Human Activity Recognition using Raspberry Pi Camera with Fall Detection and Notification System

Deva sai kumar Bheesetti
Dept. of Computer Science
University of Massachusetts Lowell
devasaikumar_bheesetti@student.uml.edu

Sai Sri Tej Palacharla
Dept. of Computer Science
University of Massachusetts Lowell
saisritej_palacharla@student.uml.edu

Prathap Ramachandra
Dept. of Computer Science
University of Massachusetts Lowell
prathap_ramachandra@student.uml.edu

Sri Latha Injam
Dept. of Computer Science
University of Massachusetts Lowell
srilatha_injam@student.uml.edu

***Abstract*— This project presents a comprehensive Human Activity Recognition (HAR) system utilizing a Raspberry Pi camera. The core of the proposed solution lies in a Joint RGB-Pose-based algorithm, which integrates Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs), and Deep Neural Networks (DNNs) to effectively recognize and classify various human activities. Using RGB-pose information enhances the model's understanding of spatial and temporal features, contributing to robust activity recognition. Furthermore, the system incorporates a fall detection mechanism leveraging the trained model. Upon detecting a fall, the system triggers a notification mechanism through the integration of Amazon Web Services (AWS) IoT Core and Simple Notification Service (SNS). This not only enhances the real-time monitoring capabilities of the system but also provides a timely alert in the event of a potential fall, facilitating rapid response and assistance. The integration with AWS IoT Core ensures efficient communication between the Raspberry Pi and the cloud infrastructure, enabling seamless handling of fall detection events. The utilization of SNS enables the delivery of instant notifications to predefined contacts, thereby enhancing the overall safety and reliability of the system. In summary, this project demonstrates a sophisticated Human Activity Recognition system using a Raspberry Pi camera, offering an innovative approach through the amalgamation of advanced algorithms and cloud-based notification systems. The implementation showcases the potential of combining edge computing and cloud services to create a robust and responsive solution for monitoring and ensuring the well-being of individuals in various environments.**

Keywords*— *Human Activity Recognition, AWS IoT Core and SNS, RGB-Pose algorithm, Edge Computing, Fall Detection, Raspberry Pi

I. INTRODUCTION

Human activity recognition (HAR) is integral to various aspects of our daily lives. It plays a crucial role in health and fitness monitoring through wearable devices, providing insights into exercise routines. In smart homes, activity recognition enables devices to adapt to user behavior, enhancing energy efficiency and convenience. It contributes to assistive technologies, aiding individuals with disabilities by interpreting movements for tasks such as wheelchair control. Security and surveillance systems benefit from activity recognition, detecting unusual activities to improve public safety. Gesture-based interfaces in computing and entertainment leverage activity recognition for natural interactions. Retailers use it to understand customer behavior and optimize store layouts, while healthcare relies on it for rehabilitation programs. In industrial settings, monitoring human activities enhances workplace safety and productivity. Overall, human activity recognition significantly impacts our daily lives by fostering health, convenience, accessibility, and safety across various domains. In recent years, the intersection of computer vision, deep learning, and edge computing has paved the way for innovative applications in the field of HAR. This research endeavors to contribute to this burgeoning field by presenting a comprehensive exploration of a novel system designed for real-time human activity recognition utilizing a Raspberry Pi camera. The primary focus of this project lies in the development and implementation of a Joint RGB-Pose-based algorithm similar to [3], seamlessly integrating Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), and Deep Neural Networks (DNN). The amalgamation of these advanced neural network architectures allows for robust and context-aware activity recognition, transcending the limitations of traditional approaches. Leveraging the capabilities of CNN for spatial feature extraction, LSTM for temporal dependencies, and DNN for complex decision-making. One distinctive feature of our system is the incorporation of fall detection using the Joint RGB-Pose algorithm. Fall incidents, particularly among the elderly, pose significant challenges to healthcare systems worldwide. Addressing this concern, we integrate fall detection capabilities into our system, employing AWS IoT Core and Simple Notification Service (SNS). This not only enhances the overall utility of the proposed system but also establishes a valuable link between edge computing and cloud services for

timely notifications. Furthermore, this research delves into the technical intricacies of deploying such a system on a resource-constrained device like the Raspberry Pi, emphasizing the practicality and feasibility of edge-based HAR solutions. The project also underscores the significance of edge-to-cloud integration for extending the functionality and scalability of the system. In summary, this research contributes to the evolving landscape of human activity recognition by presenting a cutting-edge solution that combines sophisticated neural network architectures with practical edge computing implementations. The integration of fall detection using AWS IoT Core and SNS enhances the system's applicability in real-world scenarios, making it a valuable asset in the realm of healthcare, safety, and ambient assisted living.

In the subsequent sections, we will navigate through the foundational principles, methodologies, and advancements in fall detection, shedding light on the pivotal role technology plays in preserving the health and safety of individuals. Through a comprehensive examination of existing literature, we will identify gaps in current research, paving the way for recommendations and potential avenues for future exploration in this critical domain.

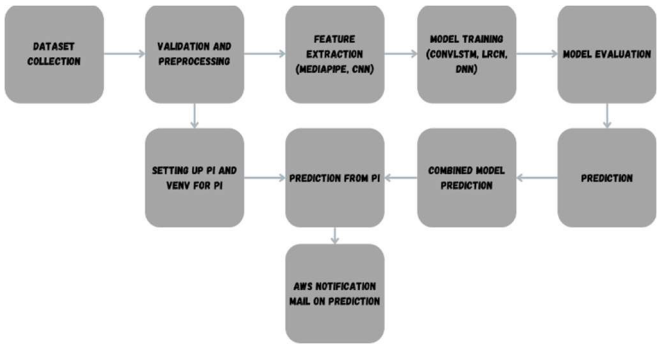


Figure 1: Overall Design Flow Diagram

II. OVERALL SYSTEM ARCHITECTURE

Fig 1 illustrates a schematic diagram of our developed system architecture which consists of three main components.

A. Data Collection and Preprocessing:

In our project, we systematically gathered data from diverse online datasets such as HMDB51, UCF50, and KTH, each containing a variety of activities. Additionally, we meticulously curated a custom dataset specifically for fall detection purposes. Following data collection, we conducted thorough analyses, leading us to define seven distinct classes for the evaluation of our activity recognition models. These classes encompass a range of activities, including boxing, hand clapping, hand waving, push-ups, jump rope, walking, and fall detection on the floor. This comprehensive approach to dataset compilation and class definition enables a comprehensive evaluation of our models' performance across various activities, ensuring the robustness and versatility of our activity recognition framework.

Figure 2: Classes Used

In our data preprocessing pipeline, we implemented several crucial steps to enhance the quality and relevance of the recorded videos for our research. Initially, we systematically removed videos containing unnecessary data, streamlining the dataset for improved efficiency. Subsequently, we developed a Python code to resize all recorded videos to a standardized resolution of 640x480, ensuring consistency across the dataset. Additionally, we conducted video trimming to eliminate any extraneous components, focusing on retaining essential information. Notably, special attention was given to videos used for training the Pose model, where we ensured that the shoulder and knee joints remained visible. This meticulous preprocessing strategy contributes to the overall effectiveness of our models by optimizing the input data for subsequent training and analysis.

B. Hardware & Software Setup:

In our hardware configuration, we employed a Raspberry Pi 4 Model B along with a Camera Module, Micro SD Card, and a dedicated power supply. The Raspberry Pi served as the core computing platform for our project. To facilitate remote management and access, we configured the Pi using SSH and VNC platforms, allowing seamless interaction with the device. For efficient file transfer between systems, we utilized the SCP (Secure Copy Protocol). On the software side, our development environment included PyCharm and Jupyter Notebook, providing powerful tools for coding and experimentation. Remote access was further enhanced through the use of RealVNC, Command Prompt, and PuTTY as optional tools. Additionally, we leveraged Amazon Web Services (AWS) to harness cloud computing resources, expanding our computational capabilities and supporting various aspects of our project. This combination of hardware and software components laid the foundation for a robust and versatile computing environment.

C. Machine Learning:

In our pursuit of effective activity recognition, we experimented with three distinct machine-learning models tailored to different facets of our dataset. For RGB-based

recognition, we implemented a ConvLSTM Model (Convolutional LSTM) and an LRCN Model (Long-term Recurrent Convolutional Network). These models were designed to capture and learn temporal dependencies and patterns in sequential data, particularly useful for analyzing RGB video frames. In parallel, for Pose-based recognition, we developed a dedicated Pose-based DNN model (Deep Neural Network). This model focused on extracting relevant features from the pose information within the videos, contributing to a more nuanced understanding of human activities. The selection of these models aimed to comprehensively address the diverse nature of our dataset, showcasing our commitment to leveraging state-of-the-art machine-learning techniques for accurate and robust activity recognition.

In our approach to RGB-based models, we employed a Convolutional Neural Network (CNN) for feature extraction, emphasizing a sequence length of 20 and a frame size of 64x64 pixels. This design choice was aimed at effectively capturing and processing temporal dependencies within sequences of RGB video frames. The CNN's ability to learn hierarchical representations from the spatial information in each frame contributed to the model's capacity to discern complex patterns in the given sequence. Simultaneously, for our Pose-based model, we utilized the Datapoints extraction feature from Mediapipe, a tool known for its accuracy in extracting pose information from images and videos. The inclusion of shoulder and knee points as mandatory data points in the pose extraction process was crucial for ensuring that the model focused on key joints, enhancing its ability to recognize and interpret human poses accurately. This meticulous configuration of feature extraction components in both RGB and Pose models aimed to optimize the models for discerning meaningful patterns in the diverse activities captured in our dataset.

RGB-based ConvLSTM Approach: The utilization of ConvLSTM cells, as described in [1], presents a sophisticated approach for video classification. By integrating convolution operations into the LSTM network, the ConvLSTM architecture adeptly combines spatial and temporal considerations. This amalgamation allows the model to discern spatial features within the data while concurrently accounting for temporal relations. In the context of video classification, the ConvLSTM proves effective in capturing both the spatial relationships within individual frames and the temporal relationships across different frames. This is crucial for understanding the dynamic nature of video data. An important distinction highlighted is the ConvLSTM's capacity to process 3-dimensional input (width, height, num_of_channels), in contrast to a simple LSTM that can only handle 1-dimensional input. This distinction underscores the ConvLSTM's suitability for modeling spatio-temporal data, making it a preferable choice for video classification tasks.

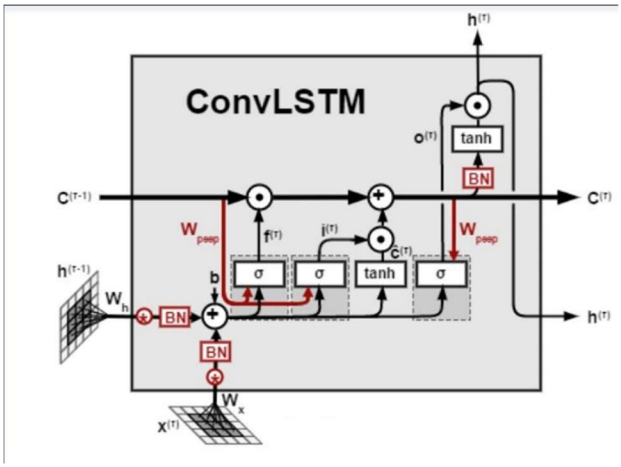


Figure 3: ConvLSTM Model

RGB-based LRCN Approach: As described in [2], the proposed approach involves implementing the Long-term Recurrent Convolutional Network (LRCN), which integrates Convolutional and LSTM layers within a single model for video classification. This design is aimed at leveraging the strengths of both CNNs and LSTMs for spatial feature extraction and temporal sequence modeling, respectively. In contrast to a separate CNN and LSTM model approach, where the CNN extracts spatial features to be later utilized by the LSTM, the LRCN approach streamlines the process by combining both spatial feature extraction and temporal modeling within a unified model. The Convolutional layers focus on extracting spatial features from video frames, and these features are then fed to the LSTM layer(s) at each time step for the modeling of temporal sequences. This end-to-end training methodology in the LRCN allows the network to learn spatiotemporal features cohesively, contributing to the development of a more robust model for video classification.

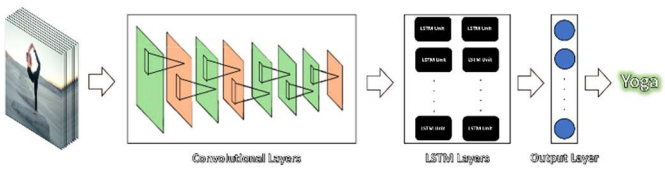


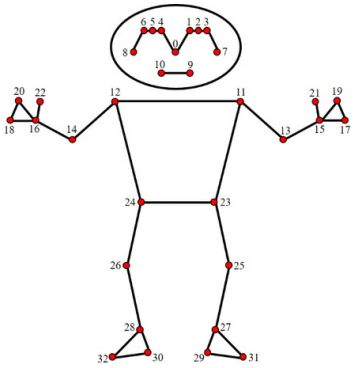
Figure 4: LRCN Model

We also incorporate the TimeDistributed wrapper layer, as a key element in the implementation of the LRCN approach. This layer plays a crucial role in enabling the application of the same layer to every frame of the video independently, effectively extending the capabilities of the underlying layer. By using the TimeDistributed wrapper, the layer's input shape can be transformed from (width, height, num_of_channels) to (no_of_frames, width, height, num_of_channels). This is highly beneficial as it allows the model to process the entire video in a single shot, considering each frame independently.

This approach simplifies the handling of video data within the model, streamlining the training process.

Figure 5: TimeDistributed Wrapper layer

Pose based DNN approach: In this approach, the Mediapipe pose detection library was employed to extract key points from video data, which were then organized into arrays and saved as .npy files. The subsequent step involved the utilization of a sequence of Dense layers in the neural network architecture. These Dense layers were equipped with the Tanh activation function and trained using the rmsprop optimizer. The input for the training process consisted of the previously generated .npy files, allowing the model to learn and capture patterns from the spatial information encoded in the pose key points. This methodology enables the training of a model that can understand and classify actions or features within videos based on the extracted pose data, providing a robust framework for video analysis.



0. Nose
1. Left_eye_inner
2. Left_eye
3. Left_eye_outer
4. Right_eye_inner
5. Right_eye
6. Right_eye_outer
7. Left_ear
8. Right_ear
9. Mouth_left
10. Mouth_right
11. Left_shoulder
12. Right_shoulder
13. Left_elbow
14. Right_elbow
15. Left_wrist
16. Right_wrist
17. Left_pinky
18. Right_pinky
19. Left_index
20. Right_index
21. Left_thumb
22. Right_thumb
23. Left_hip
24. Right_hip
25. Left_knee
26. Right_knee
27. Left_ankle
28. Right_ankle
29. Left_heel
30. Right_heel
31. Left_foot_index
32. Right_foot_index

Figure 6: Mediapipe pose datapoints.

III. EXPERIMENTAL EVALUATION:

A. Implementation:

In the design of the ConvLSTM approach, ConvLSTM2D recurrent layers are employed, allowing for the integration of convolutional and LSTM operations. This layer is configured with parameters such as the number of filters and kernel size for the convolutional operations. Subsequently, the output undergoes flattening and is passed through a Dense layer with softmax activation, yielding probabilities for each action category. To optimize the model's performance, MaxPooling3D layers are incorporated to reduce frame dimensions and minimize computational load, while Dropout layers are employed to mitigate overfitting. The overall architecture is intentionally kept simple, with a limited number of trainable parameters, as it addresses a specific subset of the dataset, obviating the need for an extensive and computationally demanding model.

In the construction of the Long-Short Term Memory with Convolutional Networks (LRCN) architecture, time-distributed Conv2D layers are employed, sequentially accompanied by MaxPooling2D and Dropout layers. These Conv2D layers serve to extract relevant features from the input data. Following the convolutional operations, the extracted features undergo flattening through a Flatten layer. Subsequently, the flattened features are fed into an LSTM layer, which captures temporal dependencies within the data. The final prediction is made through a Dense layer with softmax activation, utilizing the output from the LSTM layer to classify and predict the specific action being performed. This combination of convolutional and recurrent layers enables the model to effectively capture both spatial and temporal patterns in the input sequences, making it suitable for action recognition tasks.

In Deep Neural Network (DNN), we employed a sequential arrangement of Dense layers, each activated by the hyperbolic tangent (Tanh) activation function. This choice of activation function facilitates the model's capacity to capture intricate non-linear relationships within the data. To optimize the learning process, we utilized the rmsprop optimizer. The training data, essential for the model's learning, was provided in the form of .npy files, which were inputted into the network during the training phase. This approach enables the DNN to iteratively learn and adjust its parameters based on the patterns present in the training data, leveraging the non-linear transformations introduced by Tanh activation and benefiting from the optimization capabilities of the rmsprop algorithm. The resulting model is thus geared towards effectively capturing and generalizing complex features from the input dataset.

In our AWS notification setup, the primary tools employed are IoT Core MQTT client and Simple Notification Service (SNS). The process begins with the creation of a "thing," accompanied by the generation and download of certificates for subsequent use in our code. Following this, a Topic is established, and a subscription is created, with our email ID enlisted for notification. Subsequently, we implement Python code to facilitate the triggering of email notifications in the event of fall detection. This comprehensive setup utilizes IoT Core for handling IoT devices, SNS for managing notifications, and Python code to seamlessly integrate fall detection triggers with the configured notification system. The utilization of certificates adds a layer of security, ensuring the integrity of

communications between the IoT devices and the AWS services.

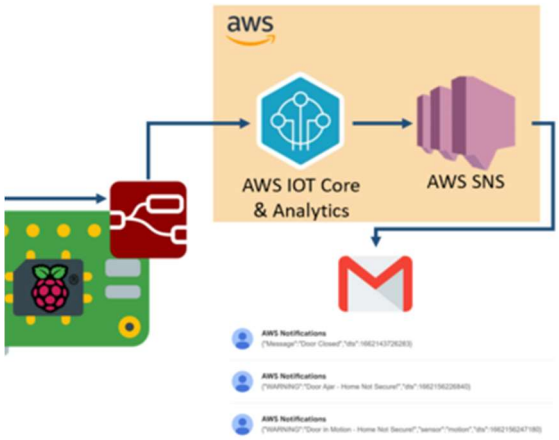


Figure 7: AWS IoT Core and SNS

B. Results & Discussion:

Throughout our model training endeavors, we encountered challenges, particularly with classes that exhibit similarities, such as Walking, Jogging, and Running. To address this, we explored various classes during the training process. In evaluating different approaches for RGB-based models, we found that ConvLSTM demonstrated slightly superior performance compared to LRCN, albeit at a higher computational cost. Consequently, we opted for LRCN due to its balance between accuracy and computational efficiency. Additionally, we implemented a strategy for combined predictions by integrating the outputs of both individual models. This entails deriving the class prediction based on the confidence levels assigned by each model, providing a more robust and nuanced prediction mechanism that leverages the strengths of both RGB and Pose in our action recognition system.

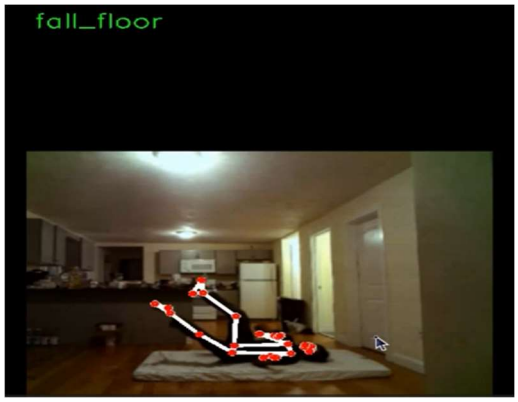


Figure 8: Fall Demo

In our LRCN model designed for action recognition across seven classes, we achieved commendable accuracy results across different datasets. The training set demonstrated an accuracy of 96.3%, showcasing the model's proficiency in learning from the training data. The validation set accuracy remained high at 90.1%, indicating the model's ability to generalize well to unseen data. Furthermore, the test set accuracy reached 88.7%, affirming the robustness of the model's predictive capabilities. On a separate note, our Pose-based Deep Neural Network (DNN) model exhibited remarkable performance, boasting 100% accuracy on the training set and maintaining above 98% accuracy in real-time predictions. These results underscore the effectiveness of both models in capturing and recognizing patterns within the input data.

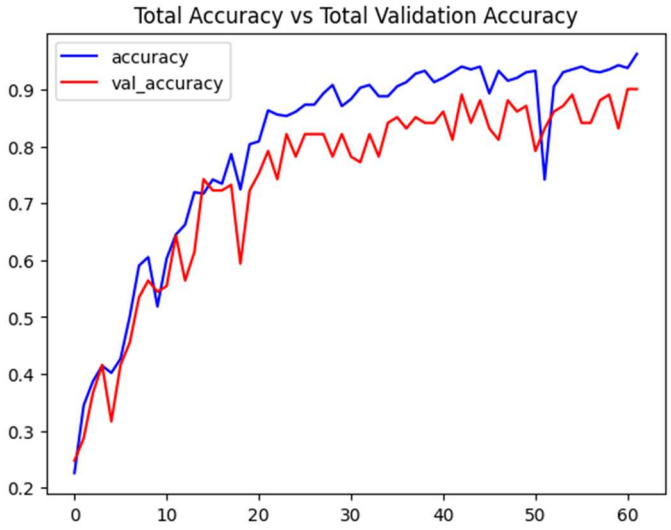


Figure 9: Accuracy graph

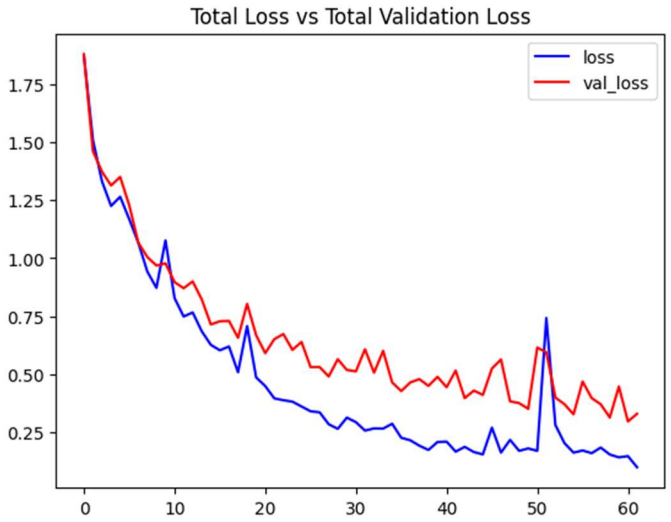


Figure 10: Loss graph

IV. FUTUREWORKS AND LIMITATIONS:

In our ongoing efforts to improve the Human-Activity Recognition system, we are prioritizing code refinement for increased reusability and user understandability. Currently, the system detects falls as anomalies, triggering email notifications; however, we are working towards providing users with the ability to define and monitor custom anomalies in the future. To enhance user interaction and control, we plan to design a user-friendly interface accessible via web and mobile platforms for configuring and monitoring the system. System accuracy will be elevated by incorporating data from multiple sensors, such as accelerometers, microphones, and environmental sensors. We are also committed to addressing ethical considerations in deploying activity recognition in public spaces, emphasizing consent, and privacy safeguards, and mitigating potential biases. Additionally, collaborative efforts with researchers in healthcare, home automation, or industrial domains are underway to adapt the system for specific applications, tailoring recognition models and features. A notable limitation is the privacy concern associated with camera usage for activity recognition, and efforts will be made to address and communicate these concerns transparently to users.

V. CONCLUSION

In conclusion, our comprehensive exploration of advanced algorithms, particularly RGB and Pose models, for action recognition using a Raspberry Pi camera has yielded promising results. Despite challenges in handling classes with similarities, our strategic approach involved leveraging ConvLSTM and LRCN models. While ConvLSTM exhibited slightly superior performance, the balance between accuracy and computational efficiency led us to choose LRCN for our final implementation. The integration of RGB and Pose models through a combined prediction strategy further enhanced the robustness and nuance of our action recognition system. The robust model results serve as a testament to the efficacy of our developed algorithms in effectively capturing and recognizing intricate patterns within live feed data. These outcomes highlight the considerable potential for real-world applications, particularly in scenarios centered around human activity monitoring. The successful incorporation of email notifications on fall detection adds a practical and relevant dimension to our system, contributing significantly to the well-being and safety of individuals. This project represents a significant step forward in the integration of advanced algorithms for real-time action recognition using accessible hardware like the Raspberry Pi.

ACKNOWLEDGMENT

OpenCV, Mediapipe, AWS Documentations and Bleed AI Academy.

REFERENCES

[1] Shi, X., Chen, Z., Wang, H., Yeung, D., Wong, W., & Woo, W. (2015). Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. ArXiv. /abs/1506.04214

[2] Donahue, J., Hendricks, L. A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., & Darrell, T. (2014). Long-term Recurrent Convolutional Networks for Visual Recognition and Description. ArXiv. /abs/1411.4389

[3] F. Angelini and S. M. Naqvi, "Joint RGB-Pose Based Human Action Recognition for Anomaly Detection Applications," 2019 22th International Conference on Information Fusion (FUSION), Ottawa, ON, Canada, 2019, pp. 1-7, doi: 10.23919/FUSION43075.2019.9011277.

[4] Thanh-Nghi Doan, "An Efficient Patient Activity Recognition using LSTM Network and High-Fidelity Body Pose Tracking" International Journal of Advanced Computer Science and Applications(IJACSA), 13(8), 2022. http://dx.doi.org/10.14569/IJACSA.2022.0130827

[5] K, P. P and J. Paulose, "Human Body Pose Estimation and Applications," 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), Kuala Lumpur, Malaysia, 2021, pp. 1-6, doi: 10.1109/i-PACT52855.2021.9696513.

[6] Yang, Y., Angelini, F., & Naqvi, S. M. (2023). Pose-driven human activity anomaly detection in a CCTV-like environment. IET Image Processing, 17(3), 674-686. https://doi.org/10.1049/ipr2.12664

[7] K M, Adarsh and U B, Nagesh and Doddagoudra, Abhishek V. and Bhat, Mayoori K. and L, Shreya, Human Action Recognition Using Deep Learning Technique. Available at SSRN: https://ssrn.com/abstract=4421669 or http://dx.doi.org/10.2139/ssrn.4421669

[8] Fuqiang Gu, Mu-Huan Chung, Mark Chignell, Shahrokh Valaee, Baoding Zhou, and Xue Liu. 2021. A Survey on Deep Learning for Human Activity Recognition. ACM Comput. Surv. 54, 8, Article 177 (November 2022), 34 pages. https://doi.org/10.1145/3472290

[9] E. De-La-Hoz-Franco, P. Ariza-Colpas, J. M. Quero and M. Espinilla, "Sensor-Based Datasets for Human Activity Recognition – A Systematic Review of Literature," in IEEE Access, vol. 6, pp. 59192-59210, 2018, doi: 10.1109/ACCESS.2018.2873502.

[10] harmi Jobanputra, Jatna Bavishi, Nishant Doshi, Human Activity recognition: A Survey, Procedia Computer Science, Volume 155, 2019, Pages 698-703, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2019.08.100.

[11] S. Gaglio, G. L. Re and M. Morana, "Human Activity Recognition Process Using 3-D Posture Data," in IEEE Transactions on Human-Machine Systems, vol. 45, no. 5, pp. 586-597, Oct. 2015, doi: 10.1109/THMS.2014.2377111.

[12] Maryam Ziaeefard, Robert Bergevin, Semantic human activity recognition: A literature review, Pattern Recognition, Volume 48, Issue 8, 2015, Pages 2329-2345, ISSN 0031-3203, https://doi.org/10.1016/j.patcog.2015.03.006.

[13] Charissa Ann Ronao, Sung-Bae Cho, Human activity recognition with smartphone sensors using deep learning neural networks, Expert Systems with Applications, Volume 59, 2016, Pages 235-244, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2016.04.032.

[14] Zhang, Shibo, Yaxuan Li, Shen Zhang, Farzad Shahabi, Stephen Xia, Yu Deng, and Nabil Alshurafa. 2022. "Deep Learning in Human Activity Recognition with Wearable Sensors: A Review on Advances" Sensors 22, no. 4: 1476. https://doi.org/10.3390/s22041476

[15] Murad, Abdulmajid, and Jae-Young Pyun. 2017. "Deep Recurrent Neural Networks for Human Activity Recognition" Sensors 17, no. 11: 2556. https://doi.org/10.3390/s17112556

[16] C. Xu, D. Chai, J. He, X. Zhang and S. Duan, "InnoHAR: A Deep Neural Network for Complex Human Activity Recognition," in IEEE Access, vol. 7, pp. 9893-9902, 2019, doi: 10.1109/ACCESS.2018.2890675.

[17] G. Forbes, S. Massie and S. Craw, "WiFi-based Human Activity Recognition using Raspberry Pi," 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), Baltimore, MD, USA, 2020, pp. 722-730, doi: 10.1109/ICTAI50040.2020.00115.

[18] Jmal, A., Barioul, R., Meddeb Makhlouf, A., Fakhfakh, A., Kanoun, O. (2020). An Embedded ANN Raspberry PI for Inertial Sensor Based

Human Activity Recognition. In: Jmaiel, M., Mokhtari, M., Abdulrazak, B., Aloulou, H., Kallel, S. (eds) The Impact of Digital Technologies on Public Health in Developed and Developing Countries. ICOST 2020. Lecture Notes in Computer Science(), vol 12157. Springer, Cham. https://doi.org/10.1007/978-3-030-51517-1_34

[19] K. Yamao and R. Kubota, "Development of Human Pose Recognition System by Using Raspberry Pi and PoseNet Model," 2021 20th International Symposium on Communications and Information Technologies (ISCIT), Tottori, Japan, 2021, pp. 41-44, doi: 10.1109/ISCIT52804.2021.9590593.

[20] Muhammad U. S. Khan, Assad Abbas, Mazhar Ali, Muhammad Jawad, and Samee U. Khan. 2020. Convolutional neural networks as means to identify apposite sensor combination for human activity recognition. In Proceedings of the 2018 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE '18). Association for Computing Machinery, New York, NY, USA, 45–50. https://doi.org/10.1145/3278576.3278594

[21] Eilar, Cody Wilson. "Distributed and Scalable Video Analysis Architecture for Human Activity Recognition Using Cloud Services." (2016). https://digitalrepository.unm.edu/ece_etds/300

[22] Alawneh, L., Al-Ayyoub, M., Al-Sharif, Z.A. et al. Personalized human activity recognition using deep learning and edge-cloud architecture. J Ambient Intell Human Comput 14, 12021–12033 (2023). https://doi.org/10.1007/s12652-022-03752-w

[Datasets]

[23] https://www.csc.kth.se/cvap/actions/

[24] https://www.crcv.ucf.edu/research/data-sets/ucf50/

[25] https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/