-Term Memory (LSTMs), and Bidirectional LSTMs (BiLSTMs), are applied for capturing temporal features. While this approach offers a richer understanding of time-dependent patterns, it comes with a higher computational cost due to the additional model. The preprocessing for RNN models in the hybrid approach encompasses two primary stages. In the first stage, dimensionality reduction is applied using a MaxPooling2D layer. The initial feature vector, extracted from the pre-trained EfficientNetB0 model, is reduced from 7x7x1280 to 6x6x1280, preserving essential features while decreasing data volume by approximately a factor of 13. The second stage of preprocessing involves preparing the videos as sequences of feature vectors. This procedure includes three primary components: (1) a feature vector array obtained by breaking down each video, (2) masks generated to manage sequences of varying lengths, and (3) encoded labels corresponding to each video. This preparation process is performed individually for each video and takes into account their varying lengths by defining a maximum frame limit and using masks to maintain uniform sequence lengths. The preprocessing stage results in the generation of feature vector arrays and mask vectors for each video, as well as the extraction of features from video frames using the EfficientNetB0 model. Ultimately, this process aims to ensure efficient computational processing and handle variable-length sequences in RNN models. Upon the completion of preprocessing, the extracted video features, masks, and encoded labels are used as inputs for RNN models such as GRU, LSTM, and BiLSTM, which are capable of analyzing temporal dependencies between frames for video classification. Figure 3 provides a schematic representation of the entire process. The figure 4 show it the architecture of the different RNN models, finally in the figure 5 it's possible observe the setting of the hyper parameters used during the training and validation process.

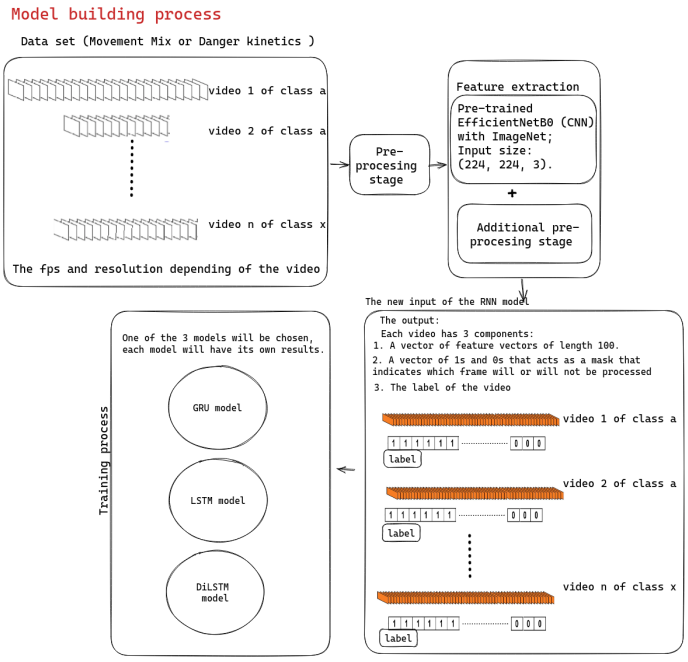


Fig. 3. Pipeline of the hybrid approach.

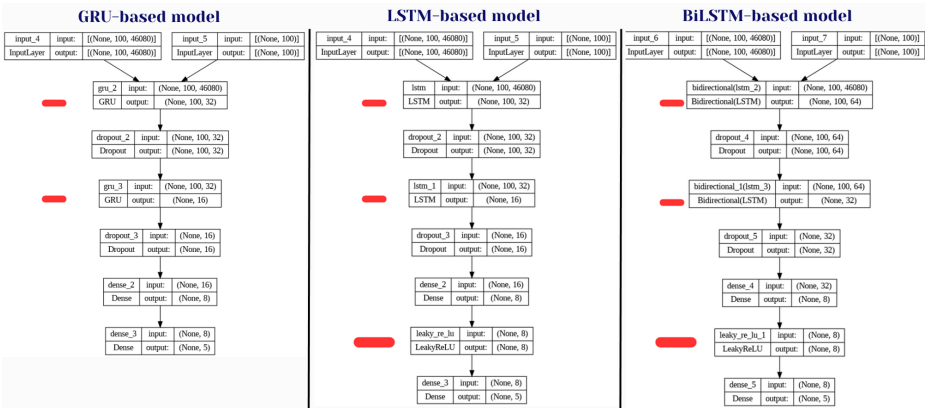


Fig. 4. Architecture of each of the RNN models.

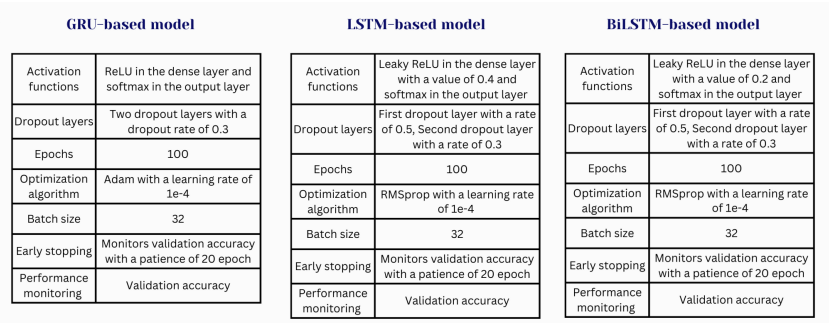


Fig. 5. Configuration for each of the RNN models during the training and validation process.

5 Summary of Experimental Analysis

The experimental analysis evaluates the effectiveness of various video classification methods on two distinct datasets. These include the simple frame approach and three hybrid models: 2D CNN+GRU, 2D CNN+LSTM, and 2D CNN+BiLSTM. The purpose is to identify the most suitable technique for video classification tasks.

5.1 Simple Approach

First Dataset (Movement Mix) This dataset is well-balanced, with a similar number of frames per class. The distribution is as follows:

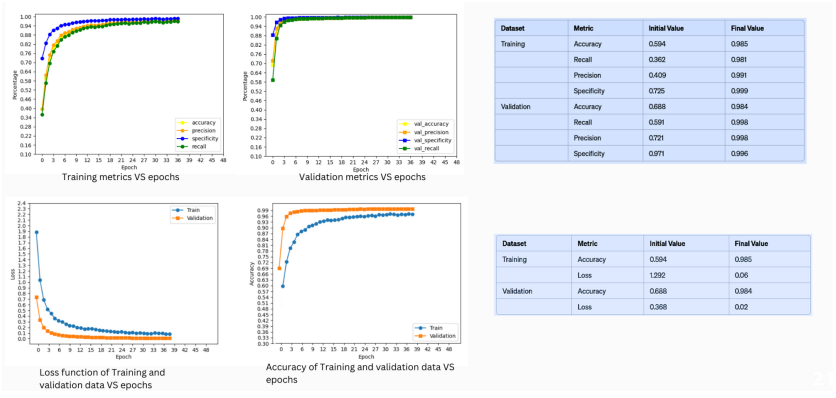
Table 2. Distribution of Frames in the Movement Mix Dataset

Data Type	Category	Number of Frames
Training Data	Dance	7,099 frames
	Exercise	6,798 frames
	Yoga	7,086 frames
Testing Data	Dance	846 frames
	Exercise	915 frames
	Yoga	920 frames

For the performance analysis, the following aspects are evaluated:

1. Examination of training metrics.
2. Evaluation of validation metrics.
3. Investigation of the relationship between accuracy and validation accuracy with the number of epochs.
4. Assessment of the loss function for both training and validation as a function of the number of epochs.

In the study, the model's efficacy was assessed over 50 epochs using training and validation datasets. Significant enhancements in accuracy, recall, precision, and specificity were observed across epochs. Training metrics improved as follows: accuracy (0.594 to 0.985), recall (0.362 to 0.981), precision (0.409 to 0.991), and specificity (0.725 to 0.999). Validation metrics also improved: accuracy (0.688 to 0.984), recall (0.591 to 0.998), precision (0.721 to 0.998), and specificity (0.971 to 0.996). The loss function consistently declined for both datasets, with training loss decreasing from 1.292 to 0.06, and validation loss from 0.368 to 0.02, indicating effective model performance. Refer to figure 6 for further details. See the figure 6 for more detail. In the study, the model's performance was evaluated using a confusion matrix and tested from two perspectives: frame-by-frame and considering the entire video, as illustrated in figure x. The frame-by-frame analysis revealed an accuracy of 79.38%, with significant misclassification issues in the yoga category, possibly due to similarities with exercise postures. When entire videos were evaluated, the overall accuracy further decreased to 69%, with 'Exercise' at 70%, 'Dance' at 100%, and 'Yoga' at 36%. This suggests potential overfitting, as the model performed well with training and validation data but struggled with testing data. The discrepancies may be due to the video classification process, which averages the sum of all probability vectors yielded by the model, potentially disregarding correctly labeled frames. The primary issue identified was the lack of diversity in the yoga category, as many videos were derived from a single long video, reducing source diversity. Additionally, many videos from a single YouTube channel performing yoga and exercise further decreased diversity, particularly for this class. These findings underscore the critical importance of dataset diversity in such problems for improving generalization and facilitating correct pattern recognition. The results are comprehensively represented in the confusion matrices depicted in the figures 7.



Dataset	Metric	Initial Value	Final Value
Training	Accuracy	0.594	0.985
	Recall	0.362	0.981
	Precision	0.409	0.991
	Specificity	0.725	0.999
Validation	Accuracy	0.688	0.984
	Recall	0.591	0.998
	Precision	0.721	0.998
	Specificity	0.971	0.996

Dataset	Metric	Initial Value	Final Value
Training	Accuracy	0.594	0.985
	Loss	1.292	0.06
Validation	Accuracy	0.688	0.984
	Loss	0.368	0.02

Fig. 6. General results for the first dataset.

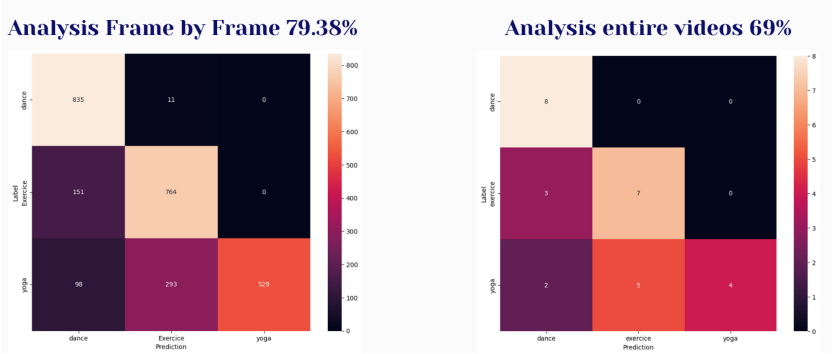


Fig. 7. Confusion matrix's for the fisrt dataset.

Second Dataset (Danger Kinetics) The balanced dataset consists of training and testing frames for five categories: Punching Bag (6,658 training, 966 testing), Punching Person (7,717 training, 646 testing), Slapping (5,843 training, 894 testing), Throwing Knife (7,493 training, 1,079 testing), and Walking Through Snow (7,654 training, 865 testing). The analysis involves evaluating training and validation metrics, assessing the evolution of accuracy and loss function across epochs for both training and validation datasets. For the Danger Kinetics dataset, all training and validation metrics improved over 50 epochs. Training accuracy rose from 0.353 to 0.850, and validation accuracy from 0.642 to 0.939. Simultaneously, loss function values decreased, with training loss falling from 1.965 to 0.408, and validation loss from 0.9394 to 0.2010. Even though these results were slightly inferior to the ones obtained with the initial dataset, they were more consistent, indicating no overfitting. The problems experienced with the previous dataset seemed to stem from overfitting due to the inadequate diversity in the yoga class data.This suggests that the problem with the previous dataset was rooted in overfitting, caused by the inappropriate generalization of the yoga class data, which lacked diversity.In the figure 8 we can see the results in the traing process and in figure 9 we can see the two confusion matrixs.

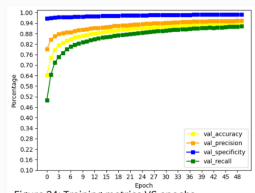


Figure 24: Training metrics VS epochs

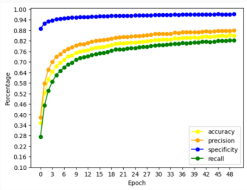
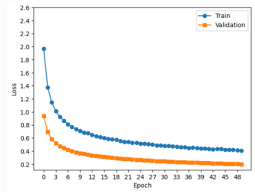
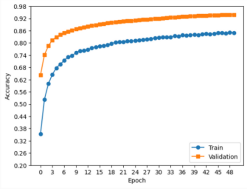


Figure 25: Validation metrics VS epochs

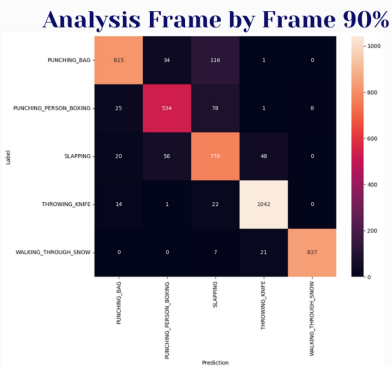
Dataset	Metric	Initial Value	Final Value
Training	Accuracy	0.353	0.850
	Recall	0.274	0.822
	Precision	0.383	0.879
	Specificity	0.890	0.971
Validation	Accuracy	0.642	0.939
	Recall	0.500	0.921
	Precision	0.791	0.953
	Specificity	0.966	0.988





Dataset	Metric	Initial Value	Final Value
Training	Accuracy	0.594	0.985
	Loss	1.965	0.408
Validation	Accuracy	0.688	0.984
	Loss	0.9394	0.2010

Fig. 8. Resutls for the second dataset.



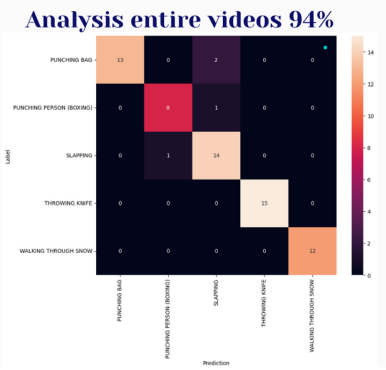


Fig. 9. Confusion matrix's for the second dataset.

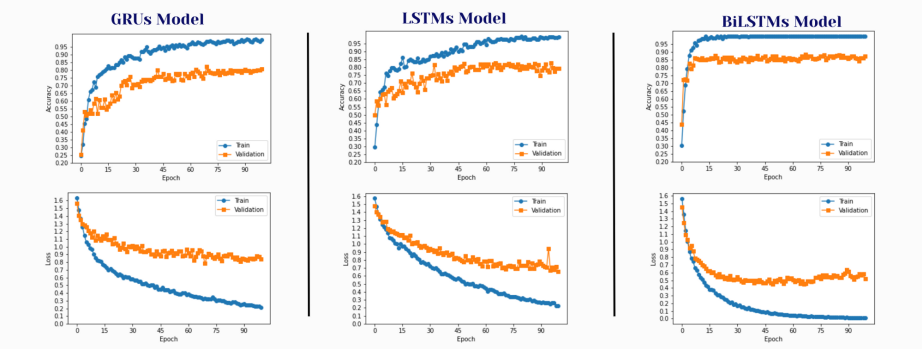
5.2 Hybrid approach

In the hybrid approach subsection, the analysis encompasses both the first and second datasets, employing three distinct models for each dataset. The primary focus is on the accuracy of testing, training, and validation data, with the models undergoing 100 epochs during training. Critical aspects for consideration in the analysis include the model's accuracy for training and validation data across epochs, the behavior of the loss function for training and validation data throughout epochs, and the model's performance with test data, as assessed through accuracy and the confusion matrix.

First Dataset (Movement Mix) In the "Movement Mix" dataset analysis, the model's performance was evaluated across three key aspects. The training data loss decreased from 1.19 to 0.46, and validation data loss reduced from 0.95 to 0.38. Training accuracy increased from 0.32 to 0.95, and validation accuracy rose from 0.41 to 0.97. However, the test data accuracy was only 59%. The yoga class exhibited poor predictions, with only one video correctly predicted, attributed to the dataset itself. This bias towards the yoga and exercise classes prevented a fair comparison of the models. Consequently, the analysis was limited to the hybrid approach with GRUs, and focus was shifted to the second dataset.

Second Dataset (Danger kinetics) The analysis of the second dataset, employing three different RNN models - GRU, LSTM, and BiLSTM, focused on three key aspects: the accuracy of the models for training and validation data across epochs, the behavior of the loss function for these data, and the performance of the models with test data. The GRU model demonstrated a consistent improvement in accuracy across 100 epochs, with the accuracy for training data increasing from 23.8% to 98.9%, and for validation data from 26.2% to 83.33%. However, the accuracy for test data was slightly lower at 87.9%. Class-specific performance varied, with 'Punching Bag' and 'Throwing Knife' classes achieving over 90% accuracy. The LSTM model outperformed the GRU model, with training data accuracy increasing from 29.5% to 99.4%, and validation data from 51.5% to 85%. The test data accuracy was higher at 90%. The LSTM model demonstrated superior performance, particularly for the 'Slapping' class. The BiLSTM model, despite being the most complex, yielded the least favorable results. The accuracy for training data increased from 30.1% to 99.8%, and for validation data from 44.3% to 88.43%. However, the test data accuracy was lower at 86.36%. Despite this, the BiLSTM model converged the fastest, stabilizing around epoch 15. The analysis also highlighted potential overfitting, as evidenced by the difference between training and validation results. However, the similar performance on test data suggested reasonable generalization. The computational cost was a significant factor, with hybrid models consuming up to 50GB of RAM and 20GB of VRAM during the data preparation and training process. The size and diversity of the dataset were also crucial for improving model performance. In summary, all models performed well with minor differences in results. The LSTM model was the best performer, while the simple approach showed unexpected good results. Despite the lower accuracy, the BiLSTM model converged the fastest, offering potential advantages in scenarios where training time is a concern. In the figure 10 and 11 we can observe details about the training and validation process and about the testing. Finally, Figure 12 presents a comparison of model sizes in terms of parameters and the accuracy achieved on the second dataset using test data. The largest difference is observed in the number of parameters, where RNN-based approaches possess significantly more parameters. Focusing on accuracy, though the simple approach scores highest, the difference is marginal. The figure illustrates the efficiency of the simple approach, concluding that despite its lack of complexity, this kind of

model remains at the forefront. It suggests that this approach may be sufficient for many video classification problems, even when dealing with complex scenarios as demonstrated in this study. The code for this work can be found in the next link.

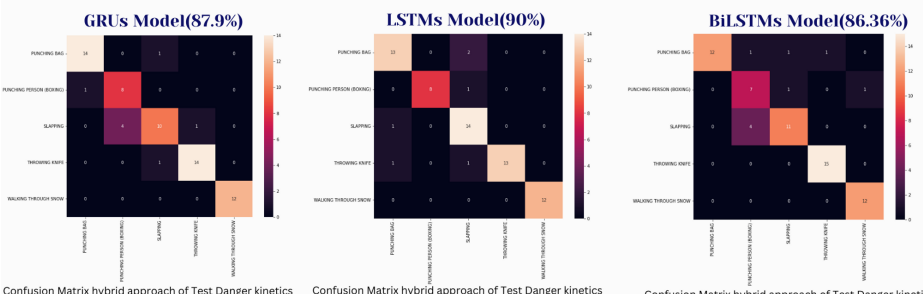


Dataset	Metric	Initial Value	Final Value
Training	Accuracy	23.8%	98.9%
	Loss Function	1.65	0.23
Validation	Accuracy	26.2%	83.33%
	Loss Function	1.58	0.82
Test	Accuracy	-	87.9%

Dataset	Metric	Initial Value	Final Value
Training	Accuracy	29.5%	99.4%
	Loss Function	1.59	0.22
Validation	Accuracy	51.5%	85%
	Loss Function	1.48	0.65
Test	Accuracy	-	90%

Dataset	Metric	Initial Value	Final Value
Training	Accuracy	30.1%	99.8%
	Loss Function	1.57	0.02
Validation	Accuracy	44.3%	88.43%
	Loss Function	1.48	0.42
Test	Accuracy	-	86.36%

Fig. 10. Results of the training process for the second data set in a the RNN models.



Confusion Matrix hybrid approach of Test Danger kinetics dataset using GRUs layers

Class	Accuracy
Punching Bag	93%
Punching Person	88%
Slapping	67%
Throwing Knife	93%
Walking Through Snow	100%

Confusion Matrix hybrid approach of Test Danger kinetics dataset using LSTMs layers

Class	Accuracy
Punching Bag	87%
Punching Person	88%
Slapping	93%
Throwing Knife	87%
Walking Through Snow	100%

Confusion Matrix hybrid approach of Test Danger kineti dataset using BiLSTM layers

Class	Accuracy
Punching Bag	80%
Punching Person	78%
Slapping	73%
Throwing Knife	100%
Walking Through Snow	100%

Fig. 11. Confusion matrix's for the second dataset in the RNN models.

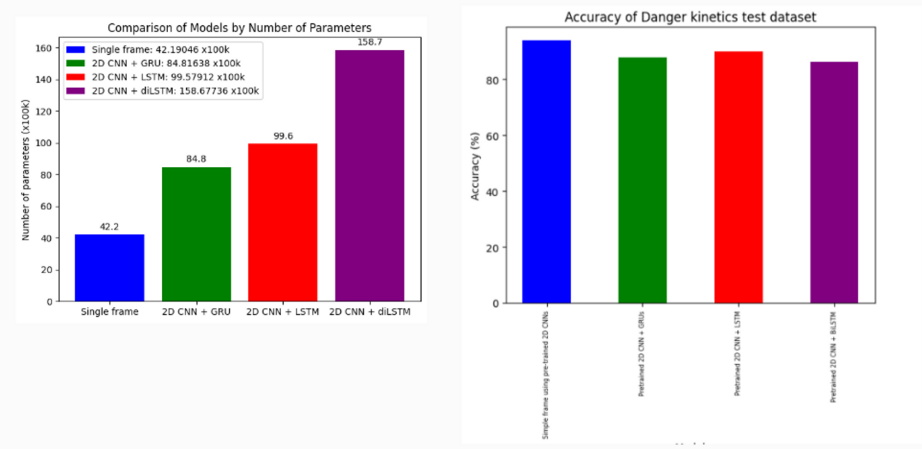


Fig. 12. Comparison of both the size and the accuracy of the different models based on neural networks.

6 Conclusions and Future Work

This study explored human activity video classification using both a simple approach (EfficientNetB0) and hybrid approaches (2D CNN + GRU, 2D CNN + LSTM, 2D CNN + BiLSTM) on two diverse datasets. Surprisingly, the simple approach, despite its simplicity, outperformed the more complex models, underscoring its effectiveness in video classification tasks. All models exhibited promising potential in classifying a variety of activities, even those characterized by high intra-class variation and low inter-class differences. However, the computational power required during the training phase posed a significant limitation, particularly for larger datasets. The study underscored the critical role of dataset variability, highlighting its profound impact when working with relatively small datasets in video classification tasks. Interestingly, the BiLSTM model, despite having the lowest accuracy, demonstrated superior convergence compared to the other models, including the simple approach. This factor is worth considering when training time is a crucial aspect. For future research, it would be beneficial to explore strategies to reduce computational resource requirements, experiment with larger and more diverse datasets, and investigate alternative lightweight pre-trained models or custom architectures. The findings of this study pave the way for further research in this direction, aiming to optimize the balance between model complexity, performance, and computational efficiency.

Bibliography

[1] Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)

[2] Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)

[3] Feichtenhofer, C., Pinz, A., Wildes, R.P.: Spatiotemporal residual networks for video action recognition. CoRR **abs/1611.02155** (2016), http://arxiv.org/abs/1611.02155

[4] Feichtenhofer, C., Pinz, A., Wildes, R.P.: Spatiotemporal multiplier networks for video action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)

[5] Hussein, N., Gavves, E., Smeulders, A.W.: Timeception for complex action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)

[6] Ke, Q., Bennamoun, M., An, S., Sohel, F.A., Boussaïd, F.: A new representation of skeleton sequences for 3d action recognition. CoRR **abs/1703.03492** (2017), http://arxiv.org/abs/1703.03492

[7] Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., Inman, D.J.: 1d convolutional neural networks and applications: A survey (2019)

[8] Kong, Y., Fu, Y.: Human action recognition and prediction: A survey. International Journal of Computer Vision **130**(5), 1366–1401 (2022). https://doi.org/10.1007/s11263-022-01594-9, https://doi.org/10.1007/s11263-022-01594-9

[9] Kong, Y., Tao, Z., Fu, Y.: Deep sequential context networks for action prediction. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3662–3670 (2017). https://doi.org/10.1109/CVPR.2017.390

[10] Laptev, I.: On space time interest points. International Journal of Computer Vision **64**(2), 107–123 (2005). https://doi.org/10.1007/s11263-005-1838-7

[11] Liu, J., Kuipers, B., Savarese, S.: Recognizing human actions by attributes. In: CVPR 2011. pp. 3337–3344 (2011). https://doi.org/10.1109/CVPR.2011.5995353

[12] Minallah, N., Tariq, M., Aziz, N., Khan, W., Rehman, A.u., Belhaouari, S.B.: On the performance of fusion based planet-scope and sentinel-2 data for crop classification using inception inspired deep convolutional neural network. PLOS ONE **15**(9), 1–16 (09 2020). https://doi.org/10.1371/journal.pone.0239746

[13] Morency, L.P., Quattoni, A., Darrell, T.: Latent-dynamic discriminative models for continuous gesture recognition. Tech. Rep. MIT-CSAIL-TR-

2007-002, MIT Computer Science and Artificial Intelligence Laboratory (2007)

[14] Rehman, A., Belhaouari, S.B.: Deep learning for video classification: A review. TechRxiv (2021). https://doi.org/10.36227/techrxiv.15172920.v1

[15] Rehman, A.u., Bermak, A.: Averaging neural network ensembles model for quantification of volatile organic compound. In: 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC). pp. 848–852 (2019). https://doi.org/10.1109/IWCMC.2019.8766776

[16] Samek, W., Montavon, G., Lapuschkin, S., Anders, C., Muller, K.R.: Explaining deep neural networks and beyond: A review of methods and applications. Proceedings of the IEEE **109**, 247–278 (03 2021). https://doi.org/10.1109/JPROC.2021.3060483

[17] Shen, J., Huang, Y., Wen, M., Zhang, C.: Toward an efficient deep pipelined template-based architecture for accelerating the entire 2-d and 3-d cnns on fpga. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **39**(7), 1442–1455 (2020). https://doi.org/10.1109/TCAD.2019.2912894

[18] Shi, Q., Cheng, L., Wang, L., Smola, A.: Human action segmentation and recognition using discriminative semi-markov models. International Journal of Computer Vision **93**, 22–32 (2011). https://doi.org/10.1007/s11263-010-0384-0

[19] Si, C., Chen, W., Wang, W., Wang, L., Tan, T.: An attention enhanced graph convolutional LSTM network for skeleton-based action recognition. CoRR **abs/1902.09130** (2019), http://arxiv.org/abs/1902.09130

[20] Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K. (eds.) Advances in Neural Information Processing Systems. vol. 27. Curran Associates, Inc. (2014), https://proceedings.neurips.cc/paper_files/paper/2014/file/00ec53c4682d36f5c4359f4ae7bd7ba1-Paper.pdf

[21] Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 4489–4497 (2015). https://doi.org/10.1109/ICCV.2015.510

[22] Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. Proceedings of the AAAI Conference on Artificial Intelligence **32** (01 2018). https://doi.org/10.1609/aaai.v32i1.12328

[23] Yu Kong, Y.F.: Human action recognition and prediction: A survey. International Journal of Forecasting **10**, 1–37 (2022)

[24] Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., Xie, X.: Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. Proceedings of the AAAI Conference on Artificial Intelligence **30**(1) (Mar 2016). https://doi.org/10.1609/aaai.v30i1.10451, https://ojs.aaai.org/index.php/AAAI/article/view/10451