International Journal of Research in Engineering, Science and Management
Volume-3, Issue-7, July-2020
journals.resaim.com/ijresm | RESAIM Publishers

Human Behavioral Action Analysis Using Deep Learning

L. Suresh[1], Shreya Nath[2*], Shabana Shaikh[3], Pratima Kumari[4], Rashmi Bharti[5]
[1]*Professor, Dept. of Computer Science and Engineering, Cambridge Institute of Technology, Bangalore, India*
[2,3,4,5]*Student, Dept. of Computer Science and Engineering, Cambridge Institute of Technology, Bangalore, India*
*Corresponding author: aviatrix97n@gmail.com

***Abstract*: The sensor network-based human activity recognition is an important Fields of study. The work uses methods to manually extract and construct functions of various motions by means of statistical machine learn. Developing Deep Learning technology requires not to extract features manually and improve efficiency in complex issues related to human activities. By migrating the deep neural network image recognition experience, we propose a profound learning model based on Inception Neural Network and recurrent neural networks in combination. The model enters end-to - end data on the waveform of multi-channel sensors. Multi-dimensional features of different kernel-based convergence layers are extracted by Inception related modules. In combination with GRU, time series modelling is carried out and is used fully. Experimental testing on three public HAR datasets that are widely used. In contrast to state of the art, our approach proposed demonstrates consistently superior performance and strong general results.**

***Keywords*: Human activity recognition, Convolution Neural Network, Accelerometer data.**

1. System Overview

The detection of human activity is the problem in classifying accelerometer sequences reported in the known well-defined movements by specialized harnesses or smart phones. Given the large number of observations produced each second, the temporal nature of the observations and lack of a clear way to associate accelerometer data with known movements, this is a difficult problem. Classical solutions to the problem include time series data hand-creating functions based on fixed-size windows and machine learning models, such as collection of decision-making bodies. The problem is that this feature design requires a thorough field expertise. Deep learning practices, like repeated neural networks and single-dimensional convolutional neural networks, or CNNs, have recently proven to produce cutting-edge results in the context of challenging activities with little or no data feature engineering. Methods of statistical learning were widely employed to resolve problems of activity recognition. In order to recognize seven motions including walking, running and jumping, Gupta P and Dallas used a naïve Bayes(NB) and KNN classification. Nevertheless, they relied on designed features to better differentiate between their different activities. they did not consider discriminatory features. Methods for extraction of features such as symbolic representation, raw data statistics and transform coding are widely used when recognizing human activity, but they are hurtful and require expert expertise to design features.

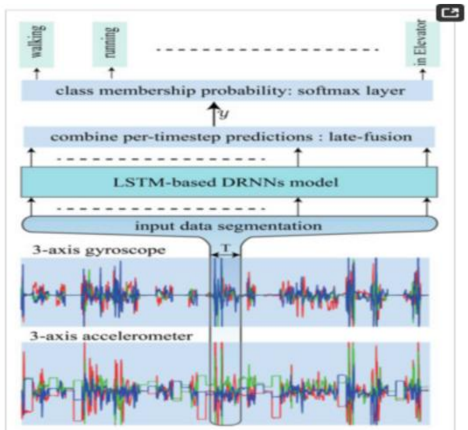


Fig. 1. Depiction of LSTM RNN for Activity Recognition

2. Proposed System

The feature extraction and splitting of waveform data are done by using specific scale-based convolution kernels like 1x1, 1x3 and 1x5 Multi-scaled extraction of the human body for different durations.

Pooling layers are used in order to reduce the noise of interference brought about by the unconscious jitter of the human body.

The above-mentioned advantages are superimposed and expanded by the concatenation of different scales based convolution layers, pooling layers and multifaceted nonlinear activation.

A. System Analysis

System analysis is the mechanism by which the problem is discovered, artifacts and specifications defined and solutions evaluated. It is the way of thinking about the company and the issue, a collection of technologies that leads to the resolution of these problems. The feasibility study plays a significant role in the analysis of structures, which is the goal for design and development.

1) Feasibility Study

The survey is now extended to include a more detailed feasibility report, according to the findings of the initial survey. "FEASIBILTY Analysis" is a test of program ideas, based on their operability, organizational effect, their ability to meet needs and their efficient usage.

Three key considerations involved in the feasibility analysis are

a) *Economic Feasibility:* This research explores the economic effect on the organization of the program. There is a limited amount of money which the organization will spend on machine research and development. The costs have to be explained.
b) *Technical Feasibility:* The research aims to check the technological viability of the program. Any built framework must not be subject to high demand for the technological resources available. This will result in high demands on the technological resources available. This results in high requirements for the customer. The developed system must have a minimal requirement, because this system needs only minimum or zero modifications.
c) *Social Feasibility:* The research is concerned with checking the user's acceptability of the program. This involves the user training to safely use the program. It is not necessary, however, that the consumer feel threatened by the program. The level of user acceptance relies solely on the approaches used to inform and familiarize the user with the program. He needs to raise trust so that he is also able to critique constructively, because he is the last user of the program.

B. System Design

Design is a creative process; the key to effective system is to have good design. The "Design" system is defined as "Process of applying different techniques and principles to define a process or system in sufficient detail to allow its physical realization." Development of the system is followed by various design features. The design specification describes system features, system components or elements, and their appearance towards end users.

C. MVC Design Model

In fact, Swing makes use of a simplified MVC design variant called the model-delegate. This design combines the view and the controller object into a single element which draws the component onto the screen and handles GUI events known as the UI delegate. Communication between the UI delegate and the model becomes a two-way street. Every part in Swing contains a model and a delegate in the UI. The model is responsible for holding information about the state of the component. The UI delegate is responsible for maintaining on-screen information on how to draw the component. The UI delegate (along with AWT) responds to various events that spread through the part.

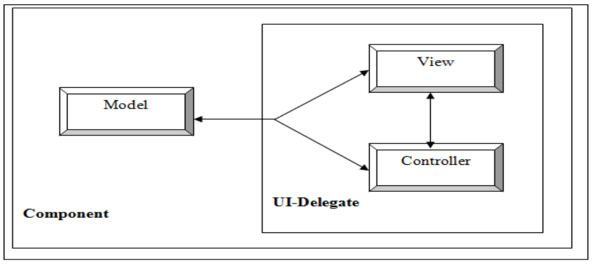


Fig. 2. Combination of View & Controller into a UI delegate object

Model: The model is the element describing the component's state and low-level behavior. It administers the state and executes all transformations on that territory. The model does not have any specific knowledge of either its controllers or their opinions. It covers the data for every portion of the state. Various models exist for different component types. For instance, a scroll bar component model might contain information about its current position of its adjustable "thumb," its minimum and maximum values, and the width of the thumb. At the other side, a menu may simply include a list of the menu items from which the user may choose.

View: The view refers to how the part looks in the image. It is the piece that manages the model's visual display of the State. Nearly all window frames will have a title bar which spans the top of the window. The title bar may however have a closing box on the left or right side. There are samples of various view styles for the same entity in the same browser. A model may have more than one view, but typically that is not the case in the Swing set.

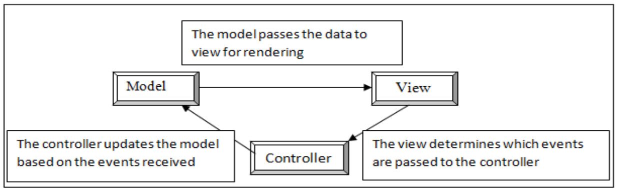


Fig. 3. Communication through the MVC architecture

Controller: The controller is the component that controls model user interaction. It provides the process by which adjustments to the model 's condition is produced. The user interface element is the one that determines how the part communicates with events. The view cannot correctly render the scrollbar without first getting information from the model. In this case, unless it can obtain its current position and width relative to the minimum and maximum, the scrollbar will not know where to draw its "thumb" Similarly the view defines if the item is the beneficiary of user activities such as clicks on the keyboard.

D. System Development Methodology

The method of device creation is a mechanism by which a project is completed or a project gets rid of some problem. The process of software development is described as a number of phases, procedures, and steps that give the software complete.

It follows sequence of measures that are used to advance items. The method of development followed in this project is model waterfall.

Means quality. Prior to writing a single line of code, the emphasis on requirements and design ensures minimal waste of time and effort, and reduces the risk of slipping the schedule.

With fewer human capital needed when if a process has been finished, these individuals will continue to work on the next step.

3. System Architecture

System architecture is the conceptual design, which defines a system's structure and behavior. A description of an architecture is a formal description of a structure, arranged in a way which supports reasoning about the system's structural properties. This describes the program elements or building blocks and includes a framework with which to purchase goods, and create processes that can operate together to execute the overall structure.

The Architecture of the System is shown below.

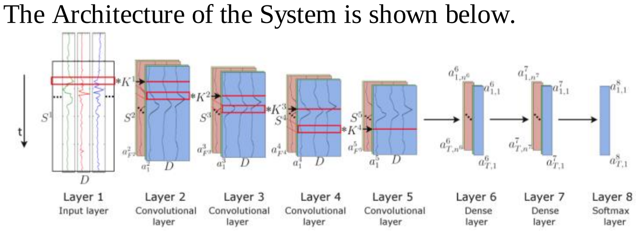
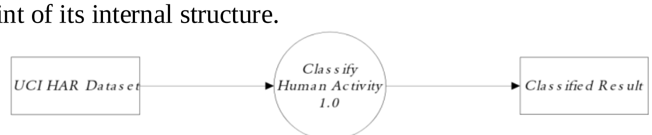


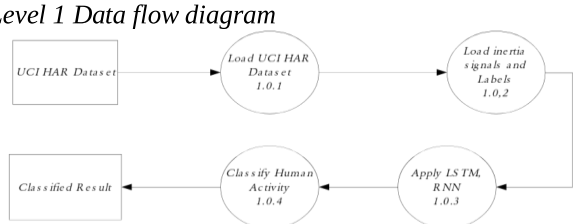
Fig. 5. CNN Working Model

1) Level 0 Data flow diagram

A data flow diagram of context-level or level 0 shows the interaction between the system and the external agents that act as data sources and data sinks. The system's relations with the outer world was represented solely in terms of data flows around the network boundaries on the background diagram (also known as the Level 0 DFD) The background diagram displays the entire program as a single operation, and gives no hint of its internal structure.



2) Level 1 Data flow diagram



The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which addresses one or more of the data flows to or from an external agent, and together provide all of the system's functionality as a whole. It also identifies internal data stores that must be present for the system to do its job, and shows the data flow between the different parts of the system.

4. Implementation

Implementation is the project level, where the abstract framework is turned into a functional program. Around this point, the main responsibility moves to the customer team and the big effect on the current program. If implementing is not planned and controlled carefully, it can cause chaos and confusion.

Algorithms Steps

1. Load data from the UCI HAR data collection with accelerometer
2. Convert and reformat the data of the accelerometer into a time frame < Visualize data from the accelerometer
3. Reshape the multi-dimensional tabular data for acceptation
4. Divide the set of data into training, validation and test set
5. Defines a deep neural network model which can be processed with ML later
6. Prepare the deep neural network for the identification of human behaviors
7. Validate trained DNN performance against test data using learning curve and confusion matrix
8. Export the DNN model trained to ML
9. Ensure that the ML model was correctly exported by performing a Python sample prediction
10. Use the ML library to predict the results of a given set of data.

5. Testing

Training is simply a continuum of specific tests whose key function is to apply the PC-based system in full. While each evaluation has an alternate purpose, all work is performed to guarantee that all the elements of the system have been lawfully organized and meet the assigned capacities. The testing process is also conducted to ensure that the element performs precisely as it will do as well. Testing is the last within of the organization itself monitoring and approval practice.

A. Unit Testing

Here every module comprising the overall system is individually tested. Unit checking centers evaluation efforts in each module even at the smallest device design level. The System modules are individually attempted. This testing is done in the very style of programming. Unit monitoring conducts specific ways of ensuring full reach and most serious blunder position in a module's control system.

B. Integration Testing

Integration testing is a systematic technique for building the structure of the programme. The main aim of this testing process is to take unit-tested modules and construct a program structure dictated by design.

C. Validation Testing

The programming is finished at the end of combination

Table 1
Integration testing table

Classes	Functions	Tests	Remarks
Class: Main	LoadUCIHARDataset() LoadInertiaSignals() Classify()	Check whether all operations that were applied are working correctly or not.	Success
Class: Training	LoadInertiaSignalsAndLables() ApplyLSTM() ApplyRNN()	Check whether all operations that were applied are working correctly or not.	Success
Class: Classify	ClassifyHumanActiviy()	Check whether all operations that were applied are working correctly or not.	Success

Table 2
Validation testing table

Functionality to be tested	**Input**	**Tests done**	**Remarks**
Working of IDE	User interaction with help of a mouse and keyboard.	Appropriate Pages and functions are tested.	Success
Working of main module	User has load the UCI HAR dataset to Classify the Human Activity.	Dataset loaded and Load Inertia Signals to predict the Human Activity.	Success
Working of Prediction	User has to apply LSTM and RNN methods to Classify.	Human activity classified	Success

testing and gathered as a bundle. Blunders that code are exposed and remedied. Checking for acceptance can be described from different viewpoints. The research here confirms the commodity quality in a manner that the consumer sensibly expects.

6. Conclusion and Future Work

We presented a method of human activity recognition based on CNN which uses data collected from the user's smart phone by accelerometer. Our system outperformed the baseline random method in the classification of human behavior, and demonstrated the highest classification accuracy when the data were used to learn the neural network for longer-duration accelerometer. We found that the dimension of the input vector affects the performance of activity recognition, and that figuring out a way in particular to disambiguate the signal will likely lead to an improvement in the performance of activity recognition. Via series of image processing operations, the basic features of human motion were extracted. The results showed a reasonable degree of accuracy in all stages of testing and validating the preparation. The overall recognition rate was 94 per cent for the designed system. The designed system can be tested for future work in more complex activities which have a high similarity in human form and movement.

References

[1] Eamonn Keogh and Shruti Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. In SIGKDD, 2002.
[2] Geoffrey E. Hinton and Simon Osindero. A fast learning algorithm for deep belief nets. Neural Computation, 18(7):1527–1554, 2006.
[3] Yoshua Bengio. Learning deep architectures for AI. Found. Trends Mach. Learn., 2(1):1–127, January 2009.
[4] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. In ICML, 2010.
[5] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. CVPR, 2011.
[6] Hong Cao, Minh Nhut Nguyen, Clifton Phua, Shonali Krishnaswamy, and Xiao Li Li. An integrated framework for human activity classific. In ACM International Conference on Ubiquitous Computing, 2012.
[7] Li Deng. A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Transactions on Signal and Information Processing, 2014
[8] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition.