Université de Montréal

Real-time Human Action and Gesture Recognition Using Skeleton Joints Information Towards Medical Applications

par

Marulasidda Swamy Kibbanahalli Shivalingappa

Department of Computer Science and Operations Research
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Informatique, option Intelligence Artificielle

September 29, 2020

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

Real-time Human Action and Gesture Recognition Using Skeleton Joints Information Towards Medical Applications

présenté par

Marulasidda Swamy Kibbanahalli Shivalingappa

a été évalué par un jury composé des personnes suivantes :

Esma Aïmeur
(Président-rapporteur)

Claude Frasson
(Directeur de recherche)

Jean Meunier
(Membre du jury)

Résumé

Des efforts importants ont été faits pour améliorer la précision de la détection des actions humaines à l'aide des articulations du squelette. Déterminer les actions dans un environnement bruyant reste une tâche difficile, car les coordonnées cartésiennes des articulations du squelette fournies par la caméra de détection à profondeur dépendent de la position de la caméra et de la position du squelette. Dans certaines applications d'interaction homme-machine, la position du squelette et la position de la caméra ne cessent de changer. La méthode proposée recommande d'utiliser des valeurs de position relatives plutôt que des valeurs de coordonnées cartésiennes réelles. Les récents progrès des réseaux de neurones à convolution (RNC) nous aident à obtenir une plus grande précision de prédiction en utilisant des entrées sous forme d'images. Pour représenter les articulations du squelette sous forme d'image, nous devons représenter les informations du squelette sous forme de matrice avec une hauteur et une largeur égale. Le nombre d'articulations du squelette fournit par certaines caméras de détection à profondeur est limité, et nous devons dépendre des valeurs de position relatives pour avoir une représentation matricielle des articulations du squelette. Avec la nouvelle représentation des articulations du squelette et le jeu de données MSR, nous pouvons obtenir des performances semblables à celles de l'état de l'art. Nous avons utilisé le décalage d'image au lieu de l'interpolation entre les images, ce qui nous aide également à obtenir des performances similaires à celle de l'état de l'art.

Mots clés - Action humaine dans un environnement virtuel, Détection des gestes, Informatique médicale, Systèmes de réalité virtuelle, Apprentissage profond, Solution de RV pour la maladie d'Alzheimer, Facteurs humains pour le traitement médical.

Abstract

There have been significant efforts in the direction of improving accuracy in detecting human action using skeleton joints. Recognizing human activities in a noisy environment is still challenging since the cartesian coordinate of the skeleton joints provided by depth camera depends on camera position and skeleton position. In a few of the human-computer interaction applications, skeleton position, and camera position keep changing. The proposed method recommends using relative positional values instead of actual cartesian coordinate values. Recent advancements in CNN help us to achieve higher prediction accuracy using input in image format. To represent skeleton joints in image format, we need to represent skeleton information in matrix form with equal height and width. With some depth cameras, the number of skeleton joints provided is limited, and we need to depend on relative positional values to have a matrix representation of skeleton joints. We can show the state-of-the-art prediction accuracy on MSR data with the help of the new representation of skeleton joints. We have used frames shifting instead of interpolation between frames, which helps us achieve state-of-the-art performance.

Keywords— Human action in Virtual Environment, Gesture detection, Medical informatics, Virtual Reality Systems, Deep learning, VR solution for Alzheimer's, Human factors for medical treatment.

Contents

List of tables

List of figures

Liste des sigles et des abréviations

3D,2D	3 Dimensional and 2 Dimensional
ADI	Alzheimer's Disease International
AHE	Adaptive Histogram Equalization
BoVW	Bag of Visual Words
BN	Branch Normalization
ConvNet	Convolution Neural Network
CNN	Convolution Neural Network
C3D	Convolution 3D
D-CNN	Deep Convolutional Neural Networks
ESPMF	Extended Skeleton Pose-Motion Feature
FER	Facial Expression Recognition

GMM	Gaussian Mixture Modelling
HCI	Human Computer Interaction
HOG	Histogram of Oriented Gradients
HOF	Histogram of Optical Flow
HAR	Human Action Recognition
HE	Histogram Equalization
iDT	improved Dense Trajectories
MBH	Motion Boundary Histogram
MFs	Motion Feature vectors
NP	Nondeterministic Polynomial
PCA	Principal Components Analysis
PFs	Pose Feature vectors
PCB	Printed Circuit Board

RGBD	RGB-Depth
ROI	Region Of Interest
RGB	Red, Blue and Green; Refers to a system for representing the colors to be used in an image
ReLU	Rectified Linear Unit
ResNet	Residual Neural Network
RNN	Recurrent Neural Network
SDK	Software Development Kit *Collection of software used for application development*
SVM	Support Vector Machine
SPMF	Skeleton Pose-Motion Feature
SkepxelRel	Skeleton Picture Relative Elements
USB	Universal Serial Bus
VR	Virtual Reality

Thanks

I acknowledge Professor Claude Frasson, the National Science and Engineering Research Council (NSERC-CRD), and Beam Me Up for funding this work.

Chapter 1

Introduction

This chapter discusses HCI technology's evolution in the direction of helping in treating patients suffering from negative emotions. The chapter is organized to include motivation factors responsible for my thesis dissertation, "Real-time Human Action and Gesture Recognition Using Skeleton Joints Information Towards Medical Applications". Furthermore, the chapter explains the problem statement, research objectives, and organization of my thesis dissertation.

1.1. Evolution of technology and motivation

The ability to provide interactions between humans and sophisticated computer applications results in an efficient automated system to address complicated real-life problems. In 1996, ACM SIGCHI gave a more formal and technical definition for HCI as a discipline concerned with the design, evaluation, and implementation of interactive computing systems for human use and the study of significant phenomena surrounding them [**1**]. Most of the time, humans exchange information and emotion through speech. Speech signals provide varieties of useful information, including speech recognition, speaker recognition, emotion recognition, health recognition, language recognition, accent recognition, age, and gender recognition information [**2**]. Machines can identify humans based on variations and unique characteristics in the voice using the Speaker recognition solutions [**3**]. Identifying humans based on their speech helps to differentiate the person as an adult or child [**4**]. The author in [**5**] well documents the impact of speech recognition solutions in the healthcare sector and explains how continuous systems helped doctors when compared to discrete systems in the process of documenting patients analysis and patients care. Speech recognition is only an assisting tool for doctors in the process of documentation and recording patient care. Speech recognition converts audio or speech signals into the most reliable form of the data like text for further analysis. If the data is in text format, like reports of patients having psychological problems, it will help a computer application diagnose for a disease like severe

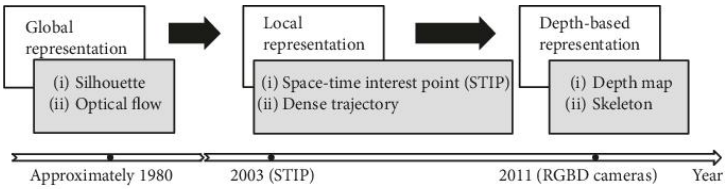

Fig. 1.1. Evolution of image representation method. Original figure was shown in [**7**].

depression, bipolar disorders, Alzheimer's, and schizophrenia. The authors in [**6**] proposed a method to prove the scope of medical records and the patient's medical history in the process of sentiment and emotional analysis. To make the prediction accuracy more accurate, the authors in [**6**] also use speech analysis and different speech pre-processing techniques. There are issues with the method of sentimental and emotion analysis using speech and text:

- There are no standards for doctor's acronyms.
- There are no known proper methods to extract hidden pain and sarcasm from the text.
- Efficient tokenization (that is grt, great, and gr8 should be replaced with a standard token) of a word having different forms is painful.
- It isn't easy to find out if a report has a biased or unbiased opinion about the patient's psychological status.

Human expressions convey the emotion and mental status of humans efficiently and better than information in the form of speech and text. Labeling human expression is more straightforward than labeling speech and text-based data. FER is used in different applications, including mental state identification, security, automatic counseling systems, face expression synthesis, lie detection, music for mood, automated tutoring systems, and operator fatigue detection [**8**]. To make facial expression detection more efficient, we need to pre-process static images or sequences of images for identifying ROI for eyes, nose, cheeks, mouth, eyebrow, ear, and forehead. Pre-processed data goes through one more level of feature extraction step, where features are extracted from ROI's [**8**]. Feature extraction steps are considered to be expensive in terms of resources and time. FER from static images and FER from the sequence of images [**9**] are two different problems since, in a sequence of images, the temporal evolution of expression exists. Executing feature extraction procedure

on all the pictures in a sequence turned out to be an expensive operation. Schizophrenia patients possesses a reduced ability to perceive and express facial expressions. Review work conducted by the authors in [**10**] shows that schizophrenia patients are highly sensitive to negative emotions like fear and anger despite a general impairment of perception or expression of facial emotions. Experiments are conducted in two phases, encoding and decoding phases. During the encoding phase, schizophrenia patient's ability to show different facial expressions is verified, and the decoding phase tests the ability to understand different facial expressions. Experiment results show that schizophrenia patients have a general impairment in the processing of emotions [**10**]. HAR introduces many applications such as automated surveillance, elderly behavior monitoring, human-computer interaction, content-based video retrieval, and video summarization. Applications capable of monitoring elderly behavior recognize "walking", "bending", and "falling", etc. from the video and takes proper action if there is a need for suggestions or informing emergencies. The hierarchical structure of human activities divides each action into three categories: primitive level actions, actions/activities, and complex interactions [**7**]. Image representation or feature representation approaches evolved consistently in recent years (evolution of image representation method is shown in figure 1.1) to extract only the useful and relevant information from a given image for improved classification accuracy. Global representation, local representation, and depth representation are the three standards followed by all recent research works for image representation. Local representation is an advanced method and more efficient compared to the global representation approach [**7**]. Image representation is a crucial pre-processing step in emotion detection and human action detection since an image carries a lot of unnecessary information, and this information needs to be filtered (as shown in figure 1.2). Unwanted information like noise, different background, and camera movements in an image makes the prediction accuracy suffer from intra-class and inter-class variance [**7**]. Depth image-based representation is more popular because of advanced RGBD cameras, which can directly provide depth representation of images in the form of depth-map or skeleton joints information. With depth image-based representation, we have less effort during image pre-processing steps and help implement efficient real-time human action recognition applications. In our proposed work, we use skeleton joints information provided by Intel Realsense camera and hand joints information provided by Leap motion camera to conduct experiments. Data given by Intel Realsense and Leap motion cameras is the depth representation of humans and hands, respectively, which in turn helps us to design applications for human action and hand gesture recognition in real-time. Every frame from the depth camera is a depth representation of a human/humans in a particular pose or human hand/hands pose at time 't'. To recognize a human action or hand gesture made by any individual, we need to encode all the frames in a sequence into an image format [**11, 12, 13, 14**] so that it can be trained using deep neural networks.

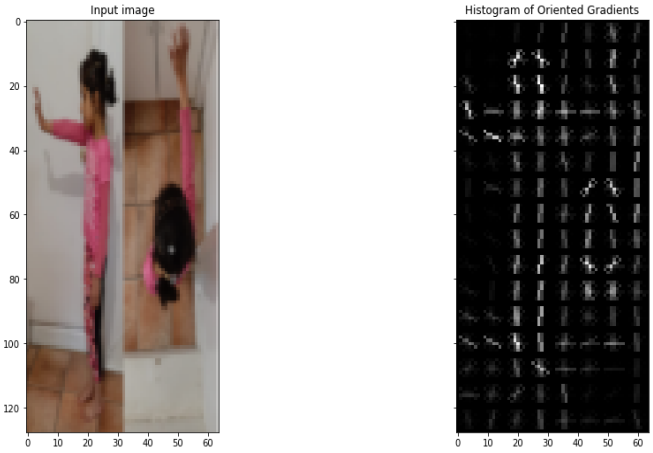

Fig. 1.2. Image representation using HOG

Representing skeleton joint information in an image format and utilizing it for human action detection is the most reliable and computationally powerful approach. Processing real images or videos for action detection requires a lot of computation resources [**15**]. There has been tremendous research effort to improve prediction accuracy in detecting human action with the help of skeleton joint information. CNN exploits the spatial relationship between pixels when they are arranged in matrix representation [**13**]. Shift invariance property possessed by CNN helps in detecting features residing in any part of the image. Encoding spatial and temporal information of skeleton frames in an image is proven to be the best representation for a deep neural network to understand human action [**13, 14**]. Detecting human action when the position of the camera and the position of the skeleton keeps changing is a challenging task [**16**]. We need to train the CNN model with many training data so that it can understand all variations in the coordinate values of a skeleton. Encoding spatial and temporal information of skeleton frames in an image is not sufficient. Hence, we need to consider encoding the distance between joints for the skeleton sequence transformation process. Depth cameras provide a limited, varying number of joints [**17**], and therefore it has become challenging to come with a more useful representation of skeleton information. We propose a method to encode the difference between 3D coordinates values in an image and train a denseNet[**18**] for better prediction accuracy. Existing practice insists on adapting interpolation between frames as the approach to fill the picture when we do not have enough frames [**13**]. CNN can learn the spatial relationship between the input features effectively. We have decided to use RGB representation so that we will be able to encode both spatial

and temporal information of a given skeleton sequence. There is a need to bring in temporal dependency of frames of the current encoded image on the skeleton action sequence's previous frames, and the same is achieved using the frame-shifting method.

1.2. Problem statement

As per ADI, there are 9.9 million new cases of dementia each year worldwide, implying one new case every 3.2 seconds. Alzheimer's patients have a deterioration of their brain connections due to negative emotions like anxiety and frustration. Negative emotions symptoms should be diagnosed in the early stages to avoid the damages caused to cognitive abilities [**19**]. Authors in [**19**] and [**20**] proposed methods to reduce the impact of negative emotions on memory and cognitive abilities by allowing Alzheimer's patients to experience a virtual environment. The virtual environment is designed in such a way that the environment helps to reduce the impact of negative emotions and helps the participants to improve memory functions. The author in [**20**] shows a method for finding negative feelings with the help of facial expression and ECG signals read from the brain. Interactive virtual environments are more effective in treating problems related to negative emotions [**21**], and existing solutions fail to provide an interactive virtual environment for treating Alzheimer's patients. We propose a method with an interactive framework for Alzheimer's patients to help them reduce their negative emotions by interacting with the virtual animal world.

1.3. Research objectives and our contribution

One interest in detecting hand gestures and body gestures relies on the interaction of Alzheimer patients into a virtual world. Virtual environments have proved their efficiency in reducing negative emotions such as anxiety. Authors in [**19**] have developed a virtual train in which Alzheimer's patients can travel and relax, which has a positive impact on decreasing negative emotions and increasing cognitive capabilities. In our proposed project, we aim to allow patients to interact with virtual animals using gesture detection mechanism. This kind of therapy would provide relaxation, motivation, and positive feelings to the patients. The quality of interaction depends on the precision of gesture recognition, which is addressed in the present thesis. In this thesis, we try to answer below research queries :

- What is an alternative to the existing solution and practical approach for treating Alzheimer's patients?
- How can an interactive based VR solution help a healthcare system to address patients suffering from negative emotions?
- How to turn an existing VR based solution into a more active and interactive based solution?

- How to utilize human action and gesture recognition technology to create an interactive-based solution for treating Alzheimer's patients?
- What is the best image representation approach for implementing real-time human action and gesture recognition?
- How to use interpolation between frames and frame-shifting effectively during the image pre-processing step?
- What is the impact of better image representation on the action and gesture prediction accuracy?

We try to answer the above research queries by our contribution, as explained below :

- We captured data from Intel Realsense camera and Leap motion camera to conduct experiments and test the effectiveness of our proposed approach in real-time application.
- Using the VR application developed by Yan AI, we examined the impact of interactive VR environment on people's negative emotions.
- We evaluated the effectiveness of different image representation approaches on human action and hand gesture prediction accuracy.
- Evaluate our proposed approaches on the MSR 3D action data set [**22**].
- I was running experiments on two sophisticated deep neural architectures for finding a more efficient deep neural architecture.
- Finally, we captured the experiment results, graphs, and discussed the results.

1.4. Thesis organization

The thesis is organized in five chapters, including the introduction chapter.

- Chapter 2 provides an insight into the literature review of the recent research work in the domain of human action recognition. I discussed the most advanced research effort in the field of human action recognition and limitations.
- Chapter 3 introduces a method for image representation using the skeleton data sequence. In this chapter, I address the problem of a limited number of joint information provided by depth cameras and using frame-shifting and interpolation between frames methods for effective image representation.
- Chapter 4 explains one more way of image representation using skeleton data and Leap motion data. The limitations of the process described in chapter 3 are addressed in chapter 4. We have done extensive experiments on Leap motion data since hand gestures are more effective in developing an interactive VR application. But, we have

also captured results after running tests on Intel Realsense data and the MSR data set [**22**].

- In chapter 5, we conclude the thesis with a discussion on experimental results, and appendixes have graphs and deep neural architecture details.

Chapter 2

Related Work

Recognizing human action from a video sequence depends on various factors, including the background of video frames, facial expression, and the rate at which position of body changes. Processing a video sequence to extract all the required information is tedious since it involves a lot of image processing. An efficient method of information extraction requires removing unwanted background noise and balancing or ignoring varying light effects in different video frames. Skeleton joint information was extensively used for predicting human action and posture detection. Kinect[**23**] of Microsoft, provided a skeleton tracking facility for a long time, and it was adopted in most of the research practice. Kinect[**23**] provides only twenty skeleton joints information; Intel Realsense camera[**24**] provides precise skeleton joints information with the assistance of third-party SDK. Nuitrack is one of the most reliable SDK in the market, with which it is easy for a Unity developer to build a skeleton tracking application. Depending on the system's hardware abilities, frame rate changes. It is effortless to develop a hardware-independent software module to capture skeleton frames in real-time with the Unity platform's help. Intel Realsense camera[**24**] instead can capture twenty-four joints 3D coordinate values. Leap Motion hardware is a dedicated camera for detecting hand joints position along with joints rotation.

There have been efforts to convert 3D coordinate values into RGB image representation for deep neural network training. The transformation step of skeleton information to RGB representation is a significant data pre-processing stage. Encoded RGB image should include extensive temporal and spatial information of skeleton frames in a sequence. Recognizing human actions using skeleton information is challenging when Alzheimer patients perform actions. There is no research work on utilizing human gesture recognition to help Alzheimer's treatment. The dataset we prepared for our experiments includes mainly hand-gestures because they are more relevant for any patient to perform. Implementing a framework that can work in real-time is challenging since most of the dataset actions look similar and have

Fig. 2.1. Camera view variations.

slight variations. The residual neural network[**25**] helps us in deriving the best prediction model to challenge the complex dataset.

2.1. Challenges involved in processing real-time video for human action recognition and prediction

There are two categories in action classification problem[**26**]: action recognition and action prediction. Predicting action requires an algorithm to watch the first frames of the video and output the accurate and most probable action class. We use action prediction algorithms[**27**] in critical scenarios wherein the situation demands the application not to wait for the complete action. Action prediction requires the beginning frames to be discriminative and right to not have redundant frames[**26**]. On the other side, action recognition algorithms[**28**] have the luxury to use all the video frames before predicting the class. A labeled action contains different versions, and different versions of an action depend on the speed at which an individual performs that action and camera view (as shown in figure 2.1). Background noise and camera movements can also create different versions of an act.

Deep neural networks help researchers exploit the neural network's ability to extract discriminative features from the training data and predicting the action label using a single framework. These discriminative features play an essential role in making an efficient prediction of unseen video frames. Deep neural networks like residual neural networks[**25**] demand massive training data. The need for more training data is one of the limitations since it is not a practical approach to collecting such enormous data for training. As opposed to the method used by authors in [**29**], authors from [**28**] uses direct video frames wherein [**29**] uses videos after segmenting human subjects. In our proposed research work, we are intended to implement gesture recognition and not gesture prediction. Hence, in our work, we used all the frames of an action video performed by a participant.

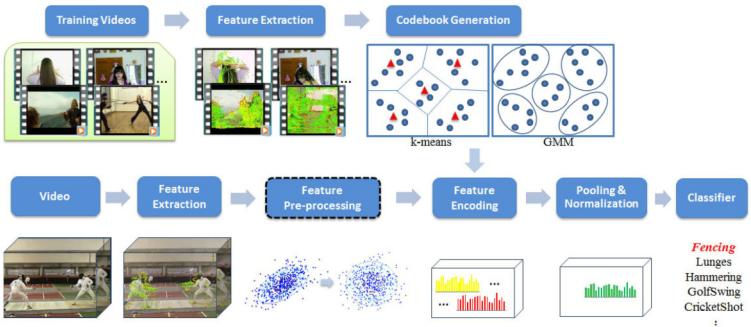

Fig. 2.2. Bag of visual words model. Original image was shown in [**33**].

2.2. Bag of visual words and fusion methods for action recognition

The BoVW model constructs a global representation of an image or video using the local features, and it is processed in five different stages, as shown in figure 2.2. Feature extraction involves collecting various features which are representing an image. Because of background noise and light effects, pictures belong to the same class generate different kinds of features. The visual pattern of a part of an image can be represented using HOG[**30**], HOF [**31**], and MBH [**32**] descriptor. HOG [**30**] captures the orientation and magnitude of gradients in any picture.

Feature extraction:

Local features are robust to background noise and helpful for image representation and video representation. Before describing the local feature, we need to detect the local region with the help of a local region detector like 3D-Harris [**34**] and 3D-Hessian [**35**]. The area detected by detectors is represented using feature descriptor methods such as HOG [**30**]. It is possible to represent a local region using multiple descriptors, and each descriptor caries different aspects of visual image patterns. iDTs [**36**] are proved to be the best local features because of their performance.

Feature pre-processing:

Local feature descriptors are high dimensional, and most of the features are highly correlated. For efficient prediction accuracy, we need to choose independent and uncorrelated features. PCA [**37**] is responsible for capturing principal components, which are linearly independent features. The whitening process is followed after PCA [**37**] to maintain the same variance across all features. Feature pre-processing is not a mandatory step, but action recognition accuracy improves if we follow the pre-processing stage.

Codebook generation:

Features mapped to the lower dimension can be grouped by dividing the feature space into groups; also, its center represents the group, and the k-means cluster algorithm is the most

used algorithm for generating codebooks. We can also generate codebooks using generative models like GMM [**37**], where different distribution is created for each feature.

Encoding methods:

The objective of an encoding method is to calculate a code given pre-processed, multi-dimensional local descriptors, and a codebook. Voting based encoding method [**38**], reconstruction based encoding method [**39**], and super vector-based encoding method [**40**] are the different available options for the encoding process. Hard assignment and soft assignment are the two rules followed while applying to encode. Hard assignment enables us to assign a single codeword to the descriptor, and soft assignment links one descriptor with many codewords. Soft encoding methods consider codeword uncertainty and reduce information loss during encoding.

Pooling and normalization methods:

The global representation of the video is extracted by applying pooling on codewords of all the local descriptors. Normalization helps global representation to be invariant to the number of local descriptors. Sum pooling and max pooling are the two varieties of pooling methods, and authors in [**41**] showed theoretically that sparse features work better with max pooling. The author in [**33**] demonstrates the impact of a different combination of methods explained in BoVW pipeline and below are the conclusions :

(1) iDTs [**36**] with more descriptors are informative compared to any other.
(2) Data pre-processing is an important step in the BoVW pipeline.
(3) Super vector encoding methods are effective compared to other encoding methods.
(4) Sum-pooling, along with power l2-normalization, is the best choice.
(5) All the steps of BoVW are greatly contributing to the final recognition rate.

Video-based human action recognition solution demands heavy processing and computer resources. And hence, in our proposed work, we are not using a video-based solution.

2.3. Learning realistic human actions from movies

Authors in [**42**] propose a method for automatic video annotation of human actions from realistic videos. The automatic annotation of video clips helps us to address the need for more data for training. The intra-class variation is a common problem associated with static image classification and video classification. The method in [**42**] explains generalized and extended spatial pyramids for feature representation to address the issue of intra-class variations. Script-based automatic annotation of a video has some known issues 1) scripts come without time information 2) need to align scripts with video 3) the action in the script does not always align with activity in the video clip and need a very sophisticated mechanism for action retrieval from the text. The author in [**42**] uses subtitles and script information to align script with video, and these are the steps followed(as shown in figure 2.3):

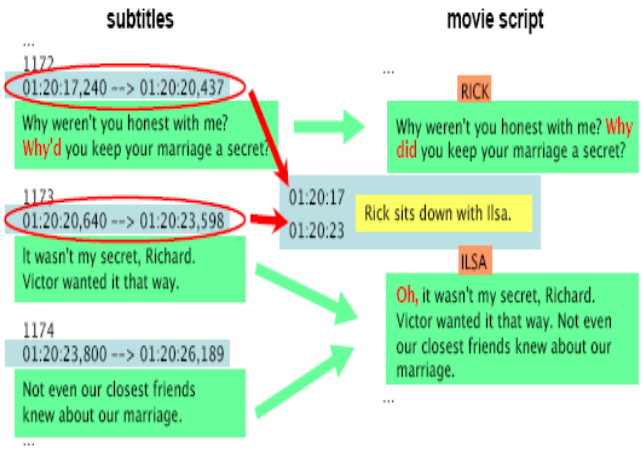

Fig. 2.3. Alignment of actions in scripts and video. This figure was originally shown in [**42**].

(1) Apply line indentation on the script.
(2) Matching words in subtitles and words in the script using dynamic programming.
(3) Extract time information from subtitles and transform to script for generating time interval of scene information.
(4) Assign a score to indicate a possible misalignment due to mismatch between script and subtitles.

Below are the steps followed to classify script description to an action label in [**42**]:

(1) Each scene description is represented as a sparse vector in a high-dimensional feature space.
(2) Remove the features which are supported by less than three training documents.
(3) Classify feature vector to a label using a regularized perceptron network.

Results show that proposed method in [**42**] outperforms by providing precision-recall value for all eight actions as [prec 0.95 / rec. 0.91] which is the best compared to [prec 0.55 / rec 0.88] using regular expression matching classification.

2.4. Learning spatiotemporal features with 3D convolutional networks

The author in [**28**] believes that preserving temporal and spatial information while performing convolution helps create ideal video descriptors. Strong video descriptors possess generic, compact, efficient, and simple to implement characteristics. The method explained by [**28**] proves that even a simple linear classifier can do an appropriate action recognition

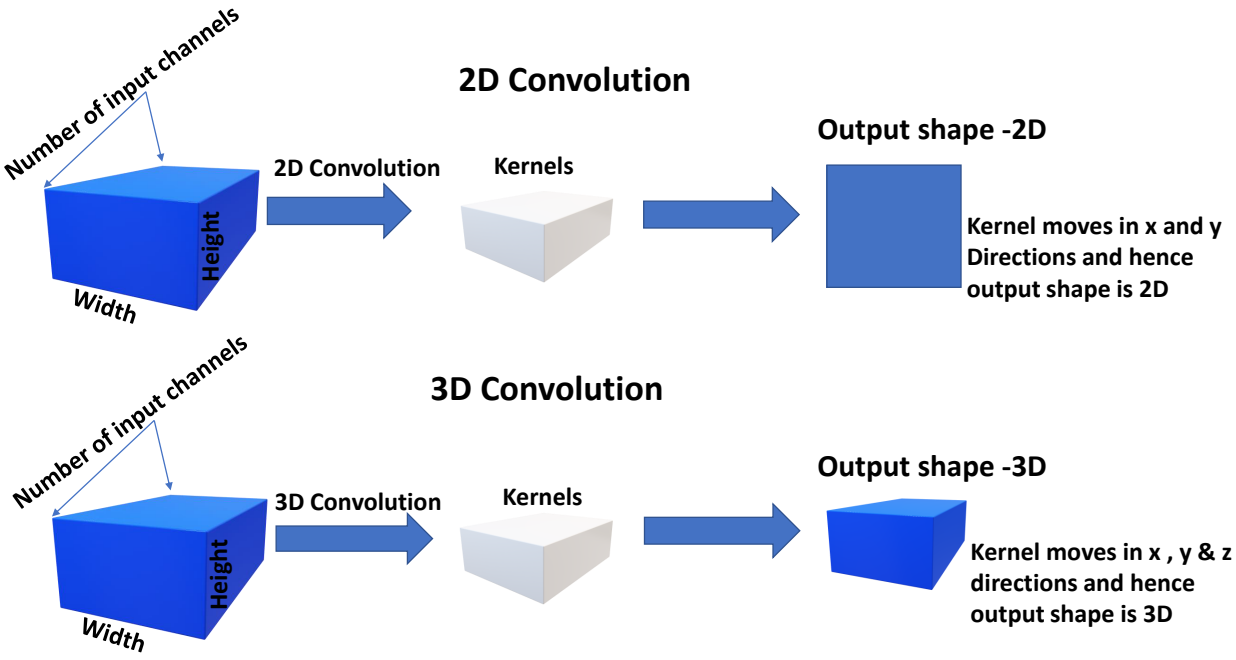

Fig. 2.4. 2D and 3D Convolution.

task with efficient video descriptors. Hence the author in [**28**] more concentrated on extracting prominent features form a given video. As shown in figure 2.4, 2D convolutional preserves the spatial relationship of the pixels. With the help of 2D convolution, it is possible to extract the relationship between different components of an image. Combining 3D convolution with 3D pooling can help the convolution network to propagate temporal information across all the layers, as demonstrated in [**28**]. Experiment results provided in [**28**] confirm that 3D ConvNet with all the convolution layers having the same kernel temporal depth along with 3 x 3 x 3 spatial receptive field is the best architecture. C3D network learns the subject's appearance in the first few frames and starts learning motion features in subsequent frames. C3D features are high-level semantic features, and iDT [**36**] are handcrafted features. The author in [**28**] has achieved 90.4% accuracy on the UCF101 dataset[**43**] after combining iDT[**36**] and C3D features.

2.5. Recognize Human Activities from Partially Observed Videos

The authors in [**27**] has attempted to provide a generic solution to predict human action from a video having an unobserved sub sequence anywhere in the video. Action prediction is challenging when the missing video frames are not at the end and appear in the middle of the video frame sequence, as shown in figure 2.5. The method in [**27**] presents a probabilistic

formulation approach for human action prediction. Sparse coding helps determine the likelihood of a particular type of activity belongs to a specific class observed in the given video. The proposed method in [**27**] divides the test activity into small segments and calculates the posterior of each segment. Combining the posterior at each segment defines the global posterior of the test activity. The training video segments with fixed length and fixed duration construct sparse coding bases. The author in [**27**] also proposes an extended procedure to construct sparse coding bases using a mixture of training video segments of different duration and different lengths. Generally, we use the mean feature vector of all the training videos to approximate test feature vectors. However, [**27**] recommends to approximate test feature vectors using sparse coding. One of the advantages of the proposed method in [**27**] is that we do not need to maintain the temporal alignment of any pair of videos, and it also addresses the problems like 1) a limited number of training videos; 2) outliers in the training video and 3) intra-class variations. The author in [**27**] has conducted experiments in three phases 1) degenerate case 2) special case and general case. Results show that we can achieve the state-of-the-art performance with the help of sparse coding way of feature approximation.

2.6. A Discriminative Model with Multiple Temporal Scales for Action Prediction

Early prediction of the label for a given video frames sequence demands the initial segments of the video to be discriminative and not to be redundant frames. The method explained in [**44**] attempting to exploit these discriminative segments to classify videos. Local templates [**44**] capture all the details of the current segment in the sequence. The global template [**44**] captures the evolution of the action at different temporal lengths, from the start of the frame sequence until the present time, as shown in figure 2.5. Below are the steps followed in [**44**] for action representation as shown in figure 2.6:

(1) Extract interest points [**45**] and trajectories [**46**] from a video.

(2) Apply clustering algorithms to create bag-of-visual-words.

(3) Create a histogram of all visual words for all the partial videos.

Local template(blue solid rectangles in figure 2.7) and global templates(purple and red dashed rectangles in figure 2.7) are captured by proposed algorithm in [**44**].Label consistency captures global context information which helps in improving prediction accuracy. The proposed algorithm in [**44**] demands initial segments of the video to be discriminative, and it is one of the drawbacks of [**44**]. Experiments are conducted on UT-Interaction dataset [**47**] and BIT-Interaction dataset [**48**]. The authors in [**44**] shows 78.33% recognition accuracy when only 50% frames of the testing videos are observed. This result is better than the results of [**42**]. The method in [**44**] also proposes a empirical risk minimization formulation for action prediction problems and the formulation is unique.

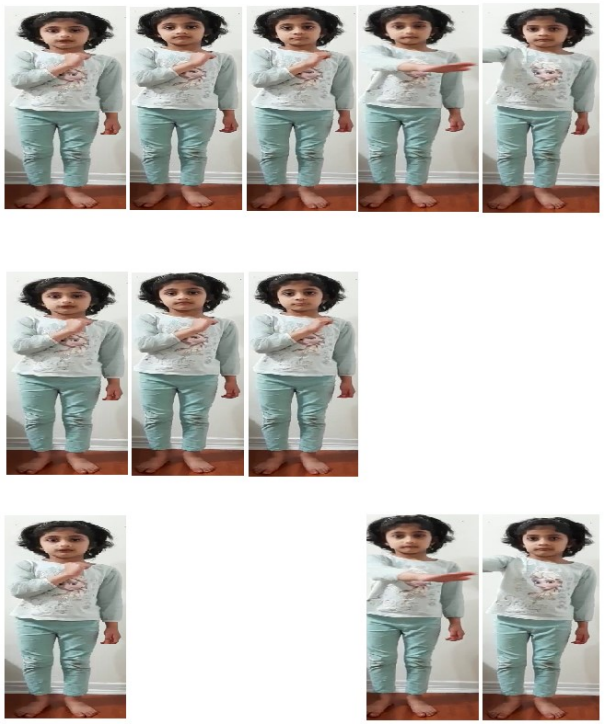

Fig. 2.5. Full video, unknown sub sequence at the end, unknown sub sequence at the middle.

2.7. What are they doing? : Collective Activity Classification Using Spatio-Temporal Relationship Among People

Some human-computer interactive applications need to consider multiple objects in a scene to understand the entire situation in the scene, and the same applies to the action classification of videos. The authors in [**49**] proposes a method to recognize action in a video based on the collective behavior of all the characters and not considering individual human action independently. The complete system discussed in [**49**] is shown in figure 2.8 and explained in 2.7.1.

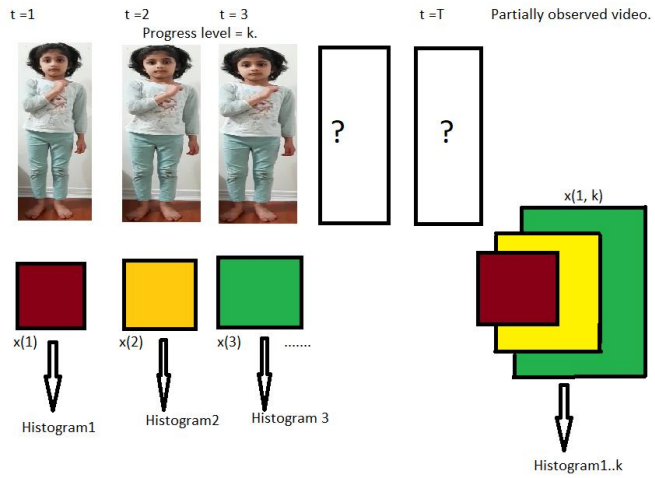

Fig. 2.6. Action representation for action prediction model. This figure created from [**44**]

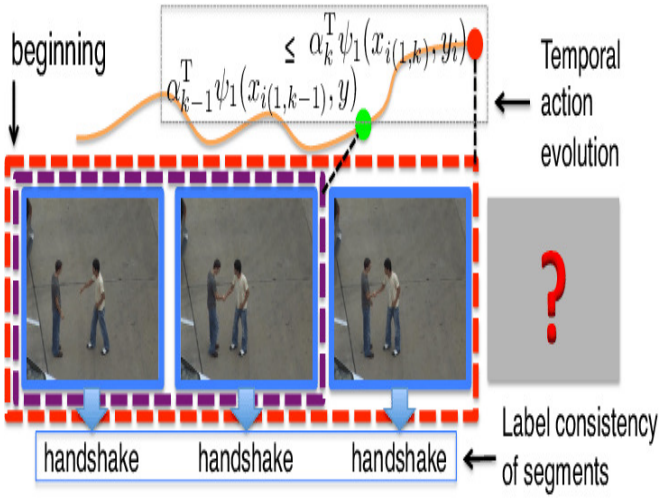

Fig. 2.7. Temporal action evolution over time and the label consistency of segments. This figure was originally shown in [**44**]

2.7.1. System overview

Deformable Part-Based Detector[**49**] is employed for human detection in all the video frames, which in turn uses HOG[**30**]. During the testing phase, if the cost of deformation to make a candidate resemble the learned model is less than a threshold, then the candidate will be labeled as human. For pose estimation, the authors in [**49**] uses HOG[**30**] descriptor and SVM classifier. To get better tracking results, the authors in [**49**] recommends using the 3D position of the target along with camera parameters. Structure From Motion(SFM)

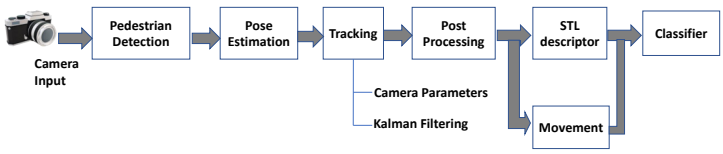

Fig. 2.8. Action representation for group of people model. This figure created from [**49**]

method has its limitations: i) 3D reconstruction is noisy ii) in the real-time background is not static iii) SFM process is computationally expensive and hence difficult to implement for real-time needs. To overcome the issues of SFM, the author in [**49**] makes two assumptions: i) all people are standing on flat surface ii) camera tilt is approximately zero. Based on these assumptions, camera parameters and the target's position are extracted using the generative model approach. First-order Extended Kalman Filter(EKF) is applied to the noisy set of bounding boxes given by the HOG [**30**] detector to estimate the target position. EKF uses one more extra parameter, the height of the individual character, provided by the modified SFM process for efficient estimation. The estimated location of the targets helps to extract the Spatio-Temporal descriptor, which is robust to the viewpoint. The temporal evolution of activities captured in a histogram with the anchor being at the center of the histogram is called the STL descriptor. Different STL descriptors are calculated by keeping each person in the scene at the center. The collection of extracted STL descriptors carries a massive amount of temporal information of each person's activity related to each other person's activity in the scene. SVM classifiers can classify the STL descriptors to different class labels.

Experiment Results:

Experiments are conducted on a dedicated dataset [**50**], which is captured in unconstrained real-world conditions. Each frame in the videos is of the 640x480 size, and videos were recorded using handheld cameras. The author in [**49**] labeled training samples manually at every tenth frame. Experiment results are shown to be performing well with the dataset.

2.8. Machine Learning for Real Time Poses Classification Using Kinect Skeleton Data

In [**16**], the authors explain the method of estimating human pose using skeleton data given by the Kinect sensor [**23**]. Instead of using the temporal sequence of 3D coordinates,

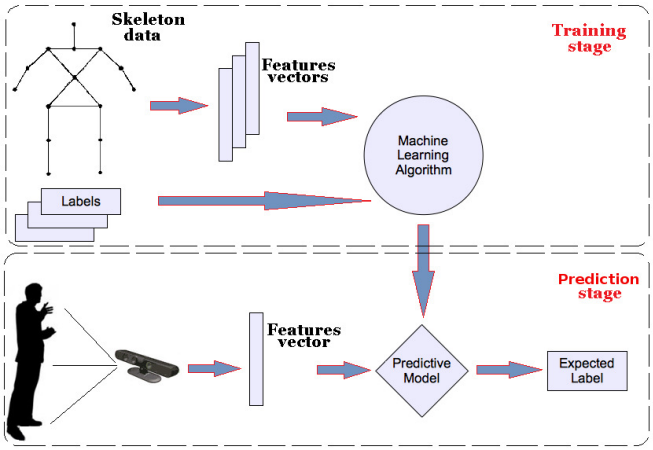

Fig. 2.9. System overview. This figure was originally shown in [**16**]

which are relative to the camera position, the authors in [**16**] uses coordinate values relative to the other joints. Relative coordinate values remove the prediction accuracy dependency on the size and location of the subject. Firstly, three-dimensional skeleton coordinates transformed into a one-dimensional feature vector as shown in figure 2.9. The feature vector is the input to a machine learning algorithm with or without pre-processing. The proposed algorithm in [**16**] is assessed on a vocabulary containing eighteen poses and employing machine learning algorithms: SVM, Artificial Neural Network, K-Nearest Neighbours, and Bayes classifier. SVM outperforms by giving 100% prediction accuracy on the dataset used in the experiments conducted by [**16**]. The method in [**16**] works excellent with a predefined set of actions and fails to consider the temporal dependency of frames in predicting human action.

2.9. Recognizing human action from Skeleton moment

RGB representation can encode rich spatiotemporal information of any skeleton sequence. That way, machine learning models like CNN and its variants can efficiently extract image features and classify the image into an available class. Transforming skeleton joint coordinate values into RGB image space is explained by the authors in [**12**]. Skelton parts are divided into five significant parts P1, P2, P3, P4, and P5. Each part will have 3D coordinate values of the set of skeleton joints (P1, P2: two arms, P4, P5: two legs, P3: trunk) at time 'T'. And hence, J'_{11} is nothing but the 3D coordinate of the first joint that comes under the P1 part at $T = 1$. The transformation module will convert skeleton joints into an image by arranging pixels in the order of P1->P2->P3->P4->P5, as shown in figure 2.10. The proposed method

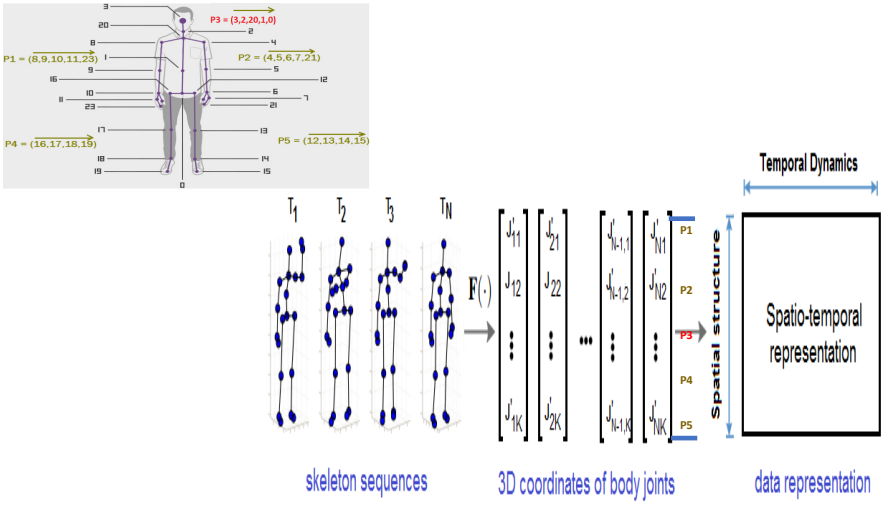

Fig. 2.10. System overview. This figure was originally shown in [**12**]

of data transformation in [**12**] helped the author to achieve the best prediction accuracy with the help of three different variants of ResNet models. The author in [**12**] achieved state-of-the-art performance on the MSR dataset[**22**]. This paper [**12**] fails to explain how to effectively incorporate spatio-temporal information of skeleton motion when the skeleton motion has a higher number of frames. The image representation method explained by [**12**] is based on global representation and ignores to capture the relationship between adjacent frames. In our proposed solution, we encode global information along with local relationships between frames. The method explained in [**12**] archives 99.47% test accuracy with the MSR dataset [**22**].

2.10. Skeleton Based Action Recognition Using Translation-Scale Invariant Image Mapping and Multi-Scale Deep CNN

Transforming from skeleton information to image representation is a crucial step in the process of human action classification using skeleton data. A sophisticated and promising method of transformation is discussed and demonstrated in [**11**] with the help of results. Very few parameters, which play a vital role in the transformation process, are extracted from each video sequence instead of referring the whole data. The proposed method in [**11**] helps in preserving scale invariance and translation invariance of the training data. In [**11**], the authors claim that the complete process of transformation becomes dataset independent.

Translation-scale invariant image mapping: Below are the steps followed to extract scale-invariant features by [**11**]:

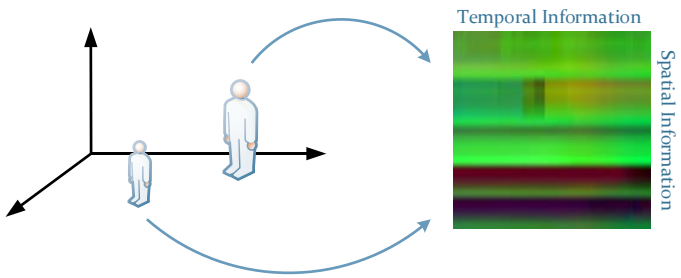

Fig. 2.11. Translation-scale invariant image mapping. This figure was originally shown in [**11**]

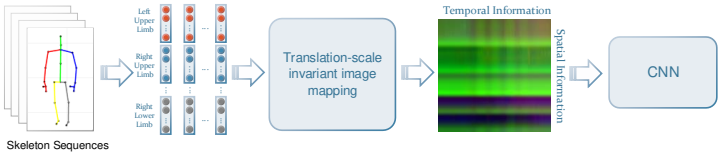

Fig. 2.12. System overview. This figure was originally shown in [**11**]

(1) The human skeleton in each frame is divided into five parts: two arms, two legs, and a trunk.
(2) 3D coordinate values are normalized, considering only the current video sequence and not the complete training data to achieve a translation-scale invariant image mapping, as shown in figure 2.11.
(3) Normalized 3D skeleton coordinates are arranged into RGB channels, as shown in figure 2.12.

The proposed method in [**11**] exhibits state-of-the-art performance on NTU RGB-D [**51**], UTD-MHAD [**52**], MSRC-12 Kinect Gesture dataset [**53**] and G3D [**54**] dataset.

2.11. Skepxels : Spatio-temporal Image Representation of Human Skeleton Joints for Action Recognition

We need an effective method to represent skeleton 3D coordinates. That way, deep learning models can exploit the correlation between local pixels, which in turn helps us to have better prediction accuracy. The paper [**13**] discusses a method that can help us arrange skeleton information as skepxels [**13**] in the horizontal and vertical direction. Skepxels [**13**] in horizontal direction carry the frames in skeleton data. Similarly, the rate of change of joints position are captured in velocity frames, as shown in figure 2.14. Spatial information of the skeleton frame sequence was captured in the vertical direction of the transformed image by rearranging pixels of a skeleton frame at a time 't' as shown in figure 2.13.

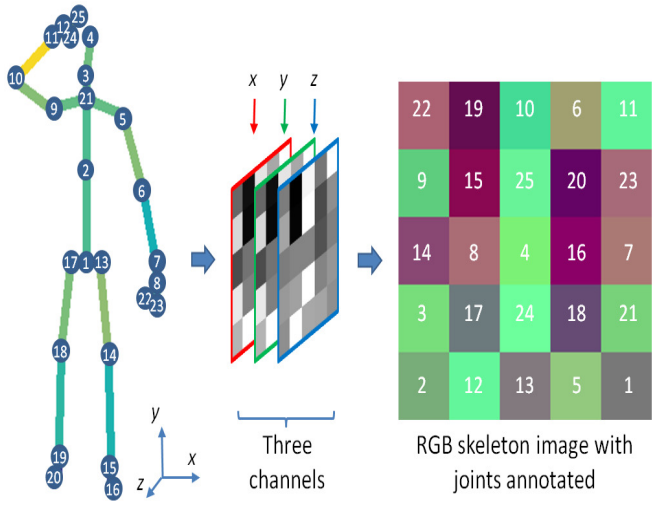

Fig. 2.13. Generating skepxel from skeleton joints. This figure was originally shown in [**13**]

The proposed way of arranging skeleton frames in [**13**] will increase the prediction accuracy as each image is carrying rich spatiotemporal information (as shown in the figure 2.15). The author in [**13**] also explains how using image interpolation between frames can create a full image even though we have a smaller number of frames in a skeleton motion. If the number of frames exceeding the number of frames required to make an image, then the rest of the frames are moved to the next image and labeled with the same class name. NTU 3D action data [**51**] was used to evaluate the proposed method in [**13**], and the transformation process generates millions of pictures after the transformation step. For data-augmentation, the author in [**13**] has recommended adding Gaussian noise samples to each frame and double the training data size. With all the proposed strategies in [**13**], the Resnet model [**25**] can achieve a state-of-the-art performance. Experiments conducted by [**13**] show that the proposed method in [**13**] can achieve 85.4% average test accuracy on the NUCLA dataset [**55**] and 97.2% average test accuracy on UTD-MHAD dataset [**52**]. If the position of the camera and skeleton changes, then model prediction accuracy will change to a great extent. With the proposed method in [**13**], to ensure better test accuracy, we need to take more data with all possible positioning of the skeleton. When the skeleton frame sequence is long, dividing sequence into multiple images as suggested in [**13**] will ignore temporal dependency information of the current image on the previous frames in the series. The method explained in [**13**] archives 97.2% test accuracy with the UTD-MHAD dataset [**52**].

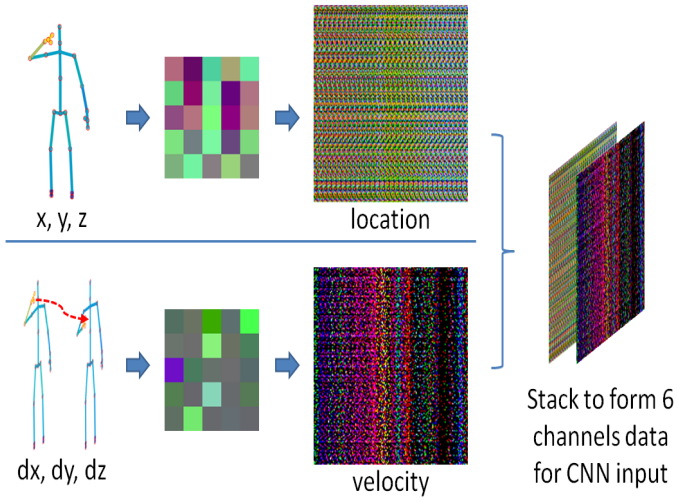

Fig. 2.14. Position and velocity frames. This figure was originally shown in [**13**]

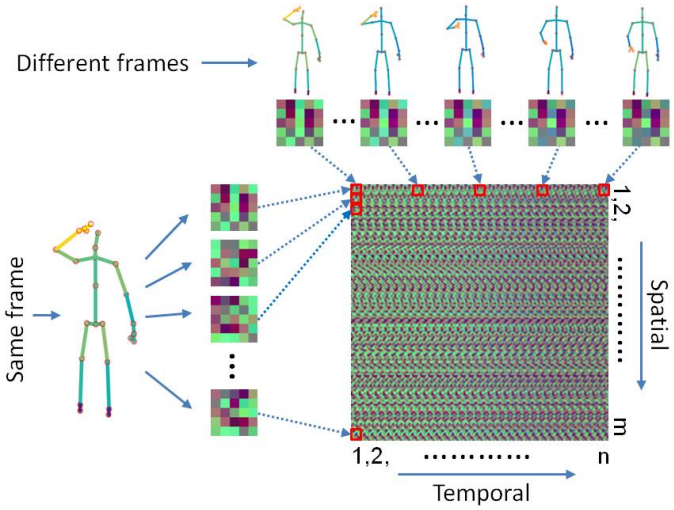

Fig. 2.15. Spatial and temporal arrangement of skepxels. This figure was originally shown in [**13**]

2.12. A Deep Learning Approach for Real-Time 3D Human Action Recognition from Skeletal Data

Surrounding distractions, along with viewpoint changes, make the HAR task challenging. Depth sensor based-HAR is the best solution to overcome the above challenges. The authors in [**14**] talks about the effective method for transforming the temporal sequence of human skeleton movements into RGB representation. The technique proposed in [**14**] does not

become dependent on the length of the skeleton sequence and efficiently can extract global features. Below are the steps followed in [**14**] for skeleton sequence to RGB transformation :

(1) Encode human poses into RGB images.
(2) Enhance local textures of the RGB images by applying AHE [**56**].
(3) Before feeding images into D-CNN, a smoothing filter is applied to reduce the input noise effect.
(4) Discriminative features can be learned by feeding images to DenseNet [**18**].

The proposed method in [**14**] is built based upon the below hypotheses:

(1) Human actions can be represented using skeleton movements.
(2) Spatio-temporal evolution of skeletons can be transformed into RGB images.
(3) Skeleton joints information is more efficient and less complicated compared to depth images for training D-CNN.
(4) DenseNet [**18**] is one of the most effective CNN architectures for image classification.

ESPMF is an enhanced version of SPMF [**57**], which in turn includes encoded PFs and MFs. The PFs encode skeleton joints position information, and MFs encode the rate of skeleton joints changes concerning all other skeleton joints. The complete process followed by [**14**] is shown in figure 2.16. The proposed method in [**14**] achieves the state-of-the-art performance on MSR action data [**22**] and NTU RGB+D dataset [**51**]. ESPMF representation shows a 1.42% increased prediction accuracy when compared to SPMF[**57**] representation(99.10% test accuracy with MSR dataset[**22**]). The method explained by the authors in [**14**] is considered to be the best image representation method. We need a mechanism to start and terminate the sequence of skeleton frames when the prediction has to be made continuously in real-time. An interactive framework for treating Alzheimer's patients demands a continuous prediction of human action or hand gestures. To facilitate continuous prediction in a healthcare application, we consider dividing the sequence of skeleton frames into blocks with an equal number of frames. Each block of frames contains the same amount of frames, and this number is the average number of frames in the training data. We understand that not all video sequences are of the equal number of frames, and to address this issue, we propose using frame-shifting and interpolation between frames method. We propose a problem-specific solution to recognize human action in real-time using Intel Realsense camera in Chapter 3. We experimented with the method explained in Chapter 3 to know the possibility and impact of considering relative joint values instead of 3D coordinate values. In Chapter 4, we try to address the issues of manually choosing a responsible list of joints for generating relative skeleton joint values. Finally, we show the effective image representation method and its impact on prediction accuracy with results in Chapter 4.

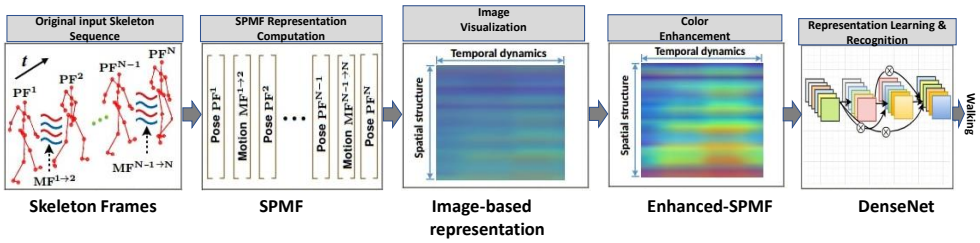

Fig. 2.16. Skeleton data is encoded in RGB image. RGB images are fed to D-CNN for action classification. This figure was originally shown in [**14**]

Chapter 3

Image representation for Intel realsense skeleton sequence data

Authors in [**13**, p. 13] realized after experimenting with large values of λ and small values of λ that, relative positions of the joints is more important for CNNs than absolute positions. For experiments, authors in [**13**] consider using the joint's absolute positions but using the best set of skeleton joints arrangement. Arriving at the best set of skeleton joints arrangement is an NP-hard problem, and hence authors in [**13**, p. 13] use a realistic strategy to find a suitable set of arrangments of skeleton joints. In this chapter, we want to show the impact of a combined approach on prediction accuracy, where we try to use relative joint values and the best arrangement of relative skeleton joint values for generating RGB images from the skeleton sequence. We adapt frame-shifting and interpolation between frames to fill RGB images when encountering a variable number of skeleton frames. Chapter 3 is organized as follows: Section 3.1 introduces the hardware setup we are using for our experiments; Section 3.2.1 explains SkepxelRel construction using frame-shifting and interpolation between frames. Section 3.2.1 also introduces us to image pre-processing techniques and data augmentation methods we followed; A brief introduction to Residual network [**25**] is given in section 3.2.2. A detailed discussion of the experimental results conducted on Intel Realsense data is provided in section 3.2.3.

3.1. Intel Realsense Camera and Leapmotion Camera

Intel Realsense camera has five necessary hardware modules, as shown in figure 3.1 :

(1) Right Imager
(2) Left Imager
(3) IR Projector
(4) RGB Module
(5) PCB and components

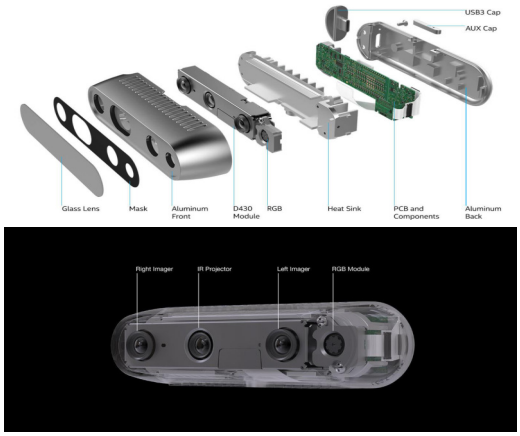

Fig. 3.1. Intel Realsense camera. This figure was originally shown in [**24**]

Right Imager and Left Imager:

Right and Left Imagers are the camera sensors with identical configurations. Sensors are named "left" and "right" from the perspective of the camera looking outward. The active pixels of the sensors is 1280 x 800.

Infrared projector:

Infrared projectors project a static infrared pattern on the low texture scene to increase the texture. The power delivery and laser safety circuits are on the stereo depth module [**24**].

RGB Module:

Data from the left and right imagers along with the data from RGB sensors is used for creating color point cloud and for 3D model reconstruction [**58**]. The active pixels of the color sensor is 1920 x 1080.

PCB: PCB contains Intel Vision Processor, D4, capable of sending high-quality depth and RGB images over USB channel.

Nuitarck SDK:

Nuitrack SDK [**59**], along with Intel realsense camera, can generate skeleton joints in real-time. Nuitrack is a skeleton tracking middle-ware, and its a multi-language, cross-platform framework. Nuitrack helps to capture twenty-four human body joints in real-time.

Leap motion camera:

Leap motion hardware contains two cameras and three infrared LEDs. The camera viewing range is 2.6 feet and struggles to track hand movements beyond 80 cm. LED light propagation through space limits the camera viewing range. The Leap motion controller reads images through the USB controller, applies resolution adjustments, and sends it to

Leap motion service running on a computer through a USB connection. Leap motion service does not generate a depth map; instead makes 3D reconstruction of what the camera sees. The tracking layer of the Leap motion service infers the details of fingers. After applying filters to tracking data, the transport layer sends the data to native or web-based client applications. Leap motion camera can capture the position and rotations of twenty-six significant hand joints for each hand.

3.2. Skeleton sequence to RGB transformation of Intel Realsense data

Using pixels of training images, CNN tries to build minor and significant features of images. CNN models are translation invariant, and they can recognize trained characteristics anywhere in the pictures. In this paper, we demonstrate how to generate images from skeleton joints information by creating building blocks of a picture called SkepxelsRel. We do not use skeleton joint coordinates; instead, we use a list of 3D coordinate values generated after taking the difference between two joints. We group a set of pair of joints which contribute more in deciding the class of action. This combination of a couple of skeleton joints is also a hyper-parameter during training a ResNet model [**25**]. Velocity frames generated uses the speed at which the difference of considered skeleton joints changes. As demonstrated in [**16**], when we take the reference point as other joints, the prediction accuracy does not depend on the position of the camera and skeleton. We explain the approach as followed.

3.2.1. Constructing SkepxelsRel

SkepxelsRel have a similar structure of Skepxels explained in [**13**]. SkepxelsRel tensors encode differences of coordinate values along the third dimension, as shown in figure 3.2. We follow the same strategy explained in [**13**] in choosing the best pixels arrangements for filling spatial information of a skeleton frame at time 't'(The algorithm to generate the required number of pixel arrangements is shown in algorithm 1). As shown in figure 3.3, RGB channels encode spatial-temporal information of skeleton joint differences and create an image. Velocity frames are constructed using a similar method, as explained in [**13**]. However, we use SkepxelsRel values to calculate the rate at which the differences between reference joints change, as shown in figure 3.4. With our proposed method, we can generate any number of joints required for image representation using equation 3.2.2. As shown in figure 3.7, we created thirty-six relative data points, which play an essential role in deciding human actions from Intel Realsense data. As shown in figure 3.4, velocity frames are generated by taking the difference of adjacent frames in a sequence and dividing them by frame rate (equation 3.2.3). In our experiments, we considered the frame rate as thirty frames/second for Intel Realsense data (data is available here : Intel Realsense Data).

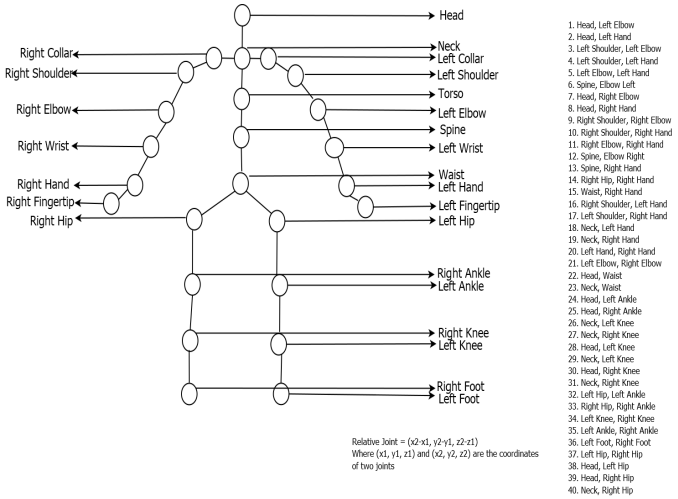

Fig. 3.2. Skeleton structure and generating relative joints

If we are encoding 30 skeleton frames in an RGB image, then we generate 30 random pixel arrangements so that the final image has equal height and width. The Random pixel arrangements of a skeleton frame at time 't' are stacked vertically, and subsequent skeleton frames in the skeleton action sequence are stacked horizontally (as shown in figure 2.15). The equation to calculate the image dimension is given in equation 3.2.1. We choose $K = L$ in equation 3.2.1 to achieve equal width and height in the final RGB representation. Hence, with $J = 36$, final RGB image dimension is $(180 \times 180 \times 3)$ with $K = L = 30$ and the final image dimension is $(300 \times 300 \times 3)$ with $K = L = 50$.
$K =$ The number of skeleton frames to encode in an RGB image.
$L =$ The number of random arrangements.
$J =$ The number of relative joints for every frame

$$\text{Image Dimension} = (K * \sqrt{J}) \times (L * \sqrt{J}) \times 3 \tag{3.2.1}$$

When the number of frames required to form the image is more than needed, we recommend using frames shifting (as shown in the figure 3.6) instead of moving the remaining skeleton frames (shown in the 3.5) to the next image. Frame shifting way of image construction helps in real-time prediction wherein each image encodes only the original frames of the skeleton motion without adding interpolated frames in between. The frame-shifting method also helps us to encode temporal dependency information of previous frames in the current image. If the available number of frames for constructing an image is less than the required, then we can go with interpolation between frames approach. Figure 3.6 demonstrates the steps involved in adjusting the frames to accommodate all the available skeleton frames.

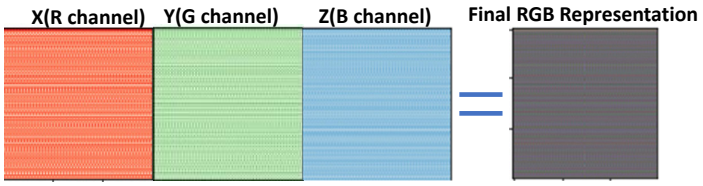

Fig. 3.3. RGB Channels generated with (x, y, z) coordinates of skeleton sequence.

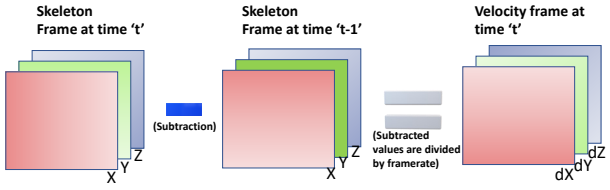

Fig. 3.4. Velocity frames calculated by subtracting frames.

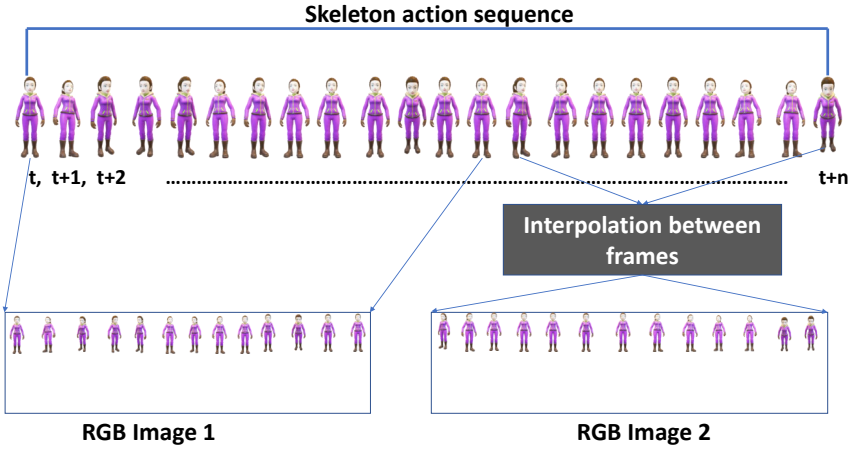

Fig. 3.5. Interpolation between frames applied.

$$\text{Relative Joint12}(x', y', z') = \text{Reference Joint1}(x, y, z) - \text{Reference Joint2}(x, y, z) \quad (3.2.2)$$

$$\text{Velocity of relative joint at time t} = \frac{\text{Joint}(x', y', z')\text{at time t} - \text{Joint}(x', y', z')\text{at time t-1}}{\text{frame rate}} \quad (3.2.3)$$

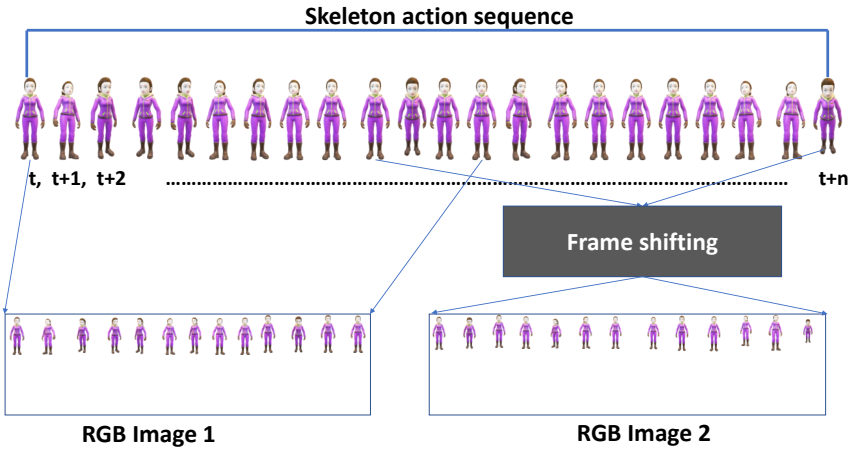

Fig. 3.6. Frames are shifted to right and temporal dependency of frames is not ignored.

```
class Joint:
    HipCenter = 6
    Spine = 3
    ShoulderCenter = 2
    Head = 19
    ShoulderLeft = 1
    ElbowLeft = 8
    WristLeft = 10
    HandLeft = 12
    ShoulderRight = 0
    ElbowRight = 7
    WristRight = 9
    HandRight = 11
    HipLeft = 5
    KneeLeft = 14
    AnkleLeft = 16
    FootLeft = 18
    HipRight = 4
    KneeRight = 13
    AnkleRight = 15
    FootRight = 17
```

```
LIST_OF_JOINTS = [(Joint.ShoulderLeft, Joint.ElbowLeft), (Joint.ElbowLeft, Joint.HandLeft),
        (Joint.ShoulderLeft, Joint.HandLeft),(Joint.Spine, Joint.ElbowLeft),
        (Joint.Spine, Joint.HandLeft), (Joint.HipCenter, Joint.HandLeft),
        (Joint.Head, Joint.ElbowLeft), (Joint.Head, Joint.HandLeft),
        (Joint.HipLeft, Joint.HandLeft),

        (Joint.ShoulderRight, Joint.ElbowRight), (Joint.ElbowRight, Joint.HandRight),
        (Joint.ShoulderRight, Joint.HandRight),(Joint.Spine, Joint.ElbowRight),
        (Joint.Spine, Joint.HandRight), (Joint.HipCenter, Joint.HandRight),
        (Joint.Head, Joint.ElbowRight), (Joint.Head, Joint.HandRight),
        (Joint.HipRight, Joint.HandRight),

        (Joint.HandLeft, Joint.HandRight), (Joint.ElbowLeft, Joint.ElbowRight),
        (Joint.HandLeft, Joint.ShoulderRight),(Joint.HandRight, Joint.ShoulderLeft),
        (Joint.HandLeft, Joint.ShoulderCenter), (Joint.HandRight, Joint.ShoulderCenter),

        (Joint.Head, Joint.HipCenter), (Joint.ShoulderCenter, Joint.HipCenter),
        (Joint.Head, Joint.AnkleLeft), (Joint.Head, Joint.AnkleRight),
        (Joint.ShoulderCenter, Joint.KneeLeft), (Joint.ShoulderCenter, Joint.KneeRight),

        (Joint.KneeLeft, Joint.KneeRight), (Joint.AnkleLeft, Joint.AnkleRight),
        (Joint.FootLeft, Joint.FootRight), (Joint.HipLeft, Joint.HipRight),
        (Joint.HipLeft, Joint.AnkleLeft), (Joint.HipRight, Joint.AnkleRight)]
```

Fig. 3.7. List of thirty-six relative joints generated for every frame from Intel Realsense data.

Data Pre-Processing:

To conduct experiments in real-time, we generated skeleton sequences for primary actions using the Intel Realsense camera [**24**] with the help of Nuitrack SDK [**59**]. Each skeleton frame is normalized by making the center of the frame as the center of the coordinate system (0, 0, 0) [**15**].

Data Augmentation:

To increase the training data size, we sampled from a gaussian distribution with mean zero and a standard deviation of 0.02 and added those noise samples to actual skeleton frames. We have also applied random cropping, horizontal flip, and vertical flip data augmentation strategies (shown in figure 3.9).

1	29	7	19	4	18
14	16	24	28	11	23
21	2	8	15	32	27
25	20	12	26	3	36
30	5	17	31	33	22
10	13	34	6	35	9

Fig. 3.8. Example of a random arrangement of a frame's thirty-six relative joints in a 6×6 matrix.

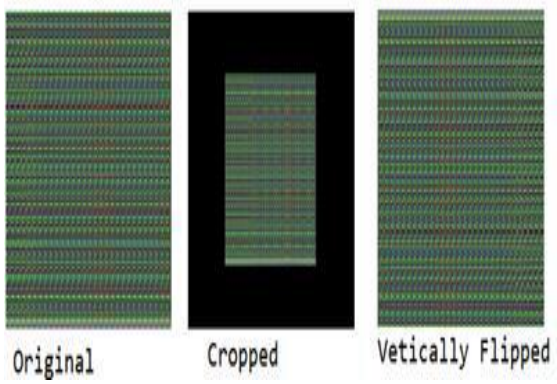

Fig. 3.9. Data Augmentation.

3.2.2. Residual Network

Deep neural networks with many more layers help a deep neural network model to have a higher number of parameters, and by that degree of freedom of that model increases. With the increased complexity of the model, the ability to learn new sophisticated features will also increase. When a neural network has the freedom to choose parameters without regularization, then the chances of finding a global minima is less, and the model ends up finding local minima. Hence, it is useful to include regularization methods to regulate the most in-depth neural networks and try avoiding model over-fitting behavior. Recent experiments and research show that even after having regularization methods in a deep neural network, it is inevitable to have an over-fitted model. To avoid such behavior without losing the benefits of deep neural networks, researchers have come up with new architecture called Residual Network [**25**]. Our experiment results show that the residual network model outperforms in real-time prediction.

Algorithm 1 Algorithm to generate random arrangements

▷ Because of the values we have chosen for lambda_thr, we always get desired number of random arrangements
mat_list ← *Generate list of random arrangements* ▷ as is shown in figure 3.8
lamda_thr ← 2500 ▷ We found this value by calculating radial distance for every pair of matrices
result_mat_list ← []
final_list ← []
num_of_joints ← 36 ▷ Number of joints in each matrix
thr_nu_frames ← 30 ▷ We need 30 arrangements
for `i in range(length(mat_list))` **do**
    *mat*1 ← *mat_list*[*i*]
    for `k in range(length(mat_list))` **do**
        if i == k **then**
            continue
        *mat*2 ← *mat_list*[*k*]
        total_sum = 0.0
        for `j in range(num_joints)` **do**
            x, y ← *get position of joint 'j' in mat1*
            x_ta, *y_ta* ← *get position of joint 'j' in mat2*
            total_sum ← $total\_sum + max(abs(x - x\_ta), abs(y - y\_ta))$
    if *total_sum* > *lamda_thr* **then**
        result_mat_list.append(*mat*1)
        if *length(result_mat_list) == thr_nu_frames* **then**
            return (result_mat_list).tolist() ▷ we get required arrangements
final_list ← (*result_mat_list*).*tolist*() ▷ we don't get required arrangements
return final_list

One of the significant problems associated with deep neural networks is vanishing gradients problem, wherein gradients at the last layer will not be able to propagate back to initial layers. Hence, learning will be prolonged and improper. Shortcut connections provided in Residual blocks (as shown in figure 3.10) make a model to learn identity mapping of the input very quickly. Also, the shortcut connection helps to carry gradients back to initial layers without vanishing gradients problem. Hence, we have adopted the Residual network [**25**] in our experiments to learn significant features of the skeleton sequence.

3.2.3. Experiments

We used a setup having Intel Realsense camera [**24**] for capturing skeleton frames on the Unity platform. 20-layer ResNet model [**25**] was trained for six basic human actions, including No Movements, Wave Hands, Soothing, Come, Go, and Clap. These actions are parts of movements that an Alzheimer patient could show for interacting with an animal such as a horse or a dog. We observed that the trained model was able to predict all the

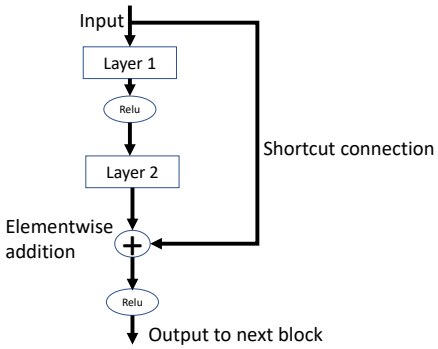

Fig. 3.10. Residual Network.

actions with 100% accuracy in real-time (accuracy graph is shown in figure 3.11). As stated already, we have used a set of joints that are responsible for predicting pre-decided actions. Other factors also behave as hyper-parameters like frame rate (number of frames per second captured by the Unity platform, and it is hardware dependent) and size of the transformed image. As the data available is less with only six human actions, we had to augment data to satisfy ResNet [**25**] requirements. We tried with different frame rates:30, 20, 10 and 30 outperformed compared to other frame-rates. Since we are using differences of coordinates, changing camera position, and skeleton position did not have any impact on prediction accuracy. We tried with different image sizes 180x180, 300x300, and 180x180 outperformed compared to other image dimensions. As explained in equation 3.2.1, the number of skeleton frames (K) to be encoded in RGB representation decides the image size. If we use a higher value for K, actions with lesser frames will be extended using interpolation between frames. Interpolation between frames results in action looks slower compared to the action without the interpolation process. If we use less value for K, chances are high that we lose important details of the action in RGB representation. Hence we decided to use moderate values for K, which are near to the average number of frames calculated for the training dataset. The average number of frames calculated for Intel Realsense training data is 36, and $K = 30$ shows better results.

Our proposed method does not need the number of joints to be equivalent to the required number of joints to form a SkepxelsRel. We can generate the required number of data points by taking differences among responsible joints. High-frequency spatial components are essential features in an image for useful prediction accuracy [**60**]. The SkepxelRel method generates high frequencies in the final RGB representation. The SkepxelRel representation also work well with DenseNet architecture [**18**], as shown in figure 3.12. DenseNet architecture is the better option for image classification since it avoids the overfitting behavior of deep neural networks [**18**]. The random arrangement of pixels [**13**] is not a more reliable way of skeleton joints representation since random arrangement ignores spatial and temporal

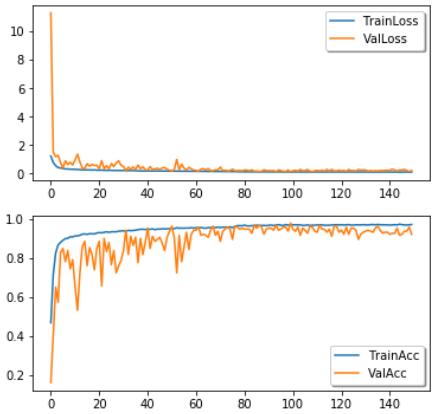

Fig. 3.11. Intel Realsense data, ResNet, SkepxelRel, performance graph.

relationships among pixels representation [**11**]. Authors in [**11, 12**] consider arranging skeleton joints in a particular way to retain the spatial relationship between human joints when they are transformed into RGB form. Hence the loss function is unstable with DenseNet [**18**] architecture as shown in figure 3.12. We run experiments on MSR dataset[**22**] using Skepxels representation, and the accuracy results are not up to the state-of-the-art performance. Figure 3.13 and 3.14 shows accuracy graphs of ResNet[**25**](Resnet-20 architecture is given in A.1) and DenseNet[**18**] (DenseNet architecture is given in B.1)models. We need an even more efficient representation of skeleton frame sequence, which can capture spatial-temporal information of skeleton movements, preserve high-frequency spatial data, and retain pixel spatial relationships. We will be able to derive an efficient model for prediction with the help of DenseNet [**18**] even though we have fewer data like the MSR dataset [**22**]. All experiment details conducted with SkepxelRel representation is listed in the table 3.1. In the next chapter, we discuss the issues with SkepxelRel representation and an alternative approach to replace the SkepxelRel method for an efficient skeleton sequence to RGB transformation.

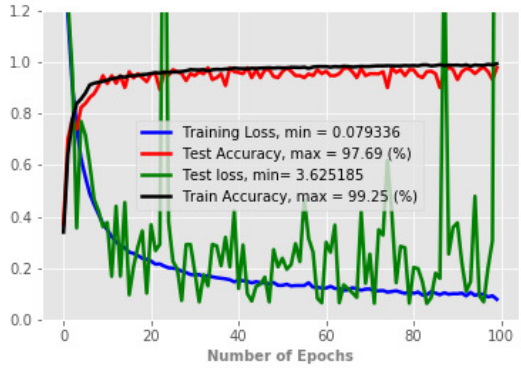

Fig. 3.12. Intel Realsense data, DenseNet, SkepxelRel, performance graph.

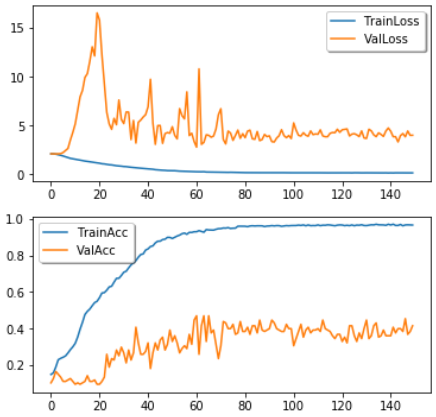

Fig. 3.13. MSR data, ResNet, SkepxelRel , performance graphs.

Intel Realsense DataSet and Image Dimension	Best Test Accuracy	Best Test Loss
(180x180), 30 frames, ResNet-20	95.351%	12.681%
(300x300), 50 frames, ResNet-20	96.4%	8.928%
(180x180), 30 frames, DenseNet	100%	7.843%
(300x300), 50 frames, DenseNet	100%	8.008%

Tableau 3.1. SkepxelRel experiment details along with results

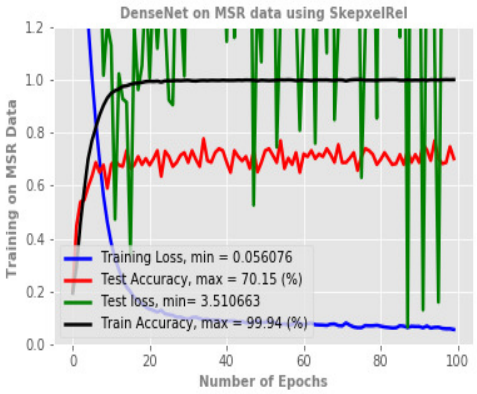

Fig. 3.14. MSR data, DenseNet, SkepxelRel , performance graphs.

Chapter 4

Enhanced SkepxelRel representation for skeleton joints sequence

Skeleton sequence to RGB transformation process needs to capture spatial-temporal information of skeleton joints movements effectively. SkepxelRel provides the flexibility to choose the number of relative joint values for RGB transformation and ignores the spatial relationship between joints during motion detection. The set of actual joints to be considered to generate relative joints is a hyper-parameter. It is not a practical solution to choose a set of real joints that can contribute more to the action detection process. The process of selecting the set of actual joints is highly dependent on the human action type. Chapter 4 explains a generic solution for transforming the skeleton sequence into RGB image representation after considering the complexity of generating the best list of relative joints and their arrangements.

The total number of possible combinations that can be generated is given by the equation 4.1.1. Equation 4.1.1 is a complex equation, and it is not feasible to generate all such combination of joints for a real-time prediction application. The amount of time required to evaluate all possible combinations is not practical, and to avoid the unnecessary computational effort, we can choose $M = \frac{N!}{(N-2)!\times 2!}$ in the equation 4.1.1. Hence with N=20, we will have 190 relative joints in one frame(as per the equation 4.1.4). All the relative joints can be stacked in columns to generate RGB representation of the skeleton sequence. In section 4.1, we discuss implementation details of enhanced SkepxelRel using the Leap motion data(data is available here : Leap motion data) that we captured. Experimental results, the impact of enhanced SkepxelRel representation on Leap motion data classification and a brief introduction to densely connected convolution network architecture are provided in section 4.1.1.

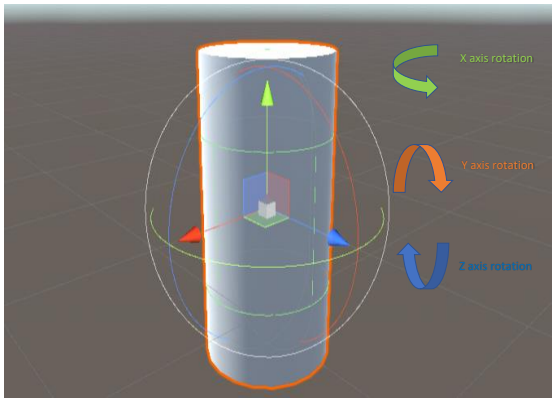

Fig. 4.1. Rotation in X, Y, Z axes.

4.1. Enhanced SkepxelRel formation and experiments

.

The Extended SkepexelRel approach is motivated by the method explained in [**14**] wherein the proposed way by [**14**] produces efficient RGB representation of skeleton sequence. We need a few more modifications on top of the method explained by [**14**] for real-time prediction applications. Alzheimer's treatment demands a continuous prediction of the patient's hand gestures. Hand gesturing made by the patient changes the virtual environment, and these changes will have a direct impact on the patient's medical status. Data for training is captured from a Leap Motion camera with the Unity platform for basic hand gestures including 1) "Left Hand Call" 2) "Right Hand Call" 3) "Left Hand Go" 4) "Right Hand Go" 5) "Version2 Left Hand Call" 6) "Version2 Right Hand Call" 7) "Left Hand Wave" 8) "Right Hand Wave" 9) "Left Hand Still" and 10) "Right Hand Still"(as shown in figure 4.2). There are two versions of the "Call" gesture since both have different rotation values. Data from the Leap Motion camera provides twenty-six joints information. Each frame data will have the position and rotation values of twenty-six joints. The position represents the joint's actual position in the Unity scene, and rotation values are the rotation of a joint relative to the world coordinate system wherein rotation in all three axes is generated for each joint, as shown in figure 4.1. With Leap Motion and Unity setup, it is effortless to capture all the details of hand gestures made by Alzheimer's. Rotation values play a very significant role in the hand gesture recognition process. Most of the actions listed above have little variation in position values but will have differences in rotations. Unity updates rotation in all three axes with the help of Euler angles [**61**]. We need a mechanism to transform both position and rotation values into RGB representation, similar to the method explained by [**14**].

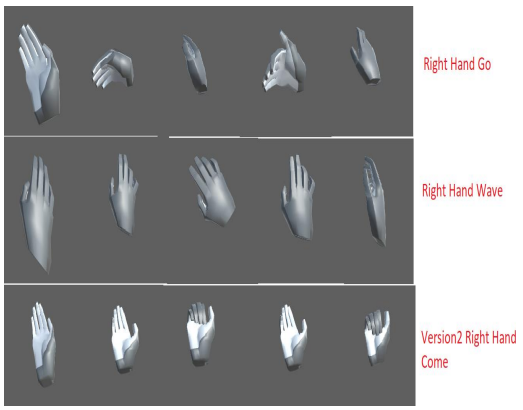

Fig. 4.2. Examples of Hand Gestures captured with the help of Leap Motion camera

Some hand gestures like "Clap Hands" and "Hold an Object" demand encoding gestures made by both hands, and hence we need to encode one hundred and four data points for each frame. To encode the speed at which the position and rotation of hand joints change, we have to consider differences between consecutive frames. SkepxelRel is formed by taking the differences of actual joints, and for generating extended SkepxelRel, we can use the Euclidean distance of 3D coordinates(4.1.2).

$$\text{Possible number of combinations} = \frac{(\frac{N!}{(N-2)!\times 2!})!}{((\frac{N!}{(N-2)!\times 2!}) - M)! \times M!} \tag{4.1.1}$$

N = Number of actual joints detected per hand by the hardware = 26
M = Total number of relative joints in each frame

$$\text{Euclidean Distance} = ||\overrightarrow{J_iJ_j}|| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{4.1.2}$$

Wherein (x_i, y_i, z_i) and (x_j, y_j, z_j) are 3D coordinates of joints ($\forall i, j \leq \frac{N!}{(N-2)!\times 2!}$ and $i \neq j$). A skeleton representation of a human body includes orientation and magnitude of vectors connecting each joint, as shown in figure 4.3. The magnitude can be measured using the Euclidean distance of joints in Euclidean space. Similarly, the orientation of the vector connecting two joints is given by the direction of a unit vector, as in equation 4.1.3.

$$\text{Orientation of } J_iJ_j = \overrightarrow{u} = \frac{J_j - J_i}{||\overrightarrow{J_iJ_j}||} \tag{4.1.3}$$

Euclidean distance transforms 3D coordinates into one-dimensional values, and hence these values must be mapped back to 3D. Remapping to 3D is necessary because RGB representation demands 3D data. We apply JET mapping(shown in figure 4.4) to transform one-dimensional Euclidean distance values to 3D values, and we adapt this mapping method

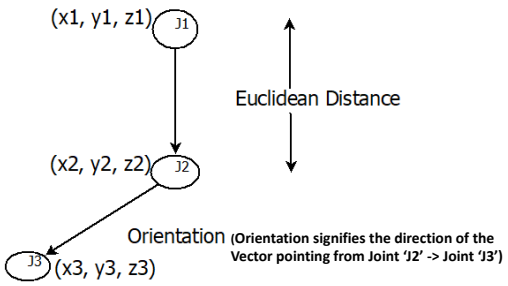

Fig. 4.3. Magnitude and Orientation of vectors connecting skeleton joints

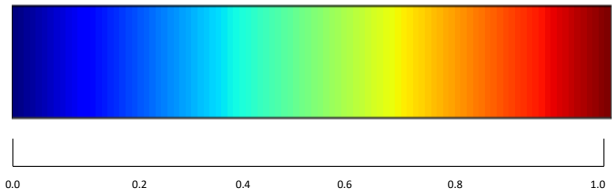

Fig. 4.4. JET Color Map

from [**14**]. Euclidean distance values are normalized to the range [0, 1], and normalized values are mapped to RGB values using JET mapping. JET mapping maps values in the range [0, 1] to RGB color values, starting from blue and ends at red. Euclidean distance and orientation calculation are applied for both position and rotation values of both hands data. We use maximum and minimum values of the training data of position and rotation values for normalizing Euclidean distance values. Hence training images generation is data-dependent, and we need to evaluate minimum and maximum values before generating images for training. Skeleton motion is encoded by calculating the Euclidean distance and orientation values considering two adjacent frames. Equations for encoding motion information are same as equation 4.1.2 and 4.1.3 but the i^{th} joint should be from the frame at time 't' and j^{th} joint should be considered from the frame at time 't+1'. The number of data points in each motion frame is more than that of Euclidean data points from pose frames; hence we enhance the number of Euclidean data points in pose frames by appending the nearest data points. The equation for calculating the number of data points in each motion frame and number of data points to append to pose frame [**14**] are given by equation 4.1.5 and 4.1.6 respectively(data points arrangement is shown in fig 4.5). Since the pose frame is extended by replicating the nearest pixel value, it has no bad impact on final image quality and prediction accuracy.

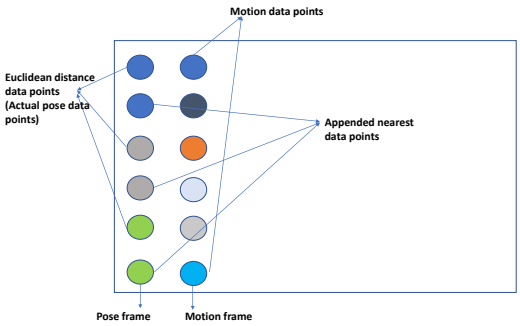

Fig. 4.5. Data point arrangement of pose and motion frames

$$\text{Number of data points in each pose frame} = \frac{N!}{(N-2)! \times 2!} = 325 \tag{4.1.4}$$

$$\text{Number of data points in each Motion frame} = N^2 = 676 \tag{4.1.5}$$

$$\text{Number of data points to append to pose frame} = N^2 - \frac{N!}{(N-2)! \times 2!} = 351 \tag{4.1.6}$$

Pose, and motion frames are generated for both position and rotation values, and we arrange them in a specific order for better RGB representation, as shown in figure 4.6. The number of actual frames from the camera to be accommodated in an image is decided based on the average number of frames evaluated from the training dataset. For 'K' number of actual frames($K = 40$) from the camera, we need to generate 'K' number of pose frames and 'K-1' number of motion frames. Hence, with 'K' number of frames, we create an image of size: width $= 8 \times ((2 \times K) - 1) = 632$ and height $= N^2 = 676$. Generated images are resized to 40×40 to maintain the same image size across training and testing data. Before resizing, images go through AHE[**56**] to enhance local contrast of the picture by applying histogram equalization to multiple parts of the image. Enhancing the local contrast of the image by Applying AHE[**56**] before resizing the images help us in keeping the color variations. Hence even after image resizing we can see the different color variations in images for every action. Figure 4.7 shows sample images generated using Leap Motion data. Figure 4.8 shows accuracy graph generated using DenseNet [**18**] on Leap motion data.

We have used the same DenseNet architecture proposed by [**14**] along with hyper-parameters such as batch size-32, learning rate-0.0003, Adam optimizer. Data augmentation methods like horizontal and vertical flip, height-shift, and width shift techniques are applied to increase the data size and avoid overfitting. The average number of frames of the Leap motion training data is 36 frames. During real-time testing, we can use interpolate between frames or image resize technique to adjust the image size if the number of frames required to form the RGB representation is not available. From the real-time test results on the experiments we conducted, we see that proposed RGB representation captures and encodes

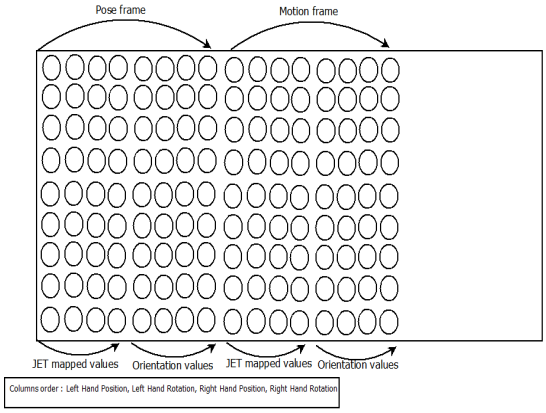

Fig. 4.6. Pose and motion frames arrangement

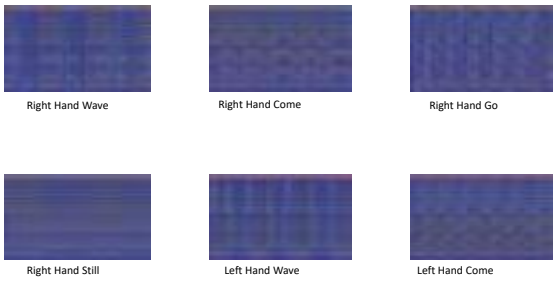

Fig. 4.7. Sample Images generated for Leap Motion data using extended SkepxelRel method.

position and rotation of hand joints efficiently and provides 100% test accuracy. We conducted different experiments by changing a few critical hyper-parameters, and the results are listed in table 4.1.

4.1.1. Densely Connected Convolutional Networks

.

Recent research works demonstrate that providing a short-cut connection from the input of the first layer to the input of the next layer will avoid vanishing gradient problem. Existing work (ResNet [**25**])ignores maximum information flow between previous layers to subsequent layers, and the information flow problem is addressed by DenseNet architecture [**18**]. DenseNet [**18**] allows each layer obtain feature maps from all its preceding layers and pass its feature maps to all its subsequent layers. Instead of summation of feature maps from the previous layer [**25**], DenseNet [**18**] concatenate feature maps for efficient information flow, as shown in figure 4.9. Number of parameters to learn in DenseNet [**18**] is less than the

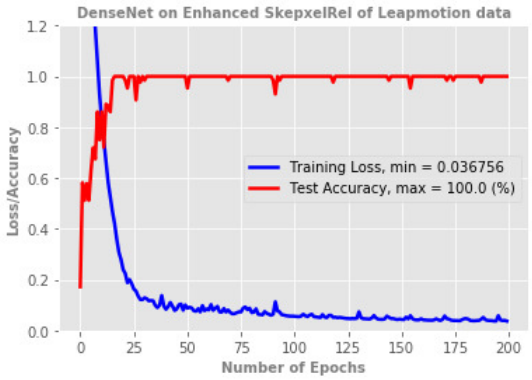

Fig. 4.8. Accuracy Graph, Leap Motion data

DataSet and Image Dimension	Best Test Accuracy	Best Test Loss
Leap Motion data : 30 frames	100%	4.355%
Leap Motion data : 36 frames	100%	3.5%
Leap Motion data : 50 frames	100%	4.085%
Leap Motion data : 80 frames	100%	4.594%
MSR data : 40 frames	100%	10.594%

Tableau 4.1. Extended SkepxelRel experiment details along with results

number of parameters learned by ResNet [**25**]. Every layer in the DenseNet [**18**] have direct access to gradients, and input signal hence helps to deep supervision. The total number of connections in a DenseNet is given by the equation 4.1.7. DenseNet [**18**] possesses very narrow layers and provides few feature ('k') maps for every layer. 'k' is a hyper-parameter in DenseNet is called growth rate of the network. Authors in [**18**] shown that, with a minimal value of $k = 12$, it is possible to achieve state-of-the-art results.

$$\text{Total number of connections} = \frac{L \times (L+1)}{2}. \text{ Where L is total number of layers.} \quad (4.1.7)$$

DenseNet [**18**] architecture can learn unique features for every action from the Leap motion training dataset, as shown in figure 4.10. Trained DenseNet can predict test images in real-time with 100% accuracy. We evaluated trained DenseNet in real-time, and it is possible to predict every two-second action in less than 0.5 seconds. Results shown in the table 4.1

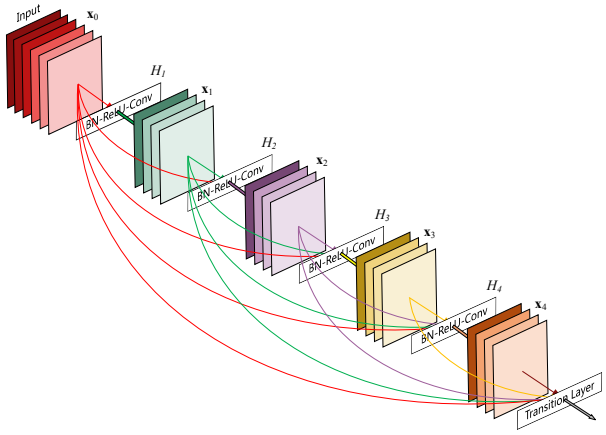

Fig. 4.9. Dense Network, Original figure was shown in [**18**]

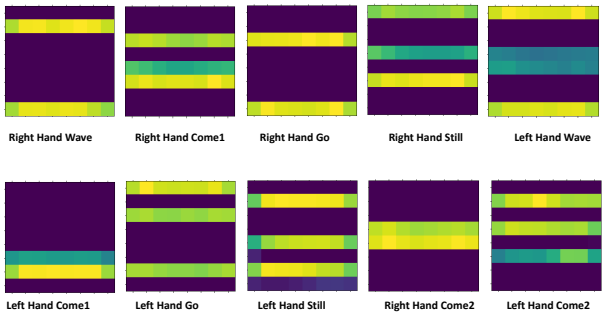

Fig. 4.10. Feature maps learned by DenseNet

proves that even with an increased number of frames encoded in RGB representation, the test error does not change significantly. Leap motion data that we captured for our experiments include actions that can be performed in a short duration. Data were obtained at different frame rates supported by the camera and we used the best frame rate for our experiment purpose(30 frames per second). Hand gestures are performed by ten different subjects, and data from subjects [1, 3, 5, 7, 8, 9, 10] are used for training, and remaining subjects are used for testing. We make sure that testing data has unseen rotations and positions of hand-joints to validate the trained model. We experimented with different image sizes by changing the Leap motion training data's average frame size, and the results are captured in table 4.1(Accuracy graphs are shown in C.1). MSR skeleton sequence data is transformed using extended SkepxelRel and trained using DenseNet [**18**], and we can see the state-of-the-art performance (accuracy graph is shown in D.1).

Chapter 5

Conclusion and Future Work

In this chapter, I conclude my thesis by giving justification to all the research objectives listed in Chapter 1 and giving an insight into future work.

5.1. Conclusion

There are challenges to address when implementing a solution for treating Alzheimer's patients. Existing solutions only provide assisting tools and a virtual environment to help improve cognitive abilities and avoid negative emotions in the participants. Recent research shows that an interactive virtual environment helps a healthcare system treat Alzheimer's effectively, and hence, we have proposed an interactive virtual environment solution for treating Alzheimer's.

We can create even a very sophisticated virtual environment for training purposes, but the environment should help Alzheimer's patients overcome negative emotions and improve cognitive abilities. Research work proves that Animal Assisted Therapy allows Alzheimer's patients to improve their mental status. In this project, we have created a virtual dog and a horse character in the VR environment. Research has proved that the Alzheimer's patients will have reduced agitation, increased physical activity, improved eating, and improved pleasure feeling behavior after a real dog visits into the patient's environment. Yan AI has developed a virtual environment for our experiments, and we have used the same environment for treating Alzheimer's patients.

To create a real-life situation in the virtual environment, we need to allow the participants to interact with pet animals present in the VR. HCI enables multiple ways and provides many interfaces to interact with the VR. We can have a voice-based interface, an action-based interface, or a gesture-based interface to communicate to the VR world. We have proposed using a gesture-based interface since we can easily capture hand joints information using a Leap motion camera and have developed a sophisticated algorithm for gesture prediction. It is easy for Alzheimer's patients to remember and perform simple hand gestures instead of a

need for complex human actions. The healthcare system we proposed can detect basic and straightforward to create hand gestures like wave hands, invite the VR character to come near-patient, instruct the animal character to leave the environment, and idle hands. The trained deep-learning model was tested in the virtual environment created by Yan AI, and it works efficiently. Apart from the basic gestures with which our model is trained, we can train the DenseNet model with new gestures. For doing this, we have to capture training data for new gestures using the Unity application and train the DenseNet with new training data set. We followed the same method to train our DenseNet model progressively by adding support to one gesture.

For implementing a better prediction algorithm, we face challenges like the need for an efficient representation of hand joints sequence, sufficient data for training a deep-learning model, efficient deep-learning architecture, and highly complicated parameters tuning process. We had to experiment with two of the best approaches to see the impact on prediction accuracy. SkepxelRel provides state-of-the-art performance with the Intel Realsence data but fails to perform better with MSR 3D action dataset. This failure is because of the random arrangement of relative joints and manually deciding the best list of joints for generating relative joints. We propose an enhanced SkepxelRel method that can effectively encode both position and rotation values of hand joints in an RGB format. The RGB representation using enhanced SkepxelRel is compact and efficient, and it is evident from the experiment results.

A promising skeleton sequence to the RGB representation method achieves better prediction accuracy. The random arrangement of transformed skeleton joints information does not yield good results. Skeleton joints information after the transformation process needs to be arranged in a specific manner to retain the spatial relationship between pixels. Enhanced SkepxelRel representation helps to keep the spatial and temporal correlation of pixels and helps DenseNet to learn unique features for every action. Enhanced SkepxelRel representation and training using DenseNet is a training-data dependent process. DenseNet architecture determines distinct features with minimal training data, has less impact of overfitting on the training process and avoids vanishing gradient problem. DenseNet architecture possesses a minimum number of parameters compared to other existing deep neural networks like ResNet and provides state-of-the-art performance. Arranging skeleton joints in a different order in a sequence has an impact on test accuracy. We found that adjacent joints to be placed in the same order to retain spatial relationship among them and to yield better test accuracy. Data augmentation should not change the actual data information, and hence we considered adding Gaussian noise with only 0.02 as standard deviation. We found that using the average number of frames calculated for training data as the number of skeleton frames to be encoded in every RGB image is the best choice in our experiments. Encoding very few frames or encoding a very high number fo frames results in bad test accuracy. Our proposed

method shows better results when tested with Leap motion data and MSR action dataset. The pre-processing method for skeleton sequence, SkepxeRel, required for representing the sequence in RGB form is a feasible solution to implement for real-time applications. We tested our proposed method in a virtual environment setup for treating Alzheimer patients, and our proposed method shows state-of-the-art performance in a real-time application. It is evident from the provided graphs and user experience with the interactive system that I addressed all the research objectives considered with the help of the proposed method.

5.2. Future Work

I want to extend my work to satisfy the requirements below, which in turn make the complete healthcare system more user friendly:

- It is possible to make the VR environment more interactive by including both human action detection and human gesture detection. Intel Realsence camera provides human body joints information, and the Leap motion camera provides hand joints information. We can develop an algorithm to combine hand gestures and human action detection to understand better what the participant wants to do with the VR character. With this approach, we will be able to make the VR environment more interactive.
- Our proposed method for image representation has a known limitation, wherein the process of training and testing depends on training data. We need to extract few parameters from the training data before starting the training and testing process. Hence the process of training and testing is training data-dependent, which results in predicting unseen hand gestures with low confidence. I want to extend my work in the future to make the training and testing process independent of training data.

References

[1] Jia Dan Wei. Exploration of human-computer interaction applications in hospitality industry. 2017.

[2] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan. Speech recognition using deep neural networks: A systematic review. *IEEE Access*, 7:19143–19165, 2019.

[3] Maghilnan S and Rajesh M. Sentiment analysis on speaker specific speech data. 02 2018.

[4] Ryuichi Nisimura, Shoko Miyamori, Lisa Kurihara, Hideki Kawahara, and Toshio Irino. Development of web-based voice interface to identify child users based on automatic speech recognition system. volume 6764, pages 607–616, 07 2011.

[5] Ronaldo Parente, Ned Kock, and John Sonsini. An analysis of the implementation and impact of speech-recognition technology in the healthcare sector. *Perspectives in health information management / AHIMA, American Health Information Management Association*, 1:5, 02 2004.

[6] Anneketh Vij and Jyotika Pruthi. An automated psychometric analyzer based on sentiment analysis and emotion recognition for healthcare. *Procedia Computer Science*, 132:1184 – 1191, 2018. International Conference on Computational Intelligence and Data Science.

[7] Shugang Zhang, Zhiqiang Wei, Jie Nie, Lei Huang, Shuang Wang, and Zhen Li. A review on human activity recognition using vision-based method. *Journal of Healthcare Engineering*, 2017:1–31, 07 2017.

[8] Jyoti Kumari, R. Rajesh, and K.M. Pooja. Facial expression recognition: A survey. *Procedia Computer Science*, 58:486 – 491, 2015. Second International Symposium on Computer Vision and the Internet (VisionNet'15).

[9] Yongmian Zhang and Qiang Ji. Facial expression understanding in image sequences using dynamic and active visual information fusion. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1297–1304 vol.2, 2003.

[10] Manas Mandal, Rakesh Pandey, and AB Prasad. Facial expressions of emotions and schizophrenia: A review. *Schizophrenia Bulletin*, 24:399, 06 2013.

[11] Bo Li, Yuchao Dai, Xuelian Cheng, Huahui Chen, Yi Lin, and Mingyi He. Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep cnn. *CoRR*, abs/1704.05645, 2017.

[12] Huy-Hieu Pham, Louahdi Khoudour, Alain Crouzil, Pablo Zegers, and Sergio A. Velastin. Learning and recognizing human action from skeleton movement with deep residual neural networks. *CoRR*, abs/1803.07780, 2018.

[13] Jian Liu, Naveed Akhtar, and Ajmal Mian. Skepxels: Spatio-temporal image representation of human skeleton joints for action recognition. *CoRR*, abs/1711.05941, 2017.

[14] Huy-Hieu Pham, Houssam Salmane, Louahdi Khoudour, Alain Crouzil, Pablo Zegers, and Sergio A. Velastin. A deep learning approach for real-time 3d human action recognition from skeletal data. *CoRR*, abs/1907.03520, 2019.

[15] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *CoRR*, abs/1406.2199, 2014.

[16] Y. Choubik and A. Mahmoudi. Machine learning for real time poses classification using kinect skeleton data. In *2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV)*, pages 307–311, 2016.

[17] Yong Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1110–1118, 2015.

[18] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[19] Hamdi Ben Abdessalem, Alexie Byrns, Marc Cuesta, Valeria Manera, Philippe Robert, Marie-Andrée Bruneau, Sylvie Belleville, and Claude Frasson. Application of virtual travel for alzheimer's disease. pages 52–60, 01 2020.

[20] Somchanok Tivatansakul, Gantaphon Chalumporn, Supadchaya Puangpontip, Yada Kankanokkul, Tiranee Achalakul, and Michiko Ohkura. Healthcare system focusing on emotional aspect using augmented reality: Emotion detection by facial expression. 07 2014.

[21] D. Freeman, Sarah Reeve, A. Robinson, Anke Ehlers, David Clark, Bernhard Spanlang, and Mel Slater. Virtual reality in the assessment, understanding, and treatment of mental health disorders. *Psychological Medicine*, 47:1–8, 03 2017.

[22] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 9–14, 2010.

[23] Ahmad Davudinasab. Kinect sensor. `https://dx.doi.org/10.13140/2.1.1068.5124`, 2014. Accessed: 2020-03-10.

[24] Intel realsense product family d400series. `https://www.intelrealsense.com/wp-content/uploads/2020/06/Intel-RealSense-D400-Series-Datasheet-June-2020.pdf`, 2020. Accessed: 2020-01-20.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[26] Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *CoRR*, abs/1806.11230, 2018.

[27] Y. Cao, D. Barrett, A. Barbu, S. Narayanaswamy, H. Yu, A. Michaux, Y. Lin, S. Dickinson, J. M. Siskind, and S. Wang. Recognize human activities from partially observed videos. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2658–2665, 2013.

[28] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. C3D: generic features for video analysis. *CoRR*, abs/1412.0767, 2014.

[29] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013.

[30] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.

[31] Janez Pers, Vildana Kenk, Matej Kristan, Matej Perše, Klemen Polanec, and Stanislav Kovačič. Histograms of optical flow for efficient representation of body motion. *Pattern Recognition Letters*, 31:1369–1376, 08 2010.

[32] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103, 05 2013.

[33] Xiaojiang Peng, Limin Wang, Xingxing Wang, and Yu Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *CoRR*, abs/1405.4506, 2014.

[34] Ivan Sipiran and Benjamin Bustos. Harris 3d: A robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer*, 27:963–976, 11 2011.

[35] Geert Willems, Tinne Tuytelaars, and Luc Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Computer Vision – ECCV 2008*, pages 650–663, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[36] H. Wang and C. Schmid. Action recognition with improved trajectories. In *2013 IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.

[37] Alaa Tharwat. Principal component analysis (pca) : An overview, 03 2016.

[38] J. Sivic and A. Zisserman. A text retrieval approach to object matching in videos. *ieee*, 2003.

[39] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.

[40] Kai Yu and Tong Zhang. Improved local coordinate coding using local tangents. In *ICML*, pages 1215–1222, 2010.

[41] Xingxing Wang, LiMin Wang, and Yu Qiao. A comparative study of encoding, pooling and normalization methods for action recognition. In Kyoung Mu Lee, Yasuyuki Matsushita, James M. Rehg, and Zhanyi Hu, editors, *Computer Vision – ACCV 2012*, pages 572–585, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[42] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[43] Khurram Soomro, Amir Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, 12 2012.

[44] Yu Kong, Dmitry Kit, and Yun Fu. A discriminative model with multiple temporal scales for action prediction. In *ECCV*, 2014.

[45] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005.

[46] Michalis Raptis and Stefano Soatto. Tracklet descriptors for action modeling and video analysis. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 577–590, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[47] M. S. Ryoo and J. K. Aggarwal. UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html, 2010.

[48] Yu Kong, Yunde Jia, and Yun Fu. Learning human interaction by interactive phrases. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 300–313, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[49] Wongun Choi, K. Shahid, and S. Savarese. What are they doing? : Collective activity classification using spatio-temporal relationship among people. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 1282–1289, 2009.

[50] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. In *CVPR 2011*, pages 3273–3280, 2011.

[51] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3d human activity analysis. *CoRR*, abs/1604.02808, 2016.

[52] C. Chen, R. Jafari, and N. Kehtarnavaz. Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 168–172, 2015.

[53] Simon Fothergill, Helena M. Mentis , Pushmeet Kohli, and Sebastian Nowozin. Instructing people for training gestural interactive systems. In *CHI '12 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1737–1746. ACM, May 2012.

[54] V. Bloom, D. Makris, and V. Argyriou. G3d: A gaming action dataset and real time action recognition evaluation framework. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 7–12, 2012.

[55] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning and recognition. *CoRR*, abs/1405.2941, 2014.

[56] Stephen M. Pizer, E. Philip Amburn, John D. Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart [ter Haar Romeny], John B. Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, 39(3):355 – 368, 1987.

[57] Huy-Hieu Pham, Louahdi Khoudour, Alain Crouzil, Pablo Zegers, and Sergio A. Velastin. Skeletal movement to color map: A novel representation for 3d action recognition with inception residual networks. *CoRR*, abs/1807.07033, 2018.

[58] Priyanka Mandikal and R. Venkatesh Babu. Dense 3d point cloud reconstruction using a deep pyramid network. *CoRR*, abs/1901.08906, 2019.

[59] Nuitrack sdk. `https://github.com/3DiVi/nuitrack-sdk`, 2020. Accessed: 2020-04-10.

[60] Haohan Wang, Xindi Wu, Pengcheng Yin, and Eric P. Xing. High frequency component helps explain the generalization of convolutional neural networks. *CoRR*, abs/1905.13545, 2019.

[61] Rotation and orientation in unity. `https://docs.unity3d.com/ScriptReference/Transform-rotation.html`, 2019. Accessed: 2020-01-15.

Appendix A

Twenty layer Residual Network Architecture

A.1. Appendix: Resnet-20

We avoid the ReLU unit after adding information from the residual unit and bypass connection so that gradients can easily flow to early layers during back-propagation [**12**]. The baseline model has a ReLU connection after every residual block. Resnet-20 formed by one convolution block at the beginning and nine residual units, each unit have two convolution blocks and a final dense block. The number of residual blocks in Resnet-N model is given by $Numberofresidualblocks = \frac{2\times N-2}{2}$.

A.1.1. Baseline architecture of Resnet-20

3×3 Conv, 16 filters, BN, ReLU
Residual Unit : BN-ReLU-Conv,16 filters -BN-ReLU-Dropout-Conv, 16 filters-$\bigoplus -ReLU$
Residual Unit : BN-ReLU-Conv,16 filters-BN-ReLU-Dropout-Conv, 16 filters-$\bigoplus -ReLU$
Residual Unit : BN-ReLU-Conv,16 filters-BN-ReLU-Dropout-Conv, 16 filters-$\bigoplus -ReLU$
Residual Unit : BN-ReLU-Conv,32 filters-BN-ReLU-Dropout-Conv, 32 filters-$\bigoplus -ReLU$
Residual Unit : BN-ReLU-Conv,32 filters-BN-ReLU-Dropout-Conv, 32 filters-$\bigoplus -ReLU$
Residual Unit : BN-ReLU-Conv,32 filters-BN-ReLU-Dropout-Conv, 32 filters-$\bigoplus -ReLU$
Residual Unit : BN-ReLU-Conv,64 filters-BN-ReLU-Dropout-Conv, 64 filters-$\bigoplus -ReLU$
Residual Unit : BN-ReLU-Conv,64 filters-BN-ReLU-Dropout-Conv, 64 filters-$\bigoplus -ReLU$
Residual Unit : BN-ReLU-Conv,64 filters-BN-ReLU-Dropout-Conv, 64 filters-$\bigoplus -ReLU$
Global Pooling -Dense Layer

A.1.2. Experimented architecture of Resnet-20

3×3 Conv, 16 filters, BN, ReLU
Residual Unit : BN-ReLU-Conv,16 filters-BN-ReLU-Dropout-Conv, 16 filters-$\bigoplus$
Residual Unit : BN-ReLU-Conv,16 filters-BN-ReLU-Dropout-Conv, 16 filters-$\bigoplus$

Residual Unit : BN-ReLU-Conv,16 filters-BN-ReLU-Dropout-Conv, 16 filters-$\oplus$
Residual Unit : BN-ReLU-Conv,32 filters-BN-ReLU-Dropout-Conv, 32 filters-$\oplus$
Residual Unit : BN-ReLU-Conv,32 filters-BN-ReLU-Dropout-Conv, 32 filters-$\oplus$
Residual Unit : BN-ReLU-Conv,32 filters-BN-ReLU-Dropout-Conv, 32 filters-$\oplus$
Residual Unit : BN-ReLU-Conv,64 filters-BN-ReLU-Dropout-Conv, 64 filters-$\oplus$
Residual Unit : BN-ReLU-Conv,64 filters-BN-ReLU-Dropout-Conv, 64 filters-$\oplus$
Residual Unit : BN-ReLU-Conv,64 filters-BN-ReLU-Dropout-Conv, 64 filters-$\oplus$
BN-Global Pooling -Dense Layer

Appendix B

DenseNet Architecture

B.1. Appendix: DesneNet-40, k=12

As per [**14**], the number of dense blocks implemented in a Dense-N model is three. Each dense block implements $\frac{N-4}{3}$ convolution blocks. Each convolution block is a combination of ReLU-Conv and an optional dropout layer. Every convolution block is followed by a transition layer and includes ReLU- 1×1 Conv - optional dropout layer - Average pooling layer. The final dense block is followed by a dense layer. Complete Dense-40 model can ge given as below :
Conv layer- Dense Block(12 Conv layers)- Transition layer (1 Conv layer)- Dense Block(12 Conv layers)- Transition layer (1 Conv layer)- Dense Block(12 Conv layers)- Dense Layer.

Appendix C

Accuracy graphs SkepxelRel

C.1. Appendix:Accuracy graphs for Intel Realsense data with SkepxelRel representation

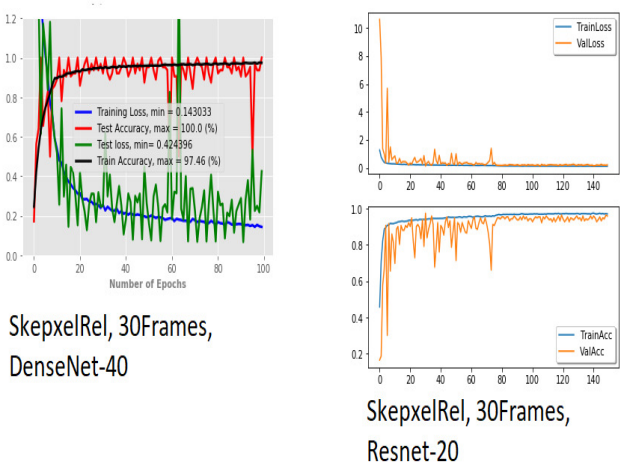

Fig. C.1. SkepxelRel accuracy graphs with average frames as 30

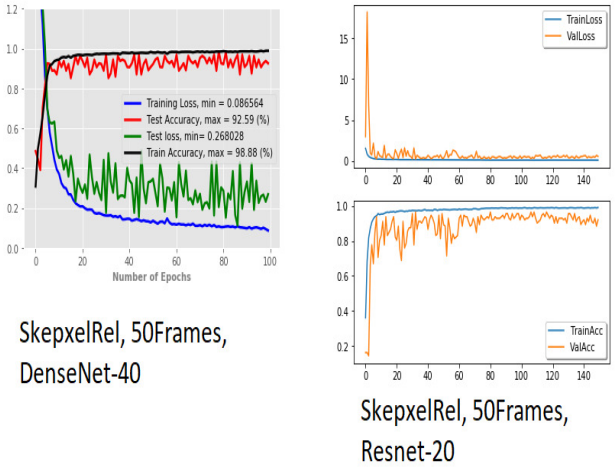

Fig. C.2. SkepxelRel accuracy graphs with average frames as 50

Appendix D

Extended SkepxelRel accuracy graph with MSR dataset :AS1

D.1. Appendix:Accuracy graphs for MSR dataset AS1 with extended SkepxelRel representation

We conducted experiments on MSR dataset [**22**] along with extended SkepxelRel representation and Densenet-40 architecture, and figure D.1 shows the performance graph for MSR dataset. MSR dataset is divided into three sections: AS1, AS2, and AS3. The author in [**22**] captured data when ten different subjects performing actions and the actions performed by subjects with ID's 1, 3, 5, 7, and 9 are used as training data rest are used for preparing test data. We could achieve 100% best test accuracy with a 10% test loss.

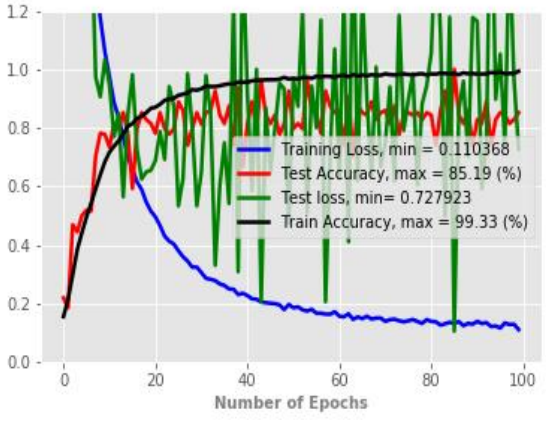

Fig. D.1. Extended SkepxelRel accuracy graphs with average frames as 50

Appendix E

Conference accepted

We have submitted a paper based on the SkepxelRel method to "KS Marulasidda Swamy, Hamdi Ben Abdessalem and Claude Frasson (full paper submission accepted) Real-time Gesture Recognition Using Deep Learning Towards Alzheimer's Disease Applications, Brain Function Assessment in Learning(BFAL) 2020, Crete, Greece, 2020", and below are the details.

Our proposed healthcare system is a combination of deep learning and virtual environment modules. Preparing training and testing data for training deep neural network models is done by Marulasidda Swamy. Training the deep neural network model by fine-tuning hyperparameters and testing the model in real-time is also done by Marulasidda Swamy. Hamdi integrated the human gesture recognition model into the VR environment and tested it in a real environment. Hamdi also took part in checking Alzheimer's mental status after allowing Alzheimer's experience in the healthcare system we developed. Professor Claude Frasson guided us through the complete project while taking significant design changes and implementation decisions. Marulasdidda Swamy and Hamdi contributed to writing the paper and were reviewed and concluded by Professor Claude Frasson.

Real-time Gesture Recognition Using Deep Learning Towards Alzheimer's Disease Applications

KS Marulasidda Swamy, Hamdi Ben Abdessalem and Claude Frasson

Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Canada
{marulasidda.swamy.kibbanahalli.shivalingappa,
hamdi.ben.abdessalem}@umontreal.ca, frasson@iro.umontreal.ca

Abstract. There have been significant efforts in the direction of improving accuracy in detecting human action using skeleton joints. Determining actions in a noisy environment is still challenging since the Cartesian coordinate of the skeleton joints provided by depth sense camera depends on camera position and skeleton position. In a few of the human-computer interaction applications, skeleton position and camera position keep changing. The proposed method recommends using relative positional values instead of actual Cartesian coordinate values. Recent advancements in the Convolution Neural Network (CNN) help us achieve higher prediction accuracy using image format input. To represent skeleton joints in image format, we need to represent skeleton information in matrix form with equal height and width. With some depth sense cameras, the number of skeleton joints provided is limited, and we need to depend on relative positional values to have a matrix representation of skeleton joints. We can show near the state-of-the-art performance on MSR 3-Dimensional(3D) data and the new representation of skeleton joints. We have used image shifting instead of interpolation between frames, which helps us have state-of-the-art performance.

Keywords: Human action, Gesture recognition, Real-time, Skelton-joint, Deep learning, Resnet.

1 Introduction

Representing skeleton joint information in an image format and utilizing it for human action detection is the most reliable and computationally powerful approach. Processing real images or videos for action detection requires a lot of computation resources [1]. There has been tremendous research effort to improve prediction accuracy in detecting human action with the help of skeleton joint information. CNN (Convolution Neural Network) exploits the spatial relationship between pixels when arranged in matrix representation [2, 3, 4]. Shift invariance property possessed by CNN helps in detecting features residing in any part of the image. Encoding spatial and temporal information of skeleton frames in an image is proven to be the best representation for a deep neural network to understand human action [2, 3, 4].

Detecting human action when the camera's position and the position of the skeleton keeps changing is a challenging task [5]. We need to train the CNN model with a lot of

training data to understand all variations in the coordinate values of a skeleton. Encoding spatial and temporal information of skeleton frames in an image is not sufficient, and hence we need to consider encoding the difference between joints for the skeleton transformation process. Depth sense cameras provide a limited, varying number of joints [6], and therefore it has become challenging to come with better representation of skeleton information.

We propose a method to encode the difference between 3D coordinates values in an image and train a deep residual neural network [7] for better prediction accuracy. Existing practice insists on adapting interpolation between frames as the approach to fill the picture when we do not have enough frames [3, 4]. CNN can only understand static images, and hence we need to bring in temporal dependency of frames of the current image on previous frames of the skeleton action sequence. We can achieve exploiting a better representation of the picture by shifting earlier frames to the current image.

This method could be used to detect hand gestures and body gestures in many fields, especially for medical applications like to create applications for Alzheimer's disease. The rest of this paper is organized as follows. In section 2, we give an overview of the related works. In section 3, we describe our methodology. In section 4, we detail the Residual Network. In section 5, we detail the experiments, and finally, in section 5, we present the obtained results.

2 Related Works

Skeleton joint information was extensively used for predicting human action and posture detection. Intel Realsense camera [10] provides precise skeleton joints information with third-party SDKs (Software Development Kit). Nuitrack is one of the most reliable SDK's in the market, with which it is easy for a Unity developer to build a skeleton tracking application. Depending on the system's hardware abilities, framerate changes, and it is effortless to develop a hardware-independent software module to capture skeleton frames in real-time with the help of the Unity platform. Kinect [8] of Microsoft provided a skeleton tracking facility for a long time, and it was adopted in most of the research practice. Kinect [8] provides just twenty skeleton joints information; Intel Realsense camera [10] instead can capture twenty-four joints 3D coordinate values. Leap Motion hardware is a dedicated camera for detecting hand joints position along with rotation. There have been efforts to convert 3D coordinate values to RGB (Red, Green, Blue. A color model represents a pixel's color by combining Red, Green, and Blue in different ranges) image representation for training deep neural networks. The transformation step of skeleton information to RGB representation is a significant data preprocessing stage. Encoded RGB image should include extensive temporal and spatial information of skeleton frames in a sequence.

2.1 Realtime pose detection

The authors in [8] explain estimating human pose using skeleton data given by the Kinect sensor. Instead of using the temporal sequence of 3D coordinates, relative to the

camera position, the authors in [8] uses coordinate values relative to the other joints. Relative coordinate values remove the prediction accuracy dependency on the size and location of the subject. Firstly, three-dimensional skeleton coordinates transformed into a one-dimensional feature vector. The feature vector is the input to a machine learning algorithm with or without pre-processing. The proposed algorithm is assessed on a vocabulary containing eighteen poses and employing machine learning algorithms: Support Vector Machines (SVM), Artificial Neural Network, K-Nearest Neighbors, and Bayes classifier and SVM outperforms on the data set used in experiments. The method explained in [8] works excellently with a predefined set of actions and failed to consider the temporal dependency of frames in predicting human action.

2.2 Skeleton Based Action Recognition Using Translation-Scale Invariant Image Mapping and Multi-Scale Deep CNN

Transforming from skeleton information to image representation is a crucial and significant step in human action classification using skeleton data. A sophisticated, promising method of transformation is discussed and demonstrated by the authors in [2] with the help of results. Very few parameters, which plays a vital role in the transformation process, are extracted from each video sequence instead of referring the whole data. The proposed method in [2] helps preserve scale invariance and translation invariance of the training data. The authors in [2] also claim that the complete process of transformation becomes data set independent.

2.3 Recognizing human action from Skeleton moment

Deep learning algorithms need data to be represented in image format so that machine learning models like CNN and its variants can extract image features and classify the image into an available class efficiently. Transforming skeleton joint coordinate values into RGB image space is explained by the authors in [3]. Skelton parts are divided into five significant parts P1, P2, P3, P4, and P5. Each section will have 3D coordinate values of the set of skeleton joints (P1, P2: two arms, P4, P5: two legs, P3: trunk). Transformation module explained by the authors in [3] will convert skeleton joints into an image by arranging pixels in the order of P1->P2->P3->P4->P5. The proposed transformation method helped the authors in [3] achieve the best prediction accuracy with three different variants of Resnet models [7]. The paper [3] fails to effectively incorporate Spatio-temporal information of skeleton motion when the skeleton motion has a higher number of frames.

2.4 Skepxels: Spatio-temporal Image Representation of Human Skeleton Joints for Action Recognition

We need an effective method to represent skeleton 3D coordinates so that deep learning models can exploit the correlation between local pixels, which helps us have better prediction accuracy. The paper [4] talks about a method that can help us arrange skeleton information like Skepxels [4] in the horizontal and vertical directions. Skepxels [4]

in the horizontal direction, carry the frames in skeleton data. Spatial information of the skeleton frame sequence was captured in the vertical direction of the transformed image by rearranging pixels of a skeleton frame at a time 't'. The proposed way of arranging skeleton frames in [4] will increase the prediction accuracy as each image carries rich temporal-spatial information. In [4], the author also explains how using image interpolation between frames can create a full image even though we have a smaller number of frames in a skeleton motion. If the number of frames exceeds the number of frames required to make an image, then the rest of the frames are moved to the next image and labeled with the same class name. NTU 3D action data [9] was used to evaluate the model, and the transformation process generates millions of pictures after the transformation step. For data-augmentation author in [4] has recommended adding Gaussian noise samples to each frame and double the training data size. With all the proposed changes in [4], the Resnet model [7] can achieve state-of-the-art performance. If the position of the camera and skeleton changes, then model prediction accuracy will change to a great extent. With the proposed method in [4], we need to take more data with all possible skeleton positioning to ensure better test accuracy. When the skeleton frame sequence is long, dividing sequence into multiple images will ignore the current picture's temporal dependency information on the previous frames in the series.

3 Our Methodology

Using pixels of training images, CNN tries to build minor and significant features of images. CNN models are translation invariance, and they can recognize trained characteristics anywhere in the pictures. This paper demonstrates how to generate images from skeleton joints information by creating building blocks of a picture called SkepxelsRel. We don't use skeleton joint coordinates; instead, we use a list of 3D coordinate values generated after taking the difference between two joints. We need to group a set of pair of joints which contribute more in deciding the class of action. Combining a couple of skeleton joints is also a hyperparameter during training a Resnet model [7]. Velocity frames generated uses the speed at which the difference of considered skeleton joints changes. As demonstrated in [5], when we take the reference point as other joints, the prediction accuracy does not depend on the camera and skeleton's position. We explain the approach as follows:

3.1 Skeleton Picture Relative Elements (SkepxelsRel)

SkepxelsRel does have a similar structure of Skepxels explained in [4]. SkepxelsRel tensors encode differences of coordinate values along the third dimension (Fig. 1). We follow the same strategy in choosing the best pixels arrangement for filling spatial information of a skeleton frame at time 't.' As shown in Fig. 2, RGB channels encode spatial-temporal information of skeleton joint differences and create an image. Velocity frames (Fig. 3) are constructed using a similar method, as explained in [2], but we use SkepxelsRel values to calculate the rate at which the differences between reference

joints change. With the proposed method, we can generate any number of joints required for image representation. As shown in Fig. 1, we created forty relative skeleton joints, which play an essential role in deciding human action. As shown in Fig 3, velocity frames are generated by taking the difference of successive frames and dividing them by frame rate. In our experiments, we considered the frame rate as 30 frames/second.

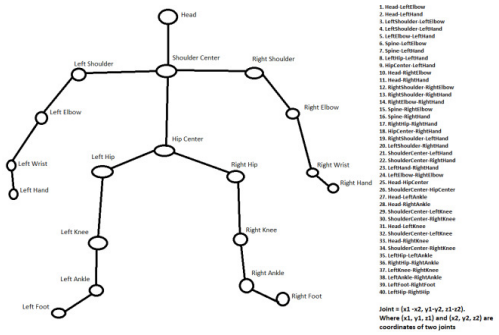

Fig. 1. Skeleton Example with relative joints

$$Relative\ Joint(x`, y`, z`) = Reference\ Joint1(x, y, z) - Reference\ Joint2(x, y, z) \quad (1)$$

$$Velocity\ of\ a\ relative\ Joint\ at\ time\ t = \frac{Joint(x`,y`,z`)\ at\ t - Joint(x`,y`,z`)\ at\ t-1}{frame\ rate} \quad (2)$$

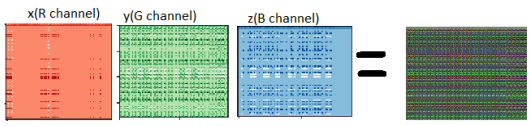

Fig. 2. RGB Channels generated with (x, y, z) coordinates of skeleton sequence

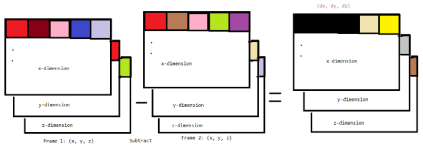

Fig. 3. Velocity frames calculated by subtracting frames

When the number of frames required to form the image is more than needed, we recommend using frames shifting (Fig. 5) instead of moving the remaining skeleton frames (Fig. 4) to the next image. This way of image construction helps in real-time prediction wherein each image encodes only the original frames of the skeleton motion without adding interpolated frames in between. And this method also helps us to encode temporal dependency information of previous frames in the current image. If the available number of frames for constructing an image is less than the required, we can go with interpolation between frames approach. Fig. 5 demonstrates the steps involved in a proper way of adjusting the frames to accommodate all the available skeleton frames.

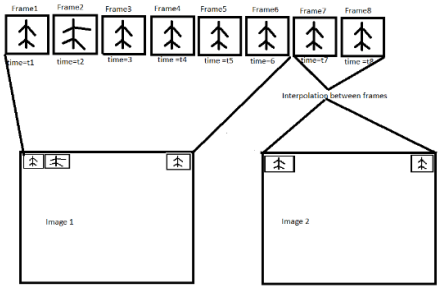

Fig. 4. Existing method: Interpolation between frames applied

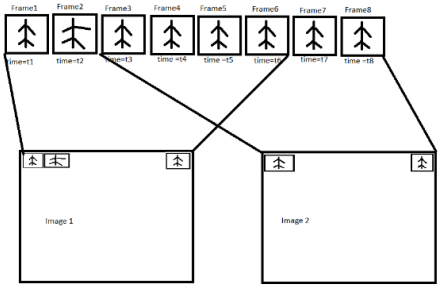

Fig. 5. Frames are shifted to right and temporal dependency of frames is not ignored

3.2 Data pre-processing

We generated skeleton sequences for primary actions using the Intel Realsense camera [10] with the help of Nuitrack SDK. Hand gesture recognition experiments are conducted on multiple channel images. Each skeleton frame is normalized by making the center of the frame the center of the coordinate system (0, 0, 0) [1].

3.3 Data augmentation

To increase the training data size, we sampled from a gaussian distribution with mean 0 and a standard deviation of 0.02 and added those noise samples to actual skeleton frames. We have also applied random cropping, horizontal flip, and vertical flip data augmentation strategies (Fig. 6).

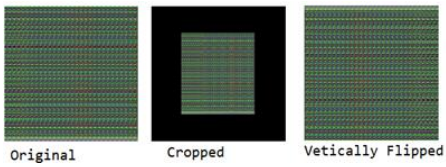

Fig. 6. Data Augmentation

4 Residual Network

Deep neural networks with many more layers stacked, help the model to have a greater number of parameters, and hence degree freedom of a model increases. With the increased complexity of the model, the ability to learn new sophisticated features will also increase. When a neural network has the freedom to choose parameters without regularization, then the chances of finding global minima are less, and the model ends up finding local minima. Hence, we include regularization methods to regulate the most in-depth neural networks and try avoiding model overfitting behavior. Recent experiments and research show that even after having regularization methods in the deep neural network, it is inevitable to have an overfitted model. Researchers have come up with new architecture called Residual Network to avoid such behavior without losing the benefits of deep neural networks [7].

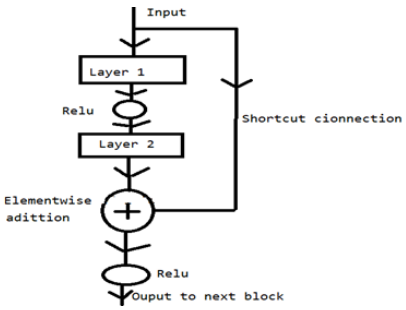

Fig. 7. Residual Block

One of the significant problems associated with deep neural networks is vanishing gradients problem, wherein gradients at the last layer will not be able to propagate back to initial layers. Hence, learning will be very slow and improper. Shortcut connections provided in Residual blocks (Fig. 7) make the model learn identity mapping of the input very easily. Also, the shortcut connection helps to carry gradients back to initial layers without vanishing gradients problem. Hence, we have adopted the Residual network [7] in our experiments to learn significant features of skeleton sequence.

5 Experiments

5.1 Real-time prediction using Intel Realsense camera

We used a setup having Intel Realsense camera [10] for capturing skeleton frames on the Unity platform. 20-layer Resnet model [7] was trained for six basic actions, including Still, Wave Hands, Soothing, Come, Go, Clap. We observed that the trained model could predict all the actions with 100% accuracy in real-time. As stated already, we have used a set of joints that are responsible for deciding pre-decided actions. Other factors also behave as hyperparameters like frame rate (number of frames per second captured by the unity platform, hardware dependent), and image size. As the data available is less with only six human actions, we had to augment data to satisfy Resnet [7] requirements. We tried with different frame rates:30, 20, and 10 and 30 outperformed compared to other framerates. Since we are using differences of coordinates, changing camera position, and skeleton position did not impact prediction accuracy. We also tried with different image sizes 180*180, 250 * 250, and 180*180 outperformed compared to other image dimensions.

The proposed method does not need the number of joints to be equivalent to the required number of joints to form a SkepxelsRel since we can generate the required number of values by taking differences among responsible joints. Leap motion camera can provide hand joints information along with hand joints rotation information. This set up is used in a different application wherein rotation and moment of joints are very important in deciding hand gestures. Hence, we encoded hand joints position information in the first three channels and rotation information in the next three channels.

5.2 MSR Action 3D Data set

MSR data set [11] is divided into three data sets, and model performance is evaluated on each data set type. There are twenty actions performed by ten different subjects in generating each dataset type. We use actions from five subjects for generating training data and remaining data used for testing. There is a total of 557 action files having 20 actions performed by different subjects. Generated data is trained and tested with Resnet-20 [7] and Resnet-50 [7] models, and Resnet-20 [7] model outperformed the rest of the models.

6 Results and Discussion

Intel Realsense data: We captured skeleton data for six necessary actions using Intel Realsense depth camera [10]. We have trained 20-layer and 50 Resnet models [7] with a batch size of 64, optimizer as Stochastic Gradient Descent, initial learning rate as 0.01. The accuracy graph (Fig. 8) shows that the model converges very slowly with a lot of variation in validation accuracy. Validation accuracy fluctuation is not an issue. The variation is due to low validation data, the high degree of freedom of the model, large batch size, and high learning rate. This fluctuation gets stabilized with a greater number of epochs.(Data is uploaded here: https://github.com/creative-swamy/IntelRealSenseData). It is evident from the accuracy graph (Fig. 8) that the model can predict the unseen action data efficiently since we see 100% test accuracy with the loss nearing to zero. The data is captured from seven different subjects. Seven different people perform each action, and actions performed by four subjects are considered for training data, and the remaining are regarded as validation data. We made sure that the data used for testing is unseen data and has noise and variation compared to training data. If the model performs better with the test data, then it can be considered for testing in real-time need. We tested the model performance in the Virtual Real environment with two unknown subjects, which are not part of training and testing data, performing actions. The model can predict all the trained actions with 100% prediction accuracy.

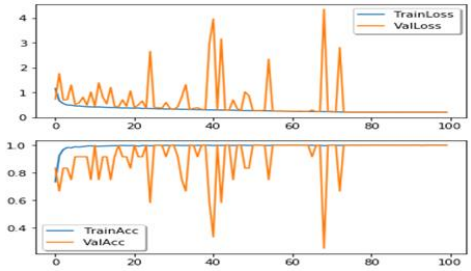

Fig. 8. Intel Realsense data, accuracy graph

Leap Motion data: Leap Motion data: Leap Motion camera provides information about hand joints position and their respective rotation values. We have captured all joints position of two hands and individual rotation values for ten different hand gestures. The data is obtained from ten different subjects. Six subjects are considered for training data, and the remaining subjects are regarded as validation data. We trained the Resnet-20 model [7] incrementally by adding more hand gesture data, and the model behavior is very consistent concerning validation accuracy (Fig. 9). We tested the trained model's performance in a real-time Virtual Reality environment with two unknown subjects, which are not part of training and testing, performing actions. The model can predict all six gestures made by unknown subjects with 100% prediction accuracy, and it is

evident from the accuracy graph shown in figure 9. (Data is uploaded here: https://github.com/creative-swamy/Leap-Motiondata-for-experiments/).

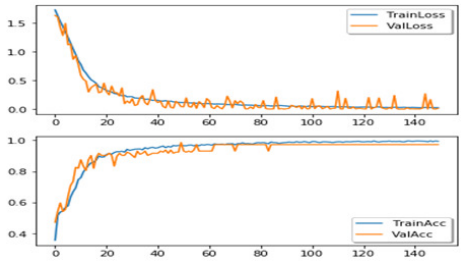

Fig. 9. Leap motion data, accuracy graph

Running Experiments with MSR 3D Action data [11]: We started exploring proposed algorithm behavior with one of the benchmark data set, MSR 3D Action dataset [11]. We tested the model's functioning with a cross data strategy and found that validation accuracy stops at 91%. We have not yet considered converting the skeleton data to scale-invariant and view-invariant [1]. Efficient pre-processing of the skeleton data will make sure proper learning curves to establish. Our research aims to address the moving object and camera position while implementing real-time action or gesture prediction algorithms. We wanted to experiment on the standard dataset to show that our proposed method performs near the state-of-the-art model. We can improve the model performance by enhancing the training data size using more advanced data augmentation methods and extensive hyper-parameters tuning. Our one more research aim is to implement and using a sophisticated, real-time compatible machine learning model in a medical application environment. We concentrated more on experimenting with our prepared dataset and hence did not get more time to tune the model for MSR 3D action dataset. We consider enhancing our model performance in the future to work even better with standard datasets like the MSR 3D action dataset.

7 Conclusion

This paper demonstrated a skeleton-based action detection mechanism using the residual neural network model with a unique way of data representation. The experiments on data captured from Intel Realsense camera [10] and Leap motion prove that the algorithm outperforms real-time prediction. The analysis conducted on a challenging data set, MSR 3D human action dataset, also shows that the proposed algorithm provides near the state-of-the-art performance. Results show that considering relative positional values to construct images provide better accuracy in real-time human action prediction using skeleton joint information. Also, using this method, we can create a medical application for Alzheimer's disease. There are challenges to address when implementing

a solution for treating Alzheimer's patients. Existing solutions only provide assisting tools and a virtual environment to help improve cognitive abilities and avoid negative emotions in the participants. Recent research shows that an interactive virtual environment helps a healthcare system treat Alzheimer's effectively, and hence, we have proposed an interactive virtual environment solution for treating Alzheimer's. We can create even a very sophisticated virtual environment for training purposes, but the environment should help Alzheimer's patients overcome negative emotions and improve cognitive abilities. Research work proves that Animal Assisted Therapy allows Alzheimer's patients to improve their mental status. In this project, we have created a virtual dog and a horse character in the VR environment. Research has proved that the Alzheimer's patients will have reduced agitation, increased physical activity, improved eating, and improved pleasure feeling behavior after a real dog visits into the patient's environment. We aim to use our proposed method of human action prediction in a sophisticated Virtual Environment created for Alzheimer's patients and study the impact of a virtual treatment on Alzheimer's mental status.

Acknowledgment. We acknowledge NSERC-CRD (National Science and Engineering Research Council Cooperative Research Development) and BMU for funding this work.

References

1. Simonyan Karen, Zisserman Andrew: Two-Stream Convolutional Networks for Action Recognition in Videos, NIPS, 2014.
2. Bo Li, Mingyi He, Xuelian Cheng, Yucheng Chen, Yuchao Dai: Skeleton Based Action Recognition Using Translation-Scale Invariant Image Mapping and Multi-Scale Deep CNN, CoRR, 2017.
3. Huy-Hieu Pham, Louahdi Khoudour, Alain Crouzil, Pablo Zegers, Sergio A. Velasitin: Learning and Recognizing Human Action from Skeleton Momement with Deep Residual Neural Networks, 8th International Conference of Pattern Recognition Systems (ICPRS 2017), 2017.
4. Jian Liu, Naveed Akthar, Ajman Mian: Skepxels: Spatio-temporal Image Representation of Human Skeleton Joints for Action Recognition, CoRR, 2017.
5. Youness Choubik., Abdelhak Mahmoudi.: Machine Learning for Real Time Poses Classification Using Kinect Skeleton Data, 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV), 2016.
6. Yong Du, W. Wang, L. Wang: Hierarchical recurrent neural network for skeleton based action recognition. CVPR, 2015.
7. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun.: Deep Residual Learning for Image Recognition, CoRR, 2015.
8. Jamie Shotton, Andrew W. Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images, CVPR, 2011.
9. Amir Shahroudy, Jun Liu, Tian-Tsong Ng, Gang Wang.: NTU RGB+D: A Large Scale Dataset for 3D Huaman Activity Analysis, CoRR, 2016.

10. Anders Grunnet- Jepsen, Dave Tong.: Depth Post-Processing for Inetl RealSense D400 Depth Cameras.
11. Wanqing Li, Zhengyou Zhang, and Zicheng Liu: Action recognition based on a bag of 3D points, CVPR Workshops, 2010.
12. Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, Xiaoou Tang: Residual Attention Network for Image Classification, CoRR, 2017.
13. Andre Karpathy, Justin Johnson, Li Fei-Fei.: Visualizing and Understanding Recurrent Networks, CoRR, 2015.
14. R Lun, W.Zhao.: A survey of applications and human motion recognition with microsoft Kinect, International Journal of Pattern Recognition and Artificial Intelligence, 2015.
15. Microsoft. Kinect for Windows - Human Interface Guidelines v2.0. Technical report, 2014.
16. Chen Chen, Kui Liu, Nasser Kehtarnavaz.: Real-time human action recognition based on depth motion maps. J. Real-Time Image Processing, Journal of Real-Time Image Processing, 2013.
17. Sepp Hochreiter, J¨urgen Schmidhuber : Long shortterm memory, Neural Computation, 1997.
18. Raviteja Vemulapalli, Felipe Arrate, Rama Chellappa.: Human Action Recognition by Representing 3D Skeletons as Points in a LieGroup. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014.

Appendix F

Conference accepted

We have submitted a paper based on the Extended SkepxelRel method to "Ben Abdessalem, H., Ai, Y., Kibbanahalli Shivalingappa, M. S., Frasson, C. (full paper submission accepted) Virtual Reality Zoo Therapy for Alzheimer's Disease Using Real-time Gesture Recognition, Genetics Geriatrics and Neurodegenerative diseases (GeNeDis) 2020, Crete, Greece, 2020", and below are the details.

Preparing training and testing data for training deep neural network models is done by Marulasidda Swamy. Training the deep neural network model by fine-tuning hyperparameters and testing the model in real-time is also done by Marulasidda Swamy. Hamdi integrated the human gesture recognition model into the VR environment and tested it in a real environment. Hamdi also took part in checking Alzheimer's mental status after allowing Alzheimer's experience in the healthcare system we developed. Yan AI contributed to developing a sophisticated VR environment, which includes animal characters. Professor Claude Frasson guided us through the complete project while taking significant design changes and implementation decisions. Marulasdidda Swamy, Yan AI and Hamdi contributed to writing the paper and were reviewed and concluded by Professor Claude Frasson. Professor Claude Frasson funded the complete work.

Virtual Reality Zoo Therapy for Alzheimer's Disease Using Real-time Gesture Recognition

Hamdi Ben Abdessalem, Yan Ai, Marulasidda Swamy KS and Claude Frasson

Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Canada
{hamdi.ben.abdessalem, yan.ai,
marulasidda.swamy.kibbanahalli.shivalingappa}@umontreal.ca,
frasson@iro.umontreal.ca

Abstract. Alzheimer disease affects almost 10 million people every year. Negative emotions such as frustration and anxiety can have impact on brain capability in term of memory functions. Alzheimer's patients experience more negative emotions then healthy older adults. Non-pharmacological treatment such as animal therapy could help Alzheimer patient but has restrictions and requirements. We propose a Virtual Reality Zoo Therapy system in which the patients are immersed in a virtual environment and can interact with animals using their hands. With the immersive experience of Virtual Reality, patients feel that they are in a real therapy room and can freely interact with animals. This system is controlled by an intelligent agent which tracks the patients' emotions using electroencephalography and commands the animals according to their hand gesture and emotions. Experiments have been done and preliminary results show that it is possible to predict patients' hand gesture and interpret them in order to interact with virtual animals and the Zoo Therapy system can reduce the negative emotions.

Keywords: Virtual Reality, EEG, Intelligent Agent, Immersive Environment, Gesture Recognition, Zoo Therapy, Emotions.

1 Introduction

There is an increasing number of people with Alzheimer's disease (AD) and, unfortunately, there is no effective pharmacological treatment that can stop or reverse the disease's progression. It is known that negative emotions such as frustration and stress have an impact on the brain capability in term of memory and cognitive functions and this is visible also in adults with AD.

Non-pharmacological approaches to reduce the impact of symptoms may be interesting. For instance, animal-assisted treatment, can temporarily relieve or improve symptoms. The interaction between an animal and human results in an increase of neurochemical initiating, a decrease in blood pressure and relaxation. This may be beneficial for ameliorating agitate behavior and psychological symptoms of AD [1].

This treatment method has very strict requirements and restrictions on the treatment environment and the animals participating in the treatment. Virtual reality (VR) has proven to be efficient in treating some disorders and could eliminate the restrictions of

the real-world methods using its immersion. Thus, a VR system with virtual animals could eliminate the restrictions and requirements of real animal therapy.

However, how the virtual animal can recognize the interaction? Gesture recognition of the patient is a way to detect what command could be communicated to interact with animals present in a VR environment in order to make the environment more immersive and feels like a real work which could attract the patients to actively cooperate with the treatment and increase the treatment effect.

As the system is intended to calm the patient, we need to measure his-her emotions and their evolution. For that we use electroencephalography (EEG) with a portable device that can track the emotions of the patients in real-time. According to the evolution of the emotions we change the behaviour of the virtual animals in order to continue to reduce negative emotions.

In order to create this system, we need to combine VR for immersion, gesture recognition for animal reactions, EEG for measuring patients' emotions and neurofeedback for adapting animal comportment to the patients' emotions. The creation of such a system is complicated because we have to use three different devices at the same time: VR headset, EEG headset and a Hand Tracking device. Each device needs to have a module in order to communicate with it: a virtual environment with the VR headset, a measuring module with the EEG headset, and a gesture recognition module with the Hand Tracking device. The challenge is to synchronize between these modules in order to have a real-like user experience.

Our research questions are: **Q1- is-it possible to predict hand gesture in or-der to interact in a virtual reality environment?** and **Q2 - is-it possible to reduce negative emotions while interacting with animals?**

The rest of this paper is organized as follows. In section 2, we give an overview of the characteristics of AD. In section 3, we present our approach and detail the different modules of the system that we developed. In section 4 we detail the experimental procedure. Finally, in section 5 we present and discuss the obtained results.

2 Related Works

2.1 Animal Therapy for Alzheimer's Disease

Alzheimer's disease is a chronic progressive neurodegenerative disease that usually starts slowly and gradually worsens over time, it is the cause of 60–70% of cases of dementia [2][3]. It has three primary groups of symptoms. The most common symptom is cognitive dysfunction. The second group comprises psychiatric symptoms and behavioral disturbances (for example, depression, hallucinations, delusions, agitation collectively) termed non-cognitive symptoms. The third group comprises difficulties with performing activities of daily living.

The symptoms of Alzheimer's disease progress from mild symptoms of memory loss to very severe dementia [2]. When the situation deteriorates, patients often be-come withdraw from their families or society [2] and gradually lose their physical function, eventually leading to death [4]. The cause of Alzheimer's disease is poorly understood

[2]. No treatments stop or reverse its progression, though some may temporarily improve symptoms [3]. Most of these treatments are palliative.

Current treatments can be divided into pharmaceutical, psychosocial and caregiving. Pet therapy (animal-assisted therapy (AAT)) is a Stimulation-oriented treatment of psychosocial, which is an interaction between humans and animals for therapeutic purposes. It can help someone recover from a health problem or mental disorder. The most used types of AAT are dog assisted therapy and horse assisted therapy. AAT aims to improve patients' social, emotional or cognitive function. A growing body of research shows the social, psychological and physical benefits of animal-assisted therapy in health and education [5]. In aged people, AAT can be used for ameliorating agitate behaviors, psychological, occupational, social and physical disorders especially in Alzheimer and Dementia. AAT can be increase social interactions by initiating decrease the agitate behaviors of patients with Alzheimer and Dementia [6]. People with Alzheimer may have an easier time decoding the simple repetitive, non-verbal actions of a dog. Animals can act as transitional objects, allowing people to first establish a bond with them and then extend this bond to people. Most of the study results revealed that AAT especially dog therapy had an “calming effect” on the patients with dementia and Alzheimer disease [7].

2.2 Virtual Reality and Alzheimer’s Disease

Over the last years, Virtual Reality started to be used in many fields due to its re-markable advantages such as the immersion. There have many reports revealing the benefits of VR for AD patients. Some researchers showed that VR intervention with computerized cognitive training can improve cognitive domains in individuals with mild cognitive impairment or AD [8,9]. Additionally, AD patients prefer completing cognitive training tasks in VR over its pencil-paper counterpart [10]. This technology has been applied in the field of psychology to treat various disorders, including brain damage [11], and alleviation of fear [12].

Most studies focus on the use of VR to help users improve cognitive performances [13,14]. However, several researchers are investigating the importance of VR at a more physiological level [15,16].

2.3 Gesture Detection

Recognizing human action from a video sequence depends on various factors, including the background of video frames, facial expression, and the rate at which position of body changes. An efficient method of information extraction requires removing unwanted background noise and balancing or ignoring varying light effects in different video frames.

There have been efforts to convert 3D coordinate values into RGB image representation for deep neural network training. Encoded RGB image should include extensive temporal and spatial information of skeleton frames in a sequence. Recognizing human gestures using 3D coordinate information is challenging when Alzheimer patients

perform gestures. The dataset we prepared for our experiments includes mainly hand-gestures because they are more relevant for any patient to per-form.

The DenseNet [17] helps us derive the best prediction model from challenging the complex dataset. The authors in [18] talks about the effective method for trans-forming the temporal sequence of human skeleton moments into RGB representation. The technique proposed in [18] does not become dependent on the length of the skeleton sequence and efficiently can extract global features. Below are the steps followed in [18] for skeleton sequence to RGB transformation: Encode human poses into RGB images; Enhance local textures of the RGB images by applying AHE [19]; Before feeding images into D-CNN, a smoothing filter is applied to reduce the input noise effect; Discriminative features can be learned by feeding images to DenseNet [17]. DenseNet [17] is one of the most effective CNN architectures for image classification. ESPMF is an enhanced version of SPMF [19], which in turn includes encoded PFs and MFs. The PFs encode skeleton joints position information, and MFs encode the rate of skeleton joints changes concerning all other skeleton joints. The proposed method in [18] achieves the state-of-the-art performance on MSR action data [20] and NTU RGB+D dataset [21]. ESPMF representation shows a 1.42% increased pre-diction accuracy when compared to SPMF [19] representation.

3 Our approach: Zoo Therapy System

In order to reach our goals, we propose a Zoo Therapy System. This system is composed of 4 main components: Zoo VR environment, EEG Measures, Gesture Recognition and an Intelligent Agent.

The users/Ad patients are immersed into Zoo VR environment and an EEG measuring module measures their emotional reactions to the environment. The gesture recognition module tracks hand gesture in real time. The intelligent agent receives the hand gesture and users' emotions in real-time and intervene in Zoo VR by commanding animals depending on the emotions and the gestures of the users. Figure 1 illustrates a general architecture of our zoo therapy system.

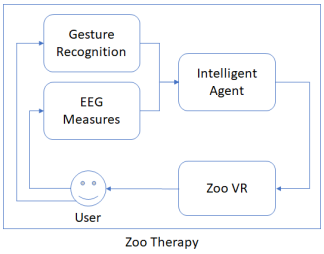

Fig. 1. Architecture of Zoo Therapy

Following is a detailed description for each module of our approach.

3.1 Zoo VR

We propose to create "Zoo VR" which creates a safe and economic environment including animals. In this environment, the user can call the animal to approach, eat or ask it to leave at any time. Animals respond as soon as they receive instructions from users. In addition to the user's gestures, the animals in the environment can also determine the next inter-action by sensing the user's emotions, that is, the changes of the user's emotions will affect the animals' coming and going and some other actions in real time.

Our environment can be divided into five Functional Modules, namely, scene module, animal module, sound effect module, map module and human-computer interaction module. Following a description for each module.

Fig. 2. 3D treatment room

Fig. 3. 3D horse

Scene Module - The overall appearance of the environment. In the scene module, we created a 3D treatment room (shown as figure 2)

Animal models-animals in the environment. The most common forms of AAT are dogs and horses. Therefore, we created a 3D treatment horse and a 3D treatment dog. (Figure 3, the horse is in the treatment room). In order to match the interaction between animals and users, we made some animations while generating 3D animals, such as walking, running, eating, etc.

Sound effect module-environment and animal sound effects. We not only play soothing background music in the entire 3D environment; we also add different animal sound effects. For example, horse and dog can make several different calls in response to different commands from users, and the sounds of horse's walking and eating, the dog's panting. The sound effect will accompany the animal's movement and change in real time according to the different actions of the animal.

Map Module-the trajectory of animal movement and the generation of interactive routes. The function of the map module is to calculate and generate a feasible path according to the animal's real-time position and state, so that the animal will update its state under the path and approach or go away from the user.

Human-computer interaction module-user interaction with the environment. First, the user can interact with the animals by selecting the 3D button in the environment. Then, after completing the model training of the gesture recognition system, we will directly update the 3D button to gesture recognition. Users can use gestures to make animals come, walk, eat or leave. In addition, a neurofeedback system has been added

to the environment to influence animal behavior by identifying emotional changes in the user's EEG. When the two instruction modules work together, we give priority to the gesture recognition system.

Finally, we integrate the above five modules. The scene module adds the map module and the animal module, the animal module adds the sound module, and finally connects with the human-computer interaction module. The gesture recognition system and neurofeedback system will then be linked to the map module, the scene module and the animal module to form the final zoo treatment.

3.2 EEG Measures

In this research we use the Emotiv Epoc EEG headset to track emotions. The headset contains 14 electrodes spatially organized according to the International 10-20 system, moist with a saline solution. The electrodes are placed in antero-frontal (AF3, AF4, F3, F4, F7, F8), fronto-central (FC5, FC6), parietal (P7, P8), temporal (T7, T8) and occipital (O1, O2) regions with two additional reference sensors placed behind the ears. The detailed position of the measured regions is shown in figure 4.

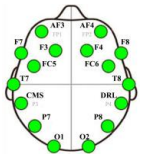

Fig. 4. Emotiv headset sensors placement

The Emotiv system generates raw EEG data (in μV) with a 128Hz sampling rate as well as the five well-known frequency bands, namely Theta (4 to 8 Hz) Alpha (8 to 12Hz), low Beta (12 to 16 Hz), high Beta (16 to 25 Hz) and Gamma (25 to 45 Hz).

The system uses internal algorithms to measure the following mental states: meditation, frustration, engagement, excitement and valence. Although we don't have access to the system's proprietary algorithms, studies have provided evidence showing the reliability of its output [22].

3.3 Gesture recognition system

Our proposed extended SkepexelRel approach for skeleton sequence to RGB representation is motivated by the method explained in [18] wherein the recommended way by [18] produces an efficient RGB image of skeleton sequence. We need a few more modifications on top of the method explained by [18] for real-time prediction applications. Alzheimer's treating use demands a continuous prediction of the patient's hand gestures. Hand gesturing made by the patient changes the virtual environment, and these changes will have a direct impact on the patient's medical status. Data for training is captured from a Leap Motion camera with the Unity platform for basic hand gestures including:

- "Left Hand Come" and "Right Hand Come": will make the animals come near the user,
- "Left Hand Go" and "Right Hand Go": will make the animals goes away from the user,
- "Version2 Left Hand Come" and "Version2 Right Hand Come": will make the animals come near the user,
- "Left Hand Wave" and "Right Hand Wave": will make the animals exit the room,
- "Left Hand Still" and "Right Hand Still": will not affect the animals.

There are two versions of the "Come" gesture since both have different rotation values. Data from the Leap Motion camera provides twenty-six joints information. Each frame data will have the position and rotation values of twenty-six joints. The position represents the joint's actual position in the Unity scene, and rotation values are the rotation of a joint relative to the world coordinate system wherein rotation in all three axes is generated for each joint. With Leap Motion and Unity setup, it is effortless to capture all the details of hand gestures made by Alzheimer's. Rotation values play a very significant role in the hand gesture recognition process. Most of the actions listed above have little variation in position values but will have differences in rotations. Unity updates rotation in all three axes with the help of Euler angles. We need a mechanism to transform both position and rotation values into RGB representation, similar to the method explained by [18]. The figure 5 shows features learned by the DenseNet model with the extended SkepxelRel RGB representation. We experienced a 100% real-time human gesture prediction accuracy in the VR environment.

Fig. 5. Feature maps learned by DenseNet

3.4 Intelligent Agent

In order to personalize the zoo therapy to every participant, The Intelligent Agent tracks the emotions and the gesture of the patient while they are immersed into Zoo VR and intervene in the environment in order to command the animals. The com-mands send to Zoo VR depends on the emotions and the hand gesture of the participants.

The agent uses a rule-based system in order to adapt the environment to the participants. For instance, if the frustration of the participants increases when the animal approaches them, the agent makes the animal go away.

The agent combines hand gesture and emotions as an input in order to make a decision of an action. It starts by given priority to the gestures and tracks at the same time the participants' emotions in order to intervene in case of negative emotions. For instance, if the participant performs a "Come" gesture in order to make the animal come and then the agent detect that his negative emotions are increasing while the animal approaching him, it will command it to go away.

The weight of the rules is updated after each intervention in order to adapt the system to the participant. For example, if the agent makes the animal go away when the participant is frustrated but the frustration doesn't decrease, the agent will understand that the animal is not the reason for the frustration and will decrease the weight of the rule so next time another rule with higher weight will be applied.

4 Experiments

In order to analyze the effectiveness of our approach we started by training the hand gesture prediction module. Hand gestures were performed by ten different subjects, and data from subjects [1, 3, 5, 7, 8, 9, 10] were used for training, and remaining subjects are used for testing. We make sure that testing data has unseen rotations and positions of hand joints to validate the trained model. Leap motion data that we captured for our experiments include actions that can be performed in short duration. Data were obtained at different frame rates supported by the camera and we used the best frame rate for our experiment purpose (30 frames per second).

After that, we aimed to experiment the entire Zoo Therapy system with participants which has these following criteria:

- Older than aged 60 of age
- Francophone
- Normal or correct-to-normal vision
- Normal hearing
- Met the Consortium for the Early Identification of Alzheimer's Disease – Quebec (CIMA-Q) criteria for SCD:
  - Presence of a complaint defined as a positive answer to the following statements: "my memory is not as good as it used to be" "and it worries me"
  - MoCA 20-30
  - No impairment on the logical memory scale based on the education-adjusted CIMA-Q cut-off scores.

Unfortunately, we were not able to perform experiments due to COVID-19 circumstances, but we were able to test our system on one participant. We started by equipping the participant with an EEG headset. When the exercises were complete, we added the Fove VR headset in which we installed the Leap Motion devise (used for gesture prediction) and the participant started the immersive experiment.

5 Results

The first objective of this research was to discover whether **it is possible to predict hand gesture in order to interact in a virtual reality environment**. Results shows that DenseNet architecture can learn unique features for every action from the Leap motion training dataset. Trained DenseNet can predict test images in real-time with 100% accuracy (accuracy graphs are shown in figure 6). We evaluated trained DenseNet in real-time, and it is possible to predict every two-second action in less than 0.5 seconds.

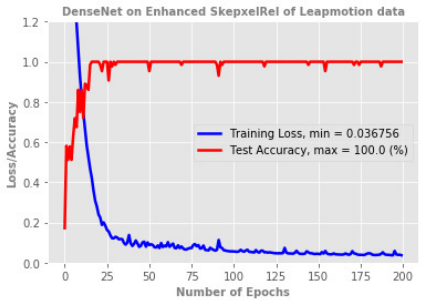

Fig. 6. Learning graphs for Leap motion dataset

The second objective of this research was to analyze **if it is possible to reduce negative emotions while interacting with animals**. To this end, we analyzed the mean frustration of the participant before, during and after Zoo Therapy. Results shows that, before the therapy the mean frustration was 0.524, during Zoo Therapy, the mean frustration was 0.429 and after the mean frustration was 0.486. Figure 7 shows the difference between the mean frustration before, during and after Zoo Therapy.

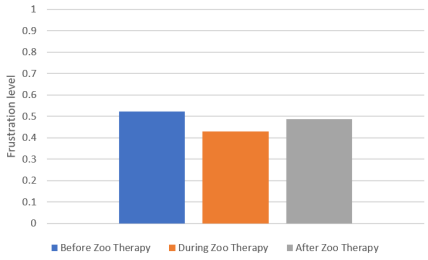

Fig. 7. Histogram of general mean frustration

Even though these results are only from one participant, these results show that Zoo Therapy has the potential to reduce negative emotions and thus reduce AD symptoms.

6 Conclusion

In this paper, we presented a novel approach which could be used to improve AD patients' memory performance by reducing their negative emotions using Zoo Therapy system. We created a VR environment in which we can interact with animals using gesture recognition module. An intelligent agent intervenes in real-time in order to control the animals depending on participants' emotions and gesture. Experiments were conducted during which we collected hand gesture data in order to train the gesture recognition module. We tested our system and results showed that we can predict hand gestures and we might reduce negative emotions with Zoo Therapy system. These results indicate that our system might be used to reduce AD symptoms.

Acknowledgment. We acknowledge NSERC-CRD (National Science and Engineering Research Council), Prompt and Beam Me Up Labs for funding this work.

References

1. Sibel Cevizci, Halil Murat Sen, Fahri Güneş and Elif Karaahmet. Animal Assisted Therapy and Activities in Alzheimer's Disease.
2. Burns A, Iliffe S (February 2009). "Alzheimer's disease". BMJ. 338: b158.
3. "Dementia Fact sheet". World Health Organization. 12 December 2017.
4. GBD 2015 Mortality Causes of Death Collaborators (October 2016). "Global, regional, and national life expectancy, all-cause mortality, and cause-specific mortality for 249 causes of death, 1980-2015: a systematic analysis for the Global Burden of Disease Study 2015". Lancet. 388 (10053): 1459–1544.
5. 10. Fung S (2017). "Canine-assisted reading programs for children with special educational needs: rationale and recommendations for the use of dogs in assisting learning". Educational Review. 69 (4): 435–450.
6. Nancy E Richeson." Effects of animal-assisted therapy on agitated behaviors and social interactions of older adults with dementia." PMID:14682084
7. Barbara W McCabe 1, Mara M Baun, Denise Speich, Sangeeta Agrawal. "Resident dog in the Alzheimer's special care unit. West J Nurs Res".
8. Coyle, H., Traynor, V., & Solowij, N. (2015). Computerized and Virtual Reality Cognitive Training for Individuals at High Risk of Cognitive Decline: Systematic Review of the Literature. The American Journal of Geriatric Psychiatry, 23(4), 335–359
9. Hill, N. T. M., Mowszowski, L., Naismith, S. L., Chadwick, V. L., Valenzuela, M., & Lampit, A. (2016). Computerized Cognitive Training in Older Adults With Mild Cognitive Impairment or Dementia: A Systematic Review and Meta-Analysis. American Journal of Psychiatry, 174(4), 329–340
10. Manera, V., Chapoulie, E., Bourgeois, J., Guerchouche, R., David, R., Ondrej, J., … Robert, P. (2016). A Feasibility Study with Image-Based Rendered Virtual Reality in Patients with Mild Cognitive Impairment and Dementia. PLOS ONE, 11(3), e0151487.

11. Rose, F. D., Brooks, Barbara. M., & Rizzo, A. A. (2005). Virtual Reality in Brain Damage Rehabilitation: Review. CyberPsychology & Behavior, 8(3), 241–262.
12. Gorini, A., & Riva, G. (2008). Virtual reality in anxiety disorders: The past and the future. Expert Review of Neurotherapeutics, 8(2), 215–233.
13. Appel, L. (2017). How Virtual Reality could Change Alzheimer Care. Technologie. Retrieved from https://fr.slideshare.net/TechnoMontreal/how-virtual-reality-could-change-alzheimer-care
14. Biamonti, A., Gramegna, S., & Imamogullari-Leblanc, B. (2014). A Design Experience for the Enhancement of the Quality of Life for People with Alzheimer's Disease. What's On: Cumulus Spring Conference.
15. Todd, R. M., & Anderson, A. K. (2009). The neurogenetics of remembering emotions past. Proceedings of the National Academy of Sciences, 106(45), 18881–18882.
16. Vindenes, J., de Gortari, A. O., & Wasson, B. (2018). Mnemosyne: Adapting the Method of Loci to Immersive Virtual Reality. In L. T. De Paolis & P. Bourdot (Eds.), Augmented Reality, Virtual Reality, and Computer Graphics (Vol. 10850, pp. 205–213).
17. Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks.
18. Huy-Hieu Pham, Houssam Salmane, Louahdi Khoudour, Alain Crouzil, Pablo Zegers, and Sergio A.Velastin. A deep learning approach for real-time 3d human action recognition from skeletal data.
19. Huy-Hieu Pham, Louahdi Khoudour, Alain Crouzil, Pablo Zegers, and Sergio A. Velastin. Skeletal movement to color map: A novel representation for 3d action recognition with inception residual networks
20. W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, pages 9–14, 2010.
21. Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3d human activity analysis.CoRR, abs/1604.02808, 2016.
22. Aspinall, P., Mavros, P., Coyne, R., & Roe, J. (2015). The urban brain: Analysing outdoor physical activity with mobile EEG. British Journal of Sports Medicine, 49(4), 272–276.