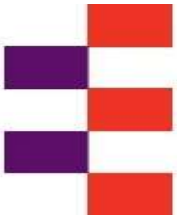

University of Essex

DEPARTMENT OF MATHEMATICAL SCIENCES

MSC APPLIED DATA SCIENCE

HUMAN ACTIVITY RECOGNITION FROM VIDEOS USING DEEP NEURAL NETWORK

Under the supervision of **Dr Mario Gutierrez Roig**

Student ID : 2111897

HUMAN ACTIVITY RECOGNITION FROM VIDEOS

ABSTRACT

Human Activity recognition has been an important topic of research due to its wide range of applications in the field of smart environments. With increase in the use of smart home sensors, the necessity to effectively recognise and classify human acitivity has become more prominent. With an increasingly large size of datasets, the use of machine learning methods to perform activity recognition has been preferred. Also, these algorithms work by learning features from the data without any prior knowledge. This research here explores the effectiveness of deep neural networks in the task of activity recognition. For the purpose of evaluation, the LSTM based RNN model and AlexNet model were implemented on the publicly available, UT interaction dataset. The results thus obtained show that these models outperform the existing machine learning models in terms of accuracy and model performance. An overall accuracy score of 0.87 has been observed for the deep learning models.

INTRODUCTION

The significance of the Human Activity Recognition system is a result of the rise in the use of security cameras. The objective of activity recognition is to recognize an action of the objects and intentions after a series of investigations into behavior of the object and its circumstances. The main use of this technology include context based retrieving, sports, surveillance monitoring and choreography. A single task is associated with numerous simple actions. Because of its importance in numerous fields and the growing necessity for smart home automation and convenient amenities for the elderly, Human activity recognition had emerged as an dynamic study field in recent years [(Roecker, 2011)]. In the specific field of smart environments and providing service by living technologies, activity recognition has played a vital role in Smart Homes with widely available sensors and detectors that has attracted a great attention for improving the standard of life for the inhabitants in the smart home environments [(D.J., 2012)].

Human Activity Recognition aims to recognise and detect both straight forward and intricate behaviours in natural environments by utilising the sensor data. Since the data produced by the sensors is occasionally confusing with regard to the occurring activities has become a difficult task. This makes it more challenging to interpret the actions. The data that is produced can usually be unclear or noisy data. Human error or a networking system error that results in inaccurate sensor reading can both introduce noise into the data. These real world situations are completely uncertain and necessarily make use of techniques for learning from the data, extracting knowledge and assisting in decision making. In addition to this using inverse probabilities, it is possible to predict the future and infer the patterns [(Holzinger, 2017)]. As a result, numerous probabilistic as well as non - probabilistic systems has been introduced for the

recognition of human activities. Sensors like accelerometers, passive infrared sensors, gyroscopes etc. are used to detect patterns that corresponds to the activity. These patterns are then identified using either Hidden Markov Modeling (HMM) [(Duong, CVPR 2005)] or feature extraction upon a sliding window proceeded by the classification process [(Roggen, EWSN 2015)]. Deep learning approaches have drawn more and more attention in recent years. Deep learning is a catch-all phrase for neural network techniques that depend on extracting representation using the unprocessed input and have multiple hidden layers. To extract and transform features, the network is trained with multiple layers of non - linear cognitive processing. The output from the preceding layer serves as the input for the following layers. In the domain applications like computer vision [(Lee, ACM 2009)], speech recognition [(Hinton, 2012)] and audio recognition [(Lee H. P., 2009)] deep learning algorithms has achieved outstanding performance than previous machine learning algorithms.

A recurrent neural network model is developed for identifying human activities in this research. On readily accessible benchmark dataset of UT - Interaction, the classifier Long Short Term Memory (LSTM) is used to classify human behaviours including eating, taking a shower, and sleeping [35]. The outcomes have been assessed by comparison with well-known machine learning techniques as Random Forests, Hidden Markov Model (HMM) and Naive Bayes model.

Algorithms for recognizing human activity often operate in a hierarchical way. The low level includes background removal, extracting the features, tracking and finally detecting the activity. The action recognition is present in the intermediate phase which is known as mid level in the approach after which the reasoning engines module that is present on the high level encodes the contextual information of the activity depending on the number of units in lower levels. Figure 1 shows the overall layout of the Human Activity Recognition system.

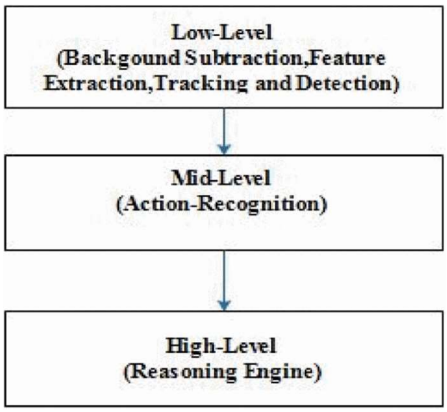

Figure 1: HUMAN ACTIVITY RECOGNITION FRAMEWORK

The retrieved frames are subjected to background removal at the low level using either among pixel based, block based or a mix of both the methods. The most often used pixel based

backdrop models include Gaussian models, combinations of Gaussians, Kalman filter, ViBe and hidden markov models. Block based approaches often come under the categories of Local Binary Patterns in Histogram, Incremental PCA, Histogram Correlation, and Distance of Normalized Vector. Either of the two procedures is used to do feature extraction after foreground detection. When using a model based approach, a human model is created to identify activity, but when using a feature based approach, the features either from global or local features or both of the behavior are obtained to help with the mathematical calculations of activity detection. It is delivered to the classifier enabling action recognition at the mid-level following the detection and then tracking step. After actions are identified, they are delivered to a higher level that has a logic engine which can decipher human activity. This study targets on offering a thorough analysis of the high level strategy which examines how human activity is perceived from the numerous writings of many scholars. Also their research limitations and applications are considered.

CHALLENGES OF HUMAN ACTIVITY RECOGNITION

The following are key applications for the Human Activity Recognition system.

A) Domain Application

Depending on the domain application, different activity tasks and specific details will have different weights. Supposing a surveillance monitoring system, the primary focus is often on identifying any unexpected behaviour (for example jumping from a building, falling down)

B. Differences between Inter class and Intra class

The effectiveness of a system is reliant on the wide variances in activity class. For an illustration, there will be very little difference between the activities of walk and jog. A decent human activity recognition system should be capable to distinguish between the actions of different classes.

C. Usage of the Learning Paradigm

To identify various human activities, a learning based strategy is applied. The greatest benefit is resistance to fluctuations within the class. Depending on the kind of training dataset that is available, the use of the learning paradigm may be supervised otherwise unsupervised.

D. Occlusion by self or external objects

Recognizing activities is significantly impacted by occluded components, whether caused by self or by external objects. As the characteristics present in the occluded body parts are frequently lost, the model may recognise an activity as incorrect even though it would actually be accurate because the features present in these parts only have a small impact on the output instance.

E. Recording and Background Settings

It might be challenging to discern human activity against a busy or dynamic environment. The effectiveness of the device is also greatly influenced by the video quality. Even with fluctuating video quality and a busy background, an effective activity identification system shall be able to identify the human activity.

(T. Subetha and S. Chitrakala, 2016)

APLLICATIONS OF HUMAN ACTIVITY RECOGNITION

The main applications for human activity recognition includes video analytics based on useful information or content such as robotics, fall detection of human, human computer interaction, ambient intelligence, smart home applications, video indexing, video surveillance etc. The rise in the website platforms number has achieved by sharing videos that is made of content based video analytics has got more significant. Therefore, it is necessary to create an efficient methods and indexing for storing the videos effectively. The most important uses for creating human and machine interfaces are assisting the elderly by creating context aware computing system, keeping track of fitness and health, creating smart homes that would react to user movements and so on. Unauthorized person detection, aberrant crowd behaviour, ATM detection of fraud are only a few examples of the deviant behaviours that can be automatically detected in video surveillance. Particularly behavioural biometric, which entails comprehending techniques and associated algorithms and procedure to uniquely identify people according to their behavioural indicators, can benefit from the application of activity recognition.

(T. Subetha and S. Chitrakala, 2016)

LITERATURE REVIEW

OVERVIEW OF MACHINE LEARNING

Machine learning is a subfield of Artificial Intelligence, that is used for building algorithms which helps to recognise and infer the patters produced during the training phase for the training dataset [(Bishop CM, 2006)]. These algorithms can be divided into two categories.

1) Supervised learning
2) Unsupervised learning

Creating a mathematical representation depending on the connection between input and the output data by using it to anticipate unknown future data points is the aim of supervised method of learning the data. Unsupervised machine learning algorithm aims to find patterns for the input data by actually not knowing the results [(Liu Y, 2016)]. Data pre - processing procedures which includes feature extraction, segmentation or vectorization, projection and standardisation or normalisation are frequently needed as well [(Domingos PM, 2012)].

Naive Bayes, Support Vector Machine (SVM), k - Means Clustering, k - Nearest Neighbors (k-NN), Linear Regression, Random Forests, Logistic Regression, Decision Trees are some of the significantly used supervised machine learning algorithms for classification purpose. Decision Tree classifier classifies data instances according to their properties and data values to categorize them. Each branching in the decision tree indicates a value in which the node can take on and each node indicates a feature or characteristic that should be classified. Naive Bayes classifiers are also known as probabilistic classifier that use the Bayes Theorem and make significant predictions about the independence of the features. For Support Vector Machine classifier, the idea of a margin along the either direction of a hyperplane separating two classes of the data, serves as the foundation. It has been demonstrated that maximising the margins reduces the upper bound upon the estimated generalized error by establishing the largest possible space between the hyperplane separation and instances present on any side of the plane. K - Nearest Neighbor classifier is a method that categorises new instances using similarity metric function (for example, distance functions like Euclidean or Manhattan) [(Bishop CM, 2006)]. Such classifiers with the exception of Support Vector Machine classifier are suitable for environments with lower in resource rate, due to their less memory and computational prerequisite also the special constraints that Human Activity Recognition that impose includes reduced latency, computational constraints and memory constraints.

The significantly known unsupervised learning algorithms, notably clustering algorithm techniques include Mixture models, k - Means clustering and Hierarchical clustering. K - Means clustering seeks to divide sample groups into k number of clusters depending on the intra group that is measure of similarity in clusters and the inter group that is measure of non similarity in clusters. In a clustering prototype, each instance belonging to a particular cluster with the closest to the central cluster or centroid cluster. A clustering analysis technique called hierarchical clustering analysis aims to create a hierarchy of clustering groups by combining or dividing them according to the degree of dissimilarity across sets. Specific probabilistic model called a mixture model is used to describe sub populations of the data within a larger population [(Liu Y, 2016)].

Such methods are especially useful while working with the datasets having no labels or where the primary result is the measurement of similarity or dissimilarity across the classes. [(Dobbins C and Rawassizadeh R, 2018)], [(Vaughn A, 2018)], [(Abdallah ZS, 2015)].

OVERVIEW OF DEEP LEARNING

On the other hand, because of their higher performance, Deep Learning algorithms has recently gained popularity across several fields [(Liu Y N. L., 2016)]. Since Deep Learning is founded on the concept of data representation, therefore these techniques could produce optimised features automatically. From the unprocessed input data, by not requiring any human involvement, enabling it possible for recognising the unidentified patterns which would otherwise survive hidden or unidentified [(Shickel B, 2017)]. Deep Learning models do have several limitations as mentioned earlier [(Marcus G, 2018)]:

- Black box models
- Intrinsic difficulty in interpretation
- Huge datasets are needed for training
- Computing costs are high.

Due to these restrictions, machine learning approaches are nevertheless chosen in some fields, particularly if the training set is limited or when quick training is necessary for the dataset. Convolutional Neural Networks (CNN), Long Short Term Memory Networks (LSTM), Recurrent Neural Networks (RNN), Stacked Autoencoders, Gated Recurrent Units (GRU), VAriational Autoencoders (VAE) and Temporal Convolutional Networks (TCN) are a few of the significantly used Deep Learning techniques [(J, 2016)].

Human activity is the semantic classification of human and object movements. Identifying the segments of video that contain these motions depends on activity recognition. This section covers many methods used for effective feature action recognition, and Figure 2 provides an illustration. A detailed comparison for several approaches is produced.

REFERENCES	METHODOLOGY	ADVANTAGES	FUTURE WORK
[1]	multiple instance-SVM	Improves the local feature based activity recognition by using advanced machine learning techniques which are different from bag-of-features based representation	incorporates spatio-temporal informations and different descriptors
[2]	subspace clustering approach	can handle multi-dimensional data that are not possible with the typical clustering method	aims in fusing a large contextual information such as emotions, health conditions.
[3]	non-parametric comparison of trajectory data with the fusion of bayes net	ability to detect activity even in medium resolution videos	The commentary can be extended to security applications
[4]	Kinematics Model-Based	Takes only the prominent human poses that dispatches the information and eliminates the rest	handling of similar action recognition
[5]	Kinematics Model-Based	Depends on Contour points for learning keyposes.	This method shows high tolerance to inter actor variance handling of occlusion and view-invariance
[6]	Physics Model-Based	Dynamic features are computed and by using these features action classes are classified in terms torques	apply dynamic features to human-gait recognition
[7]	Kinematics Model-Based	Adaptive vision-based human action recognition method is proposed.	adaptive learning should be compared to other benchmarking incremental learning and continuous adaptation methods.
[8]	Kinematics Model-Based	A new skeletal representation that specifically models the 3D geometric relationships between various body parts using rotations and translations in 3D space	increment the system to model complex activities

Figure 2: A comparison for different methods

A) FEATURE BASED APPROACHES

Improved local feature basing on the activities description is provided by [(S. Umakanthan, Aug 2014.)]. Videos are used to extract the histogram of similar patterns and dense HOG (Histogram of oriented Gradients). Thus, rather than creating a single codebook for each activity class, scholars used multiple instance - Support Vector Machines (mi-SVM). Following the pooling layer of Spatio temporal characteristics, the Locality restricted Linear Coding (LLC) is employed to illustrate each input feature using collective codebook elements. For classification purpose, SVM is used after creating the dictionary. The application of subspace clustering by [(H. Zhang and O. Yoshie, July 2012.)] enhances the action recognition because the standard clustering approach was unable to manage the complex or multi dimensional data. SUBCLU is a density related clustering technique that finds clusters in axis parallel subspaces and SCAR (subspace clustering based approach) uses SUBCLU [(K. Kailing, 2004.)] as its main clustering algorithm. The sensors collect the data, and utilising SUBCLU, the traits necessary for identifying activity of human is being extracted and produced as clusters. According to [(N. Robertson and I. Reid, 2006)], a method for recognising human activity is based on non parametric variants of trajectory data and concurrent motion properties, which are then merged with bayesian network system. The information was obtained using a colour based tracker. Hidden Markov Model is used to infer the activities.

B) MODEL BASED APPROACHES

In the model based strategy, a human model is built to recognise actions. By taking a kinematic method and extracting information from a series of [(L. Liu, Dec 2013)] build a model from which they project actual postures of human. By building a kinematic model out of layer tint, [(A. A. Chaaraoui, 2013)] are able to learn a series of positions. The HAR is expanded by [(A. A. Chaaraoui, 2013)] to include adaptive and incremental learning.

Scholars project the stance interpretation of human actions from video clips by extracting information from a series of frames. Basic postures are primarily used to train the model to identify actions in a human like manner. The kinematic model is built by [(R. Vemulapalli, June 2014)] utilising skeleton representation. Using geometrical operations which include translating and rotating in 3 Dimensional space, scholars explicitly design the model using geometrical relationship between distinct bodily parts. Despite being effective, kinematic features are higher in dimensional and have little between class variance. The environment while the movement will not be taken into consideration by these models. Therefore a dynamic feature based on physics model is built by [(A. Mansur, Aug 2013.)]. Since scholars can account for gravitation, ground impact, as well as various types of physical interactions with the ground, physics related models are much discriminative than kinematic models.

ACTIVITY RECOGNITION OF HUMAN OBJECT INTERACTION

Because of various factors such as shape, size, dimensions, location, and colour of an object handling interaction among the objects and humans in human activity recognition, is a difficult process. The handling of human and object interactions in still photos and videos is illustrated here. Figure 3 displays a sample of human object interaction.

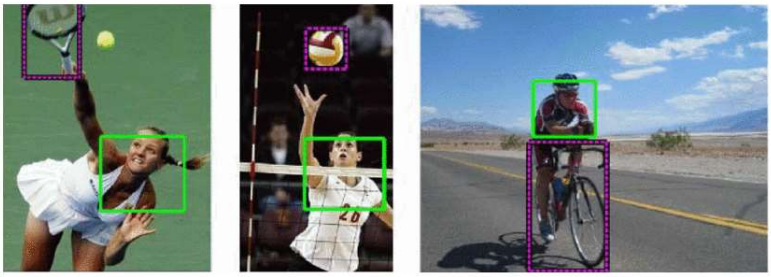

Figure 3: Human object interaction

A. ACTIVITY RECOGNITION OF HUMAN OBJECT INTERACTION IN STILL IMAGES USING SVM

[(V. Delaitre, 2011)] create a co-occurrence framework utilising the human-object interaction recognition system that are present in still photos for some portions of the object detectors that have been trained. Sparse Support Vector Machine classifer is used by them for classification purpose. Due to the tiny size of the objects and the significant impact that interference has on recognition, the system's success solely rely upon the object detectors. The method in the paper [(B. Yao and L. Fei-Fei, 2012)] described for overcoming this issue that involves simultaneously detecting human body parts, positions, and objects while operating under the presumption that recognising one person will result in the discovery of others. Due to the majority of these

methods rely just on detectors, adding relevant information, as suggested by [(C. Desai, 2010)], could improve the system's identification rate.

B. ACTIVITY RECOGNITION OF HUMAN OBJECT INTERACTION IN VIDEOS USING FRAMES

A method developed by [(A. Prest, 2013)] involves extracting the frame also using the detectors of objects and humans that identifies both the individuals and items in a specific frame. The relative motion is used to facilitate learning. However, the system's performance suffers as a result of the abundance of detectors. In order to get around this, [(S. M. Amiri, 2014)] use a kernel function to compare the connections between the human body components as well as the objects in two video instances in order to determine how similar they are. The system's key benefit is that it doesn't depend on labelling. [(A. Gupta, 2009)] utilises the motion of the limb to approach the object and uses the motion feature. The stability of the velocity field is the system's main point of popularity.

The algorithm however, struggles with managing uncontrollable videos and requires more time for training. [(A. Prest V. F., 2013)] has given a solution to this issue. Objects and humans are both closely monitored. The precise location and mobility of an object and humans for the activity is depicted in the interaction process. The majority of the techniques covered above don't consider spatial data. Because of this, [(C.-Y. Chen and K. Grauman, 2012)] has contributed spatial information while failing to capture the temporal features that [(Niebles, 2013)] resolve using spatio temporal cues. The tracker is low grade association based. Because of the tracker failing in certain videos, the system's restriction is the manual annotation addition.

ACTIVITY RECOGNITION OF HUMAN HUMAN INTERACTION

Another difficult challenge in the identification of human activity is human and human interactions. The interactions are categorised into one to one and group class interaction. One to one interaction may be a logical activity that takes place when two people are in charge of one another. The ability to recognise a group of objects under the social aggregate condition is a prerequisite for group interaction activity recognition. The movement aspects for both scenarios should be combined with sociological and psychological knowledge that governs interpersonal relationships.

DATASET INSIGHTS

The six various human action interactions covered by the UT-Interaction dataset [(C. Chen, 2012)] are handshake, hug, kick, punch, push and point. Subjects have shown up wearing 15 different colours of apparel. Total number of video clips are 20 that are made up of 720 x 480 resolution at the rate of 30 frames per second video clips in all. There are two equal sets of these sequential video clips. Set 1 consists of 10 parking lot video clips having static backgrounds and varying zoom levels. The remaining 10 more videos in sequences are included in set 2, which is placed against a gradually shifting background near to the meadow with camera glitches. The UT- Interaction sample dataset for activity recognition are shown in Fig 4

Figure 4: UT- interaction Dataset

For the evaluation of the deep learning models, the UT interaction dataset. This is a publicly available dataset. This dataset contains videos which are recorded under the surveillance of a real environment. These videos shows the interactions between two persons from a survilliance perspective. The different types of interactions that can be observed in the video are:

1. shaking hands
2. point the other person
3. huging the other person
4. kick the other person
5. punch the other person

This dataset contains 20 sequences of videos. These videos are further divided into two sets. The first set contains 10 video sequence which are taken in an environment of a parking lot. Here, in the parking lot, the zoom rate is different from the outdoor environment and the background in the video are mostly static. There seems to be no jitter in the camera survillance. The second set also a sequence of 10 videos. These videos were recorded in a environment of lawn. The climate

present on the day of recording was windy. Background in the images move slightly. An example of this movement is the shaking of the trees in the background due to the wind. Because of the disturbance in the background, the recording contained more jitters.

The 120 video sequences represent six different types of interactions in the dataset. These interactions are produced from the 20 sequences. These sequences are used to train the classification model that classifies all the uniquely differentiable frames in the video into one of six pre-defined labels of interaction.

METHODOLOGIES

LSTM MODEL

The architecture using recurrent neural networks is called as Long Short Term Memory (LSTM) that was introduced in [(Hochreiter, 1997)]. Eventually, a different version with no forget gates is introduced in [37] and further developed in [38]. Whenever the LSTM model is unrolled over the time, it resembles to a similar architecture of deep neural networks and was created to address gradient blow up issues or gradient decay issues. The memory block is the important component of the LSTM layer. Input gate, output gate, and forget gate are the three gates that make up an LSTM block. These significant gates are considered as read, write and reset functions. The most essential component that transmits information among the LSTM block is known as LSTM cell state. The three gates mentioned here governs the changes to the state of a cell. Figure LSTM depicts a single LSTM cell and the connections between the gates, and the state of the cell.

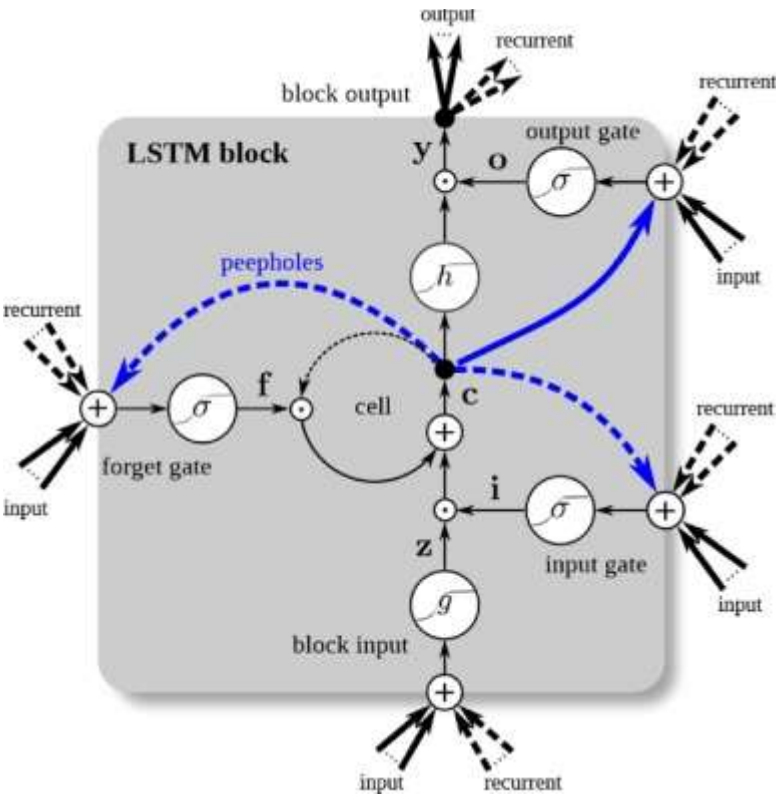

Figure 5: LSTM block

(Zhang)

Figure shows the LSTM cell structure and the following description explains how the gates work. Choosing which data that should be forgotten by the forget gate in the cell is the first most step in an LSTM cell. The forget gate layer technically known as Sigmoid layer, decides on this action. For each integer in the cell state C suffix (t-1), the system generates the output as a number from 0 to 1 after looking at h suffix (t-1). While 0 means completely eliminate this and 1 means preserve it entirely. The formalised output f suffix t of the cell gate is computed by:

$$f_t = \sigma\left(Wf \cdot [h_{t-1}, x_t] + bf\right)$$

The cell then selects the new data that would be kept inside the cell state. Two sections make up this. The input gate layer or a sigmoid layer, first determines what values are undertaking for an update. A tanh layer then generates a vector parameter of newly assigned candidate values Ct cap, that can be included to the cell state. To update the state, the following two values are joined to produce the following result:

$$it = \sigma(Wi \cdot [h_{t-1}, x_t] + b_i)$$

$$\widehat{Ct} = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right)$$

The old cell state C suffix (t-1) is then updated into the latest cell state Ct as seen below:

$$C_t = C_{t-1} * ft + i_t \widehat{C_t}$$

A Sigmoid layer determines which elements of the cell state is considered as output before the output is produced. The state of a cell is then amplified using the output value of the Sigmoid gate after being fetched by tanh function (to set the values in the range of -1 to 1) as follows.

$$ot = \sigma(Wo \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = \tanh\left(C_t\right) * o_t$$

(Eiman Kanjo, 2019)

Frame Extraction using Differential Evaluation:

Algorithm:

The following parameters are used in defining the differential evaluation algorithm:

1. The dimensional size - D
2. The size of the population - NP
3. The factor of Scaling - F
4. The weight crossover - Cr
5. The maximum number of generations (also called as 'MaxGen') that will be used in each run

The dimensional size (D):
The dimensional size here is the size of each vector defined for the frame in the video. It is the total count of the frames that would be required for the deep learning algorithm to effectively recognise the human activity in the video. This value is dependent on the algorithm that is used and can be taken as an input.

The size of the population (NP) :
It is the count of the total number of possible candidates in the video. This size has to be defined such that it is moderate enough. If the value selected is higher , then the frame selection process explores more possible frames which can include potentially unwanted candidates. Also, if the value selected is lower, then there is a possibility that this selection process can miss some of the potentially optimum frames. Hence, this value is generally selected moderatively.

The factor of Scaling - F:
It is the number of steps that is used for the process of differential mutation.

The weight crossover - Cr:
It is the value of the probability used to check if the candidate frame that can be selected as a optimal frame is from a source vector or a muted vector. Because this value of weight is a probability , the possible range of values is 0 till 1. This is also the same for the factor of scaling. Since, the performance of this algorithm depends on scaling factor and weight crossover, these parameters are tuned using the validation data.

The set of all possible candidate frame in the video is called the 'population'. Initially, the population is defined by randomly selecting frames from the input video.

Objective Function used:
A function that calculates the average SSIM value of the solution is defined and used as an objective function. This function is used to evaluate the goodness of each possible solution. This value is optimised to merge into a best solution.

Algorithm:

The population is first initiated with some frames in video chosen as the key frames at random. Each solution vector has a dimensional size. An evolutionary search process is used to find the fittest solution.

Each cycle of this search process involves 3 steps in the sequence of mutation, crossover operations and intersection. At the end of execution of each cycle, a child vector is generated for each of the input vector. Among the source vector and the child vectors, the one that is fittest is chosen for the next cycle. this process is repeated until the optimal solution is reached using the pre defined stopping criteria. After the process is terminated, the vector with the most fitted values is chosen as the optimal solution. Because, the chosen fitness value is SSIM, the key frame with a least average of SSIM is chosen as the best combination vector. The values of this vector are the frames that are chosen to represent different scenes in the video.

Organising the dataset:

The dataset contains 120 videos that involves 6 human activity interactions namely,

1. handshake
2. hug
3. kick
4. punch
5. point
6. push

This dataset is split in a proportion of 8:2 , where 80% of the dataset is used as training data and 20% of the dataset is used as the test data. So, this makes the size of training dataset to be 96

videos and the size of testing dataset to be 24 videos. From these videos, keyframes are extracted for each of the interaction type. The interaction here is between two persons.

In the dataset used, there are equal number of videos for each type of interaction. So, this eliminates the problem of class imbalance, in which differently sized classes can make the model to predict in the favour of classes with large data.

Key frame Extraction:

Using the differential key evaluation, the key frames in each of the videos are extracted. This algorithm loops through all the possible frames and provides the best possible set of key frames in the video. This removes the burden of selecting key frames manually. The two possible metrics for the extraction algorithm are entropy and euclidean distances. Here in this implementation, euclidean distance is used as a metric because it can help in extracting frames which are repetitive in nature.

Creating Data arrays for Frames:

Each frame in the video is represented using a 3 dimensional array. The three dimensions used here are the number of samples, the dimensional size of each image and the number of channels. The number of channels represents the colour format used for the images. Mostly used systems are 3 and 2, which represent RGB (Red Green Blue) and B / W (Black and White) systems respectively. Even though the RGB system can show a wide range of colours, the BW format is used because it requires lesser computational time in training and validating the model.

Feature Extraction Models:

Feature Extraction process:

It is the process of reducing the number of dimensions using an initial set of raw and unprocessed data with large dimensions to produce processed data within a smaller set of groups. A large number of dimensions in the data would require huge computational time to build a model using these features.

This process of feature extraction process helps in selecting and / or combining variables to produce features that accurately represent the original population represented by the dataset. This process also reduces the amount of redundant data in the dataset. Alos, This process also helps in removing the redundant variables in the dataset. It is also useful in reducing the total amount of data required to be processed to build the model. Building reduced number of features and thereby, decreasing the amount of data to be processed helps in speeding the process to learn and build a model that accurately generalises the selected dataset.

Deep Learning model:

Long term short memory (LSTM) network is used to predict the human interaction action from the dataframes. LSTM which is a special type of Recurrent Neural Network (RNN) is used

because it is known to effectively learn possible long term dependencies in the data. The input to a LSTM model is expected to a 3 dimensional tensor. Hence, the dataset is reshaped to represent using the following 3 dimensions:

1. Number of samples
2. Number of sequences
3. Number of features

For the dataset chosen, the shape of the test dataset is 96 * 5 * 512.

Building the Model:

The LSTM model is trained using the reshaped dataset. To train the dataset faster, the normalisation and dropout layers are used. Different number of epochs and learning rates have been experimented to find the optimum value for each hyper parameter. For this, the process of hyper parameter tuning is used. The number of epochs and learning rate parameters are tuned. The adam and adagrad optimisers were evaluated for the gradient descent algorithm.

<can be added>

Result Analysis:

Graphs that shows the change of accuracy metric in each epoch for both training and testing datasets are plotted. Also, the progress of the value of loss function with the number of epoch is examined. This will help in analysing how the model has progressed for both training and testing dataset in comparsion. This can also assist in selecting the exact stopping point before the data gets overfitted on the training data.

Environmental Setup:

Anaconda is used to create a virtual environment specifically to build and evaluate different models. All the necessary packages were installed in the newly create environment. Also, Anaconda provides default Integrated Development Environments (IDEs') like Spyder and Jupyter Notebook. For the execution and debug of the code, the spyder IDE is selected.

For the implementation of the models, the following packages were installed,

- python version 3.7
- pip version 19.0
- jupyter notebook
- TensorFlow - gpu=2.0
- SciKit - Learn
- SciPy
- Pandas
- matplotlib
- pillow
- tqdm
- h5py

- pyyaml
- flask
- opencv-python

IMPLEMENTATION:

Importing Modules:

All the necessary modules were imported. the numpy module is imported to work with array data. Matplotlib will help in plotting the graphs to evaluate the models using the metrics like accuracy. Keras module is imported to build the models. Different sub modules like model check point, early stopping, train test split, model, sequential, layers and optimisers were imported seperately.

Conversion of Videos to Data frames:

The key frames are extracted from the video using the implementation of the differential evaluation. The metric used here is euclidean distance. Each video is converted to 5 key frames that represent the actions depicted in the videos. Hence, the number of frames parameter is set to 5.
All these frames are then converted to numpy arrays. the data type of the array is 32 bit sized float.

Deep learning model:
The following layers were used in building the deep learning model:

1. A two dimensional convolutional layer: This layer is used to perform convolutional operations on the input vectors.
2. Batch Normalisation layer: This layer performs normalisation operation to scale the input data. The scaled data can ease the computational requirement of training.
3. Dropout layer: This layer removes a portion of neurons in random.
4. MaxPooling layer: Using the given size of the kernel, the max pooling layer reduces the size of the input tensor.
5. Dense Layer: This layer propagates the data without reduction in the dimension.

AlexNet:

AlexNet, a special type of deep neural network is used to evaluate its performance on ut interaction dataset in recognising human interaction. The AlexNet was inititally defined in [(Hinton G. 2., 2017)] as an image classification algorithm. The depth of the model helps in

achieving high accuracy at the cost of high computational requirements. This makes the requirement of having a graphical processing units (GPU) for training the model.

The Alex Network is a sequential deep neural network that contains a combination of two dimensional convolutional networks, max-pooling and batch generalisation. The general architecture of the Alex network defined in [(Hinton G. 2., 2017)] contains 5 convolutional layers and a total of 3 fully connected layers. The standard dimension of the input that is given to the network here is 224 * 224 * 3.

The Alex network implemented here is optimised to use only 4 convolutional layers. Also, the network uses only 2 fully connected layers instead of 3 layers. Here, the size of the input vector is 64 * 64 * 3. This is done to reduce the requirement of total computational time for training the model. In addition to these layers, the model also contains batch normalisation and two dimensional max pooling layers in each group. The activation function used is the relu function. However the 'softmax' function is used in the final group of layers.

Convolutional Neural Network:
For comparing and evaluation purpose, a convolutional network that uses 2 convolutional layers which is followed by a max pooling layer. This approach is similar to the alex network in that both of them use convolutional and max pooling layers. The activation function used is 'relu'. After the continous application of these layers the size of the input is reduced to 8 * 8 * 3.

After this, two fully connected layers are to the defined neural network. These connected layers are followed by a layer of flatten. Also, each connected layer is combined with a layer of dropout with a weight of 0.5. The activation function used for the last layer is 'softmax'.

Feature Extraction:

The model for feature extraction is defined as taking full sized data frame as an input and the size of the output is the dimension of the layer just before the final layer of the CNN model. this model is used to predict the features of the model. The prediction is performed on both the training and testing data. This produces a total of 512 features. These features are extracted and saved to another data frame.

Data Preparation:
Before the data frame is fed to the neural network, this frame has to be reshaped to match the required format of LSTM and Alex networks. The two dimensional data array is reshaped into 3 dimensions. The third dimension is introduced by creating blocks of dataframes. The number of blocks chosen here is 5.
As discusses To reshape the input data, two functions are defined:

create_data_blocks():

This function creates blocks of sequences of the input dataframe. This function has two parameters, the dataframe and the number of sequences.After the split is done, the reshapped dataframe is given as the output.

Create_Block_Labels():
This function returns the labels in the blocked format. All the 5 labels are combined to represent a single label because all of these represent the same data row in the frame.

This function is reused for both LSTM and Alex networks because the shape of the input is the same for both networks.

Training the Model:

Now the reshaped data frame is ready to be consumed for the neural networks.
Firstly, the AlexNet model is compiled using the cross entropy loss function. The adam optimiser is used to calculate the stochastic gradient descent in the training. Accuracy is used as the evaluation metric. the total number of epochs for training is 15. The compiled model is fitted on the training dataset. The Accuracy and loss values are calculated for both training and testing data for each epoch.

Then, the LSTM model is compiled using the cross entropy loss function. Like in Alex model, the adam optimiser is used to calcuate the gradient descent for training the model. The evaluation metric used here is Accuracy. The total number of epochs for training is selected as 20. This complied model is then fitted on the training data.

Plotting the Metric Values:

Once the Alex model is trained, the accuracy score for each epoch is plotted against the epoch number. Also, the value of the loss function is plotted against the epoch. This is done for both training and testing data.

Similar graphs are created for LSTM model for the accuracy and loss function scores.
To plot the graphs, the matplotlib library is used.

Results and Discussion:

Alex Model:

At the end of 15 epochs, the accuracy score was observed as 0.875. The cross entropy loss value was observed as 0.41.

Metric Evaluation:

Accuracy:
For the test data, The model started with an accuracy of 0.55 and rapidly increased for each epoch until the second epoch. After that, the accuracy score remained constant with the highest value of 0.875.

In the case of the training data, the model started with an initial accuracy score of 0.2. Until the epoch number 2, the increase in accuracy score was rapid and then the accuracy score increase was minimal. Even though there was a glitter in the accuracy in between, the maximum accuracy score of 0.92 was observed at the final epoch.

It can be seen that the model performed with higher accuracy on the training data than on the test data.

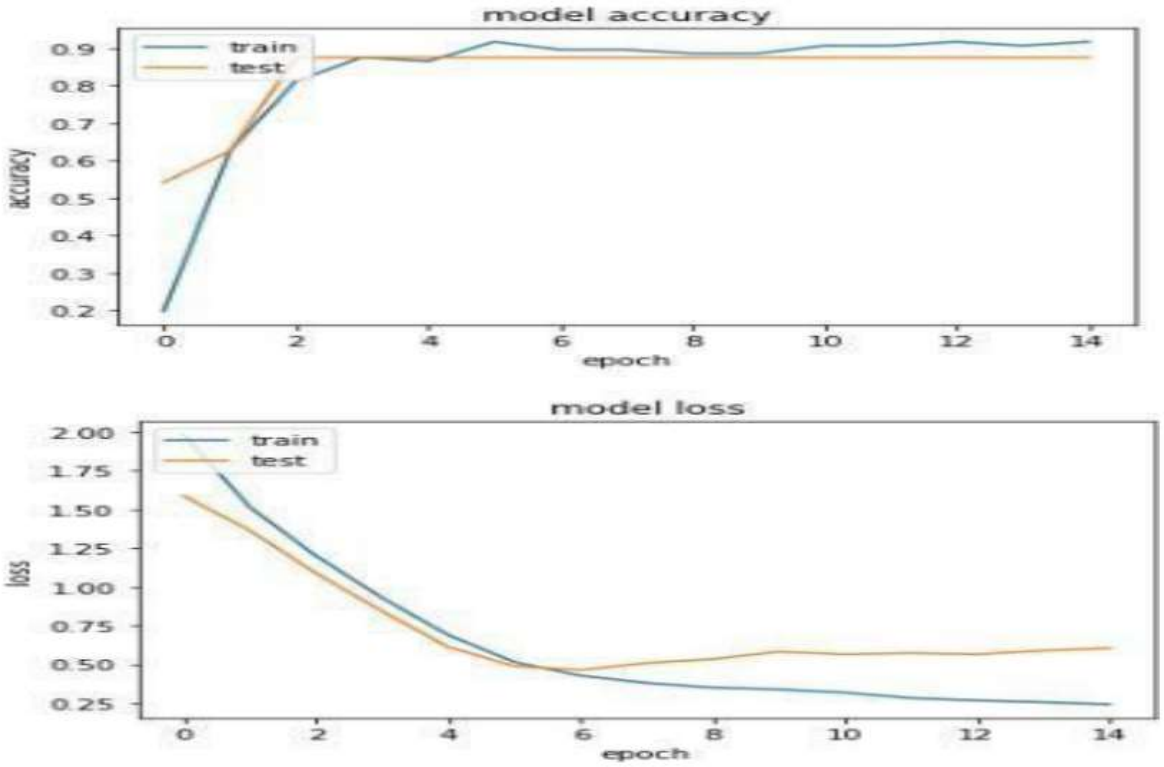

Figure 6: Alex network model

Loss Function:

For the test data, the cross entropy started with a value of 1.60. This value was observed to be decreasing largely until it reached a minimum value of 0.55 at the epoch number 5. Then this value slightly increases till the epoch number 14 to reach a value of 0.75.

For the training data, the value of loss function was initially observed to be 2.00. This value rapidly decreased until the epoch number 5 to reach a value of 0.55. This value further decrease gradually until a minimum value of 0.19 is reached in the epoch number 15.

It can be observed that both the loss function curves for test and training data meet at the epoch number 5 for a value of 0.55.

Also, the gradual decrease in the loss function in the latter stages of the model training shows that the model is able to learn spatial and temporal patterns in the data.

LSTM model:
At the end of the training with 20 number of epochs, the model reached an accuracy of 0.87 and cross entropy value of 0.42.

Metric Evaluation:

Accuracy:

For the test data, the model started with an accuracy score of 0.1. This score rapidly increased till the epoch number 2. After this, the accuracy score remained constant at 0.91. This value decreased to value of 0.875 after the epoch number 16 and remained constant till epoch 20.

It can be observed that the maximum value of accuracy is 0.91.

For the training data, the model saw an initial accuracy score of 0.3. This value rapidly increased untill the epoch number 3. Beyound this , the increase in the value was steady. A maximum accuracy of 0.96 was observed at the last epoch.
It can be observed that even though the accuracy increased till the end of training for the train data, this value decreased beyound the equilibrium. It can be understood that the model was getting overfitted with the training data and hence, the accuracy decreased on unseen data even when the score increased on the train data. So, the use of early stopping criteria can be helpful to prevent the model to be overfitted. An early stopping criteria of decrease in the accuracy for validation data can be used as a stopping point for training.

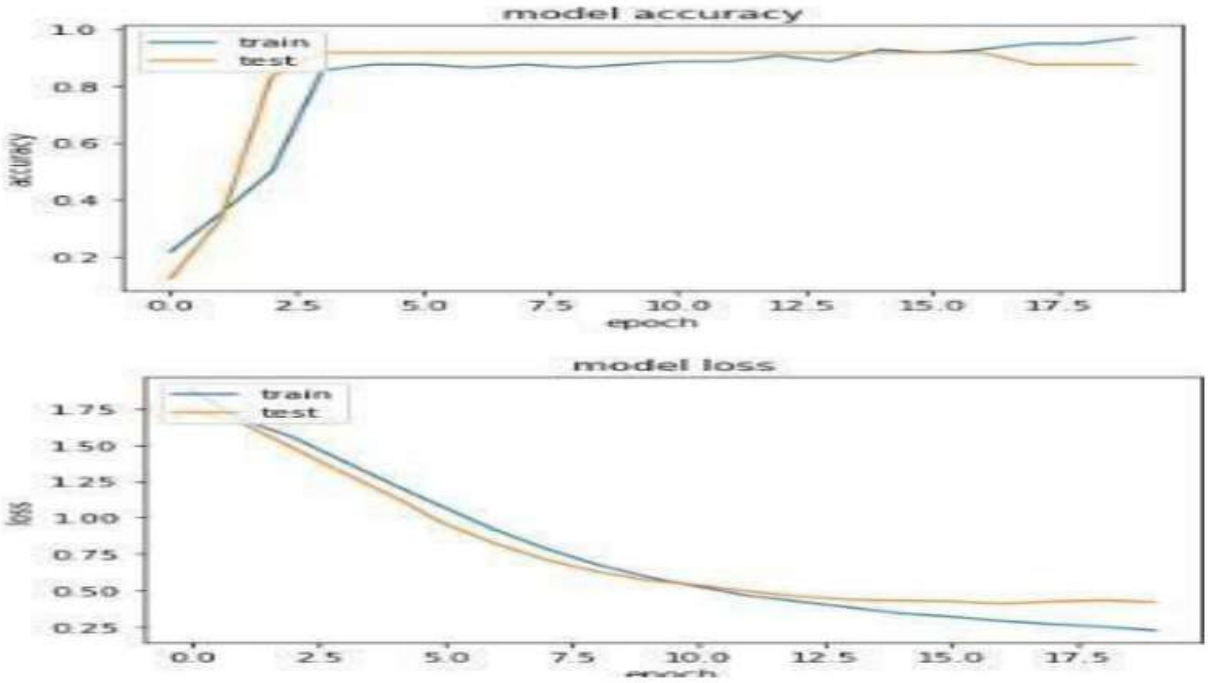

Figure 7: LSTM Model

Loss Function:

For the test data, the model started with a loss function value of 1.7. This value gradually decreased till epoch 10. From this point, the cross entropy value remained almost similar until the value of 0.42 is obtained for the last epoch.

For the training data, the model started with a loss function value of 1.75. Similar to the training data, the loss function value decreased rapidly till the epoch number 10 and then further decreased with minimal change. A minimum value of 0.22 was observed in the final epoch.

It can be seen that the curves of both the datasets are similar. Also, the loss function value and the epoch show an inverse proportional relation.

Conclusion:

Traditional methods of feature extraction required manually filtering the feature vectors from the data. However, With the help of differential key evolution algorithm, the key frames were generated for all the sample videos in the dataset. this eliminated the requirement of manually extracing the key frame from the videos. After the key frames are extracted, they are represented in the form of data frames. For the purpose of evaluating and selecting a candidate as a key frame, the euclidean distance has been chosen as measure. From these dataframes, the feature

extraction process was implemented to filter out the variable that do not provide value to building the model. The feature extraction model produced a total of 512 features for the model. |In this research, two deep learning models based on convolutional neural networks have been implemented. These models are evaluated on the publicly available dataset of UT interaction. Because the dataset was not skewed , the problem of class imbalance was not observed. This data was provided as blocks to the input of the deep learning networks, because the models require the input vectors to be in the form of 3 dimensional data arrays. Both the networks were able to successfully learn all the types of activities in the sample videos. However this research can be further extended to see working of these models on different data sets and more number of activity types. The results obtained using the LSTM and AlexNet models show that the deep learning models outperform in the task of human activity recognition when compared to the contemporary machine learning algorithms. this increase in performance thus also shows increase in the accuracy and better recognition results.The LSTM model here consisted of the cells that are connected sequentially. The parameters of the models were trained using the adam optimiser. As shown in [(Singh, 2017, August)], these models performed better in comparision with other machine learning models like HMM, HSMM, CRF and Naive Bayes. It has also been observed that both the models performed with similar accuracy scores, however LSTM model was overfit with the training data as the process of training progressed. Also, the AlexNet model was better in recognising the activities when there is some amount of noise and reflections present in the videos. The modified version of the AlexNet that is implemented here contained showed a similar accuracy scores and are better than the state of the art machine learning models. Because, the models used here were the lighter version of the original models, the full versions of the models are expected with greater accuracy and performance in recognition. however this comes with a shortcoming that they would require huge computational costs. the deep neural networks used here showed a noticable amount of validation loss when training the model. This can be reduced by creating large sequences of data for validating the model in each epoch.Also, the shorter sequences of the frames are observed to be recognised effectively by the models. However in case of large sequences, the full version of the networks is required to be used to learn deep patterns in the video sequences.

Future Work:

Different optimisational and hyper parameters of the LSTM models can be tuned to find the optimal values for each of them. the values of these hyper parameters can largely effect the performance of the deep learning models. Other deep learning models can be evaluated to check thier effectiveness in recognising activities in raw data. Even though the deep learning model is efficient in learning complex patterns in the dataset, they are slightly less consistent in capturing the uncertainty of the unseen data. However some of the machine learning models like, bayasian classification is known to offer the feature of learning the uncertainty. In [(Gal, 2016)], a framework that uses dropout training methodology that works as an approximation to the bayasian model was presented. This kind of model can help in overcoming the issue of showing the uncertainity factor in a deep learning model without needing to reduce the accuracy or increase the computational time required.

Experiments with a larger size of datasets can understand the robustness of the implemented models. These models can be more regularised by introduing more data for each activity that has to be recognised.

Also, recognising the activity in a video when more than 2 humans are involved (Multi - person Activity recognition) can also be considered as one of the future work.

References

A. A. Chaaraoui, P. C.-P.-R. (2013). Silhouette based human action recognition using sequences of key poses". *Pattern Recognition Letters, vol. 34, no. 15, pp. 1799-1807.*

A. Gupta, A. K. (2009). "Observing human-object interactions: Using spatial and functional compatibility for recognition", . *Pattern Analysis and Machine Intelligence IEEE Transactions, vol. 31, no. 10, pp. 1775-1789*.

A. Mansur, Y. M. (Aug 2013.). Inverse dynamics for action recognition. *Cybernetics IEEE Transactions, vol. 43, no. 4, pp. 1226-1236, .*

A. Prest, V. F. (2013). "Explicit modeling of human object interactions in realistic videos", . *Pattern Analysis and Machine Intelligence IEEE Transactions, vol. 35, no. 4, pp. 835-848, .*

A. Prest, V. F. (2013). "Explicit modeling of human object interactions in realistic videos", . *Pattern Analysis and Machine Intelligence IEEE Transactions, vol. 35, no. 4, pp. 835-848*.

Abdallah ZS, G. M. (2015). “Adaptive mobile activity recognition system with evolving data streams,” . *Neurocomputing, vol. 150, pp. 304–317.*

B. Yao and L. Fei-Fei. (2012). "Recognizing human-object interactions in still images by modeling the mutual context of objects and human Poses", . *Pattern Analysis and Machine Intelligence IEEE Transactions on, vol. 34, no. 9, pp. 1691-1703*.

Bishop CM. (2006). Pattern recognition and machine learning. . *springer link,*.

C. Chen, Y. X. (2012). "Semi-supervised and compound classification of network traffic", . *Proc. WDCS, pp. 617-621*.

C. Desai, D. R. (2010). Discriminative models for static human-object interactions". *Computer vision and pattern recognition workshops (CVPRW) 2010 IEEE computer society conference on. IEEE, pp. 9-16,*.

C.-Y. Chen and K. Grauman. (2012). "Efficient activity detection with max subgraph search". *Computer Vision and Pattern Recognition (CVPR) 2012 IEEE Conference on. IEEE, pp. 1274-1281*.

D.J., Y. Z. (2012). Sensor-based activity recognition. *IEEE Trans. Syst. Man Cybern., Part C (Applications and Reviews) 42, 790–808* .

Dobbins C and Rawassizadeh R. (2018). "Towards clustering of mobile and smartwatch accelerometer data for physical activity recognition," . *in Informatics, vol. 5, no. 2 Multidisciplinary Digital Publishing Institute, p. 29.*

Domingos PM. (2012). "A few useful things to know about machine learning.". *Commun. acm, vol. 55, no. 10, pp. 78–87.*

Duong, T. B. (CVPR 2005). Activity recognition and abnormality detection with the switching hidden semi-markov model. *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 838–845*.

Eiman Kanjo, E. M. (2019). Deep learning analysis of mobile physiological, environmental and location sensor data for emotion detection,. *Information Fusion, volume 49, ISSN 1566-2535, page 46 - 56*.

Gal, Y. G. (2016). Dropout as a bayesian approximation: representing model uncertainty in deep learning. . *In Balcan, M.F., Weinberger, K.Q. (eds.) proceedings of The 33rd International Conference on Machine Learning (ICML), vol. 48, pp. 1050–1059. PMLR* .

H. Zhang and O. Yoshie. (July 2012.). "Improving human activity recognition using subspace clustering",. *Machine Learning and Cybernetics (ICMLC) 2012 International Conference, vol. 3, pp. 1058-1063,*.

Hinton, G. 2. (2017). Imagenet classification with deep convolutional neural networks. . *Communications of the ACM, 60(6), pp.84-90.*

Hinton, G. 2. (2017). Imagenet classification with deep convolutional neural networks. . *Communications of the ACM, 60(6), pp.84-90.*

Hinton, G. D. (2012). Deep neural networks for acoustic modeling in speech recognition. *The shared views of four research groups. IEEE Sig. Proces*.

Hochreiter, S. S. (1997). Long short-term memory. . *Neural Comput. 9, 1735–1780* .

Holzinger, A. (2017). Introduction to machine learning and knowledge extraction (MAKE). *Mach. Learn. Knowl. Extr. 1, 1–20*.

J, B. (2016). Master Machine Learning Algorithms: discover how they work and implement them from scratch. . *Machine Learning Mastery.*

K. Kailing, H.-P. K. (2004.). "Density-connected subspace clustering for high-dimensional data", . *Proc. SDM, vol. 4.*

L. Liu, L. S. (Dec 2013). "Learning discriminative key poses for action recognition". *Cybernetics IEEE Transactions, vol. 43, no. 6, pp. 1860-1870.*

Lee, H. G. (ACM 2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 609–616.* .

Lee, H. P. (2009). Unsupervised feature learning for audio classification using convolutional deep belief networks. *In: Advances in Neural Information Processing Systems, pp. 1096–1104* .

Liu Y, N. L. (2016). “From action to activity: sensor-based activity recognition,”. *Neurocomputing, vol. 181, pp. 108–115.* .

Liu Y, N. L. (2016). From action to activity: sensor-based activity recognition,. *Neurocomputing, vol. 181, pp. 108–115.*

Marcus G. (2018). Deep learning: A critical appraisal,. *arXiv preprint arXiv:1801.00631.*

N. Robertson and I. Reid. (2006). A general method for human activity recognition in video. *Computer Vision and Image Understanding, vol. 104, no. 2, pp. 232-248.*

Niebles, V. E. (2013). "Spatio-temporal human-object interactions for action recognition in videos", . *Computer Vision Workshops (ICCVW) 2013 IEEE International Conference on. IEEE, pp. 508-514*.

R. Vemulapalli, F. A. (June 2014). Human action recognition by representing 3d skeletons as points in a lie group. *Computer Vision and Pattern Recognition (CVPR) 2014 IEEE Conference, pp. 588-595.*

Roecker, C. Z. (2011). Social inclusion in ambient assisted living environments: home automation and convenience services for elderly users. *In: Proceedings of the International Conference on Artificial Intelligence (ICAI 2011), pp. 55–5*.

Roggen, D. C.-D.-V. (EWSN 2015). Limited-Memory Warping LCSS for real-time low-power pattern recognition in wireless nodes. *In: Abdelzaher, T., Pereira, N., Tovar, E. (eds.) LNCS, vol. 8965, pp. 151–167. Spri*.

S. M. Amiri, M. T. (2014). "A similarity measure for analyzing human activities using human-object interaction context",. *Image Processing (ICIP) 2014 IEEE International Conference on. IEEE, pp. 2368-2372*.

S. Umakanthan, S. D. (Aug 2014.). Multiple instance dictionary learning for activity representation. *Pattern Recognition (ICPR) 2014 22nd International Conference, pp. 1377-1382*.

Shickel B, T. P. (2017). “Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis,” . *IEEE journal of biomedical and health informatics, vol. 22, no. 5, pp. 1589–1604.*

Singh, D. M. (2017, August). Human Activity Recognition Using Recurrent Neural Networks. *In International cross-domain conference for machine learning and knowledge extraction (pp. 267-274). Springer, Cham.*

T. Subetha and S. Chitrakala. (2016). "A survey on human activity recognition from videos," . *International Conference on Information Communication and Embedded Systems (ICICES), 2016, pp. 1-7, doi: 10.1109/ICICES.2016.7518920.*

V. Delaitre, J. S. (2011). Learning person-object interactions for action recognition in still images. *Advances in neural information processing systems, pp. 1503-1511,*.

Vaughn A, B. P. (2018). “Activity detection and analysis using smartphone sensors,”. *in 2018 IEEE International Conference on Information Reuse and Integration (IRI). IEEE, pp. 102–107.* .