2020

Machine learning approaches to physical activity prediction in young children using accelerometer data

Tuc Van Nguyen
University of Wollongong

MACHINE LEARNING APPROACHES TO PHYSICAL ACTIVITY PREDICTION IN YOUNG CHILDREN USING ACCELEROMETER DATA

A Dissertation Submitted in Fulfilment of
the Requirements for the Award of the Degree of

Doctor of Philosophy

from

UNIVERSITY OF WOLLONGONG

by

Nguyen Van Tuc

School of Computing and Information Technology
Faculty of Engineering and Information Sciences

2020

CERTIFICATION

I, **Nguyen Van Tuc**, declare that this thesis, submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Computing and Information Technology, Faculty of Engineering and Information Sciences, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institution.

Nguyen Van Tuc
2020

Dedicated to

My family

Table of Contents

List of Tables

List of Figures

ABSTRACT

Early childhood development is arguably the most significant period in the course of life. It is widely recognized that physical activity (PA) during early childhood plays an influential role on current and future developments of the child [1]. Partially based on this evidence, the Australian Government has created the Physical Activity Recommendations which recommend that, among others, preschoolers should be physically active every day for at least three hours, spread throughout the day [1]. However, difficulties in accurately measuring physical activity in preschoolers have impeded the investigations in physical activity classifications using data modelling techniques and the use of such classifications in the estimation of the metabolic equivalents (METS[1]), a measure commonly used as a proxy for measuring the extent of the physical activity performed by a subject. Therefore the issue of quantifying the extent of physical activity performed by a child is transformed to an issue of physical activity classifications into categories, like "sedentary", "light" activity, "medium" activity, "walking", or "running". Based on such classifications, the METS can be estimated, and as a result the daily recommended minimum METS can be monitored.

The research reported in this thesis is part of a larger research project which include the collection of raw data, over two separate and different small cohorts of young pre-school children, in 2014 (11 participants), and 2016 (16 participants) respectively, from accelerometry sensors mounted on various parts of the body. As these are pre-school children, they often did not adhere to the suggested activity, but instead engaged in unscripted activities during the 5 minute episodes of observations, thus introducing "noise" in the recordings. Despite such imperfection, the accelerometer recordings were labelled by the assigned activity type, irrespective of what the subject was doing during the episode thus challenging data driven modelling techniques.

Moreover, for probity reasons, as the subjects were pre-school children, consent of the parents before activity trials could be conducted was needed. Each activity trial was recorded using a video camcorder. The videos were taken as "evidence" that the children were engaged as was agreed upon with the parents and as was approved by the Ethics Committee on Experiments Involving Human Subjects. The videos were taken by a camcorder which was mounted on a tripod. The camcorder was left unattended most of the time. The videos were not meant as a data source for PA classification because accelerometers are the commonly accepted standard source of information. Nevertheless, this thesis will explore both data sources for the investigations on PA prediction. The videos are used to validate the results from using the accelerometer data.

The broad task of this thesis, is to "make sense" of these data recordings, pertaining to their ability of PA classifications independent of the subjects performing the activities. Faced with such a challenging task of "making sense" of two small cohorts of subjects, performing the assigned activity only once, and each cohort performed different types of activities, which are only broadly classified into five categories: "sedentary", "light", "medium", "walking" and "running", the first task was to normalize the recordings, and to extract features from the resulting time sequences.

As there was little understanding of the nature of accelerometer recordings of preschool children, this thesis introduces and uses a High Resolution Self Organzing Map (HRSOM),

[1] METS is an objective measure of the ratio of the rate at which a person expends energy, relative to the mass of that person, while performing some specific physical activity compared to a reference, set by convention at 3.5 mL of oxygen/kg/min, which is roughly equivalent to the energy expended when sitting quietly.

which maps the high dimensional feature vectors to a two dimensional display space, with the property that any two feature vectors that are close to each other in the high dimensional feature space are mapped to be close to one another in the two dimensional display space. High resolution in this context means that the display space should be of sufficient resolution to permit the visualization of any intricate details of interest in the display space. The unknown nature of preschool children data required us to use benchmark datasets with known properties, viz., the policemen dataset, and the network intrusion detection UNSW-NB15, to study the anticipated capabilities of the HRSOM. The visualization of the two PA datasets reveals that there is considerable overlap between some of the classes of physical activities performed by the participants, and that there are considerable variations in the grouping of samples from within the same activity class.

The samples in the PA datasets are broadly classified into five categories of accelerometer recordings. Such datasets would be ready for experimentation with classification techniques, except for one issue: the unbalanced nature and the small number of samples in each category of physical activity types. To address the issue, this thesis proposes a novel data sampling technique to generating more samples for each class where needed. To achieve this, the thesis first introduces a supervised DBSCAN (Density Based Spatial Clustering of Applications with Noise) method to label each sample based on the sample density of the region near the sample point. To help with the identification of dense regions in the high dimensional feature space, this thesis first projects the samples to a lower dimension with the help of the HRSOM. This procedure allows the identification of where more data is required to enhance differentiability of the pattern classes. For each additional point, a corresponding high dimensional feature vector is generated using a simple linear interpolation technique between two vectors from the same class which are closest to the point which is to be generated. The reason why we need to generate the corresponding feature vector of the added point is that this thesis explores how the sampling procedure is effective in enhancing the prediction accuracy of multilayer perceptron (MLP) with a single hidden layer. We call this method a synthetic sampling ensemble network (SSEN). When applying the SSEN to the PA datasets, we found that in general, the SSEN outperforms the baseline MLP method by about 10% in generalization accuracies. It is shown that the SSEN also works well with other benchmark datasets, in improving the generalization accuracies.

The thesis then explores the concept of transfer learning. Transfer learning is a method used to retain knowledge gained in one domain, the source domain, and transfers it to another domain, the target domain. This thesis uses the following architecture: the source domain is trained using a simple MLP with one or two hidden layers. When the training converges, the parameters of the classifier are "frozen" and this constitutes the source model. Then, one or more additional hidden layers are appended to the source model. The parameters of the newly added layers are trained using the target domain data to obtain the target model. The trained target model is then tested on the target domain testing dataset. Applying this methodology to the PA datasets it is found that transfer learning can improve the generalization accuracy by 2% to 5%.

We then explored the possibility of using the "evidence" videos for physical activity classifications even though their primary purposes were not intended for classification pur-

poses. We first removed the segment in which the subject did not appear in the video. It is observed in some of the videos, during some segment of the episode that the subject was being partially or completely occluded. This necessitated some kind of tracking, or re-identification of the subject when the subject was partially or completely occluded. A fast bounding box location technique called Yolo2 (You Only Look Once version 2) was used to "demarcate" the region of interest, which is where the subject was located in the video, while other subjects, e.g., the instructor, or other children are ignored, by making a simplifying assumption that the subject is furthest from the video camcorder. A Kalman filter method was used to track the "hidden" bounding boxes when they are occluded. A human re-identification method was used to re-identify the bounding box when it emerges from occlusion, and the corresponding trajectory of bounding boxes of the subject can be reconstructed. Motion based features and deep CNN based features were extracted from the bounding boxes, and the corresponding subject who was inside the bounding boxes were extracted respectively. Those features were classified using the deep neural network. It is found that, in general, the tracking with re-identification achieves approximately 2% better generalization accuracy when compared with the tracking without compensation for the partially or completely occluded segment(s).

The contribution of this thesis include the following: (1) a scalable HRSOM method for visualizing high dimensional data; (2) a data augmentation method which utilizes the HRSOM to aid the training of an MLP classifier with one or two hidden layers; (3) a novel method to generate new samples, from two close existing samples, grouped together to belong to the same class using a supervised DBSCAN method, and its incorporation into an SSEN classification ensemble system; (4) a simple and effective way to align the categories of differently labelled accelerometer recordings from two different cohorts over different time span using a K-mean clustering algorithm on mapped points of high dimensional feature vectors in the two-dimensional HRSOM display space; (5) an effective and novel transfer learning regime which retains knowledge accumulated in a source model to the target domain; (6) an effective procedure for overcoming a possible partially or completely occluded moving subject leading to good classification results.

KEYWORDS: Recognition, Physical activities, Modeling, Neural networks, Deep learning, Computer vision, Feature extraction.

List of Publications

- M. Hagenbuchner, D. P. Cliff, S. G. Trost, N. V. Tuc, and G. E. Peoples, Prediction of Activity Type in Preschool Children using Machine Learning Techniques.*Journal of Science and Medicine in Sport*, vol. 18, no. 4, pp. 426-431, 2015. (This publication forms the part of chapter 6)

- Nguyen V.T., Hagenbuchner M., Tsoi A.C. (2016) High Resolution Self-organizing Maps. In: Kang B., Bai Q. (eds) AI 2016: Advances in Artificial Intelligence. AI 2016. Lecture Notes in Computer Science, vol 9992. Springer, Cham. (This publication forms the part of chapter 5)

- S. G. Trost, D. Cliff, M. Ahmadi, N. Van Tuc, and M. Hagenbuchner, Sensor-enabled activity class recognition in preschoolers: Hip versus wrist data, *Medicine and science in sports and exercise*, vol. 50, no. 3, pp. 634-641, 2018. (This publication forms the part of chapter 6)

- Saraswati, A., Nguyen, T., Hagenbuchner, M. and Tsoi, A., High-resolution Self-Organizing Maps for advanced visualization and dimension reduction. *Neural Networks*, No 105, pp. 166-184, 2018. (This publication forms the part of chapter 5)

- N. V. Tuc, M. Hagenbuchner, and A. C. Tsoi, Synthetic sampling ensemble network. Submitting to *Neural Networks*. (From materials contained in Chapter 6)

Acknowledgements

I would firstly like to express my deepest gratitude to A. Prof Markus Hagenbuchner for being my supervisor during my PhD study. His consistent encouragement, guidance and direction is incomparable. It has been a great honour for me to study under Markus's supervision. He always provides very detailed and constructive instructions on my work. Without his continuous, meticulous directions and the uncountable-hours of work, I would not have been able to conquer different hurdles during my study. I have been deeply impressed by Markus's friendliness, politeness and endurance from my initial research program. And I am looking forward to working with Markus in the future.

I would also express my special appreciation to my co-supervisor Prof Ah Chung Tsoi for his massive help during my research. Ah Chung Tsoi selflessly shared his strong and unique research vision with me and helped me with many brilliant research ideas. His inspiration and encouragement will never be forgotten. This thesis would never be the same without the tireless working and insightful contributions from Ah Chung. In addition, I appreciate to have been on the receiving side of helpful suggestions and comments from other researchers. They are Prof. Franco Scarselli, who is very friendly and supportive, and my laboratory mate Ayu Saraswati who shared with me her knowledge and expertise from the very beginning stage of my study, and many other researchers I had the pleasure to collaborate with.

Thirdly, I acknowledge the financial support received in form of a scholarship for my studies from the ARC discovery project grant that was awarded to A. Prof Markus Hagenbuchner. I especially appreciate the technical support from the Information Technology Services staff at the University of Wollongong, allowing me to access the high performance computer clusters which were essential for my research experimental needs. They were always willingly available to help with any problems with the Internet connection and printing resources.

Last but not least, I likewise owe an eternal debt to my family for their unconditional love. I would like to thank my parents for their continuous encouragement both financially and emotionally during the hardness of my research. And certainly, life seems meaningless without the whole-hearted support of my wife Trang Thi Nguyen and without my adorable little daughter Anh Thu Nguyen. They endlessly fill me up with laughter and joy which is as invaluable as aspirin in helping through the more stressful moments. Finally, my gratitude also extends to my friends who have been assisting me in time of needs.

List of Abbreviations

2D	Two Dimension
ADAM	Adaptive Learning Rate Optimization
(A)NNs	(Artificial) Neural Networks
AUC	Area Under Curve
AVG	Average
ACC	Accuracy
BG	Background
CUDA	Computer Unified Device Architecture
CPM	Convolution Pose Machine
CNN	Convolution Neural Network
DBSCAN	Area Under Curve
DNN	Deep Neural Network
EE	Energy Expenditure
F1	F-measure
FRPN	Fully Recursive Perceptron Network
GNN	Graph Neural Network
GPU	Graphical Processing Unit
GPGPU	General Purpose Graph Processing Unit
HNBD	Hybrid Negative Binomial Distribution
HOG	Histogram of Oriented Gradients
HOF	Histogram of Optical Flow

HRSOM	Hight Resolution Self Organizing Map
KM	K-Mean
MV	Moderate to Vigorous
MLPs	Multilayer perceptrons
MBH	Motion Boundary Histograms
LRSOM	Low Resolution Self Organizing Map
L-GNN	Layered Graph Neural Network
LOPO	Leave One Person Out
LSTM	Long Short Term Memory
PA	Physical Activity
RMSE	Root Mean Square Error
RPROP	Resilient Backpropagation
RB	Roughly Balanced
R-CNN	Region Proposal Convolution Neural Network
RESNET	Residual Deep Neural Network
SSD	Single Shot Multibox Detector
SGD	Stochastic Gradient Decent
SOM	Self Organizing Map
SOMSD	Self Organizing Map for Structured Data
SSE	Sum Square Error
SSEN	Synthetic Sampling Ensemble Network
SVM	Support Vector Machine
YOLO	You Look Only Once

No	Hardware		OS	Number of cores	Core speed	Usage years
	Category	Type				
1	Workstation	Intel	Linux	2	2.1	3
2	Workstation	Intel	Linux	2	2.4	1.5
3	Workstation	Intel	Linux	4	3.5	2
4	Supercomputer	SGI	Linux	9	2.1	3
5	Cluster	AMD	Scientific Linux	240	1.5	3
6	Workstation	Intel	Linux	7	2.1	3

Notation

In this thesis, the mathematic representation is uniformly presented as follows: Lowercase script letters like n are used to indicate scalars and constants. Parameters of a learning model are shown by lowercase Greek letters such as γ. Sets and matrices are indicated by upper case letters, e.g., M. Calligraphic letters like $\mathcal{G}$, $\mathcal{N}$ and $\mathcal{E}$ are respectively used to represent graphs, a set of nodes, and a set of edges. Letters used in combination with brackets such as $h(x, y)$ denote functions. Typical examples are given below:

$x(t)$	The parameter x depends on time t.
$F_w(x, y)$	The function F takes a vector x and y as its arguments, and depends on the variable w.
$M = KL$	The multiplication of the two matrices, or the dot product.
$n = \|d\|$	n is denoted the cardinality of vector d.
$n = \|\|m\|\|$	Variable n takes the positive value of m.
$x = (x_1, x_2, ..., x_n)$	x is a vector containing n elements.
$n \in \{10, 15, 20, 24\}$	A number n can take a value from a set of four elements.

Software Usage

This thesis was completed using LaTeX for Linux version 3.14159265 ©1999 by D.E. Knuth with the Kile user-friendly editor. Some of the images were created in the format of EPS using LibreOffice 4.1 version ©2007 from the Free Software Foundation. Some other EPS format files were produced with gnuplot v4.6.5 ©2004 by Thomas Williams and Colin Kelley, or by xfig version 3.2 patch-level 2 ©1989-1998 by Brian V. Smith, and 1991 by Paul King.

Hardware Environment

The work presented in this thesis includes results from a wide range of experiments on a number of neural networks as well as kernel methods. Hardware resources which were utilized for the experiments are as follows:

Core speed is an approximate value relative to a 1GHz single-core Intel Pentium. The core speed is approximate since the actual speed of a machine dependents on the amount and speed of RAM, the speeds of permanent storage devices such as hard-discs, the number of running tasks, and other factors.

Chapter 1

Introduction

1.1 The motivation

Human activity recognition is an interesting and popular research topic which has numerous applications such in security, military and defence, health, sports, education, agent systems, robotics, systems and services optimization, and others. If activity recognition is to be performed on young children such as preschoolers and school age children then activity recognition can support health science researchers in studying human subjects from a very early stage of their lives. Corresponding applications are related to the children's development programs for diet balancing, obesity avoidance, physical and mental health improvements. The knowledge about physical activity levels in young children is central to weight control and behavior acknowledgment. For example, by predicting the activities a child performs during a period of time, one can estimate the activity-driven behavior as well as the amount of energy intake and energy expenditure of the child. More information can be derived such as whether the child is more likely to be distracted by other events or more disciplined, whether she/he is over active or over sedentary. In general, activity prediction in young children allows researchers to identify causalities for the development of mental health problems or obesity problem, respectively, in later life [2, 3].

There are clear guidelines that recommend young children should be active for reasonable periods of time during the day [4]. However, current estimation methods are not sufficiently accurate in measuring how much and when physical activities are performed by young children [3, 5]. Prior to this thesis the state-of-the-art approach to activity prediction in young children was based on linear regression models of accelerometry [3, 6]. The work in this thesis is motivated by significant advance in machine learning that bear promise of significant improvements in prediction capability as well as abilities of processing richer sources of information for the purpose of predicting activity type and duration in young children. The first approach taken in this thesis is to evaluate different machine learning methods on their ability to predict activity classes and energy expenditure from sensory data. The work is then expanded to the development and adoption of new techniques for activity recognition.

The work being carried out in this thesis is significant in supporting the national children health promotion programs. Additionally, accelerometer-based motion sensors are now used prolifically in population-based studies of physical activity and sedentary behaviour in many age-cohorts [4]. Utilizing expertise from behavioural and health sciences, and computer science the research presented in this thesis will result in the introduction of novel machine learning methods for physical activity recognition which, as will be shown, enhances prediction accuracy significantly. For long term benefits, researchers and public health agencies could enhance physical activity surveillance and more effectively identify individuals at risk. The activity recognition methods investigated here would be also directly applicable to a range of other domains such as security, surveillance, and robotics and have the potential to impact these domains as well.

In order to create a dataset for human physical activity recognition, many supporting sensory devices have been used [7] ranging from a more intrusive device like wearable sensor (e.g. some kinds of accelerometers to attach on children body's parts like chest,

hip or wrist), to a mobile-based device which is embedded in the mobile phone, and a less intrusive one like the camera network. In this research, two main kinds of data collection devices were used, namely accelerometers mounted at the hip and wrist, and tripod held video camcorders.

Accelerometers are very commonly used for the purpose of monitoring human activities particularly in sports science and other areas which require light-weight wearable sensors for data acquisition of body movements [8]. Accelerometers measure the acceleration with a single or triple acceleration axis at a regular observation frequency. The accelerometers applications can be seen in many research works such as the detection of fall [9], movement and analysis of body motion [10, 9, 11] or the prediction of human gait and postural orientation [12, 13, 14]. More details and applications of using accelerometers can be found in [7]. The drawbacks of this mode of data collection is its intrusive nature, which requires the participants wearing the sensors on their body. More intrusive approaches exist and include oxygen-based intake energy expenditure and heart beat measurement devices. Since this thesis studies activity recognition of young children we will use sensory data from low-impact sensors such as lightweight accelerometers and external video capturing devices. This research will use the data collected from wearable accelerometers mounted on children's hip and wrists. This allows us to capture data uninterrupted and unobscured. We will also explore the use of video capturing devices which are less intrusive than accelerometers but have the disadvantage of not being able to follow the child and visual information can be obscured by objects in the environment.

The use of video data is common in prior research on general human action recognition. The common approach is to extract image features from the video captured on the objects of interest. The prediction task is to issue a corresponding action class label for each video clip. Essentially, a video is formed by a number of images (frames) arranged in a chronological order. Thus, techniques in image processing are applicable to the video-based feature extrac-

tion process. Various applications of vision-based human action/activity classification can be found in literature. For example, the detection of unusual human activities [15, 16, 17], fall detection [18, 19], or generic human action and activity recognition [20, 21, 22], just to mention a few. More details about different video datasets and methods applied to human action recognition can be found in the review paper in [23, 16, 24]. The advantages of this data collection is that it is a non-intrusion way which can capture data even without a participants' awareness.

Human action recognition remains a challenging and interesting research topic. In the research presented in this thesis we find that action recognition of very young children adds new challenges. The reasons include:

- Children are not as disciplined as adults. They may not follow the laboratory protocols or experimental settings during a data acquisition phase. This increases the inter-class variations.

- Some activities cannot be performed by the child alone during a given activity trial but they need to be guided by one or more instructors. As a result, the chance that a person is occluded by another person (when using video capturing) is increased, and the chance of interference (i.e. an educator is holding the hand of a child when using accelerometers) is increased. This affects data quality and is hence creating a greater burden on the detection and tracking algorithm.

- The children are commonly more active than adults, changing activity type more frequently, they seem never staying still for long time. This makes the capturing of activity samples for activities such as sedentary, story time or quiet play more difficult and increases the intra-class variation. This increases the chance that an activity pattern may be confused with other activity patterns.

Methods would thus have to demonstrate resilience to variations in data quality in order

to be considered suitable for activity recognition in young children.

The research presented in this thesis is part of a larger project on activity recognition of young children. The project includes a data acquisition phase during which young children were invited to participate in this research. Data acquisition was performed in a controlled environment and under the supervision of qualified educators. While the data acquisition component of the project does not form a part in this thesis, we have the unique opportunity to process relatively complete sets of activity phases from collected raw high-resolution accelerometer data, and the recording of corresponding video clips. This thesis will preprocess the raw data, evaluate and develop a number of machine learning approaches suitable for the activity recognition task.

The main aims of the research presented in this thesis can be summarized as follows:

Aim 1: Investigate machine learning-based modelling approaches to estimate physical activity (PA) type (e.g. sitting, walking, running) from accelerometer data and from observational video data.

In other words, the purpose is modelling accelerometry in preschoolers using machine learning. Different artificial neural networks (ANNs) are developed and evaluated in this study. Methods studied include the standard feed-forward Multi-Layer Perceptron Network (MLP), the Self-Organizing Map (SOM), the synthetic sampling ensemble network (SSEN) and several others. We found that because the MLP tends to perform poorly when dealing with limited number of samples and high dimensional input space, it makes sense to combine the SOM with MLP, or to develop the MLPs in the form of ensemble methods where multiple MLPs can be used for evaluation. A common evaluation approach in PA prediction is leave-one-subject-out cross validation. In particular, the models will be trained on all input samples except for the data of one participant. After training, the model is then tested on the left-out data. The experiment will be repeated until each participant is considered exactly once for

testing.

Aim 2: Develop and test efficient machine learning approaches to modelling information from accelerometers and videos.

Each activity monitor collects data at regular time intervals. The results are numeric temporal sequences. It is important to note that the data collected by the monitors at each body part are not independent. For example, one activity may trigger different responses from two or more of the monitors. Hence, we expect that by taking the context of all the monitored data from one type of monitor into account, this will help to uniquely map the data to an associated activity. The model used for time-series acceleration data is much different from the one used for video-based physical action recognition since when dealing with the visual data, a good amount of image processing work is required. Some typical algorithms include object detection, object tracking and feature extraction from series of object image bounding boxes. The problem then can be simplified to be a time series where at each time step (an image in the video clip), the feature information is extracted for the classification purpose.

Aim 3: Analyse the ability of activity recognition algorithms to handle high-resolution accelerometer data and video data, and the sensitivity to measurement errors or missing measurement samples.

1.2 Research benefits

The benefits of the research in this thesis can be summarized as follows:

1. This research obtains and processes raw data from acceleration sensors and camcorder videos for the purpose of classifying physical activities in (small) children. The problem is challenging and very useful in supporting health science research. None of the

methods studied have been applied previously to activity recognition in young children. The research conducted encapsulates wide range of methods and experimental protocols. The methods produce significantly enhanced recognition accuracy compared with all prior work or methods that has been applied to the same problem or datasets.

2. A visualization method, the high resolution self-organizing map, will be introduced to support a greater level of analysis and insight. The model helps to express the intrinsic characteristics of input data space. The model is examined and experimented with a number of datasets, both artificial and real-world data. It is found that the model can not only be utilized for the purpose of visualization, but is also very helpful as an unsupervised filter for a deep-learning inspired classification models.

3. Due to the nature of input sparsity in the accelerometer data, a Synthetic Sampling Ensemble Network is developed. The new model is capable of handling the lack of training input samples or lack of training sample coverage over the testing set. The model is proved quantitatively and qualitatively better than the other well-known sampling techniques. It is shown that the model can be applied to two main disciplines, namely health science data and cyber-security dataset.

4. The benefits of transfer learning in terms of children physical activity recognition is investigated. It is found that domain background knowledge is essential. If the background model trained on a sufficiently long data set which is related to the target domain, the target model, which is employed by adding layers on top of the background model, this will perform better on the target domain.

5. Vision-based physical activity recognition is usually the most challenging prediction problem since this involves a number of image/video processing techniques, detection and tracking algorithms, solving the occlusion situations, converting the image

sequences to feature vectors, making it possible to be processed by computing devices, and thus addressing high dimensional input space issues. This will also be addressed in this thesis

1.3 Thesis contributions

The main focus of this research is to evaluate, design and develop machine learning algorithms for the purpose of predicting the activity types in children by using raw data from acceleration sensors and video sequences from camcorders recording of the children's physical activities. Since the data has typical properties such as: (1) High dimensional issue which is a consequent of the high resolution temporal accelerometer and video capture. Steps such as data analysis and visualization are required to aid proper understanding of the data before performing selecting suitable pre-processing methods and to design a suitable training procedure; (2) The highly sparse data is caused by the difficulties in collecting children's related data since the experimental protocols and procedures are not straightforward. The number of children participants can be small (i.e about 10 to 15 preschool children) which results in a small number of data samples collected. This is taken into account when selecting or designing suitable methods for data processing.

This thesis proposes two approaches respectively to the two afore-mentioned issues. First, the high resolution self-organizing map is introduced to support the higher level of data analysis. The corresponding algorithm is similar to the standard self-organizing map, however the main focus is to allow high dimensional neuron map sizes and a highly parallel optimization learning process for time efficiency. Thanks to high dimension operation, this model helps to expose the insight intrinsic characteristics of input data space. This thesis examines and tests the proposed method on a number of datasets including controlled synthetic datasets and several real-world datasets. Interestingly, it is found that the model does not only meet the envisaged application objectives, but that it can also be effective as

an unsupervised filter/pre-processor for a classification/supervised learning model. It will be shown that, by using the high resolution self-organizing map as a filter, the classification performance is improved significantly. Secondly, in order to address the sparsity in the accelerometer data, this thesis proposes a Synthetic Sampling Ensemble Network that is capable of handling the lack of training input samples or lack of training sample coverage over the testing set. The approach uses the high resolution self-organizing map in combination with the Dbscan clustering algorithm to create a new sampling technique which can be applied to sample cohorts. It is shown that such a sampling technique is better than any other well-known sampling method.

There is a sparsity of information in the accelerometer data. To address this the thesis proposes an approach by which modelled knowledge from one related dataset/domain is used as background information for another learning/classification problem. The rationale is that variations in experimental protocols when collecting data from accelerometer sensors results in different but related data cohorts. The idea is to exploit the relatedness (knowledge from a related domains, the so-called source domain) by means of transfer learning. To do this, the thesis will train a deep learning model on the source domain, then freeze the model weights in all layers excluding the layers in the output network. Then these last layers can be tuned or expanded by additional layers with trainable parameters. The trainable parameters are then trained on a given classification problem from the target domain. This produces a model that has been trained on the basis of background knowledge from a related domain and which has deep knowledge about the target domain. This thesis finds that if the source domain does contain relevant background knowledge then the target model will perform better in the target domain.

Attaching and wearing accelerometer sensors can be considered somewhat intrusive. This thesis investigates a less invasive sensory device which can provide relevant information for the activity recognition task. This thesis investigates the use of video sequences

recording of children while they were performing physical activity trials. Vision-based physical activity recognition is a challenging problem since this involves a number of techniques required in human activity recognition such as: detection of the human subject within a given scene, tracking of the subject's body parts, human re-identification to address periods of occlusion and to maintain a continuous tracking trail, and the extraction of feature vectors from sequences of frames. The proposed method is based purely on the video signal and is producing a prediction accuracy comparable to that obtained when using accelerometer data. It is thus found that, if the subject can be contained within a field-of-view of a video capturing device (i.e. within a playground, within a child's room) then video data are a viable alternative to accelerometry for activity recognition of young children.

1.4 Thesis structure

The thesis is organized as follows:

Chapter 1: This chapter gives a general overview of the research and includes the underlying ideas of the research topic, the benefit of the research, and an outline of the thesis.

Chapter 2: This chapter gives an overview of related literature, lists available approaches to modelling active play in young children, and presents some background knowledge on several machine learning models and its applications in wearable sensor-based data and computer vision.

Chapter 3: This chapter provides the description of the physical activity datasets.

Chapter 4: This chapter states the central problem which this thesis aims to solve.

Chapter 5: Approach 1: High resolution Self-organizing map for intrinsic visualization and classification purposes.

Chapter 6: Approach 2: Synthetic sampling ensemble network for classification problems.

Chapter 7: Approach 3: Transfer learning applied to time series data for children physical activity recognition.

Chapter 8: Approach 4: Video-based feature extraction for human activity understanding.

Chapter 9: This chapter offers comparisons and discussions on the results obtained by the aforementioned approaches.

Chapter 10: this chapter gives a summary of the research presented in this thesis, lists and explains limitations, and provides suggestions for future work.

Chapter 2

Background and literature review

This chapter presents some important background knowledge and reviews relevant literature on physical activity recognition in children using either acceleration or non-intrusive video captured data. A number of classic and modern machine learning based approaches will be presented. Limitations of traditional methods are examined, and approaches that pre-date this thesis and which produced respectable accuracy performances will be shown.

This rest of this chapter is structured as follows. The first part of this chapter provides a literature review on linear and traditional methods, and recent neural network models on physical activity recognition with a special focus on young children such as preschoolers, school children and adolescence using acceleration data. The basic count sample and heuristic prediction equations are considered first, then supervised models such as the decision tree, approaches based on regression, neural networks and support vector machines are shown. For each algorithm, a comparison with the previous models is made in order to overview the incremental capability, robustness, or complexity of the learning system. Then relevant video based physical activity recognition methods are reviewed.

The second part of this chapter describes in some detail several well-known machine learning algorithms which will be evaluated in this thesis. Particular attention is given on (1) unsupervised clustering models, (2) supervised neural networks and deep learning based

models. This will include the Convolution Neural Networks and object detection models. Relevant to modelling the large amount of feature extracted from the long sequences of video-based data, graph neural models and time series learning models such as the long short term memory will be described.

2.1 Literature Review

Research with the purpose of gaining knowledge from data collected from pervasive sensors defines a broad area of research. Within that area, human activity recognition has become highly attractive topic, especially for medical, military, and security applications [23]. For instance, patients with diabetes, children with high level of obesity, or heart disease are often required to supervise their daily routines as part of their treatment. Therefore, recognizing activities such as sedentary activities, walking, running, or cycling becomes significant to provide useful information for the purpose of behavior analysis, abnormal action awareness and weight control. It can also provide information about the level of physical action for activity orientation programs.

Figure 2.1 presents a common work-flow for physical activity recognition systems. There are many challenges in the system that require the attention of researchers and which motivate the development of new techniques and algorithms to improve prediction accuracy under realistic conditions. Some of these challenges include:

- The construction of a portable, unobtrusive, and inexpensive data acquisition system.
- The design of object detection, tracking, and feature extraction methods.
- The collection of data under realistic or real-life conditions.
- The design of learning and inference models to handle large dimensional input from raw acceleration and video sequence data.

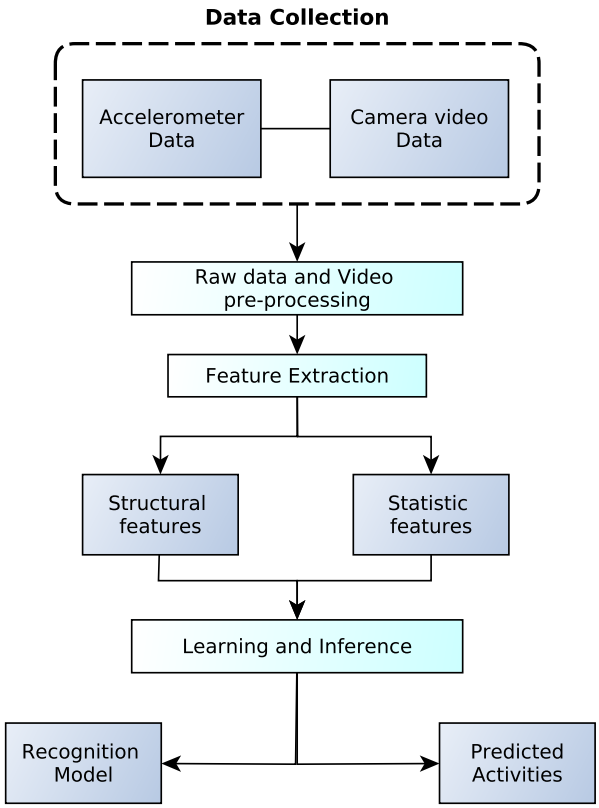


Figure 2.1: Physical activity recognition: A complete system work flow.

- The implementation for just-in-time response given a real-time data acquisition system.

The task of recognizing human physical activities has been approached in two different ways, namely using external sensing systems like video-based monitoring systems and wearable sensors as shown in Figure 2.2. For these two data acquisition approaches, different data processing algorithms and learning models are applied. More details about each are given in the following two sections.

2.1.1 Physical activity recognition on children using wearable sensors

Accelerometers are a de-facto standard type of wearable sensors for research on activity recognition in children [8, 25]. Accelerometers can either provide accelerometry count data

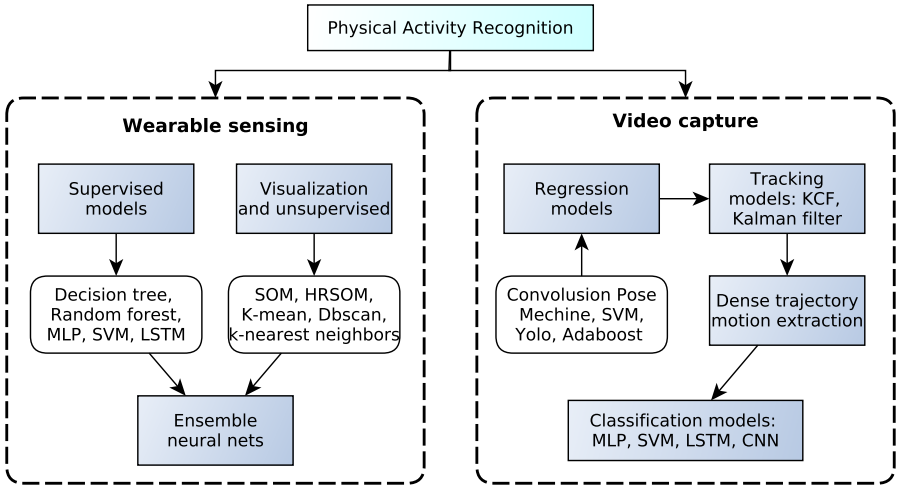


Figure 2.2: Physical activity recognition on children: relevant algorithms in use.

or raw accelerometry sampling data at a particular Hz rate. For the former, researchers have typically used cut-points developed from regression to estimate time spent in each physical activity [5]. The latter is commonly obtained from triaxial accelerometers and are likely the mostly used sensors to recognize ambulation activities e.g., walking, running, lying, etc [8]. From work presented in literature it is found that common sampling frequencies are in the range from 10Hz to 100Hz. The placement of the accelerometer is another important point of discussion. Different works explored various mounted locations on humans' body. Common are placements on the hips, arms, wrists, legs or chest. Researcher found that the best place to wear the accelerometer for activity recognition tasks is on the hip or wrists [5, 8, 25]. However, the optimal position to place the accelerometer depends on the application and the type of activities to be recognized. For example, the activities involved with moving forward and backward of the humans' body can be recognized with high accuracy if using hip mounted accelerometers [25]. On the other hand, when the users frequently move their hands or arms, the arm or wrist attached accelerometers would be of more helpful than the other attached locations. The reason is that well placed sensors capture unique or distinguishable body's part movements which can then be converted into

useful and separable input feature for the classification model.

In terms of prediction models, the traditional method for predicting PA type and energy expenditure is based on cut-point data [5] using regression model. The traditional method has been vastly outperformed by machine learning algorithms such as decision tree, random forest, MLP and SVM [5, 6, 8, 25]. Most recent studies limit the attention to predicting physical activity type in children aged from 5 years old, or school aged and older children using triaxial accelerometers [5, 6, 8]. Our first attempt in recognizing very small children physical activities has been published in [25] where the triaxial acceleration data was collected via laboratory experimental settings for children in the preschool stage. In order to model the very young children activity, both the pre-processing techniques and prediction models need to be taken into account. The traditional methods have not performed well given that the data is affected by noise and by body movement patterns that are irrelevant to the actual activities being performed. The reasons for the poor prediction accuracy is possibly that the traditional methods might not generalize well for younger children when the collected data contains noise. It would thus be interesting to investigate the suitability of machine learning algorithms when applied to data capturing very young children activities. This research studies classification problems such as predicting PA type in various children aged cohorts.

Several works have applied machine learning models to predicting PA type [5, 6, 26, 27, 28]. Authors also applied deep learning inspired models where an unsupervised self-organising map was used as pre-processing stage before the multi-layer perceptron neural network model is applied [25]. The data based on the combination of triaxial accelerometers mounted at hips and wrists was explored for several machine learning algorithms such as multi-layer perceptron networks and support vector machines [8]. The prediction accuracy is better than the case where data from an individual accelerometer (rather than from a set of accelerometers mounted at different locations) is used. In this research, a new ensemble

approach will be introduced which makes use of both unsupervised and supervised methods. It will be found that the proposed model's prediction accuracy is around 3% to 5% better than the other machine learning models and for several children physical activity datasets. As illustrated in Figure 2.2, the ensemble model can be used with the involvement of supervised models and unsupervised ones. There are choices for each type of model that can be made.

It can be important that the data samples in the input space can be appropriately visualized in order to expose their intrinsic characteristics. Since a data visualization tool can be very helpful in data analysis and in data preparation this research also proposes a new visualization and clustering technique for high dimensional input spaces, called high resolution Self-Organizing Map. The thesis will find that the algorithm is not only a useful visualization tool but also as a unsupervised filter for a classification task.

In order to be more applicable in practice, the data collection devices should not require the user to wear many or heavy sensors nor should they interact too often with the application. Even though the more sources of data available, the richer the information that can be extracted from the measured attributes. For instance, a video camera system can be used to record all visual information related to particular physical activities. However, the extraction of sensible and useful information for a given classification model is not straightforward. A review of recent and relevant methods and algorithms for image/video processing and recognition is presented in the following section.

2.1.2 Physical activity recognition on children using captured videos

Camera systems are a typical example of external sensing. In fact, the recognition of human activities and gestures from video sequences is of a great research interest [20].Camera systems are especially suitable for security purposes such as in intrusion detection, human action monitoring such as detecting unusual human activities [15] and fall detection [18]. Even though, the data collection using camera system is less obstructive that the wearable

sensors, video-based monitoring system certainly have some disadvantages:

1. The privacy issue is a problem since there is a greater reluctance to be permanently monitored and recorded by cameras.

2. A camera can only cover a restricted area, meaning that the information will be missed if the subject of interest is out of the field of view. A video capturing device can also not obtain images of the entire body during daily living activities. The subject being monitored would need to stay within a perimeter defined by the coverage capability of the camera. Hence, in recording video sequences involving small children as subjects of interest, the camera location and orientation would need to be non-static or several suitably placed cameras would be needed.

3. The complexity of processing and learning algorithms, since video processing techniques are relatively expensive, and it is hard to make the learning model scalable and operate in real-time.

Due to these issues, video sources are a much less frequently considered alternative to accelerometers as the source of data for PA in children.

An important step in visual information processing is feature extraction. Children activities are performed during relatively long periods of time like in the order of seconds or minutes. While the single sampling data point normally does not provide sufficient information to describe the actual activity, the sliding time window (with overlap or without overlap) is often used for the creation of feature vectors which can then be used as input to the learning or classification system [23]. Different sizes of windows are another point of argument. In practice, short time windows may not provide sufficient information about the activity being performed [25]. Conversely, if the window size is long, there might be more than one activity within a single time window. Whatever the window size is, the feature extraction method should be applied on each window to filter out relevant information and

to obtain quantitative measures. In general, two approaches have been proposed to extract features from time series data, namely the statistical and the structural method. The first one can include some kinds of data transformation such as the Fourier transform and the Wavelet transform which use quantitative characteristics of the data to extract features [29]. The second method takes into account the inter-relationship among data such as auto-correlation or entropy [8]. One can choose to use either of these methods or using them in combination manner, however being dependent with the nature of the given signal type. We note that, as at the time of writing this thesis, Convolutional Neural Networks have not been deployed to video sequences for the purpose of PA prediction of young children. This should be mainly due to the fact that accelerometers are the preferred choice of information in this field of research [5, 8, 25].

Among the many applications of video recording and video processing systems, human action recognition especially with high-level behavior recognition comes out to be one of the most interesting one. An physical activity is a sequence of human body movements, and may simultaneously involves a number of body parts' co-interaction. In terms of the computer vision field, as can be seen in Figure 2.2 the recognition of human action on video sequences need to go through several steps. Major components of such systems include human body and body parts detection, tracking the subject of interest possibly among many other non-interest objects, feature extraction from the detected bounding boxes, action learning, and classification [20]. More details on each step are as follows:

Subject detection: The problem of detecting the body of a child can be divided into (1) whole body detection, (2) body part detection, and (3) corresponding skeleton detection. OpenPose is a well-known library for real-time multi-person key-point detection. OpenPose is computationally efficient by using a multi-threading GPU model. The corresponding algorithm has gone through a number of development stages [30, 31, 32]. It can in a real-time fashion jointly detect a human body, hands, and facial

key-points on a single image. This thesis will use the method to extract the coordinates and correlation features from the children skeleton detected and then by using a time-series prediction model for recognizing the physical activity.

Another approach is to use the whole body detection model. Amodel for this purpose which stands out from the rest can be found in the recently introduced Yolo (You only look once) object detection method [33]. Yolo is a state-of-the-art, real-time object detection system. The method is robust in the sense that it not only offers object detection, but also allows to do classification of the image, say to know whether the detected object is human or some other object. Yolo allows us to know what is exactly the objected being detected (know the class of each detected object). Yolo outperforms other state-of-the-art methods like Faster R-CNN [34] with ResNet [35] and SSD [36] while still running significantly faster [33]. The regression mechanism in the learning stage tries to minimize the error that occurs between the object bounding box and the ground truth bounding box. The use of the whole body detection approach would result in a series of body bounding boxes for further image processing steps.

Object tracking: This step commonly follows the object detection step and functions on an object's bounding box over a sequence of image frames. The fundamental idea behind tracking algorithms is to consider the past movement patterns and changes around the object to predict a future movement direction. There are many tracking algorithms. For example:

- Kalman filter [37]: The functionality of Kalman Filter is to take the current known state (i.e. position, heading, speed and possibly acceleration) of the target and predict the new state of the target at the next time step. In making this prediction, it also updates its estimate of its errors in this prediction. This method was used originally in radar tracking because it takes into account the position, heading, speed and possibly acceleration of the object. Being similar

to the Kalman filter the Multiple Hypothesis Tracker (MHT) [38] can also be an alternative.

- Correlation filter tracking is a Minimum Output Sum of Squared Error (MOSSE) filter, which results in stable correlation filters. The model can work while only a single frame is initialized. An advantage of this tracker is that it is robust to variations in lighting, scale, pose, and deformations while it can operate very fast [39, 40].
- The Tracking Learning and Detection method (TLD) is based on Median-Flow tracker. A Median-Flow tracker uses a bounding box of the object and interprets its motion between sequential frames. Basically, the tracker estimates displacements of points within the bounding box covering the object. The drawback of this tracker is that it is not robust when the object is very small, blurry or of low resolution [41]. In addition, we found that the model performs poorly in conditions where subject's appearance is changing significantly. This, for example, occurs when a kid performs a vigorous PA as this can significantly change the scale, appearance, and orientation of the subject. Nevertheless, the method does work well i.e. tracking a human face in videos since a human face is not rotating as much.

Children re-identification: Periods of occlusion can create great challenges to tracking a human subject. Occlusion occurs more frequently when tracking young children since their smaller size is more easily occluded by other objects or other people. Moreover, children are very active. During occlusion their body can promptly move in a different direction and thus change the appearance of the body. It is thus difficult for an algorithm to re-identify the body of a small person once it re-emerges. Moreover, children act more commonly in the presence of others since they are commonly under the supervision of parents or educators. It is thus more common that others are present

while performing an activity. It is common that a childs' body is from time to time hidden and then re-appears from the perspective of the video capturing device. When they re-appear, their movement direction, distance from camera and their body pose might not the same as at the time when occlusion started. This raises a challenging problem in object re-identification. The tracking trail of a subject is ultimately connected throughout all of a video segment which represents a single physical action. Thus, the use of human re-identification is important, and accurate re-identification methods are required to help the track algorithm to work properly.

There are a number of approaches to human re-identification. The more traditional methods engage feature matching [42], or a part-based mixture of models [43] which requires the detection of body parts in the image. These models are comparable with more recent methods such as the one based on deep convolutional neural network for person re-identification [44, 45]. The method builds up the deep convolutional neural network from scratch or uses pre-trained models that are distributed publicly on the Internet. The state-of-the-art model presented in [46] is an example which uses the pretrained resnet model (having been trained with Imagenet dataset) and which is then re-trained on human re-identification datasets [46].

Feature extraction: The outcome of object detection and tracking is a series of subject's image without much background. It is possible to apply a 3D CNN directly on the image sequence for a classification task. However the method is computationally expensive and may not produce a prediction accuracy that justifies the computational expense. A common approach to reducing computational time is by reducing the feature space via feature extraction. There are numerous feature extraction methods for video sequences. For example the 3-dimensional sift descriptor (3D-SIFT) [47], dense and scale-invariant spatio-temporal interest point detector (extended SURF) [48], the 3D histogram of gradient (HOG3D) [49] and local trinary pattern [50]. The most

impressive feature extraction which is state-of-the-art (at the time of writing this) for a number of video sequence classification benchmarks, is based on the dense trajectory based approach [51, 52] and the improved model of dense trajectory [53] which includes spacial-temporal based feature like HOG, histograms of optical flow (HOF) and motion boundary histograms (MBH). The feature computation is quite fast so that the method can be applied to real-time human action recognition as i.e. in [54, 55]. The calculation of these feature are based on a number of consecutive frames when the track's lengths of individual interest points are long enough.

Classification models: Classification models can be used right after suitable video preprocessing steps such as person detection, tracking, and feature extraction. Various options exist. For example, one can apply 3D CNN directly on the image sequence after the object tracking task. One can also apply time-series prediction models by using the coordinate data collected from the children's skeleton for each image. A problem however is that the recognition accuracy can be compromised because of a very high computational demand and because of confusions that arise out of inaccuracies in human re-identification.

Given that the time-series data is available for the classification task, the choices of recognition models can be various. One could choose to use traditional model such as simple recurrent neural networks. However a more recent method better suited to solve the time-series problem would be long short term memory (LSTM) [56], or Graph Neural networks [57], or CNN models [58]. These models might handle well the long term dependencies given the feature sequence is long for a single physical activity being performed [56].

2.2 Background knowledge on selected machine learning algorithms

This section will present several relevant machine learning models found in the literature. Here, the term ”relevance” is with respect to algorithms that either (a) have been deployed to PA recognition in young children, or (b) are considered in this thesis for the PA recognition task. We will distinguish between unsupervised and supervised learning paradigms. The mentioned methods will be evaluated, developed, and deployed later in this thesis.

2.2.1 Unsupervised machine learning algorithms

Unsupervised learning methods can be any kind of topology projection models or any clustering algorithms. Because one of the main focus of this research is to explore the intrinsic characteristics given the high dimensional and complex structural input space, Self-Organising Map (SOM) and DBScan are selected. The reason for this is these algorithms can be computationally very time efficient when exploiting the parallel capability nature of these algorithms and by implementing them on massive parallel (i.e. GPU) computing infrastructure. Other common clustering algorithms such as K-means, PCA, and others are less suitable because of unrealistic assumptions (i.e. K-means assumes that clusters are globular in shape) or limitations in preserving topological relationships among input attributes.

2.2.1.1 The Self-Organising Map

Teuvo Kohonen proposed the Self-Organising (feature) map, sometimes called a Kohonen map, 30 years ago [59]. The SOM and its many variations is one of the most successful and most widely used methods for dimension reduction and visualization. The SOM is a type of artificial neural network [59]. The SOM performs a projection of high-dimensional signal spaces onto low-dimensional display spaces, usually two-dimensional spaces. The

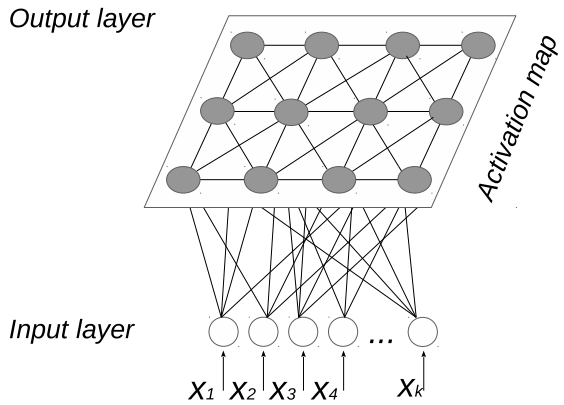


Figure 2.3: An example of a SOM model: The 12 neurons are organized on a two-dimensional display space.

two dimensional display space is parametrized by a two dimensional grid. At the intersection of the grid points, it is assumed that there is a vector of weights, which is of the same dimension as the input vectors. The main purpose is to enable these weights to approximate the training input, such that the vectors which are nearby each other in the high dimensional feature space will remain close in the low dimensional display space. Generally, the SOM is capable of preserving the topological properties of the input space [59].

The learning process encompasses two major steps: The competitive step and the co-operative step. A winner neuron is identified in the competitive step and a neighbor set of neurons is updated using a neighborhood function in the latter step. Formally, let an input vector be defined as $\mathbf{x} = [\xi_1, \xi_2, ..., \xi_n]^T \in \Re^n$ and the parametric real reference vectors or codebook vectors be defined as $\mathbf{m}_i = [\mu_{i1}, \mu_{i2}, ..., \mu_{in}]^T \in \Re^n$, $i = 1, 2, \ldots, N \times M$, where N and M are respectively the dimension of a two-dimensional grid. There is one codebook vector associated with each neuron in the feature map. T denotes Transpose. Note that the vectors $\mathbf{m}$ have the same dimensions as the input vectors. All vectors $\mathbf{m}$ may be initialized with arbitrary values. However, the probability density function $p(x)$ of the input data is often used as initial values for m_i. The number of output units is chosen by the user. The

method of normalizing $\mathbf{x}$ and $\mathbf{m}$ before their use in the algorithm may enhance the numerical accuracy as the reference vectors are implemented in the same dynamic range. However only the dot product used in measuring the similarity between two vectors is needed rather than requiring the input to be normalized. In finding the winning neuron, the best-matching unit (BMU) $\mathbf{m}_c$ has the maximum value in the matching criterion compared with other neurons. Many matching criteria could be used especially the Euclidean distance $d = \|\mathbf{x}-\mathbf{m}_i\|^2$ and the dot-product, $d = \mathbf{x}^T\mathbf{m}$. The BMU is determined as follows:

$$\mathbf{c} = \arg\min_i\{d(\mathbf{x}, \mathbf{m}_i)\} \tag{2.1}$$

where $\mathbf{c}$ is a two dimensional vector, denoting the location of the winning neuron.

In the cooperative phase, the elements in a neighborhood set N_i of node $\mathbf{m}_i$ are modified by using the neighborhood function $h_{ci}(t)$ where t denotes the iteration. The learning equation is as follows:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}(t)[\mathbf{x}(t) - \mathbf{m}_i(t)] \tag{2.2}$$

A widely used neighborhood function is the smooth Gaussian kernel function:

$$h_{ci} = \alpha(t)\exp\Big(-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}\Big) \tag{2.3}$$

where $\alpha(t)$ is a scalar learning-rate factor and the parameter $\sigma(t)$ represents the kernel size. $\alpha(t)$ and $\sigma(t)$ are both monotonically decreasing functions of time t. $\mathbf{r}_c$ and $\mathbf{r}_i$ are the location vectors of winning neuron $\mathbf{c}$ and a neuron i respectively.

The corresponding learning algorithm can be given as follows:

Step 1: Initialize the map node weight vectors.

Step 2: Select then present one input vector to the network.

Step 3: Calculate the distance between the given input vector and all weight vectors, then find the winning neuron that corresponds to the smallest distance to the input vector. This step identifies the BMU.

Step 4: All neurons in the neighborhood set of the BMU are updated by moving them closer to the input vector using Equation (2.2).

Step 5: Increment the time step then repeat Steps 2 through to Step 4 until a stopping criterion is met.

In order to increase the mapping effectiveness, the batch map algorithm can be run for several iterations first [59]. A few iterations of the K-mean algorithm can be effective in eliminating border effects in the two-dimensional map [59]. h_{ci} can be shrunk to a constant value when it is close to the convergence stage in order to achieve a better approximation of $p(x)$ [59].

Weight initialization: Beside the random method, weights can be initialized via linear initialization. This popular method utilizes the eigenvectors of a few of the largest eigenvalues which are calculated based on the autocorrelation matrix of the input space. The eigenvectors span the linear subspace that contains the centroid which is the mean of the rectangular or hexagonal lattice array (feature map). The size of feature map is then set the same as the two largest eigenvalues. $\mathbf{m}_i(0)$ are now ordered and their point density functions "loosely" approximates $p(x)$ [59].

Optimal learning rate factor: It has been suggested that $\alpha_i(t+1) = \frac{\alpha_i(t)}{1+h_{ci}\alpha_i(t)}$ or $\alpha(t) = \frac{A}{t+B}$ where A and B are some reasonable constants [59].

2.2.1.2 Density-based spatial clustering of applications with noise

Density-Based Spatial clustering of applications with noise (DBSCAN) is a data clustering algorithm [60]. The unsupervised mechanism allows DBSCAN to be used as a data preprocessing tool. Basically, DBSCAN can help to identify outliers in the input data space and

help to investigate the similarity between several data categories. A particular strength of DBSCAN is that, unlike many other popular clustering algorithms, it can find clusters of arbitrary size and shape. In particular, given a set of points in some space, it would group together points that are packed nearby in high-density regions (points with many nearby neighbors), marking as outliers that lie alone in low-density regions (points with very far away nearest neighbors). DBSCAN is one of the most common clustering algorithms and also is also most cited in scientific literature [60]. This density-based clustering and non-parametric algorithm can be formally presented as follows:

The input and models' parameters include:

1. There is a set of points D in some space that one wishes to cluster.
2. Let ϵ be a parameter specifying the radius of a neighborhood function.
3. Let $minPts$ be the minimum number of neighbors to identify the core point

With DBSCAN the data points are classified as core points, density reachable points, and outliers as follows:

1. A point p is classified as a core point if at least $minPts$ points lies within ϵ distance from p.
2. A point q is directly reachable from core point p if q is within distance ϵ from p. Points are only said to be directly reachable from core points.
3. A point q is reachable from p if there is a path $p_1, ..., p_n$ with $p_1 = p$ and $p_n = q$, where each p_{i+1} is directly reachable from p_i. Please note: all points on the path must be core points, with the possible exception of q.
4. All points not reachable from any other points are maked as outliers or noise points.
5. If p is a core point, then it forms a cluster together with all points (core or non-core) that are reachable from it.

6. Each cluster contains at least one core point; non-core points can be part of a cluster, but they form its "edge" (boundary points), since they cannot be used to reach more points.

The DBSCAN algorithm can be abstracted into the following steps:

- Step 1: Find the points in the ϵ neighborhood of every point, and marks points as *core* if they have more than $minPts$ neighbors.
- Step 2: Find the connected components of core points on the neighbor graph, ignoring all non-core points.
- Step 3: Assign each non-core point to a nearby cluster if the cluster is an ϵ neighbor, otherwise assign it to noise.

2.2.2 Supervised machine learning models

Supervised methods have been extensively studied and used in the machine learning literature. The MLP is well established example of a supervised algorithm [61, 62]. A vast number of algorithms have evolved on the basis of MLP over the years. Many recursive, recurrent, deep learning, graph neural, and convolutional neural network architectures use concepts and elements of the MLP. For example, a version capable of processing time sequence input data was presented in [63], which is denoted as Elman recurrent network. Several approaches dealing with structured data have been proposed such as the back-propagation through structure [64] and the extended cascade-correlation in [65]. More generic model for data structures is proposed in [66]. However, those models are restricted in processing acyclic and directed graphs. Some other extensions in addressing the cyclic and labeled-link graphs were introduced in [67]. The graph neural network, a recent generation of recursive neural network which can handle more general types of graphs such as cyclic, directed and

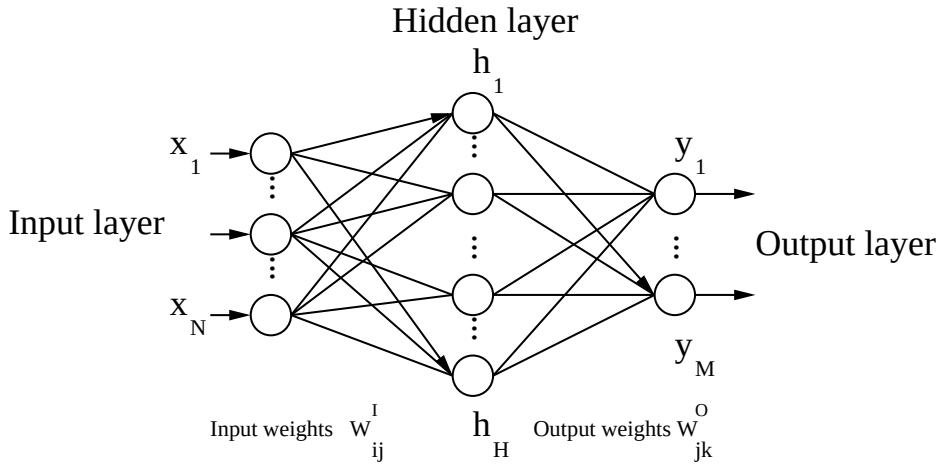


Figure 2.4: A common architecture of the MLP with three layers. N input neurons, H hidden neurons and M output neurons. Neurons are connected via weighted and directed links.

undirected graphs, was proposed in [68, 69, 70]. The followings will explain some well-known and relevant supervised neural processing prediction models.

2.2.2.1 Multilayer perceptron (MLP)

The MLP is a fully connected feedforward neural network [61, 62]. The main characteristic of the MLP is the activation function and the training mechanism called the error backpropagation. The activation function can be briefly described as follows: Given an input x as defined before, and a corresponding target t: $0 < t < 1$. When x is normalized to lie in [-1 1] then we want to observe that the most significant change of an output occurs in the neighborhood of $x \approx 0$. This property is applied in the input data preparation stage called squashing or scaling the input data. A widely used activation function which exhibits such property is the sigmoidal *activation* function:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.4}$$

We will consider the most common MLP architecture which features: One input layer

(N neurons), one hidden layer (H neurons) and one output layer (M neurons) as illustrated in Figure 2.4. The error or cost function is defined based on the Least mean squared error such that $E = \frac{1}{2}\sum_{k=1}^{M}(t_k - o_k)^2$ where, o is the vector of actual outputs and t is the vector of desired outputs. The activation function is associated with every neuron in the output layer and the hidden layers. W^I of dimension $N \times H$ is a weight matrix between the input and the hidden layers and W^O of dimension $H \times M$ is the weight matrix between the hidden and the output layer. Then, in the feedforward stage, the output of hidden units is computed as follows: $h_j = \sigma(\sum_{i=1}^{N} w_{ij}^I x_i)$. Similarly, the outputs of the output layer neurons is computed as follows: $o_k = \sigma(\sum_{j=1}^{H} w_{jk}^O h_j)$. The computation of an output for a given input as described here constitutes the *feedforward* step of the algorithm.

Learning through backpropagation mechanism: Learning occurs through the backpropagation stage in which the output of the network is compared with a given target value, and then all elements in the weight matrices are updated by using a gradient descent method with a given cost function. A common cost function is defined as $E = \frac{1}{2}\sqrt{(t-o)^2}$. The amount by which weights change can then be computed by $\Delta w = -\gamma \nabla_w(E)$ where γ is a learning rate and ∇ is the gradient term. In the case of using a sigmoidal function hidden layer neuron activation, we note that the corresponding derivative is $\sigma'(x) = \sigma(x)(1-\sigma(x))$. Computing the gradient using the error back propagation algorithm for the output layer, we have:

$$\frac{\partial E}{\partial w_{jk}^O} = (t_k - o_k)o_k(1 - o_k)h_j \tag{2.5}$$

The gradient in the hidden layer is given by:

$$\frac{\partial E}{\partial w_{ij}^I} = \sum_{k=1}^{M}(d_k - y_k)y_k(1 - y_k)\Big(\sum_{j=1}^{H} w_{jk}^O \frac{\partial h_j}{\partial w_{ij}^I}\Big) \tag{2.6}$$

where $\frac{\partial h_j}{\partial w_{ij}^I} = h_j(1 - h_j)x_i$. Weights are updated, guided by the learning rate, into the negative direction of the gradient.

The MLP has some known limitations:

1. The network requires the input samples to be normalized, therefore a pre-processing step should be taken into account for the effective functioning of the activation function. The value range of inputs needs thus to be know a-priori.

2. The data fed into the input layer is a vector, so that in the cases of the input data featuring sequences, tree structure or graph, more generic models are required.

3. The algorithm is prone to be trapped in a local minimum of the error function due to the complex landscape of the error surface.

2.2.2.2 The Graph neural network (GNN)

The GNN was first introduced in [68]. The method has been applied to a number of practical applications, for example to XML document and sentence classification [71, 72], to web page ranking and processing [73, 74, 75], to image recognition applications [76], and others. A comprehensive explanation of the GNN learning model and computational complexity is presented in [69, 70]. The GNN is considered one of the most generic models which can accept various types of input including vectors, sequences, and graphs. Sequences and graphs can be directed or undirected, ordered or unordered, edges can be labeled. Cyclic graphs can also be processed by this method.

The GNN consists of two network components: The encoding network and the output network. In the encoding network, consider the a node c of a given graph and its neighboring (connected) nodes ne, then x_c denotes the state of current node in the given graph, x_{ne} is the states of neighbors of x_c. Let l_c be the label of c, l_{ne} be the labels of ne. Linked-edge labels between c and a node u of ne is $l_{(c,u)}$. s is the dimension of the nodes' state. For non-positional GNN, the current node's state and the output o corresponding to each node

at time step t are calculated as follows:

$$\begin{aligned} x_c(t) &= \sum_{u \in ne} h_w(x_u(t), l_u, l_c, l_{(c,u)}) \\ o_c(t) &= g_w(x_c(t), l_c) \end{aligned} \tag{2.7}$$

where h_w and g_w are local transition and output functions, respectively. Note that the state x is computed by a dynamic system called the *encoding* network whereas the output o is computed by a feedforward network called the *output network*. The output network is an MLP which takes the state x and a node's feature vector l as input. Function h_w is introduced in order to make the GNN to be applicable to un-ordered graphs. For simplicity we can reduce the representation of 2.7 and 2.8 as follows:

$$\begin{aligned} x &= F_w(x, l) \\ o &= G_w(x, l_c) = G_w(F_w(x, l), l_c) \end{aligned} \tag{2.8}$$

Here F_w and G_w are global transition and output functions, respectively. l is stacked by all labels or current node, edge and neighbor node labels. However, note that x in Equation 2.8 the left hand side is not the same as x in the right hand side. At time step t, we compute the current state x_c of a node in the left of Equation 2.7, then in the next time step $t + 1$, that value of x_c would become x_{ne} in the right hand side, if at this time step we consider the activation of the c neighboring node. Because of the mutual dependencies between nodes, the state value x_c is iteratively calculated until a stable solution (called stable state) is obtained. The states are guaranteed to converge. The corresponding proof is provided in [68].

Target values may be associated with any node in an input graph. The GNN training algorithm uses an error function similar to that of the MLP to apply the gradient descent method for adjusting the internal weight parameters. The error function is calculated as

follows:

$$E_w = \sum_{i=1}^{p} \sum_{j=1}^{q} (t_{ij} - o_{ij})^2 \tag{2.9}$$

where p is the number of graphs in the dataset, q is the number of supervised nodes in a particular graph.

2.2.2.3 Support vector machine

Unlike neural networks, SVM is a non-parametric machine learning algorithm. The SVM is one of the most popular kernel methods [77]. The SVM algorithm is based on a supervised learning regime. The fundamental idea is that given a set of input feature vectors and associated class labels, SVM will construct a hyper-plane to separate the data in high-dimensional space into binary categories [77]. The basic form of SVM is defined as a non-probabilistic linear classifier. However, SVM can in practice efficiently perform a non-linear classification by the application of kernel functions. One of the most widely used kernel function is the radius basic function $k(x_i, x_j) = exp\Big(- \frac{\|x_i - x_j\|}{2\sigma^2}\Big)$, where σ denotes the kernel function parameter, x_i and x_j are two arbitrary input samples. The model would be formally defined as follow. Given a set of training examples $\mathcal{D}$ and corresponding class labels, $\mathcal{D} = \{(x_i, y_i)|x_i \in R^n, y_i \in \{-1, 1\}\}_{i=1}^N$. The output of SVM is defined as $y = \sum_{i=1}^N \alpha_i y_i K(x, x_i) + b$, where $K(.)$ denotes a kernel function. (x_i, y_i) is the i-th training sample and corresponding class label in N training inputs. If an unseen sample x is present, the output y of SVM is computed accordingly. The model parameters $\alpha = \{\alpha_i\}_{i=1}^N$ are learned by solving the optimization problem raised in the dual form:

$$\min_{\alpha_i} \Big(\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i y_i \alpha_j y_j K(x_i, x_j) \Big) \tag{2.10}$$

satisfying the constrains $\sum_{i=1}^N \alpha_i y_i = 0, 0 \leqslant \alpha_i \leqslant C, i = 1, ..., N$, where C denotes an upper bound for the soft margin of the optimal hyper-plane.

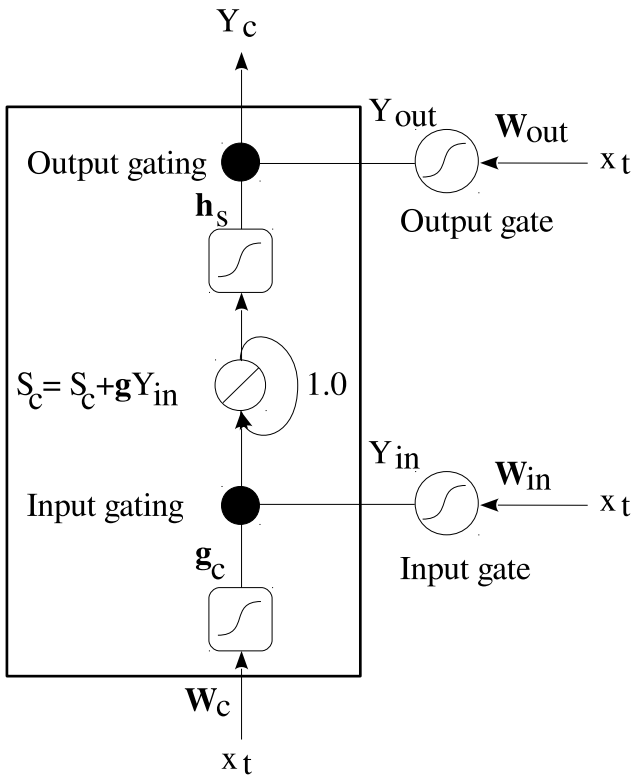


Figure 2.5: LSTM cell structure of the LSTM neural network model

2.2.2.4 Long Short Term Memory

The LSTM is a recurrent NN model which is effective in solving the long-term dependency problems [56]. The LSTM architecture contains special memory blocks located at the hidden layer. Each memory block may include one or more memory cells. The memory is built with a fixed self-connection. The model is learned by seeking an appropriate way to open and shut the input and output gates. For instance, the gate remains close if the model assesses the input information as not useful and vice versus.

Figure 2.5 illustrates one memory block with a single cell. The input x_t at time step t is given to each the input, output gate and the memory cell. The corresponding weights are W_{in}, W_{out}, W_c. The squashing function used in the input gate and output gate are sigmoidal $f = \frac{1}{1+e^{-x}}$. The squashing function at for input of the memory cell is the logistic sigmoidal function $g = \frac{4}{1+e^{-x-2}}$ and for the output of the memory cell it is a centered sigmoid $g =$

$\frac{2}{1+e^{-x-1}}$. Lets denote Y_{in}, Y_{out}, Y_c to be respectively the outcome of the input, output gates, and the memory cell. We can then define formally:

$$Y_{in} = f(\sum W_{in} \times x_t)$$
$$Y_{out} = f(\sum W_{out} \times x_t)$$
$$S_c = S_c + Y_{in} \times g(\sum W_c x_t)$$
$$Y_c = Y_{out} \times h(S_c)$$

A major problem with gradient descent for standard recurrent NNs is that the error gradients vanish exponentially quickly with the size of the time lag between important events [56]. With the LSTM memory blocks, however, when the error values are back-propagated from the output, the error becomes trapped in the memory portion of the block. This is referred to as an "error carousel", which continuously feeds error back to each of the gates until they become trained to cut off the value. Thus, regular back-propagation become more effective by training LSTM blocks to remember values for a longer duration.

This method has been very successful in solving the long-term dependency problem thus contributing to advancements in a range of pattern recognition learning problems such as in speech signal recognition, handwriting recognition, and general time-series problems [56]. The LSTM can handle learning problems with considerable long term dependencies by utilizing the special memory unit located at the hidden layer. Each memory cell is built with a fixed self-connection. The truncated update rule is as follows: Error signal is trapped in the cell and cannot be changed. The output gate of the memory cell has to learn which error to trap by properly scaling them. Meanwhile, the input gate learns when to release the error, again by a scaling method. Then the error is truncated once it is allowed to leave the memory cell. The design of such memory units allows the gradient of an error function to freely back-propagate through the network with possibly infinite duration. More recent development of LSTM is deep temporal LSTM [78] which contains a finite number of hidden

layers.

2.2.2.5 Fully Recursive Perceptron Network (FRPN)

The FRPN is a recent generation of multilayer perceptron (MLP) [79]. FRPN is an effective alternative to MLPs that feature a large number of hidden layers such as those found in deep neural networks. Basically, the FRPN consists of an input layer, an output layer, and one *pool* of hidden neurons. The unique feature lies in the composition of the hidden neurons. The hidden neurons are fully connected with algebraic (instantaneous) connections. The FRPN thus eliminates the need of multiple hidden layers and hence eliminates the need of identifying the optimal number of hidden layers as well as the number of neurons for each hidden layer for a given learning problem [79].

Training an FRPN model is similar to the case of MLPs. In FRPN, however the learning mechanism is performed in a recursive manner. The mechanism is similar to the encoding network of the GNN except that the input to the FRPN remains static until a stable state is reached. Given the number of hidden neurons, we denote as a fully connected pool of neurons, each neuron in the pool computes its own weights using the weighted inputs from the input layer and from all neurons in the pool. The FRPN training algorithm uses a gradient descent method in two steps: The forward step which is analog to the feedforward step of the encoding network in the GNN and the backward back-propagation step for computing the weight updates. In the forward step, the outputs of the pool of neurons are applied repeatedly until they converge to a stable state or until a maximum number of recursions is reached. In the backward step, the weights are updated based on the gradients of an error function with respect to the weights. The training procedure is progressed repeatedly for each training sample or for each batch of training samples and for a pre-defined number of training iterations [79]. An advantage of the FRPN is that it can simulate deep neural networks of arbitrary depths without requiring the specification of the depth.

2.2.2.6 Deep learning and Convolution neural networks

Research in Deep Learning and Convolutional Neural Networks (CNN) is rapidly evolving. This section presents a snapshot of relevant and current (as of date of thesis commencement) literature on deep learning models learning and CNN. The focus will be on algorithms that accept vectorial input. The earliest exploration of deep leaning originated from MLPs. Lecun proposed to use multiple hidden layer MLP with very large hidden layer size [80]. Lecun's work then led to the introduction of CNNs [81, 82]. Both of the models were successfully applied to digit and image recognition problems. Then Hinton and Bengio introduced Deep belief networks (DBN) by using either the Restricted Boltzmann Machines (RBM) [83] or Auto-Encoders [84] as the unsupervised pre-training layers, followed by a fully connected MLP for the supervised learning stage. More recently, and inspired by the biological reaction on the visual area V2 of human brain cortex, Andrew Ng proposed Sparse DBN for feature learning from images [85]. A more detailed overview and analysis of deep learning architectures can be found in [86, 87, 58, 88]

In practice, a CNN or ConvNet is an important class of deep neural networks. Its applications are mostly focused on image and video perception. The key characteristics of CNN include shared-weight architecture and translation invariance. The weight sharing and sparsity nature of the network architecture helps the model to learn high dimensional input effectively [58, 89]. Hence, given a high dimensional input of color value pixels in an input image, the number of learnable parameters is significantly smaller when compared with the fully connected and traditional multi-layer perceptron networks if provided with the same number of network layers.

In addition, CNN learns the filters that in traditional algorithms needed to be hand-engineered. The major components of CNN include different types of learning layers:

1. The convolutional layer is the most important part. The layer runs a convolution operation to the input and then passes the result to the next layer. The inspiration of the

convolution layers is to imitate the actual neuron's response to a visual signal. Each convolutional neuron only processes data for its own receptive field. Learning this ways makes the network more sparse. For this reason, CNN can resolve the vanishing or exploding gradients problem that can be encountered when training traditional multi-layer neural networks with many layers by using backpropagation.

2. The local or global pooling layer is another feature of the CNN model. The pooling layer down-samples the outputs of neuron groups from one layer to a single neuron in the next layer. For example, max pooling layers use the maximum outcome value of neuron groups at the prior layer. In addition to max-pooling, average pooling layers use the average value of neurons in the cluster at the prior layer.

3. The fully connected layers. These are exactly same as the traditional MLP layers. It should be noted that in a fully connected layers, each neuron receives input from every element of the previous layer. However, in a convolutional layer, neurons only receive input from a restricted sub-area of the previous layer. The sub-area is basically of a square shape, and is called the receptive field. In the other words, in a fully connected layer, the receptive field is the entire previous layer whereas in a convolutional layer, the receptive area is smaller than the entire previous layer.

Weights and bias: A neuron in CNNs computes an output value by applying an activation function to the input values coming from the receptive field in the previous layer. The activation function is specified by a vector of weights and a bias value which is usually a real number. Learning in a neural network is a procedure to make incremental adjustments to the biases and weights. The vector of weights and the bias are called a filter. The filter, similar to encounter filter in image processing, helps to identify some feature of the input e.g., a particular edge, arc or shape. In CNN architecture, the same filter is shared between many neurons. This reduces the memory footprint since a single bias and a filter is used

across all receptive fields, rather than each receptive field having its own bias and vector of weights.

2.2.2.7 Yolo network

Yolo (You only look once), is a typical type of deep neural network for the purpose of object detection and classification [33]. Object detection requires determining the location of certain objects on a given image and identifying the class labels (i.e. object category) of those objects as well. There are a wide range of deep learning algorithms used for object detection and classification such as region based CNN (R-CNN) and its variations [90, 34], deep residual neural network [35], google inception network [91], densely connected deep neural network [92] and many others, which go beyond the scope of this thesis. Interested readers should be referred to comprehensive studies on applying deep learning in object detection and recognition [93, 89].

This section is limited to describe one representative of deep learning algorithm for object detection and classification, Yolo. To put it simple, one can take an image as input, pass it through the network and ultimately one can get a vector of bounding boxes and class predictions in the output. The input image is divided into an $S \times S$ grid of cells. Each grid cell predicts B bounding boxes and C class probabilities. The bounding box prediction contains the coordinates representing the center of the box and the confidence score. The confidence score reflects the probability that the object is presence or absence on the image. Given an object present on the image, a grid cell is said to be responsible for predicting the object if the center of the object falls into that grid cell. Since Yolo considers all the grid cells and *looks* at the entire image once when making predictions (hence the name of the algorithm), the model implicitly encodes contextual information about object and hence is less likely to predict false positives on background.

Loss function: The Yolo loss function contains 4 different components:

1. An element which computes the loss related to the predicted bounding box position.
2. An element which represents the loss related to the predicted width and height of the box. The error metric reflects that small deviations in large boxes matter less than in small boxes. To address this the square root value of the width and height of the bounding box is used instead of the width and height directly.
3. A component that computes the loss associated with the confidence score for each bounding box predictor.
4. A component that is similar to a normal sum-squared error for classification.

Thus, the first three components compute the loss for the regression task to detect the right object on the image while the final component is for the purpose of identifying which class label the object belongs to.

2.3 Conclusion

This chapter has given a brief review of approaches to children physical activity recognition based on two respective types of input data, namely the wearable accelerometer sensors and video sequences. For each approach, we have listed a number of relevant work in literature and pointed out some limitations. Advantages and disadvantages for these problems have also been presented. Additionally, several well-known machine learning algorithms that will be used later in this thesis, have been described in some detail. Special attention was given on the unsupervised learning models such as SOM and DBSCAN since one of the main aim of this thesis is to develop a visualization tool for data analysis and data knowledge expression. Supervised models including MLP, SVM, LSTM and deep learning algorithms have also been explained since the aim of the thesis is to classify samples into activity classes. Moreover, the deep learning models can be utilized as a regression model for object detection or used for the classification task and is thus relevant to this thesis.

Chapter 3

Description of the Physical Activity Datasets

3.1 Introduction

This chapter describe the three physical activity datasets which were collected in the course of this research, either by our collaborative partners, or by ourselves. These datasets were created to support research in metabolic consumption rate of humans, engaged in physical activities, and they were collected under controlled practical situations. Therefore, they are far from the well curated datasets which one often encounters in machine learning benchmarks. The nature of these datasets dictates to a large extent the approaches which we will deploy for their processing, in order to answer the question: can we classify such datasets into different categories of physical activities. By categorizing these activities into different categories together with the time duration in which the physical activity was conducted by a participant, sports physiologists will be able to use the information, to compute the metabolic consumption rate involved in performing a particular physical activity type.

Three sets of physical activity data were collected, one involving adolescents and preschool children, (PA2012 dataset), while the other two involved only preschoolers (PA2014 dataset

and PA2016 dataset). The physical activities conducted during these trials were different in each trial, and they could be different within a trial, as they were conducted over different dates. While conducting the trial on preschool children, because of their age, it was quite difficult to ensure that they will perform the designated category of physical activities, for example, it was observed that a child, in the middle of a trial on 'cycling' got off the bike, ran around, then went back onto the cycle to continue. The whole sequence of which would, however, still be labelled as "cycling".

Owing to the imperfect conditions under which the data was collected, that is the reason we decided to introduce the dataset collected first, before discussing the problem which we wish to solve in this thesis, as the nature of such datasets dictate to a large extent the type of information which may be extracted, pertinent to solving the classification problem, and more specifically, they dictate on the types of approaches which would lead their resolutions.

3.1.1 Accelerometer Data from a Wearable Device

An accelerometer is an electromechanical unit used to measure acceleration forces which might be static such as the continuous gravity force (G-force) or be dynamic to measure changes in movements or vibrations of an attached object. Acceleration is the measure of how large is the change in velocity, or in other words, the change in speed within an amount of time. For example, a car accelerating from a stopping point to a speed of 60 mph in six seconds will be calculated as an acceleration of 10 mph per second (60 divided by 6).

Accelerometers are used widely and in a multitude of disciplines. For example, accelerometers are used in portable computing devices like laptops to protect hard drives from damage. If the laptop is suddenly dropped while being in operation, the accelerometer will detect the free fall and will temporarily park the sensitive read-write head(s) of the hard drive to prevent them from hitting the hard drive platter(s). In another example, accelerometers are used in cars that would help in car crash situations. They can be used to detect a crash

and to instantaneously relieve airbags.

In essence, a dynamic accelerometer measures G-force to determine the angle at which the wearable device is tilted with respect to the Earth. Given the acceleration information, users may know how their wearable device is moving, such as moving uphill, falling over or tilting at any angle. For example, smartphones can rotate their display to meet the user viewing angles based on their 3D position [94].

The principal functionality of an accelerator is as follows. Most commonly the accelerator's circuit operate either on the piezoelectric effect or via capacitance sensors. The piezoelectric effect is the most common form that uses microscopic crystal structures which are more or less stressed due to accelerative forces. These crystals result in a voltage from the stress, and the accelerometer's circuit interprets the voltage to determine velocity and orientation. Another common technique is via capacitance sensors which sense changes between microstructures located next to the device's circuitry. If an accelerative force moves one of these structures, the capacitance will change and the accelerometer will translate the changes to voltage for interpretation [94]. Typical accelerometers measure G-forces for each of three axes in which the first two are to determine two-dimensional changes in movement and the third to determine the 3D positioning.

Figure 3.1 shows some examples of 3D accelerometer data collected from an accelerometer attached on a kid's body part when performing several physical activities. Smartphones commonly make use of three-axis models, whereas cars use only a two-axis to determine the moment of impact. The sensitivity of an accelerometer is quite high since it is intended to measure very moment shifts in acceleration. The more sensitive the accelerometer, the more easily it can measure acceleration.

Accelerometers, while actively used in many electronics in todays world, are also available for use in physical health disciplines as they will be used in this research. This thesis considers accelerometers that have been integrated in a lightweight wearable device such as

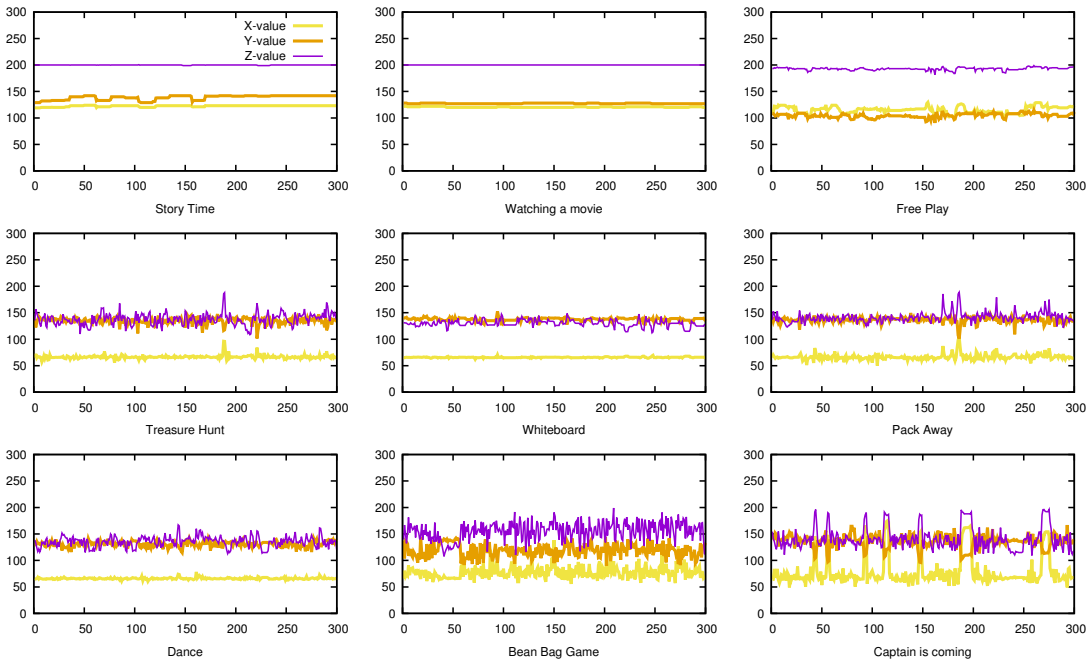


Figure 3.1: Some examples of 3D accelerometer data.

those that can be attached to a wristband or a belt clip. The 3-axial accelerometer ActiGraph GT3X+ is a light weight, low cost, precision instrument which has been used by us for data collection. The ActiGraph GT3X+ can sample G-forces at 100Hz and has been deployed as will be described in the following subsections.

3.1.2 School children and Adolescence data (PA2012 data)

This dataset was created by our research partner Professor Stewart Trost at Queensland University of Technology (QUT) in 2012. The dataset contains observations from 100 children and adolescents in the age group of 5 to 15 years old [5]. The accelerometer used in collecting this data is set at 30Hz sampling rate and positioned at the waist of participants using flexible elastic belts. Each participant performed 12 activity trials including lying down, handwriting, laundry task, throw and catch, comfortable overground walk, aerobic dance, computer game, floor sweeping, brisk overground walk, basketball, overground run/jog, and brisk treadmill walk. Each activity trial lasted 5 minutes, except for the lying down

Table 3.1: Activity classes in the PA2012 dataset.

ID	PA type class	Activity inclusion
1	Sedentary	Lying down and handwriting
2	Light activities	computer game, floor sweeping, laundry task, and throw and catch
3	Moderate-to-vigorous	Aerobic dance and basketball
4	Walking	Comfortable overground walk, brisk overground walk, and brisk treadmill walk
5	Running	Running or Jogging

trial, which was completed in 10 minutes. Based on the movement pattern and the amount of energy expenditure, these activities are categorized into 5 classes: (1) sedentary, (2) light activities, (3) moderate, (4) walking, and (5) running, as shown in Table 3.1.

All accelerometer data can be modelled as time series or temporal sequences. This then becomes a sequence to label classification problem, since each activity is composed of a series of time-step based acceleration information. It can be assumed that the information at a current time-step is influenced by the information happened in the past.

3.1.3 Preschool children physical activity cohort 2014 (PA2014 data)

This dataset was created as part of a feasibility study in 2014. Eleven children aged 3-6 years were recruited to participate in the study [25]. Data collection was performed by our project team at the University of Wollongong. Parent consent was obtained prior to participation by the child.

Participants were requested to complete 12 protocol activity trials over two laboratory visits scheduled within a 3 week period. The 12 activities performed by the children are slightly different in both visits, but for analysis purposes, they are grouped into five groups, the grouping of the activity into the same group is based on the approximate equivalence of the estimated energy expenditure by the child in performing the activity.

Children performed the following six activity trials during the **first** visit: Watching TV

Table 3.2: Activity classes in the PA2014 dataset.

ID	PA type class	Activity inclusion
1	Sedentary	Watching TV, Story time, Playing iPad game, Quiet play
2	Light lifestyle activities	Treasure hunt, Cleaning up, Collage
3	Moderate-to-vigorous	Obstacle course, Bean bags, Riding bicycle or tricycles
4	Walking	Walking
5	Running	Running

(TV), sitting on the floor being read to (reading), standing making a collage on a wall (art), walking (walking), playing an active game against an instructor (active game), and completing an obstacle course (obstacle course). The following six activity trials were performed during the **second** visit and by the same participants: Sitting on a chair playing a computer tablet game (tablet), sitting on the floor playing quietly with toys (quiet play), treasure hunt (treasure hunt), cleaning up toys (clean up), bicycle riding (bicycle), and running (running). Each trial lasted 4-5 min. These 12 activities were then grouped into five activity classes, the same five classes as those in the 2012 dataset for consistency through the activities covered by each of the five classes differed as is shown in Table 3.2. The main purpose of class division is that the PA activities of more or less equivalence in the amount of energy expended are considered to belong to the same group, while running and walking ones are distinct in terms of energy expenditure and are thus separated to two different classes. This in turn means that the class ”sedentary” contains four times as many samples as each of the classes ”walking” and ”running”.

Participants were equipped with ActiGraph GT3X+ accelerometers on three different body locations, hip, left wrist, and right wrist. The acceleration output is recorded with the user-specified sampling frequency of 100 Hz. Those sensors measured and stored triaxial acceleration of those body locations. As a result, there are three datasets extracted from those accelerometers, denoted as *Hip*, *Lwr* and *Rwr* data corresponding to the data collected by the accelerometer mounted to the hip, left wrist, and right wrist respectively.

Energy expenditure (EE) was also recorded via a room based calorimetry system and

a face mask. Two EE measured were the metabolic equivalents (METs) and activity energy expenditure (Kcal/kg/min) denoted as AEE. The face mask attached to the children's mouth measured the difference between inhaled and exhaled CO_2. The two real values of METs and AEEs were computed from those measurements accordingly. Though energy expenditure prediction will not be a focus of this thesis.

3.1.4 Video sequences captured during the PA2014 trials

In addition to the data collected via acceleration sensors, video recordings were taken during the 2014 trial. A tripod mounted video camcorder recorded the activity trials using 512×512 image resolution at 25 frames per second. The time to start the activity trials and the time displayed on the video were noted, hence one can use the video sequences for the purpose of data preparation and validation. In particular, one can examine: (1) if there is a missing bit of time when the child was doing something else rather than doing the assigned activity trial; (2) if the input signal from sensors was consistent (with the designated activity trial type) and with no loss of information; (3) if there is some unwanted inferences during the experiments. The camera is mounted on a tripod near the middle of the laboratory. The camera was left unattended most of the time but was rotated occasionally to follow the movement direction of the participant doing the PA trial. This is thus not a fixed mounted camera hence it raises some challenges for image processing, object detection, object tracking and recognition.

In total, there is approximately 5 (minutes) $\times$ 12 (activity trials) $\times$ 11 (participants) = 660 minutes (or 11 hours) of video recorded. The video sequences contain segments in which there is no presence of the performing subject (i.e. the subject left the field of view). Such segments will be removed as part of a video pre-processing step. Cutting and pasting of video sequences and labelling was performed manually in order to obtain a single video file for each individual activity trial.

The original purpose of the video recording is to serve as "evidence" of compliance

with procedures as approved by the Ethics Committee on Experiments Involving Human Subjects. Nevertheless, in this thesis we use the videos for validating the accelerometer recordings. This is because a child even if one suggests what the designated activity type is, say, cleaning up toys, a child might instead play with the toys for sometime whilst walking between placing the toys into a designated area, before recommencing cleaning up. From the visual evidence captured by the videos we know that it was common that a child deviated from a designated activity during a trial. This correspondingly affects the data quality and would make it difficult to differentiate activity classes from the accelerometer measurements. However, such episodes would be readily differentiable from the video recordings. Therefore, the video recordings are considered in this thesis as an alternate source of information in order to investigate, via a comparison of results, the limiting effects of the noise in the accelerometer data. We find that while using the video recording in this role, it is a simple step to edit the video recordings as well, so that they are approximately consistent with the accelerometer measurements; it is approximate because manually it is very difficult to align the two different modalities of recordings precisely, as they are being sampled at different sampling rates. A curios question arose: what if we use video processing techniques, like object tracking, object recognition algorithms on the edited videos, what might be their classification accuracies, when compared with those in the accelerometer measurements. In other words, the two modes of measurements are not meant to constitute basis of multiple modality fusion, because of the ways in which the accelerometer measurements and the video recordings were edited, would render this exercise futile. Had fusion been intended we would have to have the camera following and tracking the child continually, rather than intermittently.

3.1.5 The Preschool PA cohort 2016 (PA2016 data)

This dataset was created by our research group at UoW in 2016. 16 children aged 3-6 years were recruited to participate in this study. Actigraphs and GeneActivs (the accelerometers used) were mounted in the same three locations of the child, viz., hip, and both wrists, as in the 2014 study.

The datasets thus also contain triaxial accelerometer data. Nine physical activities were performed: (1) Lying down; (2) Toys at table (free play); (3) Story time; (4) Whiteboard; (5) Treasure hunt; (6) Pack Away; (7) Dance; (8) Bean Bag Game; (9) Captain is coming.

For this dataset the physical activities were not grouped into different PA classes. This would prevent comparisons of the performances of the proposed machine learning algorithms across the three datasets. A solution to this issue will be presented later in Chapter 7.

Video recordings of the trials are made under similar conditions to those described in the 2014 study.

3.2 Processing Problems Encountered in the Collected Data

These three datasets are collected for the purpose of supporting metabolic consumption, and energy expenditure of a participant performing some specified physical activities. They are not intended for bench-marking machine learning algorithms. Therefore, there are a number of problems faced when using machine learning types of algorithms to process the data collected. These problems are described as follows:

1. The secondary role played by the video recordings. As indicated, video recordings are obtained as a way to validate the label placed manually on a particular accelerometer recording. It is not designed for conducting multi-modal fusion possibilities so as to enhance the classification accuracies of either modality. Therefore, in this thesis, we will consider these two recordings as separate modality recordings and no attempt

would be made to fuse them together.

2. In the 2012 study, it involves both adolescents, and pre-school children, while the 2014 study and 2016 study only pre-school children are involved. Therefore some caution would need to be exercised in comparing the results across the three datasets.

3. In the 2012 study only one accelerometer was involved, while in both the 2014 study and the 2016 study three accelerometers were deployed. In other words, in the 2012 study, only one accelerometer recording is available while in the 2014 and 2016 studies, three accelerometer recordings are available. So some care would need to be exercised in interpreting the results obtained by a machine learning algorithm.

4. As indicated above, the accelerometer recordings are labelled according to the trial rather than by the activity actually performed by the child. This introduces considerable noise. Data cleaning was not engaged. This thesis will instead investigate the robustness of PA prediction methods to such noise.

5. In both the 2014 and 2016 studies only few participants were involved in the trial; 11 and 16 respectively for the 2014 and 2016 studies. While for the estimation of energy expenditure and metabolic rates, this would not cause any issues, but for supervised machine learning approaches, such small numbers would cause severe challenges for cross dataset comparisons.

6. The datasets are unbalanced. Some classes such as the class ”sedentary” in the PA2014 dataset are five times larger than the smallest classes such as the classes ”running” and ”walking”. This thesis will thus investigate the suitability of PA prediction methods to model unbalanced data.

7. As the types of activities vary over the three datasets: 12 in the 2012 study, 6 in the first visit and another 6 in the second visit for the 2014 study, and 9 for the 2016

study. The activity type varies in each of these four recordings. For convenience the two visits in the 2014 study are grouped as one visit, i.e., 12 types of physical activities for that study. Moreover, for convenience sake, these 12 types are grouped into five categories. For estimation of the energy expenditure, and metabolic rates, such variations in each trial would not make much impact, say, in the 2014 moderate energy expenditure category: obstacle course, and riding a bicycle or a tricycle, but such actions would make it quite different to being in the same category in the video recordings. Therefore, some care must be exercised in interpreting the results related to the video classifications.

These issues largely dictate the type of machine learning algorithms which we may use to process the data. Moreover, as we will introduce some new machine learning algorithms which have not been tried on such datasets previously, we need to use some benchmark datasets to validate the proposed machine learning algorithms first, before applying them to process these three datasets. The description of such benchmark datasets would best be described after we have provided an idea of the type of approaches which we will use to process these three datasets.

Chapter 4

Problem description

4.1 Introduction

As indicated in the previous chapter, the datasets collected during those three studies, viz., PA2012, PA2014, and PA2016 respectively can be considered as two separate modalities: accelerometer recordings and video recordings. As these are considered as two separate modalities, the processing of the data would be different. Therefore, we need to formulate two different problems, one for each modality to solve.

4.2 Physical activity recognition problem using accelerometer recordings

Given that we have a set of accelerometer recordings, either as recordings from one accelerometer, or from three accelerometers, each with an associated label (the training set), is it possible to classify them into distinct categories. Moreover, is it possible to predict the labels in a testing set which consists of accelerometer recordings but which are assumed to be without any associated labels?

When formulated in this manner, this defines a classification problem. Since these data sequences are not well-known, indeed we are probably the first one in the world who considered this classification problem, involving these three datasets, therefore the nature of these datasets are not known. We propose first to familiarize ourselves with the datasets, through visualization of the datasets, on a two dimensional display space. We propose to use HRSOM (high resolution self organizing map) to do so, as it is known that the SOM, proposed by Kohoene originally is capable of projecting from a high dimensional feature space to a two dimensional display space, with the property that any two feature vectors close to one another in the high dimensional feature space will remain close in the two dimensional display space. By being able to visualize the relationships among the features in different categories with itself (other samples in the same categories) and those samples in other categories, in the high dimensional feature space on the display space, it will inform us of those relationships. As those relationships among the feature vectors could be quite intricate, therefore, we will need the display space to be of sufficient resolution so as to display such intricate details. This is why we will need to use a HRSOM, instead of one which is lower resolution, or insufficient resolution to display those intricate relationships.

In this process of discovery we found that such visual information can be incorporated into the features, to serve as an aid in the classification scheme (for details please see the Chapter 6). We then tackle the classification problem when the dataset is severely unbalanced. For this we propose to use a new machine learning algorithm which is called SSEN (Synthetic Sampling Ensemble Network).

Then we tackle the problem of how to make use of information (learned knowledge) in a dataset, e.g., the PA2014 dataset, and apply it to a different dataset, e.g., the PA2016 dataset. Therefore we will use transfer learning techniques which can transfer learned knowledge from one dataset to another. This will be reported in Chapter 7.

4.3 Problem Formulation for Video Recordings

Given that we have video recordings concerning different subjects performing different tasks (the training set) the problem is that if the trained model can predict the class labels of videos in a testing set which consists of videos only, without any labels. Unlike the accelerometer situation, the videos have been edited, to remove segments which are inconsistent with the label on the video. Therefore the results we will obtain will be a theoretical maximum, as in practice the videos in the testing set would not be edited. The problem as posed is a standard video classification problem, like in human activity recognition problems.

In this case, as we are dealing with videos, there is no need for us to learn about the characteristics of the videos, as they are readily visible. Secondly, as we are dealing with a standard human activity classification problem, we decided to use the "best of breed" at the time with some modifications when we processed such datasets. The details of this will be contained in Chapter 8. A major issue of this would be the video classification algorithm would be subjected to the vagary of the "best of breed" approach at the time, and so the approach used would not be the latest approach in the field. We justify the deployment of the "best of breed" at that time of processing the video data by the fact that the video recordings were not of any use except that they serve a secondary role in validating the accelerometer recordings. It is just our curiosity to see if processing such information using the "best of breed" video classification algorithms at the time to see if anything useful might come out of such an exercise. In other words, the main focus of this thesis is the possibility of classifying accelerometer recordings of physical activities into categories, and not on video classification algorithms. Therefore, we can just use some "best of breed" video classification algorithms at the time when the processing was performed. As it happens, to our surprise, the results are quite comparable to those obtained with the accelerometer recordings. This is somewhat unexpected, but then this serves as a confirmatory result for the ways in which we deploy the proposed SSEN and transfer learning techniques. Details

of this discussion will be elaborated in Chapter 9.

4.4 More datasets for model validation

Since one of the research objectives is to develop and test novel classification models using the accelerometer recordings, the use of other benchmark datasets for model evaluation is considered. A set of well studied and well understood datasets is selected. Datasets were chosen which either (a) share certain properties with the PA datasets to be deemed relevant for the purposes of the study in this thesis, or (b) feature some properties which are not present in an ideal PA dataset but which would be encountered in physical activity observations in an an environment which is beyond control. This situation is especially rampant in wishing young children to do as requested. For example, the PA data were collected as part of a controlled environment. The occurrence and duration for each activity is therefore designed to be balanced. But uncontrolled (free play) settings would perform some activities much more frequently (i.e. playing games) than other activities (such as household tasks). This means for popular activity types, like game playing, there would be much longer recordings than the activity type, like, household chores, as children, as can be understood, would be less likely to be performing those designated activity which would not provide them with an incentive to perform, during the trial, and hence this would result in a shorter recording. This would create issues of disparate length recordings of different activity types. To analyse robustness of the proposed machine learning algorithms on unbalanced data we thus select suitable benchmark datasets which would have similar characteristics.

As we do not know the characteristics of the collected PA datasets, as a result we propose to use a HRSOM to help us explore those characteristics. Therefore, we need to choose a HRSOM of sufficient resolution which would display some of the intricacies which might be present in the PA datasets. Therefore we have chosen a benchmark dataset. viz., the

policemen dataset. which would have sufficient degree of intricacy (indeed the intricacy of the policemen dataset can be adjustable to arbitrarily fine precision) to give an idea of the likely resolution of the HRSOM to be used to study the PA datasets.

Other datasets selected either to feature severely imbalanced in terms of class distribution, or where the sample size is much larger in order to verify robustness and scalability of the classification models.

Hence a number of artificial and real-world challenging datasets will be used to ensure that the proposed classification models would perform in a robust fashion in the PA classification tasks including free-play situations.

We have used the following datasets: (1) the policemen dataset, apart from being used for the purpose of finding a SOM with sufficiently high resolution to visualize the intricate patterns in the physical activity datasets, it can also be used in the evaluation of the proposed supervised learning algorithm, SSEN; (2) web spam detection datasets: UK2006 and UK2007, and (3) an intrusion detection dataset, UNSW-NB15 dataset. (2) and (3) are deployed to evaluate mainly the capabilities of the proposed supervised learning algorithm, called SSEN, in particular of its ability to handle unbalanced datasets.

4.4.1 The Policement dataset

The policemen dataset is an artificial dataset which is used for the purpose of assessing visualization and clustering capabilities of machine learning algorithms [95]. The dataset consists of an arbitrarily large database of images that are produced via a given attributed plex grammar [95]. The dataset contains three categories of images: policeman, house, and sailing boat. Each category of images contains a number of sub-classes dependent on the specific features associated with the image. For example, a policemen image contains a hat, a head, a torso, two arms and two legs. All policemen whose left hand is raised belong to one sub-class whereas all other policemen belong to another sub-class. Similarly,

Table 4.1: The distribution of the 12 classes in the policemen dataset.

Cls.ID	Description	Symbol	Sample
1	Policemen with raised left arm	+	2,520
2	Policemen with lowered left arm	×	2,480
3	Ships featuring two masts	▼	3,767
4	Ships with three masts	◇	1,233
5	Houses without windows	✳	84
6	Houses with 1 window in (LL) corner	□	227
7	Houses with 1 window in (UR) corner	■	251
8	Houses with 1 window in (UL) corner	○	241
9	Houses with 2 windows in (LL) and (UL)	●	658
10	Houses with 2 windows in (UL)) and (UR)	△	726
11	Houses with 2 windows in (LL) and (UR)	▲	663
12	Houses with 3 windows	▽	2,150

(LL) = lower left; (UR) = upper right; (UL) = upper left

a house can have a roof, a chimney, a door and several windows. Sub-classes of houses are defined depending on the number of visible windows. Thus, for example, a house with two windows is considered to be in a different sub-class from that of a house with three windows. Similarly for the sailing boats where sub-classes are defined depending on the number of masts.

Each generated image is encoded into a feature input vector which is the concatenation of the center of gravity coordinates of the image's parts. For this thesis, a dataset containing 15,000 artificial images (5,000 for each of the three categories), which are then described by 15,000 corresponding feature vectors. The maximum number of parts in an image is 14. Since the center of gravity is a two-dimensional coordinate value and hence the input data dimension will be $2 \times 14 = 28$. For images having fewer than 14 parts, the corresponding input vector is padded with the value zero.

The distribution of the 12 classes in the dataset is shown in Table 4.1. It can be observed that the distribution of the data classes is considerably unbalanced. The largest class is more than 30 times larger than the smallest one.

Table 4.2: Statistical information on the two Webspam data sets.

Properties	UK2006	UK2007
Number of samples	11,402	114,529
Unlabeled samples	3,929	108,050
Labeled samples	7,473	6,479
Training set	5,622	4,275
Test set	1,851	2,204
Number of hyperlinks	730,774	1,885,820

4.4.2 Web spam detection problems

The Webspam datasets UK2006 [96] and UK2007 [97] are severely unbalanced *real world* datasets with an input dimension larger than that of the policemen dataset. Web spam detection problems were provided for advanced research on detecting the spam websites. The spam and normal are basically two main categories of web-pages. There also exists a number of unknown/unlabeled pages, for which it is uncertain if should be classified as spam or normal. It is interesting to note that the category distributions of these datasets are severely unbalanced. In particular, the number of spam class samples is 10 to 15 times smaller than that of normal class samples. General properties of these datasets are summarized in Table 4.2.

It can be observed that the Webspam dataset contains unlabelled samples. Unlabelled samples correspond to samples which are not known whether they are normal or spam. The dimension of the input feature vectors for these two datasets is 137, consisting of 96 features that describe the content of a web page and 41 hyperlink-based features [96, 97].

4.4.3 Intrusion detection problems

Intrusion detection concerns the processing of large amount of data in real-time in order to classify network traffic into *normal* or *attack*. Corresponding learning problems are thus suited to evaluate scalability limitations of a given machine learning method.

There are many other large scale datasets that could have been chosen for this purpose. We chose data from intrusion detection problems due to the acute nature of the problem in current literature. Due to the rapid growth in computer network applications, the challenges in cyber security research have increased. Intrusions and attacks can be defined as events that compromise availability, authority, confidentiality or integrity of a computer system. A network intrusion detection system (NIDS) monitors network traffic flow to identify attacks. There are misuse/signature based and anomaly based intrusion detection systems. The signature based system uses the knowledge of known attacks to detect intrusions. However, in the anomaly based system, a normal profile is created from normal network behaviors, and any deviations from these normal behaviors are considered attacks.

Some older benchmark data set such as the popular KDDCUP99 dataset [98] were widely adopted for evaluating NIDS algorithms performance. Evaluating a NIDS algorithm using such data sets does not reflect realistic performances due to (1) a large number of redundant records in the training set, (2) multiple missing records that are a factor in changing the nature of the data, and (3) the dataset was created artificially *by simulation rather than from actual network measurements).

We will, instead, use the much newer dataset UNSW-NB15 [99]. The number of records in the training set is $175,341$ and the testing set contains $82,332$ records which can be grouped into two classes: attack and normal. Nine families of attacks are covered by this dataset (see Table 4.3). The dataset was the result of observations during a 16 hour period on Jan 22, 2015 (training set) and observation during a 15 hour period on Feb 17, 2015 (testing set) during which 100 GBs of data was collected. The data set is labelled from a ground truth table that contains nine known attack types. A key characteristic of the UNSW-NB15 dataset is that it is a hybrid set consisting of both real modern normal behaviors as well as simulated (synthetic) attack activities.

Table 4.3: Class distribution of the UNSW-NB15 dataset.

Category	Training set	Testing set
Normal	56,000	37,000
Analysis	2,000	677
Backdoor	1,746	583
DoS	12,264	4089
Exploits	33,393	11,132
Fuzzers	18,184	6,062
Generic	40,000	18,871
Reconnaissance	10,491	3,496
Shellcode	1,133	378
Worms	130	44
Total Records	**175,341**	**82,332**

4.5 Evaluation methods

The performance of models needs to be quantified in order to allow a formal evaluation of their capabilities. To cover various aspects of performance evaluation, various evaluation metrics will be applied. For the classification problems, Accuracy (ACC), (macro/micro) Recall, F1, and Area under the ROC curve (AUC) indicators will be utilized. The root mean square error (RMSE) and (absolute) mean bias are the evaluation metrics when evaluating the results of the regression and visualization tasks. The metrics are defined as follows.

4.5.1 Accuracy (ACC)

ACC represents the percentage of correctly classified examples over the dataset size. On the basis of the confusion matrix given in Table 4.4, the accuracy is calculated as follows.

$ACC = \frac{TP+TN}{TP+FN+TN+FP}$, where TP and TN are true positives and true negatives respectively and FP and FN are false positives and false negatives respectively..

Despite its popularity, the ACC performance measure is limited in expressing the true performance of a classifier on unbalanced learning problems.

Table 4.4: Confusion matrix.

	Classified Positive	Classified Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

4.5.2 Recall

Recall is defined as the proportion of target documents returned in a document classification task. There are two conventional methods of calculating the performance of a text categorization system based on recall, namely micro-averaging and macro-averaging. Micro-averaged values are calculated by constructing a global contingency table and then calculating the recall using these sums. In contrast, macro-averaged scores are calculated by first calculating precision and recall for each category and then taking the average of these. The difference between these is that micro-averaging gives equal weight to every document while macro-averaging gives equal weight to every category.

$$R_{micro} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + FN_i} \tag{4.1}$$

$$R_{macro} = \frac{1}{|C|} \sum_{i=1}^{|C|} \frac{TP_i}{TP_i + FN_i} \tag{4.2}$$

4.5.3 F-measure (F1)

F1 can reflect more accurately on the generalization performance of a classifier in an unbalanced dataset. The larger the F-measure value the better the performance on the positive class. Its calculation is a balance between precision $Pr = \frac{TP}{TP+FP}$ and recall $Re = \frac{TP}{TP+FN}$ in that the F-measure is $F_1 = \frac{2*Pr*Re}{Pr+Re}$

4.5.4 Area under the ROC curve (AUC)

AUC is a measure which represents the accumulated performance over all possible classification thresholds. AUC can also be referred to as the probability that a learning model ranks a randomly chosen positive sample higher than a randomly chosen negative one. In fact, if a model classifies the negative examples correctly, then a poor performance in predicting the positive examples would be reflected by a low AUC value.

4.5.5 Validation methods

There are two main validation approaches that will be considered in this thesis, namely the train-validation-test split and the leave-one-subject-out (LOSO) cross validation approach. For the former evaluation method, a dataset is split into 3 non-overlapping groups for serving the training, validation, or testing of a given method. The training set is used to fit the classification model. During the learning procedure, the validation set is used for the selection purpose of hyper-parameters in the classification model so that the best model performed on the validation set would be selected to be applied to the blind testing set. This evaluation method will be used mostly in this thesis.

In the LOSO cross-validation, the classification model is trained on data collected from all participants, who perform physical activity trials, except one, which is left apart and used as the test dataset. The process is repeated until every participant has served as the test data. The model performance results are calculated as the average of all testing results.

In this thesis, because some of the datasets are very small (such as the PA2014 dataset) we will use the LOSO method whereas for larger datasets the train-validation-test split method will be used.

4.6 Conclusion

This chapter considers the problem of data processing issues arising from the characteristics of the the three sets of physical activities collected in 2012, 2014, and 2016 respectively. It is proposed to use a data visualization technique to visualize the data, to investigate what might be characteristics of the datasets at hand. Then, it is proposed to use SSEN, a proposed machine learning algorithm which can handle sparse unbalanced datasets to classify the accelerometer data. The use of transfer learning to handle the issue of small sample sizes, in the PA2014 and PA2016 datasets is suggested. It is also proposed a "best of breed" object detection, object tracking algorithm to process the video recordings.

Then as the characteristics of the PA data are relatively unknown, therefore a benchmark dataset, the Policemen dataset, which could have arbitrary fine visual patterns is suggested to provide the information of what might be a sufficiently high resolution suitable to visualize the PA datasets. Then, it is suggested that two more datasets, viz., web spam dataset, and an intrusion detection dataset, UNSW-NB15, together with the policemen dataset can be used to evaluate the capabilities of the proposed SSEN algorithm in handling severely unbalanced data classifications in a timely fashion.

A number of evaluation criteria were described which can be used to assess the efficacy of the proposed approaches to study these three PA datasets.

Chapter 5

High resolution Self-organizing Map

5.1 Introduction

The Self Organizing Feature Map (SOM) provides a convenient way for visualizing high dimensional inputs by projecting them onto a low dimensional display space. Since the objective it to visualize data, in this thesis we will limit the exposure to SOMs with a two-dimensional display space. This map has an appealing characteristic: Feature vectors close to one another in the high dimensional input space remain close to one another in the low dimensional display space. Owing to the computational requirements, the display space so far remains of relatively low resolutions.

This Chapter describes an implementation of the SOM which makes use of the highly parallel architecture of a graphic processing unit. The corresponding algorithm significantly decreases the computational time requirement thus increases the computational speed. This in turn allows a substantial increase in resolution of the map while keeping the computation to within an acceptable wall clock time. The public interest in training a SOM is to produce an output map structure that matches as well as possible the input structure of increasingly complex problems. A small neural map would not be sufficient for modelling complex input spaces. While the concept of high resolution neural map is not new, there are no significant

works proving its benefits in its applications to difficult clustering problems. This thesis will present the graphic processing unit (GPU) implementation of a high-resolution SOM (HRSOM) algorithm which will allow to adopt clustering experiments with very large neural maps. Armed with such an implementation, we find that the HRSOM can display intricate details associating the relationships among input feature vectors, which would be lost if a low resolution SOM was deployed. This property is validated through a deployment of the SOM and HRSOM to visualize an artificially generated dataset with well understood properties (viz. the policeman dataset). The experiments will confirm that the HRSOM can expose intricate relationships among input feature vectors which would remain hidden in SOMs of lower resolution. Moreover, by measuring the clustering performances for three large clustering problems, this thesis will find that the HRSOM produces maps with near optimal clustering performance.

5.2 Background

The SOM [100] is popularly used for data visualization in the exploration stage of a data mining application. One of the key properties of a SOM is that it creates a topology-preserving mapping of a high dimensional input (feature) space onto a low dimensional, usually two-dimensional, output discrete grid of resolution $N \times M$, commonly referred to as a display space [100]. The SOM is especially suitable for data visualization and analysis, as it conveniently facilitates the visualization of relationships among the input vectors in high dimensional space onto a two-dimensional display space. Through such visualization, it helps the user in understanding any intricate relationships among the input vectors via exploration in the display space. Such visualization can act as a prelude to further processing of the input data [100].

Each grid point is referred to as a *neuron*, characterized by a codebook vector of the same length as the input vectors. In this case the SOM is said to consist of NM neurons. The SOM

training algorithm aims to order the codebook vectors located at grid points so that data points represented by high dimensional vectors which are similar in input space are mapped to nearby grid points. Once the ordered codebook vectors is obtained and converged to a stable equilibrium [100], interesting and useful insights into the properties of input vectors can be made. A main problem with the SOM is that its mapping space, $N \times M$, is discrete, thus the quality of the mapping depends on the magnitude of N and M, the resolution of the grid over the display space.

A SOM consisting of a very large number of neurons, i.e. the magnitude of N and M are relatively large, is called a High Resolution SOM [101]. The reason for creating HRSOMs is to better visualize the macro as well as micro structures, indicating relationships existing among the input vectors [102, 103]. HRSOMs allow more room to separate dissimilar input patterns, and are more suitable for datasets that exhibit complex relations among its vectors. In contrast, a low resolution SOM (LRSOM) consists of N and M of relatively low magnitude. Intricate and complex relationships among input vectors will be lost if LRSOM is deployed. The intricate relationships would merge into simpler structures due to the low resolution nature of the display space. The training algorithm of SOM scales linearly with the number of neurons and the size of the dataset [100]. When implemented on a modern CPU architecture, its limited computational power prevents it from training sufficiently large SOMs, i.e. both N and M are relatively large. This also prevents the construction of a display space which could considered continuous, by having very large magnitude N and M values. To the best of our knowledge, prior to this thesis nobody had succeeded in implementing and training SOMs with N and M in the order of low thousands. There were various attempts in improving the granularity of the display space but these were based on a hierarchical SOM structure [104, 105] and by a social hierarchical structure: The tree SOM [106, 107]. The basic idea behind these approaches is that the SOM adapts the topology of each hierarchical layer to the properties of the input vectors, starting with a

very small SOM of grid size 1×1, then growing/enlarging the SOM in places where the quantization errors are high [106, 107, 104, 105]. The approach of growing SOM online and only in locations where high quantization error occurs, gives the SOM a tree-like appearance [104, 105]. The approach does deploy a number of additional neurons in order to separate the input vectors from different categories. However, there are two main drawbacks associated with this approach:

1. The SOM can only grow in restricted areas. Thus, the shape of clusters formed can be distorted so that they no longer reflect the size and shape of a cluster in the high dimensional input space.

2. The growth of the neuron number increases the computational demand.

It hence remained difficult to solve problems involving datasets with complex relationships among input vectors which require a display space with relatively large magnitudes of N and M. In contrast, the HRSOM introduced here does not suffer from these drawbacks, as it can provide a display map with relatively large magnitude of N and M. This is achieved by "collapsing" the hierarchical structure into the two dimensional display space by the provision of N and M with relatively large magnitude.

An important observation is that the SOM is an Artificial Neural Network and is a massively parallel system. The SOM has traditionally been implemented on CPU systems thus limiting the degree of parallelism to the number of cores in a CPU. There have been various recent attempts in porting the SOM algorithm to GPU (Graphics Processing Unit) in order to take advantage of hundreds and thousands of cores, though each core would have access to relatively small fast onboard memory capacity, typically found on a GPU. It was claimed [1] that this significantly reduces the execution time of the SOM algorithm [108, 109, 110]. Moreover, there is a trend to computing clusters which are equipped with multiple GPUs

[1] Some of the papers do not provide sufficient details for us to repeat their experiments to validate their claims.

internally connected together, and which are much more powerful than CPU clusters in parallel and distributed applications [111, 112].

The aims of this Chapter is to (a) describe techniques which enhance the speed of a GPU implementation and (b) analyze the effects and results of using HRSOM in a dataset which contains intricate relationships among their input vectors.

The contribution of this work is as follows:

1. An implementation of the SOM on a GPU which takes into account its architectural characteristics. To the best of our knowledge, the resulting GPU implementation is the fastest known implementation of the SOM algorithm.

2. Demonstrating that a HRSOM can display intricate and complex relationships among input vectors by applying the method to an artificially generated dataset with known intricate and complex relationships among its vectors. This demonstration is possible because the algorithm allows the revision of a sufficiently large magnitude N and M. It is shown that the unsupervised classification accuracy of the SOM approaches 100% with the increase of both N and M. This indicates that the HRSOM is able to capture and display the intricate and complex relationships among its input vectors. To our knowledge there exist no similar demonstrations to what will be shown in this Chapter.

5.3 The High Resolution Self-Organizing map

The SOM algorithm [100] performs a nonlinear and topology preserving projection of the high dimensional input data (feature) space onto a discretized display space consisting of NM neurons arranged in a $N \times M$ grid. Each neuron i of the map is associated with an n-dimensional codebook vector $m_i = (m_{i1},\ldots,m_{in})^T$, where T denotes the transpose operator. n is also the dimension of the input vectors. Neurons adjacent to neuron i belong

to a neighbourhood denoted as N_i where the neighborhood relation is often hexagonal in shape [100]. The SOM training algorithm [100] can be presented in two steps:

Step 1 - Competitive step: An input vector u is randomly drawn from the input dataset. Its similarity to the codebook vectors is computed. Most commonly, it is to find the minimum Euclidean distance $\|u - m_i\|$ between u and the codebook vector of neuron i, m_i. The winning neuron r must satisfy the relationship $r = \arg\min_i \|u - m_i\|$.

Step 2 - Cooperative step: All codebook vectors are adjusted. The best matching codebook vector m_r and its neighbours are moved closer to the input vector. The magnitude of the adjustment is controlled by the learning rate α and by the neighbourhood function $f(\Delta_{ir})$, where Δ_{ir} is the topological distance between the two codebook vectors m_r and m_i. The amount of change in the codebook vector m_i is computed as $\Delta_{m_i} = \alpha(t) f(\Delta_{ir})(m_i - u)$. The learning rate $\alpha(t)$ decreases with the training process. The neighbourhood function $f(.)$ controls the amount by which the codebooks of the neighbouring neurons are updated. Popular is a Gaussian neighbourhood function: $f(\Delta_{ir}) = \exp\Big(-\frac{\|l_i - l_r\|^2}{2\sigma^2}\Big)$, where σ is the spread or radius which controls the operating region of function $f(.)$. The two values l_r and l_i are respectively the location of the winning neuron, and the location of the i-th neuron on the map.

The two steps are repeated for each training sample and for a pre-defined number of iterations. While there is no proof of the convergence of the training algorithm [100], it is empirically found that the training algorithm always converge, as when the learning constant is reduced to a value close to 0 the training algorithm stops updating the codebook vectors.

5.3.1 GPU acceleration of the SOM algorithm

5.3.1.1 GPU architecture:

A GPU, sometimes called visual processing unit, is a specialized graphical unit. The electronic circuit design of GPU is for the particular purpose to rapidly manipulate and alter

memory to accelerate the processing of data stored in a frame buffer, and to optimize the visualizing process on a display. GPUs are widely used in various hardware architectures such as embedded systems, personal computers and workstations. In a personal computer, a GPU can be integrated on a video card, can be embedded on the motherboard. Modern computers allow to attach one or more GPUs. The more recent GPU technology allows for efficient and general purpose massive parallel computations that is not limited to graphics processing.

In software programming, a GPU is extensively used to support computational problems that require high-performance and highly-parallel computations. The GPU is constructed by a number of *stream multiprocessors*. A multi-threaded program is employed in parallel by allocating the number of threads equally to a number of processing blocks and the number of blocks is partitioned into a grid of blocks. All threads have access to the GPU's global memory which is also used as a communication channel between the multiprocessors and between GPU and CPU. Threads of an individual block have common access to another type of memory called shared memory that helps threads in a block to have access to the data of each others. A thread from one block cannot access the shared memory of another block. There are five types of memory on a GPU each with different properties and scope:

Register memory: Data stored in this memory is limited to be accessible to the thread that uses it and is destroyed when the thread terminates. The size of register memory is in the order of a few bytes.

Local memory: This has a similar role as register memory, however its size is in the order of kilobytes but it is slower than register memory.

Shared memory: Data stored in shared memory is accessible to all threads within the operating block. This type of memory allows for threads to share data and to communicate with one another. The shared memory lasts for the life-time of the associated block it is larger but slower than local memory.

Global memory: This is the most accessible type of memory. The memory is accessible by all threads and by the CPU host system. Its content lasts for the duration of the host allocation, and is in size in the order of gigabytes although it is the slowest of all.

Constant and texture memory: These are read only memory and are used to store data that is not altered during the code execution. This memory, too, can be gigabytes in size and it is faster than global memory.

GPU hardware manufacturers provide programming interfaces that allow the host system to interact with the GPU. For example, CUDA is a parallel computing platform that offers an application programming interface (API) model. CUDA is created by NVIDIA allowing software developers to use GPU for general purpose, so-called GPGPU code. CUDA is a software layer that gives direct instructions to GPU for the execution of GPU kernels. The CUDA platform accommodates various programming languages such as C and C++. A multiprocessor adopts an unique architecture called SIMT (single-instruction, multiple-thread). For this reason, all parallel threads execute the same set of instructions but can operate on different data. CUDA introduces simple kernel calls to execute code on the GPU.

The HRSOM algorithm described in this section exploits GPU characteristics that can be accessed by commonly available programming interfaces. Thus, while the algorithm has been implemented and tested using CUDA and NVIDIA GPU infrastructure, the algorithm has no dependency on any particular programming interface or type of GPU. The algorithm is thus more portable than alternatives that use GPU type specific features such as tensor cores or ray-tracing cores.

5.3.1.2 Porting the SOM algorithm to GPU

The SOM algorithm implemented on GPU is optimized for dealing with problems that require the training of significantly large maps. There are computations involving the finding

of the best matching unit and in updating the related codebook vectors for large amounts of data. All computations in the SOM algorithm can be implemented for massive parallel computation on a GPU. To achieve this we define a number of specialized kernels namely (1) a kernel for initializing the codebook vectors and assigning the map coordinates to reference variables in parallel; (2) calculating all the Euclidean distances between the current input vector and the codebook vectors in parallel; (3) a reduction kernel for finding the minimum distance; (4) a kernel for identifying the neighbouring nodes based on the radius value in parallel; (5) a kernel which updates in parallel the codebook vectors of nodes relative to the winning node.

Each of these operations can thus be implemented as a GPU kernel function. In order to optimize these kernels, a number of strategies have been applied:

1. Reduce the number of host to device and device to host memory transfers. When transferring data from CPU to GPU and back, we use one continuous memory block. For example, by concatenating all data samples into a single array there is only one instruction to transfer the data to GPU. For large datasets this can improve the transfer speed by more than 10 times when compared to transferring input samples separately.

2. For all kernel functions store run-time variables in shared memory or in local memory as much as possible, and keep kernels simple and small since the cost of kernel launches is negligible and since this improves speed due to fewer registers used. By using small kernels one can better utilize the registers, shared memory and constant memory because the memory resources are limited to each kernel.

3. The reduction kernel is a choke point of the algorithm. Its speed can be optimized by assuming that the number of threads is a power of two value. Hence the number of threads is chosen to suit the reduction kernel.

4. A stream multiprocessor can handle at most 2048 threads concurrently and can ac-

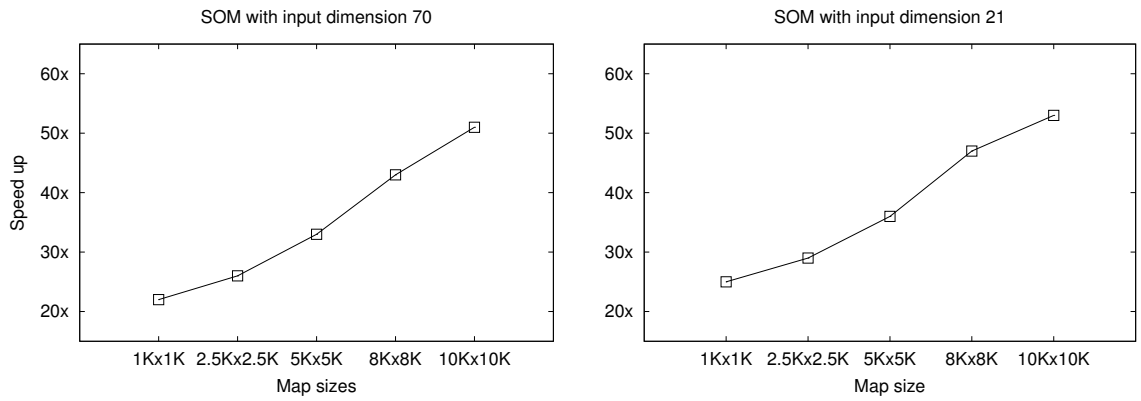


Figure 5.1: GPU rate of speed-up depending on map size. 1K means 1000 and Ax means A times speed up when compared to the CPU.

commodate 16 active blocks. If one sets a block to contain 128 threads, the number of concurrently active blocks will be $2048/128 = 16$ (the maximum threshold for the GPUs that were available to this thesis). On the other hand a block size of 256 would also use the available computation resource. We found that setting the block size to any other value seems to be wasting the resources since there will exist a block which contain fewer number of threads than the others. We found that using blocks that contain 128 threads produces the best acceleration.

5. Prevent threads from diverging by ensuring that the conditional jumps branch equally for all threads. We implemented conditional branching based on a multiple of the wrap size and, in addition, we also unroll loops.

The speed improvement of the GPU implementation increases with the degree of parallelism. This can be observed in Figure 5.1. The Figure shows that the rate of speed improvement increases with the size of a SOM. The speed comparison is relative to the Intel(R) Core(TM) i7-5960X CPU @3.00GHz (extreme edition). That CPU was the fastest consumer type CPU from Intel at the time of writing. The Figure shows that the GPU implementation can be 52 times faster than the fastest consumer CPU. The results were obtained by using GeForce GTX TITAN X (black edition) - one of the fastest GPUs for single pre-

cision computations at the time of writing. The results were obtained by using the compiler optimization flag `-O3` for both, the CPU and GPU version of the code. Moreover, the CPU version is based on the SOM software package (som_pak) which implements *tricks* to accelerate code execution such as (a) not updating codebooks that are more than 3σ away from the winning codebook, and (b) breaking the loop early when computing the Euclidean distance. Theses tricks improve the execution speed of the CPU code by approximately three to 10 times depending on network size, and are not (and do not need to be) implemented in the GPU code.

It is difficult to make a comparison with others who ported the algorithm to GPU since the others use different computation resources, do not specify the type of CPU and GPU that they used for their comparisons, do not specify whether or not optimization tricks were implemented, or do not provide the code. Nevertheless, we found that we achieved a maximum on the memory bandwidth on the GPU and hence are confident that our GPU implementation processes the data at the maximum possible rate. The latest hardware architecture of popular models have been selected for the experiments. The CPU information is Intel(R) Core(TM) i7-5960X CPU @ 3.00GHz and the GPU is NVIDIA Corporation GM200 [GeForce GTX TITAN X]. The computations are made using single precision floating point.

Limitations of the HRSOM: The size of the HRSOM is limited to the amount of memory on a GPU. GPUs generally have access to less (global) memory when compared to the amount of RAM accessible by a CPU. Since the training data needs to be located in the global memory and thus this reduces the amount of memory available for the HRSOM. Nevertheless, HRSOMs of size $10,000 \times 10,000$ and larger (depending on the size of a training set) can be trained on consumer market GPUs. This is much larger than would be feasible for a CPU.

5.4 Evaluation methods

There is a number of standard measurements which are usually used to determine the quality of clusters with regard to the known input samples' categories. The micro purity and the macro purity is commonly used in the machine learning and data mining community. While the micro purity measures the overall clustering performance with respect to given sample's classes, the macro purity calculates the average of individual clustering performance for each class. Given a SOM mapping result of a n-class clustering problem, the number of samples in a dataset is denoted as A, and the number of samples in class k is denoted as A_k. The number of samples with the majority label in class k is denoted as a_k. The calculation of micro and macro purity is as follows:

$$\text{micro-purity} = \frac{\sum_{k=0}^{n} a_k}{A}, \qquad \text{macro-purity} = \frac{\sum_{k=0}^{n} \frac{a_k}{A_k}}{n}. \tag{5.1}$$

The SOM is an unsupervised training algorithm whereas micro-purity and macro purity are computed based on available labels. These two measures thus quantify how well the mappings align with the class labels. To quantify the degree by which the mapped data is organized in clusters, a third evaluation method is used to compute the clustering quality. The quantity is computed based on a group of neurons on the SOM map. A group is defined by including the nodes that are in the direct neighborhood to a given node on the map. In other words, all nodes that are connected with a given node. For every node and corresponding group we count the number of samples with the majority label for every class. The result obtained is divided by the number of samples in the whole dataset. We denote this evaluation as the *grouping index*. In practice,the local relation between neurons on the map is considered in this evaluation method, hence reducing the sparsity problem when one uses more number of neurons than the number of input samples.

5.5 Experiments

5.5.1 The cluster forming progress

A HRSOM of size 2500×2500 is trained using $\sigma = 600$, the number of training iterations is 120 and the learning rate $\alpha = 0.6$. The progress by which the clusters formed is presented in Figure 5.2. The mappings as they were observed at iteration 2, 40, 80, 90, 100, 110, 117 and 120 are shown from top left to bottom right, respectively. It can be observed that for about half of the training period, the clusters form relatively slowly. Separation of groups of samples commenced between the 70th and 80th iteration, and become well defined between the 100th to 120th epoch. The mapping result from 115th iteration to the end of the training procedure is largely unchanged. Such a detailed clustering result has never before been seen for this dataset since the largest SOM applied thus far never exceeded the size of 256×256. The result presented here is useful since it not only provides an insight into the visually differences and similarities between sample classes, but also supports the claim that the HRSOM can give a better visualization for the complex input space.

5.5.2 Closer view on individual clusters

The fully trained HRSOM in Figure 5.2 shows well separated clusters although there exist pattern classes the mapping of which spread across a larger region on the map. For example: the two classes denoted as the policemen with lowered left arm and the policemen with raised left arm. This indicates that the policemen classes contain a variety of sample features which are not very similar. Another reason could be that the two classes contains approximate ten times the number of samples of the other classes. A closer view of some of the individual clusters is provided by Figure 5.3. Shown are the clustering of samples belonging to the four sub-classes of the class *house*, clustering of the two sub-classes of *policemen* and the clusters formed by the two sub-classes of *ships* from top to bottom,

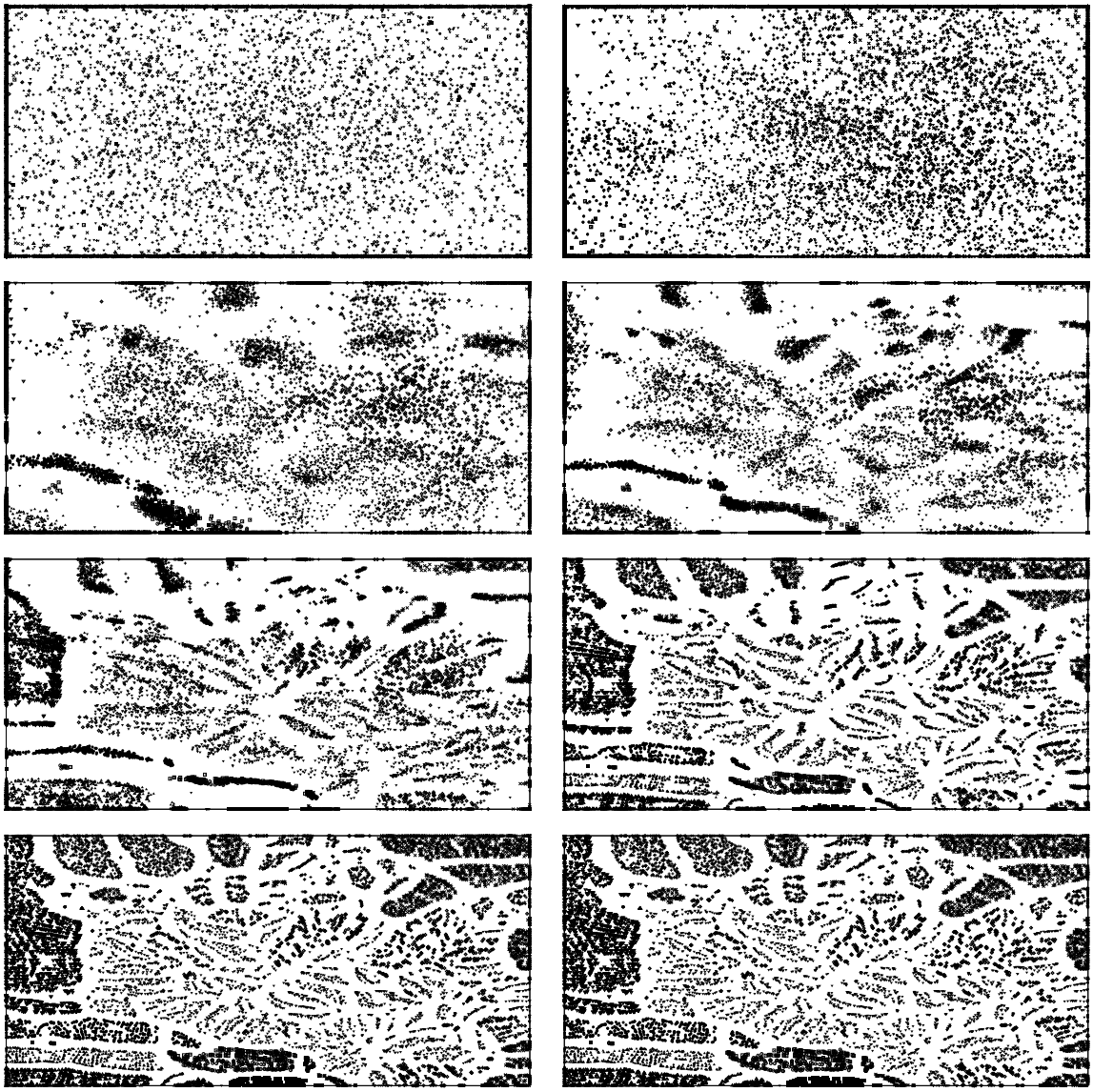


Figure 5.2: The evolution of the mapping during the training procedure.

respectively. Inside each cluster, there exists a number of sub-clusters which show the different samples' characteristics. More observation has been made for a house type class as can be seen from Figure 5.4. This cluster has some patterns which were mapped closer than the other. These patterns have similar characteristics. Samples mapped far apart, however contain different features. It can be seen from the corresponding images of these samples, the patterns mapped near by, i.e in the middle of the cluster, look very similar in the shape

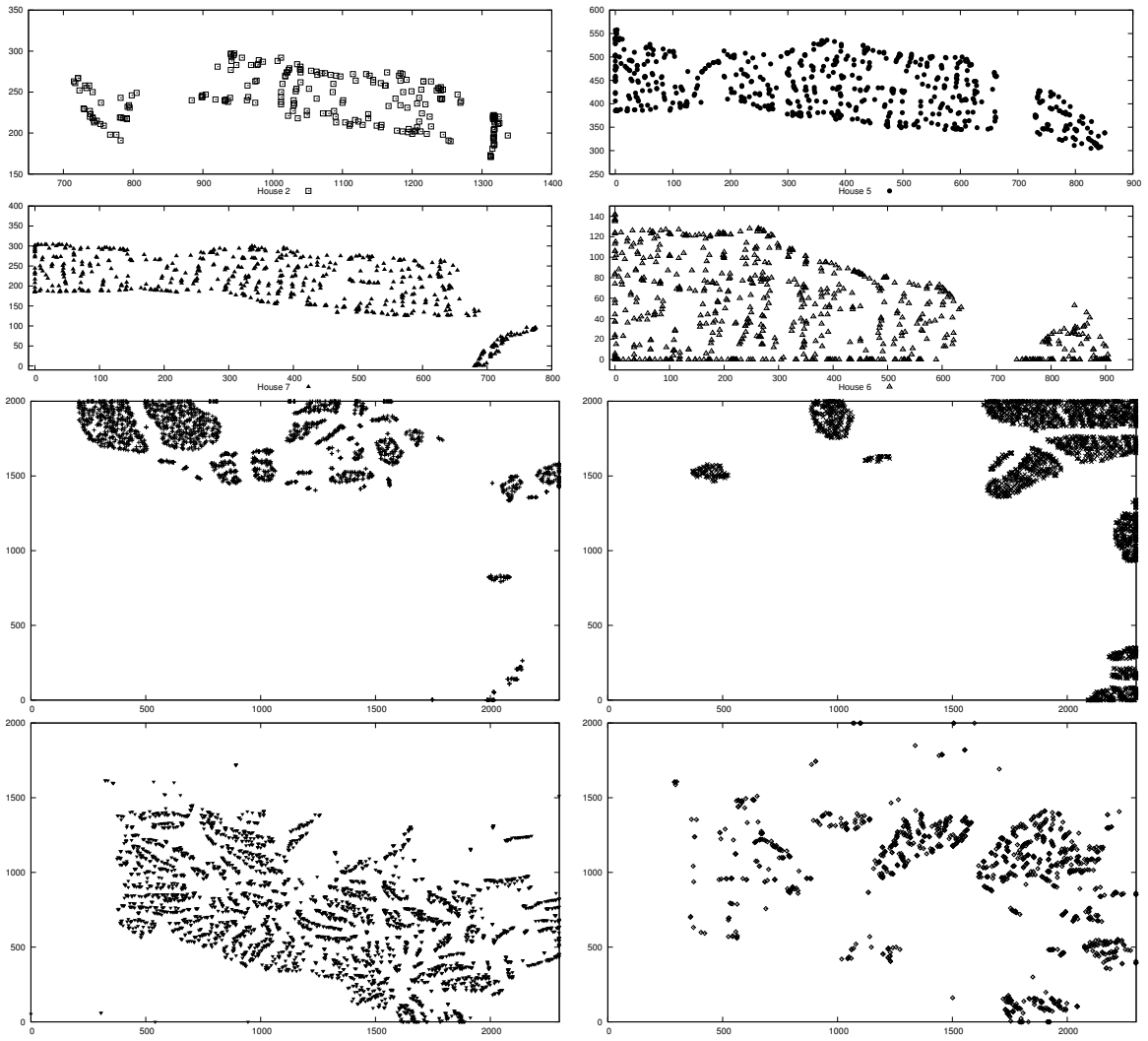


Figure 5.3: The mapping of some of the pattern classes.

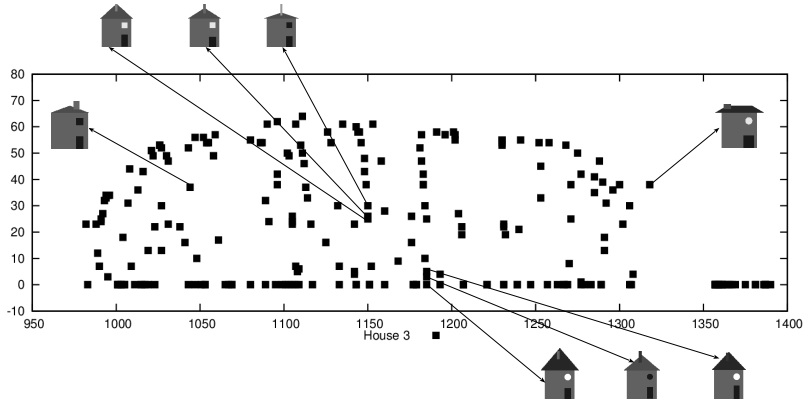


Figure 5.4: The mappings of the samples in class ”house one windows (UR)”.

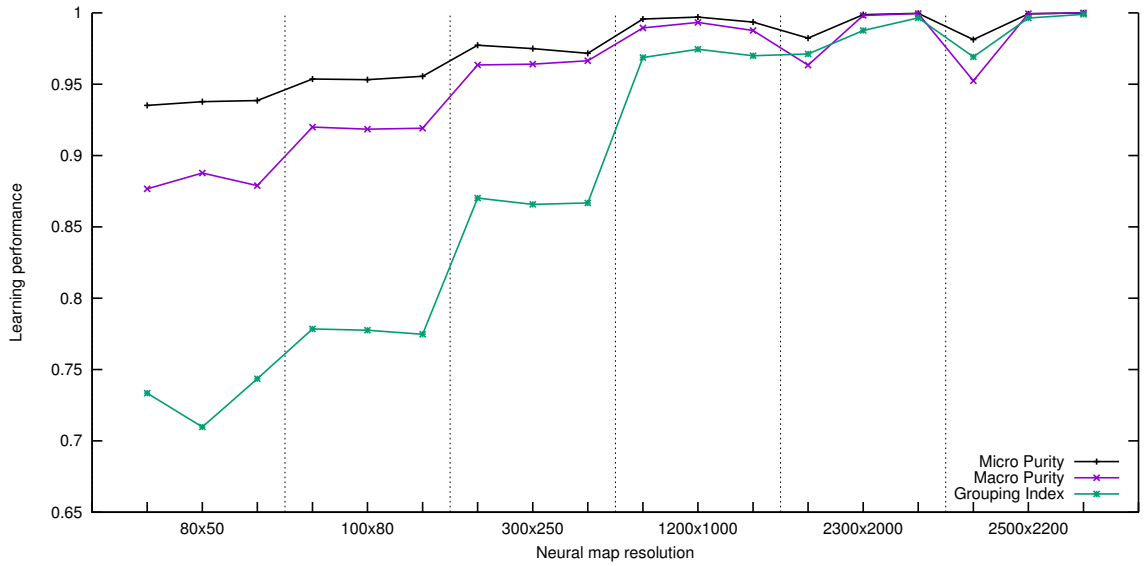


Figure 5.5: Comparing LRSOM and HRSOM performances when trained on the artificial policemen dataset.

of the windows and the relative positions of chimneys. Further to the left of the figure, there are houses with chimneys located further on their right sides. On the other hand, further to the right of the figure, there are houses with chimneys located further on their left sides. Such observations were only possible due to the HRSOM and were not previously observed using lower resolution SOMs.

5.5.3 Comparing LRSOMs and HRSOMs

Table 5.1 compares the clustering performance when training SOMs with different map sizes for the policemen dataset. The map sizes are grouped into low resolution SOMs (LRSOMs) and HRSOMs. The former group includes SOMs with as little as 400 neurons (80×50) and SOMs with up to 75,000 neurons (300×250) while the latter contains SOMs with more than 100,000 neurons and up to 5,500,000 neurons (2500×2200). It should be noted that this is the first time that SOMs of such large size have been trained on complex clustering problems such as the policemen benchmark data.

When training the SOMs we varied training parameters such as the learning rate α and

Table 5.1: A comparison of LRSOMs with HRSOMs when using the policemen dataset.

Map	Map size	Epoch	σ	α	Micro Purity	Macro Purity	Grouping Index
1	80x50	400	20	0.2	0.9351	0.8766	0.7334
2	80x50	400	22	0.5	0.9377	0.8877	0.7097
3	80x50	400	25	0.6	0.9385	0.8789	0.7435
4	100x80	400	35	0.5	0.9536	0.9200	0.7785
5	100x80	400	40	0.8	0.9532	0.9185	0.7776
6	100x80	400	37	0.3	0.9556	0.9192	0.7747
7	300x250	400	100	0.6	0.9773	0.9635	0.8702
8	300x250	400	115	0.4	0.9750	0.9641	0.8658
9	300x250	400	120	0.7	0.9717	0.9664	0.8668
10	1200x1000	200	400	0.1	0.9957	0.9894	0.9687
11	1200x1000	200	300	0.5	0.9970	0.9933	0.9745
12	1200x1000	200	500	0.9	0.9935	0.9876	0.9699
13	2300x2000	100	500	0.4	0.9823	0.9634	0.9712
14	2300x2000	150	500	0.4	0.9987	0.9982	0.9876
15	2300x2000	200	500	0.4	0.9997	0.9995	0.9965
16	2500x2200	80	600	0.6	0.9813	0.9524	0.9692
17	2500x2200	100	600	0.6	0.9993	0.9994	0.9964
18	2500x2200	120	600	0.6	1.0000	1.0000	0.9990

number of training iterations as indicated in Table 5.1 or visually in Figure 5.5. The radius (σ) was adjusted to be about 40% of the smaller side of the map although we varied σ to investigate the sensitivity of this parameter. Each experiment was repeated three times with different initializing conditions. The results shown are the average performance over the three runs. A number of interesting observations can be made from Table 5.1:

1. The performance of HRSOMs is generally much better than that of LRSOMs for all three assessment methods.

2. An almost perfect performance is obtained for the highest resolution SOM. This is an interesting observation because the SOM is trained unsupervised but evaluated on using actual class labels. Only the largest SOMs offer sufficient mapping space to serve applications which require the separation of pattern instances in low dimensional space while preserving the topology of the input data. HRSOMs are thus partic-

ularly well suited as a dimension reduction method while maintaining the information needed to separate pattern classes. A trained HRSOM is useful, for example, as a pre-processor in big data applications to reduce the dimensionality of a domain and speed up subsequent computations.

3. The grouping index experiences the most significant improvement among the three evaluation metrics. An improvement by approximate 27% indicates the improvement in quality of the clusters.

4. HRSOMs are less sensitive to the choice of σ and the learning rate α. We attribute this observation to the fact that HRSOM offer a higher degree of freedom to the mappings of the data and hence do not rely on large α and large σ as is often needed for smaller SOMs (this is needed to allow the re-organization of mappings during the early stages of network training [100]).

The training time required ranged from less than 60 minutes for the smallest of the maps to 4 days and 9 hours for the largest map. The training of the largest map would have taken nearly 8 months had it been executed on a state-of-the-art Intel CPU. Note also that the SOM only needs to be trained once and, when trained, the GPU version of the SOM can project data in O($\log N$) time, where N is the number of neurons. Provided a sufficiently large GPU the computational complexity of the HRSOM is thus independent to the number of samples that need to be projected as they can be processed independently and in parallel.

5.5.4 Clustering abilities of the HRSOM for web spam detection datasets

The following will present and compare the experimental results of LRSOMs and HRSOMs for two real-world web spam detection problems, namely, the UK2006 and UK2007 datasets. Figure 5.6 and 5.7 visualises SOMs' results when training the network containing as little as 400 neurons (80×50) and SOMs with up to 3,000,000 neurons (2000×1500) for UK2006

problem and with up to 990,000 neurons (1100×900) for UK2007 dataset. Some major derivations include:

1. The clustering abilities of the SOMs with larger maps are always better than that of lower resolution ones given three evaluation indicators.

2. The clustering performance is poor for the spam class samples with respect to the low resolution maps. This is associated with the situations that the micro purity is high while the macro purity is low. The reason for this poor clustering performance is that samples in the normal class overlapped most of the spam class samples in the case that the training map is not sufficiently large. There must be lack of room on the neural map to separate the spam samples from the normal ones.

3. The clustering performance is almost 100% for both categories when training on the very large maps such as the map of size 2000×1500 for UK2006 and the map of size 1100×900 for UK2007.

4. The greatest improvement is seen with regard to the macro purity performance, which is approximately 28% and 32% when compared the lowest with the highest resolution SOMs for the UK2006 and the UK2007 datasets, respectively.

5. The grouping index results for the web spam detection data are seen much better (by at least 12%) than the case for the policemen dataset even though the training maps are small in size. The reason for this may be that the policemen data has 12 categories which might increase the confusion level in the clustering process.

Regarding the training time requirement, the experiment on the largest map has finished in almost 4 days and 1 hours for UK2006, and in almost 13 days and 12 hours for the UK2007 dataset. Once the training is done, in the application phase, the trained HRSOM can map around 400 samples per second.

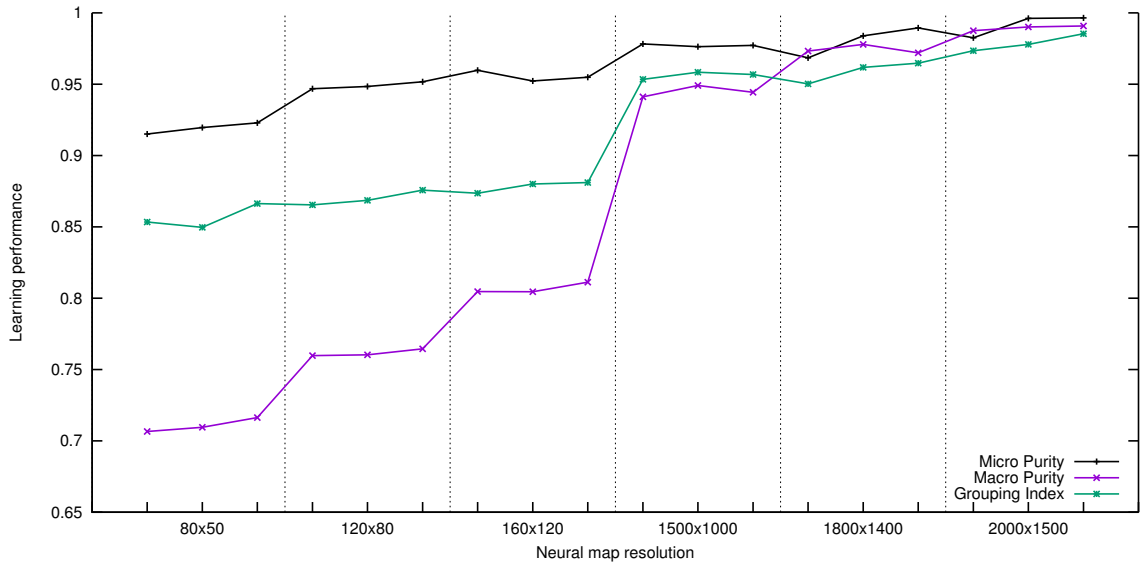


Figure 5.6: Comparing LRSOMs and HRSOM performance on the UK2006 dataset.

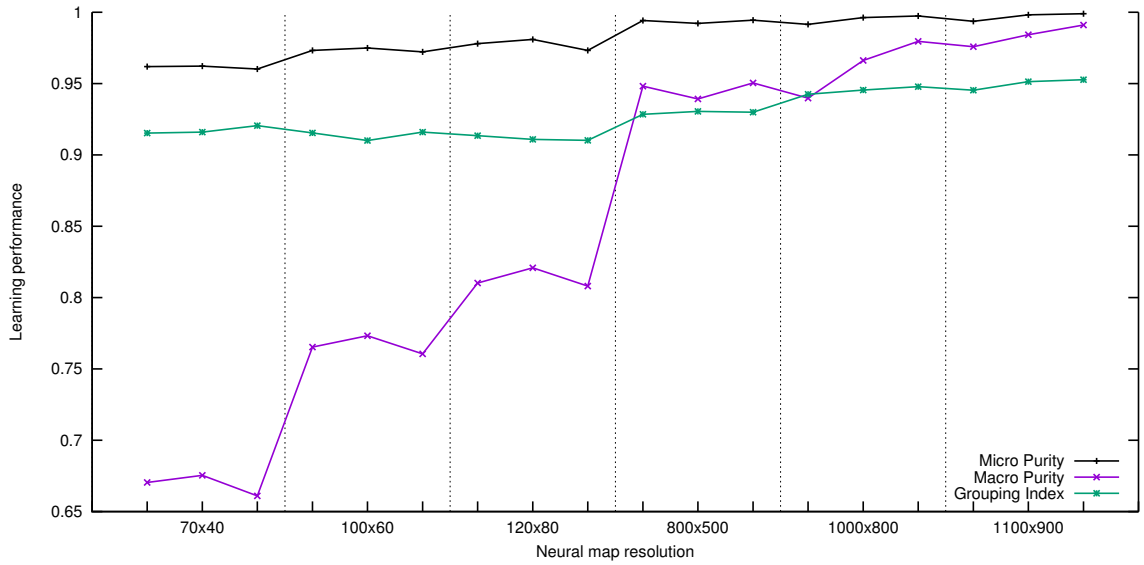


Figure 5.7: Comparing LRSOMs and HRSOM performance on the UK2007 dataset.

5.6 HRSOM in a layered classification ensemble

This section will deploy a layered ensemble model consisting of the HRSOM in its first layer and a classifier in the second layer. Two layers are trained independently. The HRSOM is trained on the available input samples. Then, once training is finished, the original feature vector of each sample is augmented with the mapping result of the HRSOM, resulting in

augmented feature vectors. There are several advantages when using the HRSOM in this manner: Since the HRSOM is trained unsupervised, the learning process is not affected by the unbalanced nature of the input data distribution. The significantly large display space will help to reveal intrinsic and complex relationships among data and also help to identify regions of likely confusion. The classifier in the second layer is then trained with the augmented feature vectors. The mapping information from the HRSOM would provide useful information that allows the classifier to learn effectively.

In this section, we would like to provide evidence that the HRSOM is not only served as a clustering or visualisation package, but also served as an effective unsupervised enrichment model.

In the following, three evaluation metrics, namely AUC, F1 and ACC are used since the experiments have been conducted on datasets with differing characteristics and since we aim to show the robustness of the learning systems. The key evaluation method is AUC which is also used in the competitions for the web spam detection. The classifier model chosen is GNN which is a recent graph-based generation of MLP, and is a suitable selection for link-based web spam detection problems.

The SOM's best training results in the previous step are taken for this experiment. The GNN model is trained using a variety of parameter settings. The number of hidden units is selected from within $\{15, 19, 25, 34, 40\}$ while the number of state neurons is within $\{8, 15, 18, 29, 35\}$. An adaptive learning mechanism is applied during training. The input data is normalized. The number of training iterations respectively is set to $2,000$ and $1,200$ for the UK2006 and UK2007. The selection of number of epochs is based on observing the network's error convergence during the training. The experimental procedure is as follows. GNN is trained by itself using different network configurations. The best set of network's parameters is selected based on the training performance, and the associated classification performance will be taken as the baseline for further comparison.

Table 5.2: Learning performance on UK2006 dataset with SOM+GNN.

Map size		Training			Testing		
		AUC	F1	ACC	AUC	F1	ACC
No SOM		0.9267[0.0031]	0.6694[0.0112]	0.9173[0.0044]	0.8603[0.0031]	0.7803[0.0056]	0.7421[0.0060]
Low Res.	80x50	0.9256[0.0039]	0.6644[0.0158]	0.9141[0.0087]	0.8574[0.0031]	0.7800[0.0068]	0.7414[0.0067]
	120x80	0.9287[0.0017]	0.6739[0.0100]	0.9227[0.0023]	0.8643[0.0053]	0.7811[0.0047]	0.7436[0.0061]
	160x120	0.9304[0.0016]	0.6794[0.0090]	0.9208[0.0036]	0.8677[0.0051]	0.7962[0.0178]	0.7573[0.0137]
High Res.	1500x1000	0.9326[0.0085]	0.6804[0.0260]	0.9243[0.0034]	0.8737[0.0258]	0.7833[0.0254]	0.7477[0.0254]
	1800x1400	0.9450[0.0010]	0.7220[0.0208]	0.9313[0.0062]	0.8905[0.0062]	0.8131[0.0105]	0.7753[0.0087]
	2000x1500	0.9356[0.0012]	0.7044[0.0063]	0.9300[0.0032]	0.8830[0.0007]	0.7827[0.0198]	0.7500[0.0174]

Table 5.3: Learning performance on UK2007 dataset with SOM+GNN.

Map size		Training			Testing		
		AUC	F1	ACC	AUC	F1	ACC
No SOM		0.7233[0.0061]	0.3034[0.0232]	0.9430[0.0014]	0.7485[0.0034]	0.3103[0.0077]	0.9231[0.0145]
Low Res.	70x40	0.7180[0.0063]	0.2810[0.0253]	0.9432[0.0010]	0.7464[0.0040]	0.2994[0.0018]	0.8938[0.0144]
	100x60	0.7242[0.0063]	0.3473[0.0259]	0.9431[0.0007]	0.7512[0.0032]	0.3277[0.0104]	0.9208[0.0116]
	120x80	0.7426[0.0309]	0.3054[0.0441]	0.9460[0.0035]	0.7547[0.0053]	0.3119[0.0279]	0.8929[0.0323]
High Res.	800x500	0.7579[0.0137]	0.3268[0.0167]	0.9433[0.0022]	0.7630[0.0011]	0.3462[0.0142]	0.9076[0.0022]
	1000x800	0.7851[0.0060]	0.2988[0.0194]	0.9478[0.0013]	0.7686[0.0015]	0.3061[0.0210]	0.8655[0.0242]
	1100x900	0.7770[0.0112]	0.3158[0.0232]	0.9456[0.0006]	0.7678[0.0040]	0.3370[0.0344]	0.8904[0.0297]

In the following, the layer-wise architecture performance will be presented. Comparisons can be made between the LRSOM+GNN and HRSOM+GNN model and with the baseline method. Each experiment is repeated three times and the average performance will be reported.

In Table 5.2 and Table 5.3, the layer-wise architecture performance is presented for the UK2006 and UK2007 datasets, respectively. Comparisons can be made between the LRSOM+GNN and HRSOM+GNN model and with the baseline performance. Without using any SOM result mapping, the AUC generalization performances for UK2006 and UK2007 are 0.8603 and 0.7485, respectively. The use of LRSOM seems not contributing much on the generalization performance while the use of HRSOM in the layer-wise model is seen more effective. In contrast, the HRSOM consistently contributes to an improvement of around 2% to 3% for all evaluation metrics for both datasets.

5.7 Conclusions

This Chapter demonstrated important capabilities of the HRSOM, which include a clustering and a unsupervised feature enrichment capability. It is shown that a HRSOM with resolution $N \times M$ for the display map, when both N and M are of the order of low thousands, intricate details of the relationships among input vectors can be observed in the display space. These details would have been lost if N and M are of the order of low hundreds. Our implementation of the SOM algorithm on a GPU is particularly efficient so that the limitation is now only the amount of available memory on the GPU. It is expected that the next generation of GPU would allow N and M to be of the order of high thousands, or low tens of thousands, thus further reduce the difference between a discrete and a continuous mapping space. Owing to the relatively low cost of the GPU to CPU, one could envisage the HRSOM to be deployed as a visualization device in its own right.

As a topic for future research, we suggest the implementation of SOM to run on a GPU cluster. This will allow the SOM to be deployed to big data applications, and to applications which require an even higher resolution of the mapping space.

Chapter 6

Synthetic Sampling Ensemble Network

6.1 Preamble

The datasets available for PA classification in this thesis are relatively small. The number of participants in the PA2012 dataset is 100, in the PA2014 dataset just 11, and in the PA2016 dataset 16. We find that such small datasets may hamper attempts for obtaining robust models that offer a good generalization performance. Adding to the problem is the unbalanced nature of the PA datasets. The class ”running”, for example, is several times smaller than the class ”light activities”. This thesis thus investigates techniques for working with small unbalanced sets of data.

Many of the common machine learning methods are unable to effectively model unbalanced data in a sparse domain. These models tend to overfit the majority class samples and to generalize poorly. Prior work on improving the generalization performances of i.e. MLPs has shown that the shortcomings of an individual method can be overcome by designing an ensemble system consisting of several methods in such a way that their complementary properties are exploited. For example, an ensemble approach which involves the incorporation of an unsupervised learning approach, e.g. a SOM as a preprocessor, concatenate the outputs to the data samples prior to processing them by an MLP. This can be effective in

addressing the problem [113, 114, 115].

Other common approaches to addressing the issue of unbalanced information use a sampling approach, where a number of subsets are created. Each subset contains approximately the same number of normal samples and abnormal samples [115, 116]. There are various sampling techniques, and, these can be categorized as: (1) balanced sampling, (2) over sampling, (3) under sampling, and (4) synthetic sampling [117, 118, 119].

Balanced sampling, under sampling and oversampling are methods which do not make any changes to the original training samples, but differ only in the way samples are selected. Synthetic sampling, on the other hand, creates new training samples from the feature information of the available training set.

This chapter investigates an effective approach to synthetic sampling using an innovative approach based on a supervised clustering method, viz. supervised DBSCAN, and the use of a visualization algorithm. The supervised DBSCAN method can identify different groups (clusters) of samples. Hence the synthetic sampling can be conducted to different data groups and in knowledge of the class membership of the samples. The procedure is described in greater detail later in this Chapter.

This thesis will apply the proposed synthetic sampling ensemble model to three benchmark datasets from two domains, namely physical activity classification and cyber security. The results are compared and analysed. It will be found that this new sampling technique performs better than other popular methods such as oversampling or under sampling. The thesis will further find that the approach leads to consistently better generalization performances when compared with previous works on these learning problems.

6.2 Introduction

Research in machine learning has long been dealing with problems where a set of training samples is either affected by unbalanced class distributions, small number of training sam-

ples relative to domain space, poor coverage of the feature space, or any combination of these. In order to address such issues, it is common to apply sampling approaches to create more training samples in the hope that the newly created samples compensate the lack of information that can be observed in a given training set [113, 120]. In terms of sampling, the bagging and aggregating idea were pioneered by Breiman [121]. The concept refers to a method of generating multiple versions of a classifier, trained individually on bootstrap replicates of the training sample set [113, 120]. Conceptually, one has a training dataset with n classes, in which class 1 has N_1 labelled samples, class 2 contains N_2 ones and so on. A balanced dataset can be obtained by selecting all $N_s = min(N_i)$ samples from all classes. This method thus under-samples data [118]. It is also possible to add a number of duplicate samples for each class so that the number of samples for each class is equal to $N_l = max(N_i)$. Such methods over-sample data [122].

A more robust approach called *roughly balanced bagging* selects all available minority-class samples while a portion of majority class is chosen based on the use of negative binomial distribution (NBD) [119]. The method is about creating a nearly (not exactly) balanced number of samples for every class in the datasets. It was shown that this method is more effective in classifier learning mechanisms than the more popular alternatives, like under-sampling, and over-sampling [119]. The reason for this improvement in performance cannot be proved theoretically. Intuitively, as we are dealing with a classification problem, with more samples in a majority class than in other classes. So by providing more training samples in the majority class this will bias the classifier towards classifying an unknown sample in the direction of the majority class. In the case of over-sampling, or under-sampling, this bias is reduced thus informing the classifier that every unknown test sample would have a likely possibility of being majority or a minority class. But this is clearly not what the underlying distribution of the data is, and therefore it makes sense to bias the classifier, towards the majority class, by providing more training samples in the majority class. Obviously such

a situation of having more majority class samples than minority classes, and the negative binomial distribution is a good probability distribution for such an application, as it provides more samples in the direction of selection of majority samples in a controlled manner.

Another approach to sampling creates distorted versions of input samples with the aim of making changes to the input so as to introduce diversity and variance to a training data set. Such approaches are commonly used in image classification (e.g., see [123, 124]). The input image is modified and distorted in some ways such as adding noise, changing the color, rotating, shearing or deleting some of its parts before sending it to the classifier for training. This procedure is recursively applied through the learning process so that at each training iteration, the distortion procedure is engaged to create further images that differ from the original ones. This kind of sampling has proves qualitatively very effective if the level of distortion is set appropriately [124, 125, 126].

A common approach to sampling is to use some form of random data selection from an original dataset in order to create a training set. But, in practice, many input samples do not contribute to the robustness of the classifier. Random selection methods may thus select samples that are not particularly useful to the training algorithm. Similarly, mechanisms that enlarge a dataset via the insertion of distorted samples are commonly ignorant to whether a generated sample would lead to enhancing the robustness of a learning system or not. As a result, those methods insert samples that are of no additional value to the classifier which is thus leading to unnecessary increase in computational demand. The main point we wish to make is that when some training samples are not useful for the classifier then it would be useful to have an algorithm which does not select such samples randomly. In other words, the value of selected samples to a classification system should be taken into account during sampling.

The small size of the PA datasets are likely to lead to overfitting when building a model. This thesis explores an idea to creating distorted samples in a fashion which targets the

robustness of a classifier. The basic idea is to introduce distorted samples in regions of the feature space where missclassifications are most likely. The newly generated samples are thus to improve the ability of a classification system to discriminate the pattern classes.

There are two main challenges that need to be overcome:

1. We first have to identify the subspaces where confusions between pattern classes are most likely. The hypothesis is: The closer a sample (or a group of samples) is to samples (or a group of samples) from another class the greater the risk of misclassification in the feature space between those samples (or group of samples). The question could be addressed via a proximity matrix. But due to the sparsity issue which arises out of the few samples in a high-dimensional feature space a proximity matrix in the high dimensional space would not be an appropriate approach. We instead engage a dimension reduction and visualization technique, the HRSOM. The HRSOM is ideally suited for identifying structure and proximities between samples in different pattern classes. We specifically engage the HRSOM algorithm to enhance granularity of the mappings and thus enhance the precision of results. We further process these mappings by engaging the DBSCAN clustering algorithm. DBSCAN will label each sample as either *core*, *border*, or *noise* points as was described in Section 2.2.1.2. The role of each sample can thus be identified. Core points correspond to samples that are embedded within a cluster of similar samples. Border points are marginal samples which are located at the edge of a cluster. And noise points are isolated cases of samples. Since we wish to identify the proximities between clusters samples in different classes and hence we alter the DBSCAN algorithm such that clusters are identified for each of the pattern classes.

2. The HRSOM will reveal the proximity of samples to each other and DBSCAN will reveal the role of each sample. Furnished with this information we then wish to create new samples in regions where confusions (between pattern classes) are most likely.

Since samples at a border of a cluster would be closest to samples from a cluster of samples in another class it should be best to create more samples in the border regions of clusters that are in close proximity and belong to different classes. When creating a distorted version of border samples then we need to take case that the newly generated sample will not fall into the feature space of samples from another class. To achieve this we first select a sample at a border region, find a set of nearby samples that belong to the same cluster, then create new samples via interpolation of the first sample with those nearby samples. The approach thus increases the density in regions of the feature space where samples from different classes are closest to each other.

A set of base learners is engaged to further enhance the representative value of the approach. The base learner we chose is a deep multilayer perceptron [127], in which the number of hidden layers and the number of neurons can be adjusted to fit the particular classification problem. The reason to why we choose a set of base learners is that this allows us to present typical results via averaging of results.

The contributions of this chapter can be summarized as follows:

1. The introduction of an unsupervised clustering mechanism consisting of a HRSOM and DBSCAN, which then is used to group input samples into a number of groups. We introduce a supervised version of DBSCAN since we wish to identify data clusters that belong to different classes. The approach will reveal proximities between pattern classes the insight of which is used later to identify samples for the selective based sampling approach.

2. The introduction of distortion based sampling method which we named *Synthetic Sampling Ensemble Network* (SSEN). The method automates means by which the cardinality of samples is increased in the areas of a feature space where confusion between pattern classes are most likely. A learning system would thus be provided with more samples in regions where the various pattern classes are closest in feature space.

The approach thus encourages the robustness of a classification and hence reducing likelihood of missclassifications.

This thesis will validate the proposed approach via an application to the PA2012 and PA2014 datasets, and to the intrusion detection dataset. The thesis will find that the approach is very effective in enhancing the generalization performance than other popular approaches used in the literature [115, 117, 118, 116, 119].

The rest of this Chapter is organized as follows. Section 6.3 describes briefly the model architectures. Section 6.3.2 and Section 6.3.1 introduce two supervised algorithms, namely synthetic sampling ensemble network (SSEN) and the supervised Dbscan algorithm, respectively. The experimental results of sampling techniques are presented in Section 6.4. Experimental results for intrusion detection data are given in Section 6.5. Conclusions are drawn in Section 6.6.

6.3 Model architectures

The SSEN will engage the HRSOM algorithm as was described in Chapter 5. The SSEN will furthermore utilize a modified version of the DBSCAN algorithm as described in the following.

6.3.1 The supervised DBSCAN

The DBSCAN algorithm was presented in Section 2.2.1.2. When deployed to the mappings produced by the HRSOM the algorithm would group points that are closely packed together or in dense region. The algorithm also identifies marginal points and outlier points that are located isolated in low-density regions. DBSCAN is one of the most popular supervised clustering algorithms in data mining since it can identify clusters of any shape, and since it can identify noise points and outliers.

Algorithm 1 The Supervised DBSCAN algorithm

Input:
Database DB
Neighborhood radius eps
Minimum number of points $minPts$ to form a dense region
Number of classes/clusters Cls
A HRSOM resulting activation map M of all training samples $P \in DB$

for each class C in Cls **do**
  for each point P in DB and $P \in C$ **do**
    Neighbors N = FindNeighbors(M, P, eps)
    if Density check $|N| = 0$ **then**
      label(P) = $C\_Outlier$
    else if Density check $|N| < minPts$ **then**
      label(P) = C_Border
    else
      label(P) = C_Core

for each pair of core points $P1, P2 \in DB$ **do**
  $C1 = Classlabel(P1)$
  $C2 = Classlabel(P2)$
  if $C1 \# C2$ **then**
    Neighbors N = FindNeighbors($M, P1, eps$)
    if $P2 \in N$ **then**
      label($P1$) = $Core\_overlapped$
      label($P2$) = $Core\_overlapped$

The SSEN algorithm will apply sampling to individual pattern classes. The class hence needs to be taken into account when computing the clusters by DBSCAN. Another reason is that by considering the class label of each sample when grouping samples into clusters then it becomes possible to identify whether pattern classes overlap. This is helpful in the group-based sampling approach that will be presented in Section 6.3.2. We propose the supervised DBSCAN algorithm as follows.

The following description of the supervised DBSCAN algorithm is based on the description of the unsupervised DBSCAN algorithm in Section 2.2.1.2 with a slight simplification: the number of points in the circle of radius eps which constitute noise/outliers is 0. Consider a set of points in a mapping space, and the two DBSCAN parameters, the radius eps and the $minPts$ (the minimum of points located in a circle defined by the radius), the points are clustered into different groups. In Algorithm 1, FindNeighbors is a function that finds

all the neighboring points given a point P, on a map M of a trained HRSOM and the given radius eps. Samples of individual class labels are applied to DBSCAN once, resulting in three sets of data points: core, boundary/border and noise/outlier points. After all class samples are separated, a final scan through all the samples on the map is deployed in order to determine the overlapped points between any two classes. These data points are named as core-overlapped points. Ultimately, samples are clustered into four main sets: overlap core points (OC, labeled as $Core\_overlapped$ in Algorithm 1) separate core points (SC labeled as C_Core in the algorithm), border points (BD, labeled as C_Border in the algorithm) and outliers (OUT, labeled as $C\_Outlier$ in the algorithm. It should be noted that in this method, the number of clusters determined by DBSCAN is equivalent to the number of class labels for the classification problem.

The use of the supervised DBSCAN serves two purposes. First, it helps to group input samples into a number of clusters denoted as OC, SC, BD and OUT for individual classes. The groups of samples will be useful for group-based sampling in the SSEN algorithm. Secondly, the resulting clustering produces a clearer visualization with more intrinsic topological information on the two-dimensional map of the HRSOM. DBSCAN is thus also very useful for the visualisation of results in this chapter.

On a side note: In theory, the supervised DBSCAN algorithm could have been applied to the raw data. However, the sparsity of the data space is amplified by the separate treatment of the different pattern classes and would thus hamper attempt to group samples. The HRSOM produces a much more compact representation of the samples. Moreover, the two-dimensionality of the display space allows a visual inspection thus leading to a better understanding of the data, and it allows for the visualization of results. Hence the application of DBSCAN to the mappings of the HRSOM is essential.

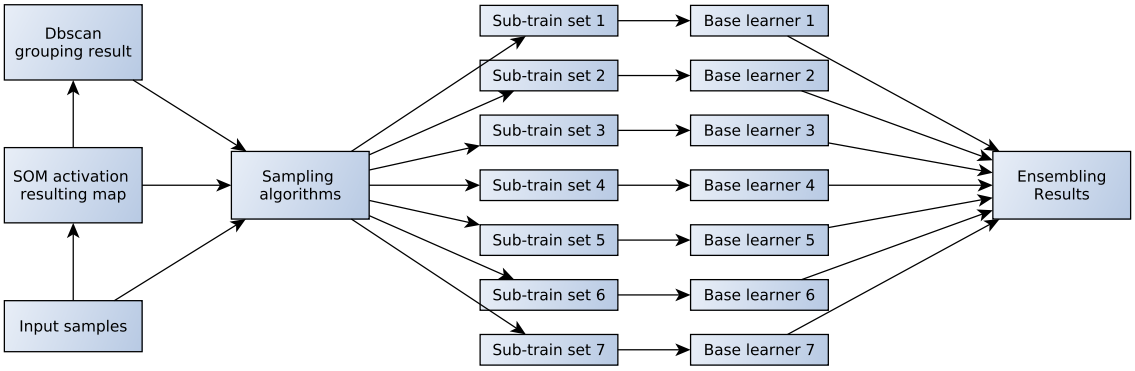


Figure 6.1: The SSEN model illustration.

6.3.2 The SSEN learning model

The SSEN algorithm is presented in Algorithm 2. The SSEN model is an ensemble learning system which uses a number (p) of base learners (here MLPs) on differently sub-sampled datasets. The final results is then the average over all the base learners outcomes.

The SSEN algorithm is based on the HRSOM activation map (M) for generating the new input samples, hence, a HRSOM needs to be trained first, in order to obtain the required activation map. The synthetic sub-dataset is then created for each base learner (L) by using the activation map. The searching radius (α) is used to find the number of neighborhood samples. The parameter $range$ controls the level of distortion. The supervised DBSCAN creates a number of groups G consisting of border samples. Note that only samples belonging to the set G are being sampled.

The purpose of using the HRSOM mappings is to find a number of nearest neighbors corresponding to samples in the original training set. Those will be used to create new synthetic samples by steps shown in line 3, line 4 and line 5 of Algorithm 2.

The SSEN is a type of ensemble learning method. The reason for selecting a MLP as a base learner in this thesis is that MLP is a excellent and well understood representative of connectionist learning system. The MLP will suffice for demonstration purposes. One can consider other model architectures here, however this is beyond the scope of this thesis.

Algorithm 2 The SSEN algorithm

Input:
Original training set T
Dbscan-based Group of samples to be used in sampling G
Base learning model L
The value decides the level of distortion $range$, in percentage
Number of classes n and corresponding numbers of input samples in each class $c_1, ...c_n$
Desired numbers of input samples in each class d
The searching radius α
A HRSOM result activation map M of all training sample $x \in T$

for $k = 1$ **to** p = number of base learner L **do**
    New training set $T_k \leftarrow T$
    for $i = 1$ **to** n **do**
        for $j = 1$ **to** $d - c_i$ **do**
            (1) Randomly select sample $x \in Class_i$ so that $x \in G$
            if $\nexists x \in G$ **then**
                $T_k \leftarrow x_s$
                continue
            (2) Find the α-NN $x_{r_1}, ..., x_{r_m}$ samples to x on M.
            (3) Randomly select a x_r in m nearest neighbors
            (4) Create synthetic sample x_s, $a \in x - attributes$ using
            (5) $x_s[a] = x[a] + rand(0, 10) * (x[a] - x_r[a]) * range$
            (6) $T_k \leftarrow x_s$
    (7) Train L_k on sampled data T_k to output O_k
Averaging is applied to p output O to get final outcome.

Some notes on the deployment of the SSEN model are as follows:

1. The learning problems: it should be reasonable to show it works on two completely different domains. In this chapter we have chosen these two domains, namely physical activity recognition and cyber-security intrusion detection problems. For this purpose we use the PA2012 and PA2014 datasets since both are small yet differ significantly in size. The PA2016 dataset was not available at the time when the experiments were conducted. The second set of data is related to cyber security which includes the intrusion detection or the UNSWNB15 dataset a much larger dataset.

2. The HRSOM uses an unsupervised learning mechanism to project all samples onto an activation map, so that similar input samples will be mapped onto nearby locations on the map. The resulting activation map is the mapping coordinates of all input samples. An input sample can be referred to a particular location on the map. In order to locate

the neighborhood samples of a given input sample within a given radius, one can use the resulting map for its reference. In particular, all the samples mapped within the radius region from the given input sample are named as neighboring samples.

3. Using the results of SOMs in the SSEN model algorithm, especially in line 2 to line 5 of Algorithm 2. In particular, in line 2, one would find the α-NN $x_{r_1}, ..., x_{r_m}$ samples to x on M (which are to find all the neighboring samples of x on the SOM resulting map). Then line 3 would randomly select a x_r in m nearest neighbors of x and in line 4 would create a synthetic sample x_s based on the calculation of every attribute of the input sample vector.

4. The supervised DBSCAN creates different groups G of samples. The groups G are then used in line 1 in the algorithm. The use of supervised mode will help us identify the class label as well as the group label of any samples on the activation map.

5. The sampling techniques will make use of all training samples, the activation map of the SOM, and the grouping of the mappings by the supervised DBSCAN method. The techniques thus include group-based, class-based and range-based approaches. A number of sub-training sets are created and a corresponding number of base learners are trained on the respective sub-training set. The final result would be the averaging results of all based learner's one.

6.4 Experimental results: Physical activity recognition

This section uses the PA2012 and PA2014 datasets which were described in Section 3.1.2 and Section 3.1.3 respectively. Each of these datasets is split into three roughly equal parts. One part is used for training purpose, the second is for validation and the rest is used as testing set. The training model is learned based on the training set, tuning its parameters and

adjusting the feature set options based on the validation set and finally testing the generalization performance on the test set.

The following presents experimental results for different sampling approaches. For the experimental setting, the number of base learners used by the SSEN is varied from 5 to 30. Each MLP is configured with two hidden layers, the number of hidden units is chosen within $\{15, 25, 37, 56, 88\}$, the learning rate $\alpha = 0.0001$ and is trained for $50,000$ iterations using standard back-propagation. The experimental results are averaged and the standard deviation of results over all MLPs is reported. For the group-based sampling, the synthetic sampling is only applied to the particular group indicated in the experiment. For example, the experiment set of the border points group will give the SSEN experimental result based on applying synthetic sampling on border points, but not for outlier and core points.

We analyse the SSEN incrementally as follows. Section 6.4.1 uses the SSEN algorithm as presented before. Then Section 6.4.2 uses the SSEN algorithm with the group selection feature disabled. But considers class memberships when creating new samples. Then Section 6.4.3 disables both the group selection feature as well as ignoring class memberships. This mode of presentation allows us to investigate the role and impact of the components of the SSEN algorithm.

6.4.1 Group-based sampling

The group-based sampling approach aims to determine the importance of these groups in solving the classification problem. The sampling is applied to individual groups (border points, outlier points, overlap core points or all core points which is the combination of groups overlap core and separate core points). The reason, why the border and outlier points are not chosen in the final combination of points, is because outlier points are considered as noise in terms of clustering discipline and needed to be removed from the training set, while the border points are considerably sparse, and empirically do not contribute to the model's

Table 6.1: The SSEN performance using the group-based sampling approach.

Group-based Sampling	**Model performance**			
	Accuracy	**Avg-Precision**	**Avg-Recall**	**Avg-F1**
PA2012 data				
Border points	0.9000[0.0020]	0.8871[0.0024]	0.8877[0.0028]	0.8874[0.0027]
Outlier points	0.9009[0.0032]	0.8890[0.0037]	0.8898[0.0049]	0.8898[0.0049]
Overlap core pts	0.9028[0.0009]	0.8880[0.0038]	0.8892[0.0033]	0.8875[0.0038]
All core points	**0.9046[0.0018]**	0.8903[0.0021]	0.8903[0.0021]	0.8903[0.0021]
PA2014 Hip data				
Border points	0.8708[0.0009]	0.8537[0.0023]	0.8612[0.0022]	0.8572[0.0009]
Outlier points	0.8767[0.0073]	0.8554[0.0064]	0.8700[0.0069]	0.8621[0.0066]
Overlap core pts	0.8816[0.0052]	0.8632[0.0042]	0.8767[0.0075]	0.8695[0.0056]
All core points	**0.8865[0.0046]**	0.8680[0.0022]	0.8821[0.0079]	0.8744[0.0049]
PA2014 Hip+Wrist data				
Border points	0.9081[0.0020]	0.8834[0.0015]	0.8970[0.0030]	0.8896[0.0021]
Outlier points	0.9083[0.0016]	0.8829[0.0023]	0.8956[0.0023]	0.8887[0.0020]
Overlap core pts	0.9101[0.0026]	0.8850[0.0025]	0.9005[0.0025]	0.8921[0.0023]
All core points	**0.9108[0.0017]**	0.8858[0.0011]	0.8989[0.0015]	0.8919[0.0026]

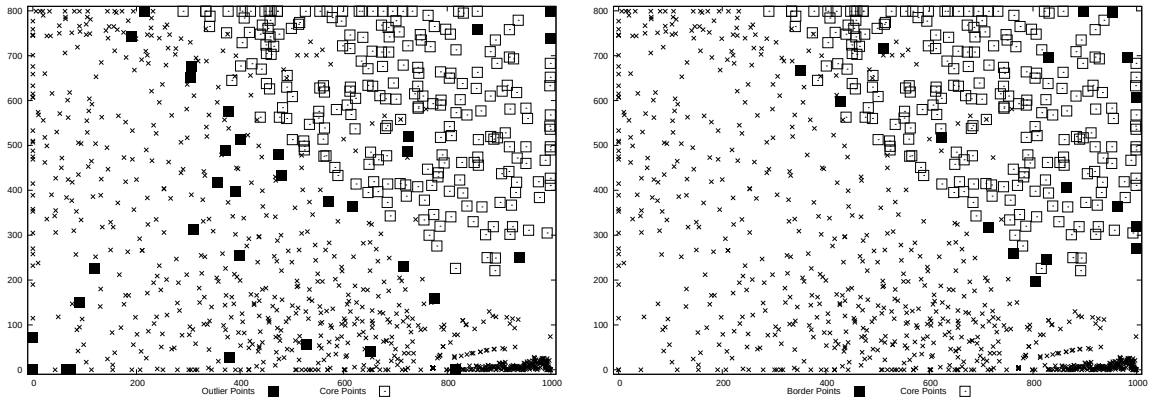


Figure 6.2: An example of outliers and borders.

performance. A combination of core points and border points (and/or outlier points) was found not better than the core points by themselves in terms of SSEN classification results.

Results are summarised in Table 6.1. Given that the ACC is the most important assessment indicator for this learning problem, it can be seen that the group of ”all core points” leads to a better accuracy than the other groups. The results are consistent for all datasets, though the improvement in accuracy for the best choice of group (all core points) is not re-

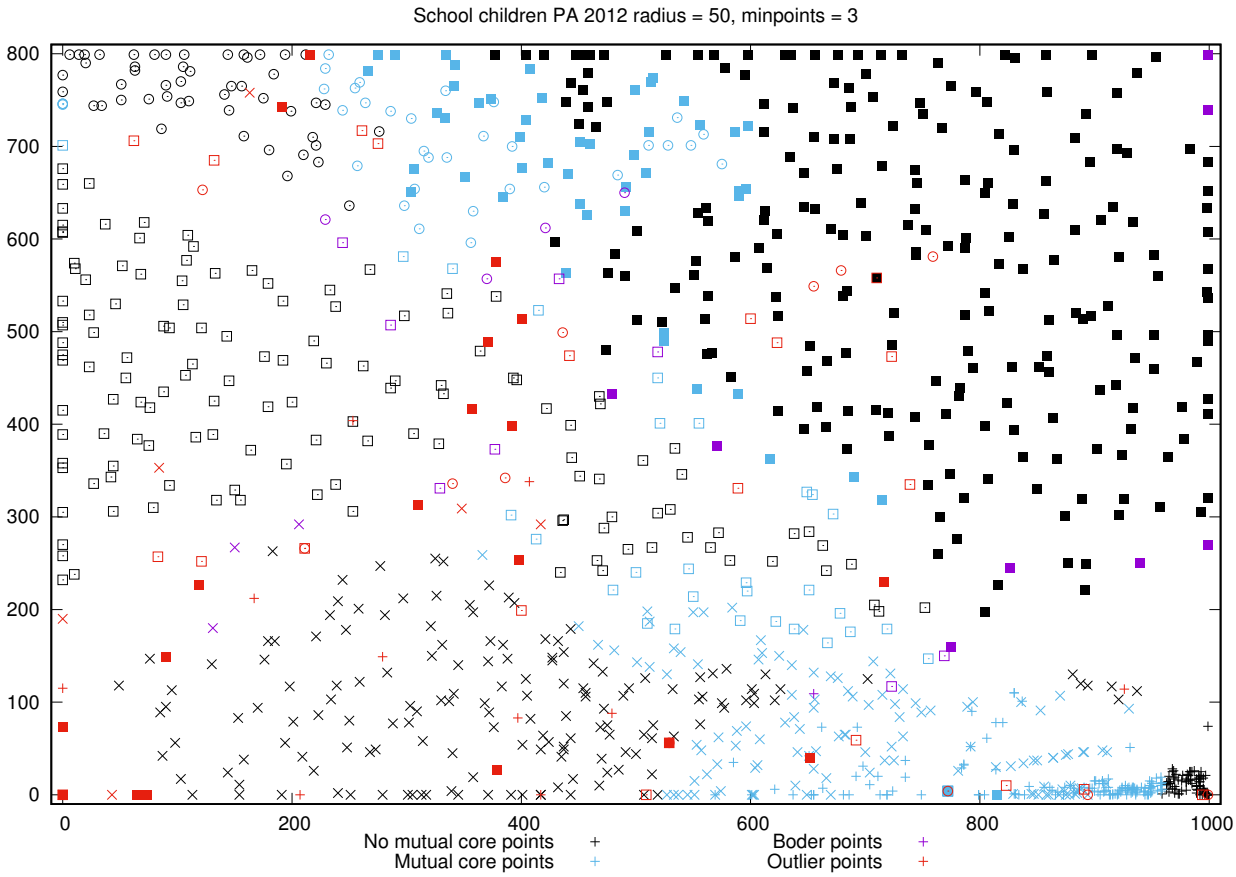


Figure 6.3: The mappings of the samples when using an HRSOM of size 1000x800 and the PA2012 dataset.

ally significant. The improvement in accuracy ranges from 0.3 to 1.0% when compared with the worst choice of groups for SSEN model. The experiments revealed that the use of all core points would result in a better classification performance. This indicates that DBSCAN helps to remove unhelpful data points in the SSEN model. The finding is important in the term of selective sampling algorithm.

To gain a better understanding of the results we need to look at the distribution of the data points and the output of DBSCAN. Figure 6.2 presents the outlier points versus core points (on the left) and border points versus core points (on the right) on the activation map of size 1000×800 for all samples in class "light activities and games" (treasure hunt, collage, clean up) in the PA2012 dataset. Points shown using square shapes belong to the class "light

activities and games" whereas samples which belong to any of the other classes are shown using crosses. It can be seen that the noise points can be considered outliers since they are located far from their class center density region.

Similarly, Figure 6.3 presents all the class sample points from the PA2012 dataset on the activation map. Each class is represented by a unique symbol. The mapping of the sample classes can thus be seen. The no-mutual/separate core points and mutual/overlapped core points are represented in black and blue color, respectively. It is observed that the border and outlier points are distributed relatively sparsely on the map. This can explain why these two groups do not contribute to improve generalization accuracy of the SSEN algorithm.

6.4.2 Class-based sampling

The previous group-based sampling did not consider the class label of the samples. This means that among all the neighboring samples, a sample of any classes could be selected. This can make a newly created sample biased towards samples of a different class, thus making it harder to discriminate the pattern classes. In this section, the class label will be taken into account so that when randomly selecting a sample within the neighborhood, we only select a sample of the same class as the class label of the given sample.

In the following experiments, the group selection step is not applied, meaning that all the samples are used by the SSEN model though class memberships are taken into account. We compare two approaches, one for the case that only sample of the same class is selected and secondly for the case that any sample of another class is selected.

Table 6.2 summarizes the results. It can be observed that for all cases the generalization accuracy is best when selecting samples from the same class. This is an expected result. Though the difference in result to those obtained from selecting samples from another class is not very significant. This is somewhat unexpected but may be attributed to pattern classes which generally overlap (as could be seen in Figure 6.3). It is interesting to observe that class

Table 6.2: The SSEN performance using class-based sampling approach

Direction-based Sampling	**Model performance**			
	Accuracy	**Avg-Precision**	**Avg-Recall**	**Avg-F1**
PA2012 data				
same class	**0.9047[0.0014]**	0.8927[0.0012]	0.8927[0.0011]	0.8927[0.0019]
other class	0.9019[0.0017]	0.8907[0.0013]	0.8907[0.0012]	0.8907[0.0013]
PA2014 Hip data				
same class	**0.8844[0.0048]**	0.8634[0.0048]	0.8808[0.0058]	0.8894[0.0052]
other class	0.8772[0.0051]	0.8564[0.0053]	0.8748[0.0050]	0.8898[0.0051]
PA2014 Hip+Wrist data				
same class	**0.9081[0.0008]**	0.8832[0.0032]	0.8961[0.0017]	0.8892[0.0016]
other class	0.9061[0.0018]	0.8813[0.0026]	0.8964[0.0028]	0.8882[0.0026]

based sampling did not improve the results when compared to using group based sampling.

6.4.3 Range-based sampling

The final approach to sampling is the range-based one. The $range$ parameter appears in the SSEN algorithm to indicate the level of distortion in the synthetic sampling step. In the aforementioned experiments, the $range$ was set to a default value 1. In the following experiments, the group selection feature is disabled and the class-based sampling is not used.

We will also investigate the effectiveness of selecting the right value of $range$ on the SSEN model's classification performance. By changing the $range$ value from 0.1 (or 10% of proposed distortion) to 1 (or 100% of proposed distortion). The higher the $range$ value is, the more distortion to feature vector is added. When $range$ = 1, we found that this is a significant distortion so that should be the maximum threshold being proposed.

Table 6.3 below only shows several value of $range$. We found that the right choice of $range$ value can help to improve the generalization accuracy by 0.4% to 0.6% on all datasets. We found that $range = 0.5$ is the best choice for all three datasets. This indicates that the random distortion of the input feature could be helpful if one know how much the change should be. Too much distortion would not the right way to add randomness to a

Table 6.3: The SSEN performance using range-based sampling approach.

Direction-based Sampling	Model performance			
	Accuracy	**Avg-Precision**	**Avg-Recall**	**Avg-F1**
PA2012 data				
range = 1.0	0.8991[0.0016]	0.8860[0.0038]	0.8860[0.0035]	0.8860[0.0032]
range = 0.5	**0.9065[0.0034]**	0.8948[0.0037]	0.8948[0.0037]	0.8948[0.0037]
range = 0.3	0.9037[0.0014]	0.8920[0.0021]	0.8920[0.0021]	0.8920[0.0017]
range = 0.1	0.9028[0.0031]	0.8914[0.0020]	0.8914[0.0041]	0.8914[0.0020]
PA2014 Hip data				
range = 1.0	0.8780[0.0052]	0.8569[0.0054]	0.8758[0.0047]	0.8906[0.0050]
range = 0.5	**0.8819[0.0036]**	0.8564[0.0040]	0.8752[0.0040]	0.8900[0.0039]
range = 0.3	0.8778[0.0035]	0.8576[0.0041]	0.8737[0.0042]	0.8900[0.0041]
range = 0.1	0.8778[0.0030]	0.8565[0.0036]	0.8738[0.0039]	0.8894[0.0037]
PA2014 Hip+Wrist data				
range = 1.0	0.9085[0.0023]	0.8832[0.0059]	0.8962[0.0019]	0.8892[0.0041]
range = 0.5	**0.9116[0.0023]**	0.8860[0.0013]	0.9016[0.0020]	0.8932[0.0015]
range = 0.3	0.9097[0.0016]	0.8846[0.0014]	0.8999[0.0020]	0.8917[0.0011]
range = 0.1	0.9062[0.0026]	0.8805[0.0021]	0.8952[0.0023]	0.8873[0.0021]

model training since this might have ruined the training feature space.

6.4.4 Comparing with other sampling approaches

Table 6.4 which compares the best result of three sampling approaches. In Table 6.4 the line "Traditional ANN" corresponds to baseline results which are obtained by deploying the MLP without any sampling. "RB Ensemple MLPs" corresponds to a well-known roughly balanced (RB) ensemble MLPs technique [119]. Also shown are results (where available) from deploying vanilla SVM for comparisons.

The SSEN results shown in Table 6.4 were obtained by using full SSEN algorithm (denoted as "group based sampling" earlier) and using range=0.5. The SSEN results shown in Table 6.4 confirm that increasing the range value to 0.5 also improves the result for group based sampling.

When compared to the ANN baseline results we find that the proposed SSEN algorithm improves the generalization ability by 2.7% to 10.6%. The improvement is greater the

Table 6.4: Comparing SSEN performance with other approach

Evaluation	Model performance			
	Accuracy	**Avg-Precision**	**Avg-Recall**	**Avg-F1**
PA2012 data				
Traditional ANN [128]	0.8840	-	-	-
RB Ensemble MLPs	0.9066[0.0011]	0.8943[0.0016]	0.8895[0.0012]	0.8950[0.0026]
SSEN model	**0.9111[0.0018]**	0.9000[0.0015]	0.9006[0.0013]	0.8990[0.0019]
PA2014 Hip data				
Traditional ANN [129]	0.8000	-	-	-
SVM model [129]	0.8400	-	-	-
RB Ensemble MLPs	0.8944[0.0013]	0.8756[0.0023]	0.8896[0.0014]	0.8800[0.0021]
SSEN model	**0.8990[0.0020]**	0.8773[0.0018]	0.8974[0.0017]	0.8865[0.0016]
PA2014 Hip+Wrist data				
Traditional ANN [129]	0.8100	-	-	-
SVM model [129]	0.8550	-	-	-
RB Ensemble MLPs	0.9105[0.0016]	0.8834[0.0019]	0.9021[0.0019]	0.8933[0.0023]
SSEN model	**0.9211[0.0015]**	0.8934[0.0015]	0.9117[0.0013]	0.9019[0.0014]

smaller the dataset. This thus confirms effectiveness and that the algorithm meets its design goals.

Table 6.4 also reveals that the SSEN generally outperforms the RB ensemble method. This is an interesting observation since RB sampling makes use of the binomial distribution to sample a relative number of individual classes so that the new sampling dataset contains roughly balanced number of samples for each class. Moreover, the number of MLPs for this ensemble method is 50 which is much greater than the number of MLPs used in the SSEN model. Yet, the SSEN is able to outperform the RB sampling method by 0.5% to 1% in accuracy. This is a surprise finding since the SSEN model only uses 20 MLPs as base learners.

SVMs often outperform MLPs because they compute decision boundaries that are located at the maximum distance between pattern classes. The SSEN significantly outperforms the SVM which implies that the optimal decision boundary is not in the center between pattern classes.

The findings in this Chapter show that the proposed SSEN is best for the PA classification

task. To verify whether the algorithm is sufficiently robust we apply it to another dataset which features significant different properties.

6.5 Experimental results: UNSW-NB15 data

6.5.1 Experimental setting

The experimental setting is similar to that for the PA classification problems though we only use group based sampling (the full SSEN algorithm with all features enables) here. The dataset was standardized to make each attribute mean zero and standard deviation one. The base learner MLP has been experimented with a number of different network activation functions, training duration and the number of MLPs for SSEN. Since this intrusion detection data is much larger in size when compared with the PA problems, the following will show the experimental results with respect to the best accuracies on the validation set (randomly extracted 10% from the training set which is separated from the testing set).

The base learner MLP was trained with the following settings: the number of hidden layers are selected within $\{1, 2, 3\}$. The number of hidden neurons on each layers was selected within $\{158, 118, 78, 50, 18\}$. The learning rate is set with smaller value than 0.001 and with the decay factor of 0.95% during the training process. The number of training iteration is set up to $5,000$. Note that the number of training iterations is much smaller than when using the PA data. The reason for this is that we use online training which updates the network parameter for each sample. Since the number of training samples is by orders of magnitudes larger and hence by reducing the number of training iterations the number of updates for each base learner will remain similar.

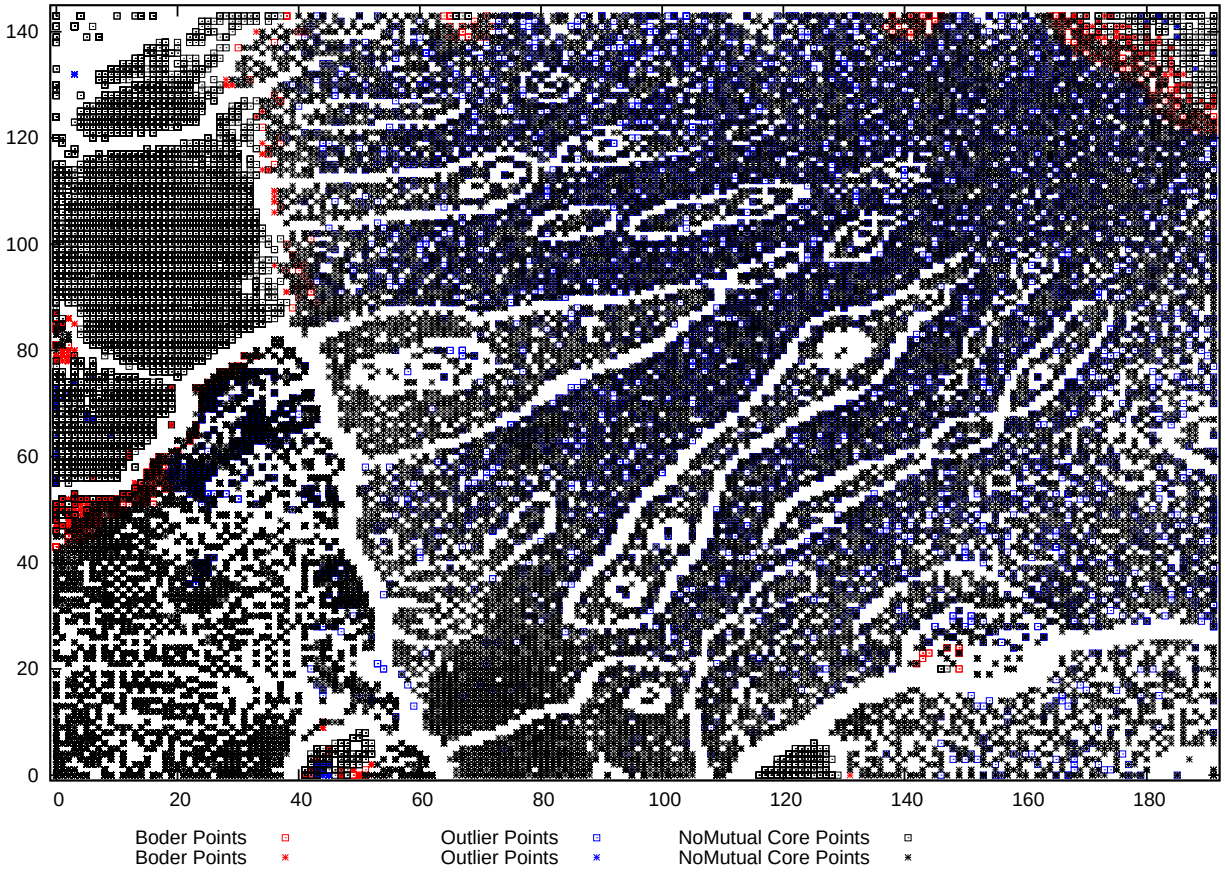


Figure 6.4: The mappings of the samples when using SOM of size 192x144 for part of UNSWNB15 dataset.

6.5.2 Experimental results

We first present the mapping of the training samples and the result of DBSCAN in order to obtain an overview of the learning problem. Figure 6.4 presents the activation map of a trained SOM of size 192×144. We also trained larger maps but chose this smaller map for visualization purposes. The mapping look extraordinary and very different to that observed with the PA datasets. It can be observed in Figure 6.4 that some of the pattern classes are organized in well distinct clusters (i.e. in the upper left corner and lower left corer of the map) as well as large regions that are organized in clusters of overlapping pattern classes. There is a clear distinction of border points which occur much denser than was observed for the PA data.

Table 6.5: SSEN performance with different network's settings

Parameters	Model performance			
	Acc	**Pre**	**Recall**	**F1**
Optimization functions				
Adam	83.50%	86.53%	83.50%	82.83%
RMSProp	85.85%	87.85%	85.85%	85.45%
SGD	**86.57%**	88.19%	86.57%	86.24%
Number of training epoches				
500 epoches	82.22%	84.81%	82.22%	81.54%
1000 epoches	84.20%	86.40%	84.20%	83.70%
2000 epoches	85.82%	87.46%	85.82%	85.46%
3000 epoches	85.45%	87.38%	85.45%	85.04%
4000 epoches	**86.62%**	88.16%	86.62%	86.30%
5000 epoches	86.45%	87.94%	86.15%	85.79%
Number of base learners				
5 learners	86.48%	87.81%	86.48%	86.19%
10 learners	87.01%	88.16%	87.01%	86.76%
20 learners	**87.69%**	88.61%	87.67%	87.46%
30 learners	87.28%	88.34%	87.27%	87.05%
40 learners	86.83%	88.05%	86.83%	86.57%

In the following, we will try SSEN with a number of different settings to show how much the generalization accuracy is affected by the changes the number of training iterations or the number of base learners within the SSEN.

First, given three different first order minimization algorithms, namely Adam (a first order gradient based optimization of stochastic objective functions) [130], RMSProp (a derivative of RProp) [131]and SGD (Stochastic Gradient Descent) [132]. Results are summarized in Table 6.5. It can be observed in Table 6.5 that the SGD minimization algorithm gives the best generalization performance. The difference in result is significant. SGD improves the accuracy by over 3% when compared to the popular Adam optimization minimization algorithm which performed worst on this dataset. It should be noted that each problem can be fitted to some extent by a different first order minimization algorithm.

For the second set of experiments we change the duration of training process. We vary the number of training epoches within $\{500, 1000, 2000, 3000, 4000, 5000\}$ and record

the generalization performance (on the validation set) at the end of each training run. Each experiment is repeated three times, starting from a different initial condition. Average results are reported. The result is shown in Table 6.5. It is found that when using more than 4000 training epochs this may lead to overfitting whereas the network is not fully trained when using less than 4000 training iterations. This implies that for this dataset 4000 epochs is the best one to use.

In a third set of experiments we vary the number of base learners in the SSEN. The number of MLPs is increased incrementally from 5 to 40. Results are shown in Table 6.5. It is again observed that SSEN might only require as many as 20 base learners since the best accuracy performance is obtained with 20 base learners, even though the difference is not significant.

6.5.3 Comparing SSEN with other approaches for the UNSW-NB15 dataset

Table 6.6 compares our experimental results with results obtained by using machine learning algorithms. Comparisons will be made between popular classification methods such as decision tree [133, 134], SVM [77], Naive Bayes [99] and some MLP based models [135] which are single model based algorithms, and the RB ensemble and the SSEN approach which are multi-model based algorithms.

The results are summarized in Table 6.6. It is found that the decision tree produced, a popular decision tree classification technique, one of the best results using the single model methods (i.e with the the model ensemble approach). The Decision Tree is outperformed only by a small margin by the SVM. For these experiments the SVM is trained with C (C is the parameter for the soft margin cost function) being chosen within [10, 100] and the gamma parameter (gamma is the free parameter of the Gaussian radial basis function) chosen within [0.01, 0.001]. The MLP was used with the same parameters as a single base

Table 6.6: Compare model's performance.

Models	**Model performance**			
	Acc	Pre	Recall	F1
Decision Tree [99]	85.56%	-	-	-
Naive Bayes [99]	82.07%	-	-	-
MLP [135]	81.34%	-	-	-
EM [135]	78.47%	-	-	-
SVM	85.74%	86.75%	81.37%	80.34%
MLP	85.01%	85.73%	80.90%	79.75%
RB ensemble	86.32%	88.15%	86.34%	86.81%
SSEN model	**87.69%**	88.61%	87.67%	87.46%

learner in the SSEN algorithm. The RB ensemble used 40 MLPs as its base learners, each base learner was trained on a sub-set sampled from original training set using the roughly balanced bagging approach.

It can be observed that the multi-model methods generally outperform single-model methods and that the SSEN produces best results by a good margin. The SSEN model improved the accuracy by 2.2% over the second best method which is quite significant for this UNSW-NB15 dataset.

6.6 Conclusion

This chapter introduced a new innovative sampling technique. The chapter has shown that the synthetic based sampling approach is effective in creating a richer training sample set for parametrized classifiers such as MLPs. The method is especially useful for the cases when the training set is quite small in size and does not encapsulate the properties of the unknown testing samples. It was shown that HRSOM and DBSCAN are effective in aiding SSEN to create samples that lead to a better discrimination of pattern classes. This in turn leads to a more robust classification system with improved generalization capabilities. A very useful side-effect is that the HRSOM and DBSCAN algorithms allow for visualizations of the data which helps to obtain a better understanding of the learning problem and is assisting in the

analyses of results. The approach was qualitatively demonstrated that its accuracy results are consistently much better than other sampling techniques. The findings were verified by using datasets with significantly different characteristics.

Chapter 7

Transfer learning

7.1 Preamble

This chapter investigates an alternative to dealing with the small PA datasets. Small datasets may not cover the feature space well enough to allow for a better discrimination between pattern classes. The question explored in this chapter is as follows: Can the knowledge gained from exploring the representations on the feature space of one PA dataset be used to enhance the class discrimination of another PA dataset. In other words, this chapter explores whether information from one PA dataset can be transferred to another dataset, using a technique called *transfer learning* [136] Whether transfer learning can enhance PA classifications will be explored.

7.2 Introduction

In the traditional machine learning applications, the training and testing data samples are assumed to be obtained from the same distribution in the same application domain. In other words, the input feature space share similar characteristics and possibly the class data distribution between the training and testing sets. In many real-world scenarios, however,

there are cases where the amount of labelled data is limited or costly to obtain. Hence, there have been efforts to design and train a classifier using available datasets from a related knowledge domain, or using related data that was more easily obtained [136, 137]. The underlining methodology discussed here is referred to as the transfer learning mechanism [136].

Transfer learning is inspired by the fact that a human being is able to recognize objects from a background knowledge obtained from related domains, or by only having learned from a few similar objects, not necessarily exactly the same identical shape. An example of knowledge transferability is the observation that it is faster to learn to play a piano when a person had prior knowledge on playing another musical instrument. Transfer learning is only applicable where the source domain is related to the target domain. One can not identify a car if one has never seen a vehicle. This is because there is an unique domain-independent feature when compared one domain with the others.

For PA recognition of children there might be several sources obtained at various times, of somewhat similar activities, which might have similar energy expenditure, but not necessarily similar visually. This is commonly called the source, or the background domain. Then, a classifier trained on data in the source domain, might be able to be adapted to provide the classification of data which is obtained at another time, from another cohort, or under similar recording conditions. This is often called the target, or target domain [44]. For example, the sources can be a data cohort collected from PA trials of a different cohort (i.e. adolescents). The source knowledge can be extracted from information contained in the accelerometer data movement sensor data, or data from the videos. Such information can be used to train a classifier using the information in the source domain. Then, for a target domain, data may be collected from different cohorts, like young pre-school children, performing activity which might have similar energy expenditure. The suggestion is to adapt the trained classifier from the source domain to classify the data in the target domain. This

idea is attractive since it is easier to collect data from PA trials involving adolescents than pre-school children. The reason for that is the young children are not as disciplined and are less likely to follow protocol in an experimental setting in the laboratory. This thesis does not have access to PA data involving adults. But we do have access to PA data from two age-cohorts the 2012 dataset collected in Brisbane and two datasets 2014 and 2016 datasets which were collected from young pre-school children in Wollongong. The classifications of activities into categories is based on energy expenditure levels rather than on what the physical activity is. The 2012 dataset consists of data collected from 100 participants, while the 2014 and 2016 datasets contain 11 and 16 participants respectively. Therefore, it would be an interesting question: is it possible to train a classifier on the 2012 dataset, and then adapt the classifier to classify the 2014 dataset, or the 2016 dataset? If this can be done, what might be the improvement in generalization accuracies using the 2014 dataset, or the 2016 dataset alone by itself as a standalone dataset.

There exist numerous work applying transfer learning to recognition tasks including text sentiment classification, image classification, human activity classification, software defect classification, and multi-language text classification [136, 138, 139].

It is also popular to exploit transfer learning in deep learning though for different reasons [139]. The reason is that the source data is assumed to be sufficiently large for an effective deep training procedure. One can easily collect multiple sources of e.g. publicly available images about related domain to the target domain, e.g., recognition of images which might not be present in the source domain. It was largely recognized that a deep learning model being trained by using images or related images from the source domain can be adapted much more quickly to a given target image recognition task and is thus is able to perform better than a model which was trained from scratch based on the images in the target domain [44, 45]. However, when the training speed is not a major concern in this thesis given the small datasets, the primary aim here is to investigate whether knowl-

edge transferred from one PA dataset can improve classification performance on the target dataset.

There are two main types of transfer learning: Feature-based transfer learning and parameter-based transfer learning [140, 137]

- The feature-based transfer learning approach attempts to transform the features of source (X_s) into features of a target (X_t) domain. The corresponding features are called domain-independent feature. One example of feature-based transfer learning is by using augmented latent features in the feature space [140]. Another work is based on the spectral feature alignment which use the spectral feature clustering method [141].

- The parameter-based transfer learning is performed at the model level [137] The objective is to transfer parameters of the classification model from a model that was trained on data from the source domain to adapt the model to work in the target domain. After training a model on labelled data from the source domain, the trained model is then adapted as the initial model in the training on the labelled data in the target domain. In other words, the trained model in the source domain is used as an initial model in the training of the same model in the target domain, often only for a few training iterations. In this manner, the knowledge obtained in training on the source domain is preserved and modified by the training data in the target domain. The boosting method is an example of parameter-based transfer learning [137]. In the boosting method, the learning model is driven to focus on a sample if a model misclassified that sample or is decelerated if the model correctly classified it.

The parameter-based transfer learning approach, which is performed at the model level, will be explored in this thesis. The general approach taken in this thesis is illustrated in Figure 7.1

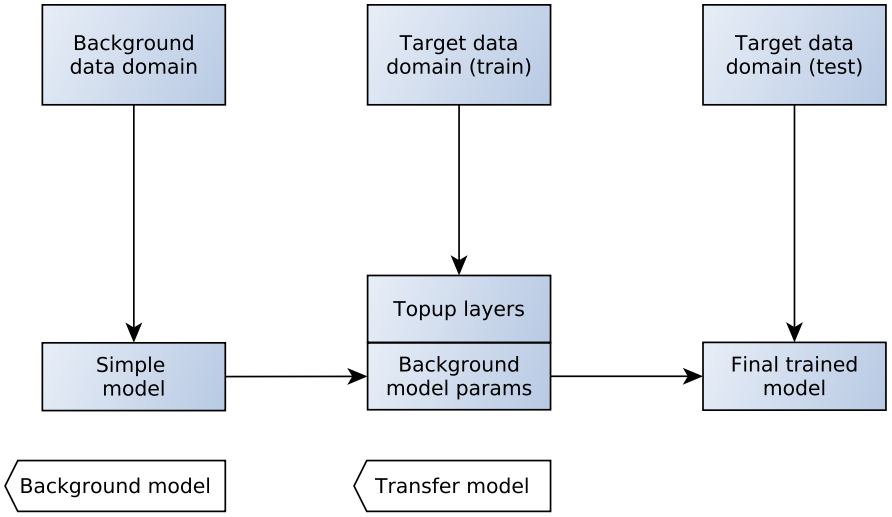


Figure 7.1: Transfer learning applied to physical activity recognition.

The Figure shows that the process commences by training a base model on the source domain. Once the base model is trained, its internal parameters are frozen. Then two options on creating the transfer model are explored.

Option 1: A number of new and randomly initialized layers of hidden neurons are added. The weights of these newly added neuron layers are then trained on data from the target domain. The transfer model is then evaluated on the test data from the target domain. We will refer to this architecture as the Expanded Transfer Learning (ETL) model.

Option 2: The FRPN (fully recursive perceptron network) deep learning architecture is added and its parameters are trained on data from the target domain. This creates a hybrid architecture consisting of a layered architecture to which a recursive model is appended. The trained transfer model is then evaluated on the test data from the target domain. We will refer to this architecture as the Stacked Transfer Learning (STL) architecture.

Here the knowledge obtained in the source domain is “frozen” in the base model, a vanilla

MLP. The knowledge on the target domain is adapted in the added feedforward layers (option 1), or added fully recursive layer (option 2), It is known that the FRPN is a data dependent expansion of the feedforward layers; in other words, a fully recursive layer can be expanded into a number of feedforward layers, and the depth of expansion is based on the incoming data. Therefore, option 1 can be considered as a fixed depth feedforward layers, while option 2 can be considered as a feedforward layer with variable depth, i.e., the depth is automatically determined by the data. Option 2 is inspired by the work reported in [79] and is presented for the first time in this thesis in the context of transfer learning. In this chapter, both option 1 and option 2 will be explored for transfer learning of PA datasets.

The rest of this Chapter is structured as follows. Section 7.3 presents the data preparation step and explains experimental settings. Experimental results are presented and analysed in Section 7.4. Conclusion will be drawn in Section 7.5.

7.3 Data preparation

7.3.1 Accelerometer cohorts

The datasets PA2014 and PA2016 will be used for the transfer learning experiments. The PA2012 dataset is not used because the PA classes differ significantly from those in the PA2014 and PA2016 dataset respectively. The PA2016 dataset consists of two subsets: One subset corresponds to data acquired by using the GeneActiv tri-axial accelerometer whereas the data in the second subset were acquired by using the ActiGraph tri-axial accelerometer. The PA2014 contains only data acquired by ActiGraph tri-axial accelerometers. Details of the types of activity trials performed for each dataset were presented in Chapter 4.

Note that the activities recorded in the PA2016 data differ somewhat from the activities in the PA2014 dataset. Moreover, the number of different activities in the 2016 dataset is nine compared with 12 activities in the PA2014 dataset. There are only three activities

that are comparable between the 2014 dataset and the 2016 dataset: Treasure hunt, Bean bag game and Story time. Some data alignment is thus required. This will be described in Section 7.3.2.

We will use the PA2014 dataset for training the base model i.e., we will consider the 2014 dataset as the source domain. The PA2016 dataset will be considered as the target domain. The PA2016 dataset is split into three parts of approximately equal size. One is used for training, one for validation and the rest serves as a test set. The non-overlapped window method as described in Chapter 4 is used for feature extraction. The window size is set empirically to 15 seconds. The features extracted include: for each accelerometer coordinate (x,y and z) we use the 5 percentiles, auto-correlation, entropy, average, standard deviation, average deviation, skew, curt and peak (13 dimensional feature vector for each accelerometer coordinate, which results in $13 \times 3 = 39$ dimensional feature vector for each input sample). The choice of features is motivated by work in [129] hence we use the same features as in [129]. We can use the combination of Hip+Left wrist for both datasets, so that the input dimension increases to $2 \times 39 = 78$. Hip+Left wrist data was shown to be the good choice of combining multiple accelerometer data [8].

7.3.2 Alignment of Pattern Classes

This section introduces a novel methodology of aligning pattern classes from two data populations. The methodology is quite generic and would work in situations where the data in multiple populations meet the following criteria:

- The populations are part of a classification problem featuring two or more classes.
- The data is normalized to within the value range $[-1, 1]$ with zero mean.
- The dimension of the features is the same in all populations.

The method has been developed to align pattern classes of the PA2014 and PA2016 datasets and will be used in this chapter for this purpose. We will thus not evaluate the method on other datasets.

Note that the pattern classes in both datasets are segmented based on the level of energy expenditure measured e.g., sedantary, light exertion, medium exertion, walking, and running. PA are categorized in five energy expenditure levels in the PA2014 dataset which differs from the 9 levels in the PA2016 dataset as shown in Chapter 4. There was very common approach to groups physical activities into 5 levels of energy expenditure which include: (1) Sedentary activities; (2) Light activities; (3) Moderate-to-vigorous activities; (4) Walking like activities; and (5) Running like activities [5, 3]. Thus, we should group these 9 activities into 5 energy expenditure levels as listed to make a consistent comparison and to enable the tranfer learning from one domain to the others.

The proposed approach aims at finding a common ground for grouping data into five categories and labelling them accordingly. The approach thus needs to be based on the level of energy expenditure of each PA. It is difficult to align activity classes by comparing feature vectors in a high-dimensional feature space. To simplify the task, this thesis will use the topology preserving characteristics of the HRSOM algorithm introduced in Chapter 5. The topology preserving characteristics of the SOM is as follows: any two feature vectors in the high dimensional feature space will remain close when they are being projected onto the two-dimensional display space [100]. Clustering of the projections is used to identify alignment of data groups. K-means clustering is engaged for this purpose. K-means clustering is similar to PCA and mean shift clustering, though the key difference is that K-means is a vector quantization method. K-means is extensively applied in signal processing, used for the cluster analysis in data mining [142]. K-means clustering aims to partition n input samples into k clusters in which each sample belongs to the cluster with the nearest mean. The nearest mean is called a prototype of the cluster. The value k, the desired number of

clusters, needs to be specified. Since we wish to identify the alignment of a fixed number (five) data groups and hence this makes K-means particularly appealing.

K-means is applied to the mappings of the HRSOM since (1) the 2D map offers visualization of results which aids observations and the understanding of results and (2) data dimension reduction reduces sparsity thus aiding K-means to identify dense clusters.

The procedures for applying this idea to align the categories between the 2014 dataset and the 2016 dataset are as follows:

- We train a HRSOM on the GeneActiv subset of PA2016.
- K-means is then applied to each HRSOM in order to identify $k = 5$ clusters. The HRSOM is thus trained on samples that belong to nine activities while K-means is performed to group the samples into five clusters. The result of K-means will reveal the correlation between clusters and the pattern classes. One should thus be able to identify how the pattern classes align with the K-means clusters. Comparisons with the two results should then reveal how the pattern classes are related.
- The pattern classes are aligned with a cluster by using majority-based assignment. This can be explained as follows. If a majority of samples from a particular class is clustered into cluster A, then all samples from that class will be assigned to cluster A and are then relabelled as class A.

The result of applying this method to the PA2016 dataset is shown in Table 7.1. From this Table it can be observed how majority-based assignment leads to an alignment with the clusters. For example, the samples from the PA "Lying down" is mapped to cluster-C while all other samples from that PA are found in cluster-E. All samples of that PA are then re-assigned to belong to the majority cluster-E and are labelled accordingly by using a unique ID (5 in this case). Notice that the ID is unique to each cluster. Since there are more PAs than clusters this means that some clusters will be the majority cluster for several

Table 7.1: Majority voting to assign nine classes to the 5 clusters.

PA trial	Cluster					New Class-Id
	A	B	C	D	E	
(1) Lying down	0	0	1	0	**139**	5
(2) Toys at table (free play)	0	**72**	20	28	20	2
(3) Story time	0	**103**	12	19	6	2
(4) Whiteboard	0	**114**	16	9	1	2
(5) Treasure hunt	2	3	11	**124**	0	4
(6) Pack Away	1	4	**135**	0	0	3
(7) Dance	19	10	8	**103**	0	4
(8) Bean Bag Game	**95**	12	18	15	0	1
(9) Captain is coming	**134**	5	0	1	0	1

PAs. For example, cluster-B is the majority cluster for the three PAs ”Free play”, ”Story time”, and ”Whiteboard”. Notice that the method re-aligned the categorization of the nine PAs. The categorization is very similar to the original categorization and we observe that new categorization organized PA according to the vigorousness of a PA. The method has thus shown its effectiveness in achieving the envisaged aim of realigning the pattern classes.

As can be seen in Table 7.1, nine activities are assigned into five clusters (from Cluster-A to Cluster-E, associated with indices from 1 to 5). To decide the class-id (class label for each activity), we need to get the index of the cluster with the largest value. For example, for (1) Lying down activity, the largest value of 139 is with Cluster-E, associated with the index of 5, thus we would assign this activity to New Class-id of 5. Similarly, for the (6) Pack Away activity, the largest value of 135 is with Cluster-C, associated with the index of 3, thus we would assign this activity to New Class-id of 3. The New Class-id here is relative since we need to arrange the classes in the energy expenditure order. That final stage should be involved with some heuristic perspective. In particular, we should assign the smaller class-id value to least energy expended activities and larger class-id value to more energy expended activities.

We analyse the results further by inspecting the projections of the data. Figure 7.2 shows the mapping of the PA2016 data on a trained HRSOM of size 600. The mappings are labelled

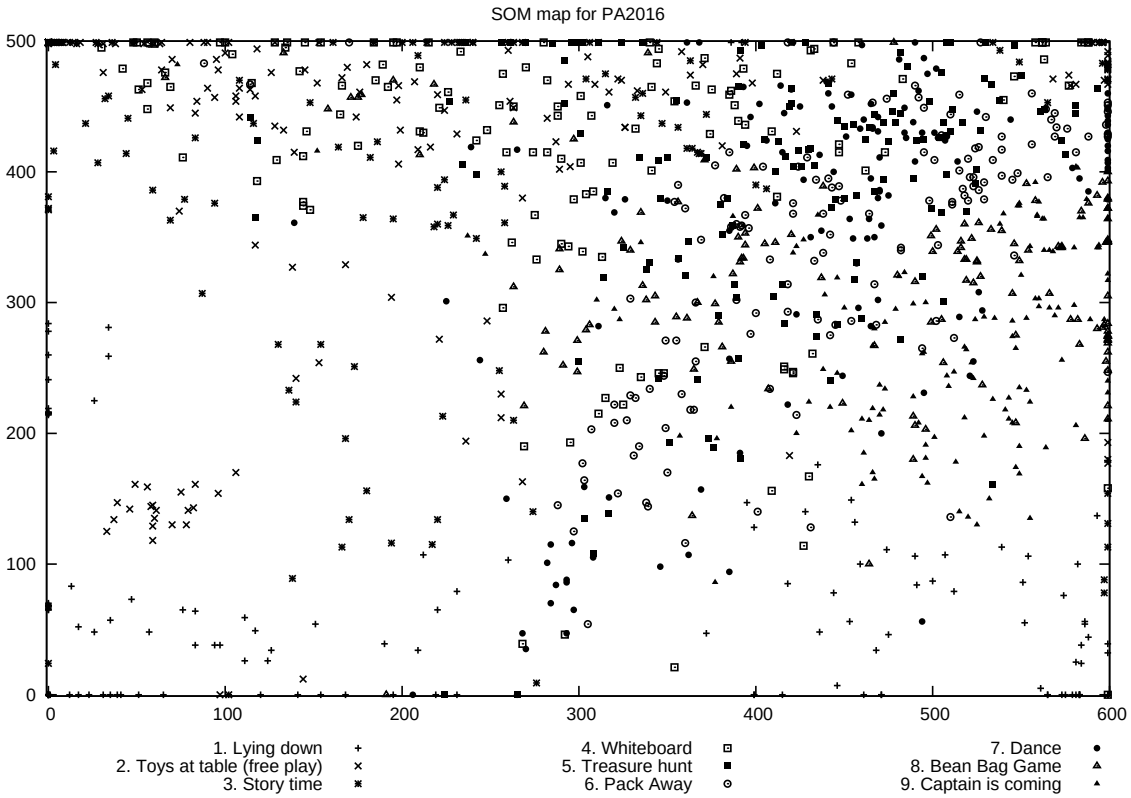


Figure 7.2: SOM mapping result.

according to the PA type of each sample. The HRSOM's mapping result is then used as the input for K-means algorithm.

Figure 7.3 illustrates the results of K-means clustering. It can be seen that the 5 clusters formed by K-means are relatively well separated and that the separation of the clusters is purer than was observed for the nine PAs. It is also observed that the clusters do largely correspond to the grouping of the data that can be observed on the SOM in Figure 7.2, in other words each model contributes to the final class annotation task.

The following observations can be made: (1) using both HRSOM and K-means with majority voting can help us to determine which PA trials to be assigned to which classes; (2) given the class label annotation shown in Table 7.2, one can observe some alignments between activities in each class in terms of level of energy expenditure; (3) the PA2016 data contains two datasets, namely GeneActiv and ActiGraph cohorts. The two datasets share

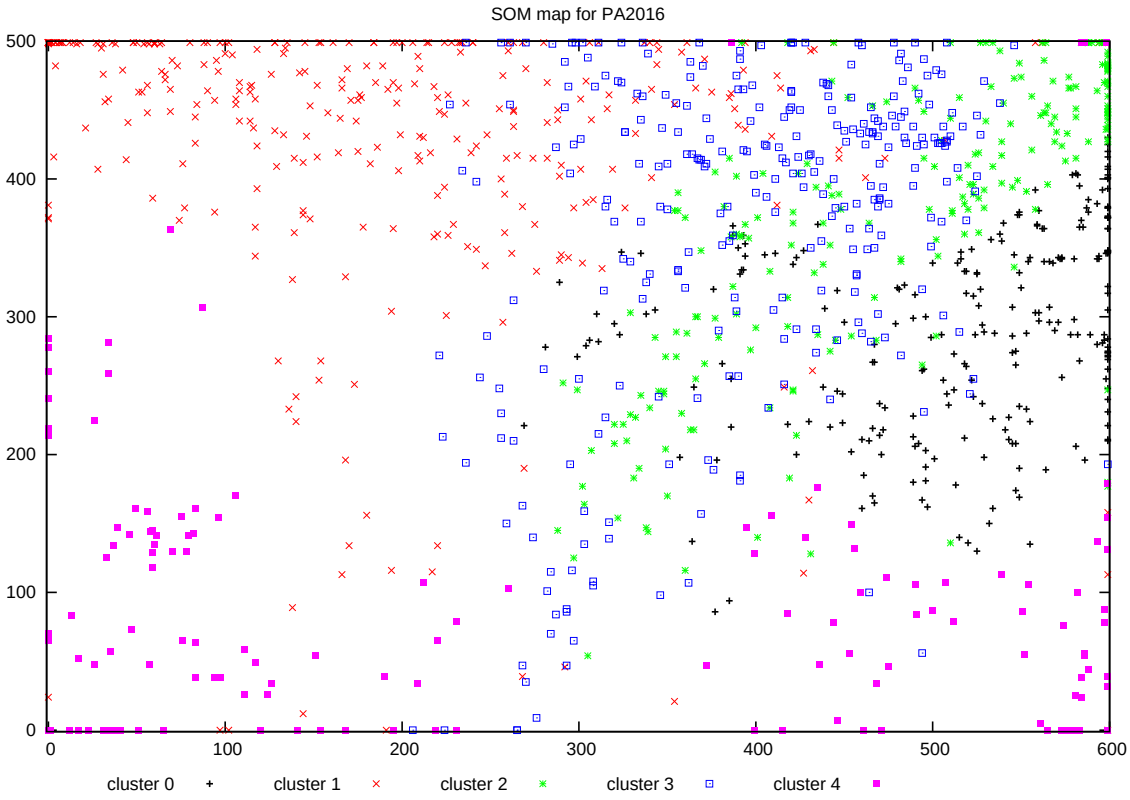


Figure 7.3: K-means with 5 clusters, mapped on SOM.

Table 7.2: Comparing class division between two datasets: PA2014 and PA2016

cID	PA2014 PA trials	PA2016 PA trials
(1)	Watching TV, Story time, Playing iPad, Quiet play	Lying down
(2)	Collage, Treasure hunt, Clean-up	Toys at table (free play), Story time, Whiteboard
(3)	Bean bag, Obstacle course, Bicycle	Treasure hunt, Dance
(4)	Walking	Pack Away
(5)	Running	Bean bag, Captain is coming

the same experimental settings and the number of physical activity trials. Hence it is not necessary to repeat the same experiments for ActiGraph dataset.

The result is that the PAs of the two datasets can now be aligned into five overarching pattern classes. Table 7.2 presents the results.

It can be seen from the Table that i.e. the activity ”Lying down” in the PA2016 dataset

aligns with the PAs "Watching TV", "Story time", Playing iPad", and "Quiet play" in the PA2014 dataset. All of these activities are of low intensity and it makes sense to have them aligned to the same class. A similar observation is made for the other newly formed classes. Even though, the activity type alignment in Table 7.2 is a heuristic procedure, the cluster determination process was made automatically using HRSOM and K-means. This is a very good result which will allow us to engage transfer learning on these two datasets. Note that this alignment of PAs is entirely data driven. This is unlike the approach taken to create the original pattern classes which were defined manually. The new classes aligns well with the original classes while concurrently aligning the pattern classes of two populations of data.

7.4 Experimental Results on the Application of Transfer Learning to PA Prediction

The following experiments will compare the recognition accuracy of the baseline models with the results of transfer learning model by using the PA2014 dataset and the two subsets of the PA2016 dataset. The target domain is defined by the subsets of the PA2016 dataset whereas the PA2014 data are to provide background knowledge to the transfer learning model. The focus of the analysis of results will be on the target domain. The results shown in this section will thus be limited to the PA2016 domain. Both, the ETL and STL modelling options are explored.

The experiments are prepared as follows: Having aligned the pattern classes as described in the previous section we then split each of the PA2016 subsets into three sets, 60% for training set, 20% for validation, and 20% for the test set. A baseline model is trained on each training set, training parameters are optimized by using the corresponding validation data, then the final result on the corresponding test set will be reported. For building the transfer learning model we first train a baseline model on the full PA2014 dataset, freeze the

weights, then build the ETL architecture as well as the STL architecture the newly added parameters of which will then be trained on the afore mentioned training sets.

7.4.1 Baseline Results

The baseline model is an MLP with either one or two hidden layers. The number of hidden neurons in each layer is chosen to be within $\{23, 35, 57, 76, 100\}$. We also trained MLPs with more than two hidden layers but found, as will be seen in Table 7.3, effects of overfitting are observed if large MLPs or MLPs with more than one hidden layer is used.We thus omit the results from MLPs with more than two hidden layers. The reason why we chose the MLP as a baseline architecture is that (a) MLPs are often used in transfer learning since the number of hidden layers can be easily expanded, (b) the properties and capabilities of MLPs are well understood, and (c) MLPs have been deployed to PA classification of young children from accelerometer data [5, 8, 25].

The training of the MLPs is performed by selecting the learning rate from within $\{0.1, 0.01, 0.001, 0.0001\}$. The final choice if the learning rate is made on the basis of the validation results. All experiments are repeated three times using different initial (random) conditions. The average test accuracy and the average training accuracy is calculated. The results are summarized in Table 7.3.

Table 7.3 reveals that, for both subsets, the best results are obtained when using just a single layer of hidden neurons and when the number of hidden neurons remains small. This implies that the model develops a tendency to overfit the training data when the number of parameters is large. To verify whether overfitting is indeed leading to these results we then train a set of MLPs where we evaluate the generalization performance at various stages during the training procedure. More specifically, the generalization performance is evaluated at the 500-th training iterations, then again at the 1000-th iterations, at the 1500-th iterations, and so on, and up to 10,000 iterations. The results are shown in Table 7.4. It can be observed

Table 7.3: Experimental results when using different number of hidden neurons and number of hidden layers.

Model architecture			**GeneActiv**		**ActivGraph**	
#hidden1	#hidden2	#params	TrainACC	TestACC	TrainACC	TestACC
23		1937	0.8645	0.6960	0.8225	0.7110
35		2945	0.8935	0.7076	0.8378	**0.7314**
57		4793	0.9082	**0.7168**	0.8555	0.7129
76		6389	0.9277	0.6865	0.8644	0.7211
100		8405	0.9360	0.6794	0.8634	0.7280
23	23	2489	0.9022	0.6738	0.8603	0.6958
57	23	5957	0.9432	0.6595	0.8635	0.6749
57	57	8099	0.9746	0.6960	0.9097	0.6711
75	57	10683	0.9845	0.6833	0.9202	0.6970
100	35	11615	0.9782	0.6762	0.9303	0.6875

Table 7.4: Experimental results when using different number of training iterations.

#epochs	**GeneActiv**		**ActivGraph**	
	TrainACC	TestACC	TrainACC	TestACC
500	0.8653	0.7286	0.8032	0.6983
1000	0.9082	0.7168	0.8378	0.7314
1500	0.9249	0.7286	0.8536	0.7362
2000	0.9201	0.7167	0.8479	0.7249
5000	0.9464	0.6929	0.9221	0.7040
10000	0.9833	0.7048	0.9506	0.7021

that the accuracy for the training data increases with the number of training iterations and that the generalization accuracy peaks at about 1500 iterations. This confirms our hypothesis that overfitting has indeed occurred.

Overfitting can be controlled by choosing smaller models or by increasing the number of training samples. Furnished with the insight that it is best to train small MLPs for a limited number of iterations we can thus eliminate the validation set as it is no longer needed for optimizing the training parameters. Furthermore, we can change the test set such that it contains only the samples of one person. This allows us to create a training set which contains all samples except the ones that are in the test set. The mechanisms thus maximizes the size of the training set. To obtain a complete overview of the generalization performance

we rotate the person in the test set so that each person appears in the test set exactly once. Since the PA2016 is a collection of samples from 16 participants and hence we train 16 MLPs where each MLP is evaluated on the data of one of the participants. This creates 16 results. Averaging the results thus produces a generalization result over all samples. The approach is called leave-one-person out (LOPO). This is similar to cross-validation that is commonly applied when working with small datasets. The LOPO approach is not new. It has been deployed previously to PA classification of young children [5, 8, 25]. By using the LOPO approach the generalization results increase to 77.34% (GeneActive) and 75.60% (ActiGraph) respectively. We will use this result as a baseline for subsequent comparisons.

7.4.1.1 Comparisons with SVM and FRPN

This section will compare the results of the MLP with results from the SVM and the FRPN algorithms in order to obtain a more complete picture on how the proposed transfer learning approach compares. The SVMs and FRPNs were trained by (a) identify the optimal training parameters by using the training and validation sets then (b) training the models using the LOPO approach.

The averaged generalization results are shown in Table 7.5. Presented are the evaluation metrics micro-recall and macro-recall [1]. For the GeneActiv dataset the generalization accuracy of the MLP is slightly better when compared with the FRPN whereas for the ActiGraph data is worse than the FRPN. This comes as a surprise because the FRPN had more parameters then the MLP and the algorithm simulates a deep learning architecture. But, as found in the previous section, more parameters and layers were observed to lead to overfitting. It is not clear why the generalization performance of the FRPN is better for the ActiGraph data. But the results are interesting because the STL method will engage the FRPN. From this result we can establish the hypothesis that the STM method will perform better for the ActiGraph data whereas the ETL method could be better for the GeneActiv data. whether

[1] Note that macro-recall can be called as "accuracy" for multi-label class classification problems

Table 7.5: Comparison of baseline results.

Model	GeneActiv		ActivGraph	
	MicroRecall	MacroRecall	MicroRecall	MacroRecall
MLP	0.7416	0.7734	0.7180	0.7560
SVM	0.7667	0.7857	0.7856	0.7895
FRPN	0.7480	0.7635	0.7467	0.7784

this hypothesis holds will be investigated in the next section.

For completeness, it can be seen in Table 7.5 that the SVM produced the best generalization performance for both datasets. This is an expected result given that the SVM establishes a decision boundary that maximises the distance between pattern classes whereas for the MLP and FRPN the decision boundary could be anywhere between the pattern classes. Another reason would be that SVM still performs well even when the number of training samples is relatively small as in the cases of our PA datasets while MLPs and FRPN might be overfitting quickly with the small data sample space. It is to be seen whether transfer learning can enhance the results and how the results of transfer learning compare with the baseline methods.

7.4.2 Results from using Transfer Learning via Model Expansion

This section investigates the effectiveness of the Expanded Transfer Learning (ETL) approach. Two avenues are explored to train the base model:

Approach A: We define the source domain as being the one which combines the PA2014 dataset with the ActiGraph data of the PA2016 dataset. The target domain is the GeneActive data. The ActiGraph and GeneActive data can be considered 2 different datasets since they were collected using different types of accelerometers and from different laboratory settings and time. The two datasets are called the as a portion of PA2016 data cohort because they contain the same types of physical activities.

Approach B: We define the source domain as being the one which combines the PA2014 dataset with the GeneActive subset of the PA2016 dataset. The target domain is the ActiGraph subset.

Both approaches maximise the amount of the source information. The source models are created and trained as before by first splitting the dataset, using training and validation data to identify optimal network size and training parameters, then use LOPO to train the final model. The source model obtained by using approach A will be referred to as model-A. Conversely, model-B is the one obtained by using approach B.

Transfer learning is then engaged as follows. The weights in the source model are frozen. The model is extended by adding one new hidden layer. The model is then trained on the (training) data in the target domain and tested on the test data in the target domain. The training procedure only updates the weights of the newly added hidden layer neurons thus retaining the information that was encoded by the source model.

Two transfer learning models are obtained: One which uses model-A as the source model and one which uses model-B as the source model. The corresponding transfer learning models will be referred to as ETL-model-A and ETL-model-B respectively.

The background model ETL-model-A and ETL-model-B here are multiple hidden layer MLPs, which were found best to contain two hidden layers. We tuned the number of hidden neurons for each case and found that for ETL-model-B, the network architecture would look like input-50-40-output where two hidden layers contain 50 and 40 hidden neurons respectively. For ETL-model-A, the network architecture has 170 and 90 hidden neurons for two hidden layers respectively. The validation set is left-aside 33% of the training set. For the case of expanding the network at the last layer, say removing the output layer and then topping up the network architecture with one hidden layer or two. We experimentally found that adding one hidden layer to the end gave better generalization accuracies. In this case, the whole background model's parameters are frozen, when training on the target

Table 7.6: Experimental results when using different number of models.

Model	GeneActiv		ActivGraph	
	MicroRecall	MacroRecall	MicroRecall	MacroRecall
MLP	0.7416	0.7734	0.7180	0.7560
model-A	0.8008	0.8012	-	-
ETL-model-A	**0.8246**	**0.8258**	-	-
model-B	-	-	0.7896	0.7901
ETL-Model-B	-	-	**0.7941**	**0.7954**

domain training set, only expanding layer's weights were updated.

The results are presented in Table 7.6. It can be observed that ETL-model-A enhances the result by about 2% over the source model and by over 5% when compared with the baseline model. This is a good improvement and is also better than the baseline SVM results.

It can also be observed in Table 7.6 that the transfer learning model significantly improves results over the baseline results. Though the difference in results between the source model and the transfer model is relatively minor.

7.4.3 Results from using Transfer Learning via Model Stacking

This section uses the Stacked Transfer Learning (STL) approach in which the output layer of the source model is replaced with an FRPN. Source model-A and ackground source model-B from the previous section are used. The corresponding transfer learning models are then referred to as STL-model-A and STL-model-B respectively. The same procedure as before is used for model optimization and model training.

Results are shown in Table 7.6. By comparing the results in Table 7.7 with those shown in Table 7.6 we can make the following observations: The STL method works much better than the ETL method. The STL method significantly enhances the generalization results over both, the source models and the baseline models for both datasets. The improvements are about 8% when compared to the baseline for both datasets, and about 2% better than the

Table 7.7: Experimental results when using the STL approach.

Model	**GeneActiv**		**ActivGraph**	
	MicroRecall	MacroRecall	MicroRecall	MacroRecall
STL-model-A	0.8404	0.8516	-	-
STL-model-B	-	-	0.8199	0.8316

Table 7.8: Confusion matrix for GeneActiv data

class	**1**	**2**	**3**	**4**	**5**	
1	**0.9929**	0.0071	0.0000	0.0000	0.0000	
2	0.0071	**0.9262**	0.0524	0.0071	0.0071	
3	0.0000	0.0964	**0.7286**	0.1071	0.0679	
4	0.0000	0.0429	0.2429	**0.6714**	0.0429	
5	0.0000	0.0250	0.0571	0.0357	**0.8821**	**0.8516**

Table 7.9: Confusion matrix for ActivGraph data

class	**1**	**2**	**3**	**4**	**5**	
1	**0.9857**	0.0071	0.0000	0.0071	0.0000	
2	0.0019	**0.9111**	0.0611	0.0222	0.0037	
3	0.0000	0.0750	**0.7444**	0.1250	0.0556	
4	0.0000	0.0500	0.2778	**0.6056**	0.0667	
5	0.0000	0.0417	0.0944	0.0111	**0.8528**	**0.8316**

ETL method for both datasets. It is a surprise finding that the STL method is better than the ETL method for the GeneActive data since the hypothesis was that the ETL method would work better on the GeneActive data. The result implies that transfer learning is effective in enhancing the generalization performance in PA classification in young children using accelerometers, and that the FRPN is much better as exploiting background information in transfer learning.

We present confusion matrices to further investigate these results. Table 7.9 presents the confusion matrix of results produced by STL-model-A and Table 7.8the confusion matrix of results produced by STL-model-B. It can be seen that the accuracy performance can vary significantly from class to class. For example, for class 4 (Walking and Pack Away) the accuracy is almost 40% lower when compared to class 1. This coincides with the fact that class 4 is the smallest class and class 1 is one of the largest (only class 2 is larger, see

Table 7.2). This indicates that the unbalanced nature of the dataset affects the overall result. The observation is true for both datasets. The way we divide the PAs into 5 activity classes may thus affect the overall generalization performance. This in turn implies that the result may further improve if a balancing technique such as SSEN is engaged. This would result in a SSEN transfer learning model. However, despite holding great promise to improve results further, this idea shall remain a subject of future research [2].

7.5 Summary

This chapter investigated the possibility of engaging transfer learning for enhancing PA classification. A background model was trained on different but related data. The background data differed in terms of protocol and in terms of the type of sensors used. Due to differences with the target domain this chapter proposed a workable approach to align the source domain with the target domain. The background model was then expanded and the newly added parameters were trained on the target domain. This resulted in significant improvements in results particularly when the FRPN was engaged to expand the background model.

While the general concept of transfer learning is not new, this chapter introduced several novel concepts to render transfer learning suitable for classifying PA of young children on available data. The work presented in this chapter confirmed that it is possible to effectively transfer prior knowledge on PAs to a model in the target domain. Improvements in results by as much as 8% were observed. This is a significant improvement given that the data is unbalanced and affected by noise. Further improvements should be possible by incorporating a data balancing technique into the adopted transfer learning approach. This, however, will be left as an objective of future research.

[2] The work could not be carried out due to time limitations and personal circumstances.

Chapter 8

Video-based children activity recognition

8.1 Preamble

Children physical action recognition problems may have two modalities of measurements: be available with two different types of collective data. The use of accelerometer sens accelerometer recordings as the primary data, and video recordings as a secondary record, for validation of the accelerometer measurements. This is because the accelerometer measurements cannot distiguish various physical activities by the young pre-school children who might not be performing the labelled activity throughout an activity episode, while such deviations from the laebled activity would be easily detected from video recordings, and manually removed from the accelerometer in a postprocessing stage, so as to preserve the consistency of the label. Therefore, the video used for validation for disposal. In this chapter, we investigate for the sake of curiosity: would the video be pliable to video processing techniques and might be used for video classifications by themselves. Obviously we do not expect the video classification task to have superior performance when compared with those obtained using the accelerometer recordings, but the energy expenditure which would be estimated from the accelerometer recordings are sufficiently coarse, which might imply that video processing of the associated video recordings might surprise us of its accuracy. In the

event if the accuracy of the video classification task is high, this could serve as a confirmation of the accuracy of the classification task using the accelerometer recordings. It is in this spirit that we investigate the video classification problem in this chapter.

8.2 Introduction

Recently, human physical activity recognition has drawn attention in the field of video understanding and knowledge acquisition from video sequence because of the growing demand from a number of applications, such as surveillance environments, entertainment environments and health-care systems [16].A good review paper about related work can be found in [143, 16, 24]

Fundamentally, steps in video-based action recognition include (referred to Figure 8.1):

- Video preparation which includes the annotation steps to assign each video segment with a class label, video image noise and blurriness reduction.
- Object detection to identify the object's boundary which is applied on the image by identifying the bounding box of an object in each image in the video sequence, ultimately resulting in a series of bounding boxes corresponding to the sequence of images in each video.
- It is necessary to track/following the object of interest in the whole video segment so that the bounding boxes are placed on the right object over time.
- Feature extraction from sequence of object's bounding boxes.
- From the sequence of feature points, we train a model to classify the human actiivity type.

We acknowledge the challenges that arises when solving small children's action recognition problem.

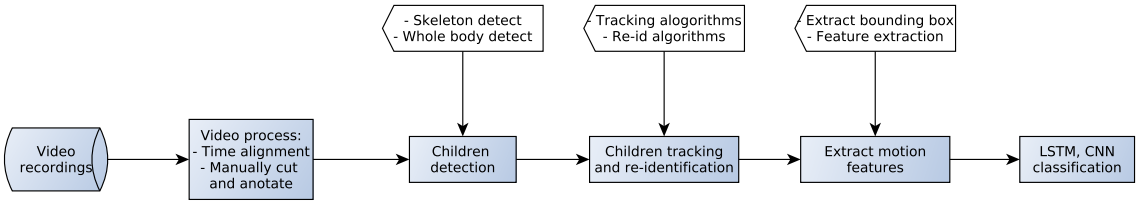


Figure 8.1: Stages in recognizing children physical activities

1. Intra- and inter-class variation challenge: For a single action/class for example, walking and running movements can differ in speed and stride length for different individuals. For several actions/classes, one action can accommodate the other actions such as the case that the clean-up contains actions that come from the walking and sitting down action/class. For increasing numbers of action classes, this will be more challenging as the overlap between classes will be higher.

2. Environment and recording settings: The environment (such as in laboratory, outdoor or free style) in which the action performance takes place is an important source of variation in the recording. The actions performed under a strict pre-setting protocol would be much easier to recognize than the actions made under free-living conditions.

3. Less disciplined children: while actions acted by a teenager or of older ages are exactly followed the setting procedures, children are more active and are not able to follow the instructions all the times. For example, while doing the walking action, children sometimes can stop walking and picking up a little toy that appears on the ground. These noisy actions can be detrimental to the classification task. In practice, instructors might request to have another replacing trial, however it is relatively hard to achieve a clean performing action by a little child.

4. There was a problem with camera position and children poses: since children are more active and hence their body parts' poses are diverse and changing more frequently. The camera position is normally fixed so that the children's body parts can hardly be

visible all the time. The detection model may find it hard to identify humans and body parts in this case.

5. Another problem is when there are more than one persons being visible from the camera. More importantly, they are moving and occluded by one another. In this case, the track lost is more likely to occur.

6. Other problems: first , the lighting conditions can further influence the appearance of the person; secondly, the same action, observed from different viewpoints, can lead to very different image observations.

8.3 Methodology

Among many applications of video recording system, human action recognition especially with high-level behavior recognition comes out to be one of the most interesting one. An physical activity is a sequence of human body movements, and may simultaneously involves a number of body parts' co-interaction. Basically, the recognition of human action on video sequences need to go through several steps. Major components of such systems include human body and body parts detection, tracking the subject of interest possibly among many other non-interest objects, feature extraction from the detected bounding boxes, action learning, and classification [20]. The methodology followed in this chapter is the "best of breed" object detection, and object tracking" algorithms at the time such experiments were conducted. More details on each step are as follows:

8.3.1 Object detection

Children body detection can be divided into whole body detection, body part detection, and corresponding skeleton detection. OpenPose is a well-known library for real-time multi-person key-point detection. OpenPose is computationally efficient by using multi-threading

GPU model. We follow closely the development of automatic human skeleton detection in [30]. Another approach includes the whole body detection model. The current state-of-the-art model for this purpose is Yolo (You only look once) object detection [33]. Yolo is a real-time object detection system. In this Chapter, we will follow the implementation of two mentioned algorithms.

8.3.2 Object tracking

This step commonly follows the object detection step or functions that give an object's bounding box on a sequence of image frames. The fundamental idea behind tracking algorithms is to consider the past movement patterns and changes around the object to predict future movement direction. There are several most accurate tracking algorithms that we will follow the development closely in this Chapter as follows:

- Tracking by detection using Kalman filter [37]
- Correlation filter tracking [39].
- Tracking by using template matching [144]
- Tracking by using human re-identification model based on deep convolutional neural network [44].

Detailed descriptions of these algorithms are shown in Chapter 2.

8.3.3 Feature extraction

The outcome of object detection and tracking is a series of subject's image with as less background noise as possible since we found experimentally that images with more noisy background would result in poorer descriptive features and hence result in poorer classification accuracy. In this Chapter we will follow the development of the state-of-the-art feature extraction approach which was based on the dense trajectory [53].

8.3.4 Classification

Once the sequence of feature vectors from the feature extraction step are available, we use them as direct inputs to a classifier model. The output of which would be the class label of physical activity class. This problem is one type of many-to-one classification problem [145]. The model would map sequence/time series data to a class label, which in other word is called supervised sequence labelling/classification [145]. Since the tradition models such MLP or SVM might not be suitable or be capable of solving this type of problem, we follow the development of the current state-of-the-art recurrent neural network model which is denoted as deep temporal LSTM [78] which is the latest generation of LSTM model [56]. For simplicity, in the following we will call the classifier LSTM.

It should be noted that at the time of this experiment, all the models selected for development in this Chapter are considered being state-of-the-art models.

8.4 Experimental settings

There will be several steps before the numerical data are extracted. One can have optional processing steps along with the required steps. In particular, the features extracted from image frames are unavoidable for every approach while the object detection, object tracking and object re-identification are kinds of optional techniques. The separated approach being used as the base line is using the human skeleton location extraction.

In the following, the required steps will be presented first, optional processes will be shown later. We will compare the experimental results when using the LSTM (long short term memory) given different combinations of pre-processing steps, namely:

- The skeleton location extraction, this would form the base line for experimental comparison
- The human detection + feature extraction (Setting 1)

- The human detection + human tracking + feature extraction (Setting 2)
- The human detection + human tracking + human re-identification + feature extraction (Setting 3)

As being stated in Methodology section, the LSTM is selected for classification task of labelling the long time series/sequence or very high dimensional input. We expect that the feature extracted would be of very high dimensionality, since each 15s contains 15*25 frames (=375), then the dense-trajectory feature extraction given those frames would be much larger in size.

The classification experiment procedure is as follows: (1) given the feature vectors extracted from the sequence of image frames using the feature extraction method stated in Methodology section (i.e each sequence input to LSTM is a series of feature vectors). The dataset we will obtain, includes these input sequences. We randomize on the dataset and split the data into training set and test set,the training set contains 60%, the validation set contains 10%, and the testing dataset contains 30% of the original dataset. (2) The LSTM network's parameters are chosen as follows. The LSTM Network architecture contains from 1 to 3 recurrent layers.The number of hidden neurons on each layer was selected in the range [30, 70], the learning rate was tuned within [0.01, 0.00025]. The number of training iterations was selected from 50,000 to 300,000. The batch size was chosen in the range [50, 200]. The input sequence length is set based on the number of image frames collected for each segment of video, since we will consider each segment of video is corresponded with a class label/a type of physical activity. The length of each video segment is selected from 5s to 15s.

The data description has been presented in Chapter 4. The video sequences are available for all participants performing their physical activity trials in the PA2014 dataset only. In total, there is approximately 5 (minutes) * 12 (activity trials) * 11 (participants) = 660 minutes (or 11 hours) of video recorded in the PA2014 dataset. The video sequences from time

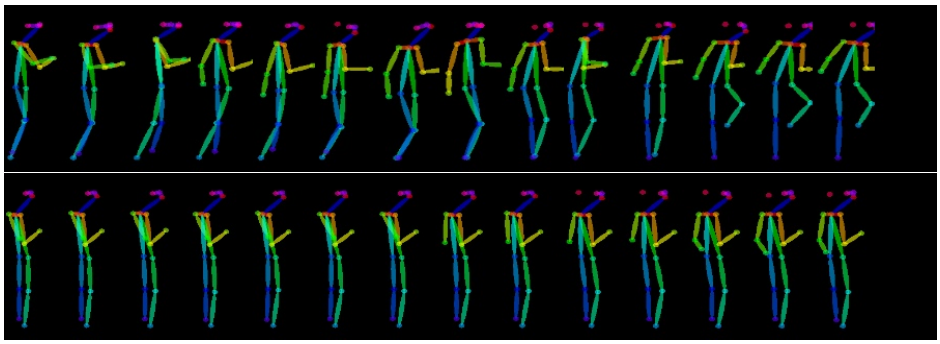


Figure 8.2: Skeletons detected given a treasure hunt activity (top) and a collage activity (bottom). The example contains skeletons from 14 consecutive frames.

to time contain segments in which there is no presence of the performing subject. The image frame in some cases contains vertical strips so the image noise removal and de-interlace approaches [146] are used to improve the frame quality.

8.5 Skeleton feature-based recognition: the base line

For human skeleton detection, we use convolution pose machine (CPM) model [30]. This method works as follows: It learns the localization context (the location of the subject and the surrounding related objects) of a body part in the image based on the model's belief map and receptive field. For the case of multiple people in the image, once body parts are found, the minimum spanning tree is used to separate a set of parts that belong to individual persons [30].

The model was trained with publicly available data, say COCO cohort [147].

There are 18 detection key points on the body (nose,left eye,right eye,left ear,right ear, neck, left shoulder,right shoulder,left elbow,right elbow, left wrist,right wrist,left hip,right hip,left knee,right knee,left ankle and right ankle), in which each was represented with a coordinate which includes 2 numbers.The window size for feature extraction is the same as the length of video segment.For example, if we take out 24 frames for each second, a feature vector is formed by concatenating all the key points' coordinates of all frames within each

5s. The final dimension for the concatenated feature vector would be 24 * 5 * 18 * 2 = 4320, which is relatively large for a classification model such as the traditional MLP model. However, we will create LSTM input in the form of sequences (series of feature vectors).

Figure 8.2 presents two examples of skeleton detection for two respective activities, namely treasure hunt (for the top row) and collage activity (for the bottom row). When there are more than one persons appearing in the video, the smaller skeleton is assumed to be of the child.This is not right in the case that the location of the child is closer to the camera than that of other people. More importantly, at some situations the model is not able to detect the child due to occlusion by others, .i.e some important body parts such as face and hands are hidden, or the child facing away from the camera. In that situation, the coordinates might become zero for that particular frame. For this problem, we interpolated the detected key points of missing values in the frames by using the frames' information before and after the missing ones. This is sensible, because if we skip the frame then the time sequence of feature information will become meaningless to some extent.

8.6 Processing 1: Detecting children in videos

In order to detect children in videos, Yolo v2.0 [33] was selected since it was at the time of processing, a state-of-the-art object detection algorithm. The model was trained with two challenging datasets, namely, PASCAL VOC [148] and Microsoft COCO dataset [147].Yolo was shown to outperform other state-of-the-art models then like Faster R-CNN [34],ResNet [35] and SSD [36] while running significantly faster [33].

The algorithm contains three components: (1) the regional selective component is learned to quickly select a region of interest where the object is more likely to be located; (2) the regression component is learned the annotated bounding box in the training set, the output of this component are the set of coordinates, the lower left hand corner coordinates, and the upper right hand corner coordinates, representing the bounding box;(3) the classification

componentis trained to classify the class label of detected objects. These three components contribute their certain role in the model accumulative loss function.

The input image is resized to fixed size, say 416×416. In some cases, however the child looks too small in the video, there would be two approaches: (1) we consider only the region in the video where the child appears which basically excludes the outer region of each image; (2) we use the network with the larger size of input layer (or image size) being 608×608.

There are situations that the people are well detected in video such as the one on the right of Figure 8.3 in which the instructor and the child are clearly in the good view from the camera. On the other hand, when the kid was facing away and crawling through the tube like the one on the left of Figure 8.3, the detection model would miss anyway. Increasing the size of input images, or excluding portion of the outer most region on images would help to boost the detection accuracy to some extent. Because of some undetectable situations, the only detection model seems not be efficient for the human action recognition problem. We need to track all kid's body movements so that the recognition model would be more beneficial using the detected information. In the following, we will investigate the tracking algorithm and how to make the tracking consistent and reliable.

8.7 Processing 2: Tracking algorithm

There are several tracking algorithms which can be divided into two main approaches, including tracking by detection and automated tracking given the initial object's bounding box.

1. Tracking by detection: it builds up the track based on the object detection model. The detection quality is identified as a key factor influencing tracking performance. A representative of this type is the Kalman filter tracking algorithm [149]. Kalman

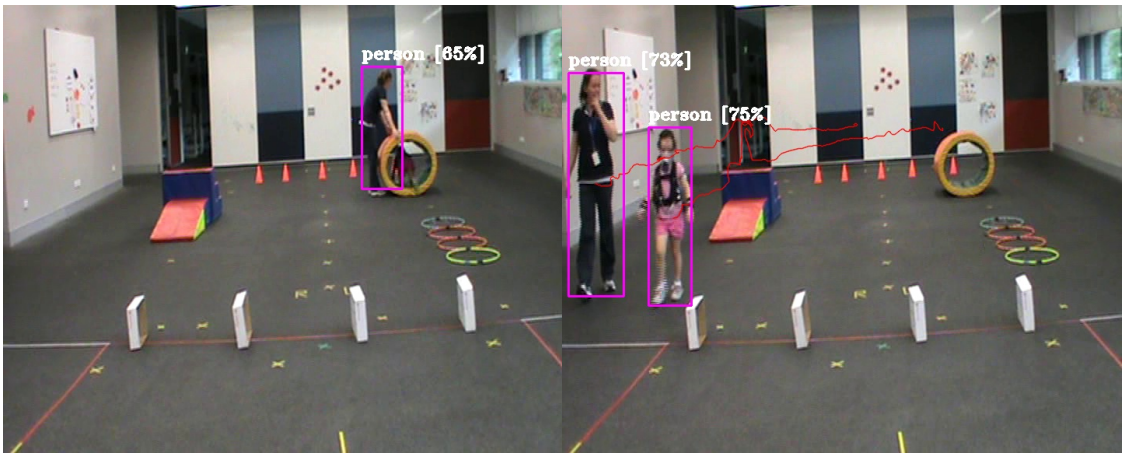


Figure 8.3: Subject detection and tracking result. On the left shows the undetectable example or a track lost. On the right shows the good detection and tracking result.

filter estimates the state and the variance or uncertainty of the tracked objects. The estimation is updated over time using a state transition model and using the actual measurements given by the detection model. If $\hat{x}_{k|k-1}$ denotes the system's state at time step k before the k-th measurement y_k has been taken into account, $P_{k|k-1}$ is the corresponding uncertainty.

The Kalman filter model is able to "filter" the state information among the noise and then is able to estomate the object's movements in the next time step, in particular the moving direction of a child in video the sequence. In some cases, there is no detection bounding boxes being given since the child's possibly is hidden behind another object, Kalman filter can estimate what is the likely position of the child. The prediction of the future direction of movement of the child is based on the historical direction of movements and positions.

Another example is simple online and realtime object tracking (SORT) algorithm [150].

2. Automated tracking: initialized by an object's bounding box, the tracking algorithm helps to track the object during its trajectory in the video. An example is a tracking

algorithm by correlation filter and online learning [151]. This type of tracking algorithm is able to learn the appearance of object and then keep searching around the current position in the next image frame to see where is the object's position might be.

Figure 8.3 shows the Kalman filter's trace results of a child and an instructor when doing their activity trials in the laboratory setting. The detection model is set to skip one image frame after processing one image frame. This means the Kalman filter tracking is required to provide one step ahead prediction. The trace shown in the figure indicates that Kalman filter does track quite well given the accurate detection results. In practice, however there are situations that the camera is rotated on tripod to follow a subject of interest. This results in a wrongly stored tracking history in the Kalman filter algorithm. The rotating camera will add some uncontrolled movements such that the tracking algorithm is not able to predict the subject's position in the next image frame. Another case is when the child is occluded sufficiently long such that the tracking algorithm, which is based very much on detection results, cannot provide accurate prediction any more. This is also challenging for any tracking algorithm if the child re-appears at different locations. The following will presents a workable solution for this problem. It should be noted that the better tracking solutions would help us extract more accurate image frames of the subject of interest since we found experimentally that the more accurate and the less background noise is in resulting image frames, the better prediction results will be.

8.8 Processing 3: human re-identification

Before going through the human re-identification model, we have tried another two popular methods for supporting the tracking algorithm in cases that the track is lost due to occlusion or no detected bounding boxes are available in several consecutive image frames.

8.8.1 Traditional methods to support a tracking algorithm

The traditional methods used to re-identify the object since its lost from the tracking algorithm include template [144] and feature matching [152]:

- Template matching: The basic idea of template matching is that we have a template image patch and a test image to search on where there are any template image parts appearing on the test image [144]. The template image is super-imposed on the test image and the correlation matrix is calculated accordingly to see the best match location on the test image. For this case we store several images of the child as templates as shown in Figure 8.4, and when the tracking algorithm is not able to maintain the track, we start matching the image region around the missing points and the stored template images. A threshold value is required to compare with the output of the template matching algorithm, and this value is tuned based on the trial and errors method so that the best matching results are obtained. Several outputs of this algorithm are shown in Table 8.1.

  For this method to work well, the subject would have been non deformable. For a human subject especially a young child this method is unlikely to work well.

- Feature matching: This method calculates ”feature description” of the object. This description, extracted from an image (template image), can then be used to identify the object when attempting to locate the object in another image (testing image) containing many other objects [152]. To perform reliable recognition, it is important that the features extracted from the training image be detectable even under deformation, noise contamination and illumination alterations. The feature matching algorithm calculates the Harris corner condition for edge detections or the Haar-like features of the object in images [152]. This approach would work well if the object consists of sharp and well defined edges. For human subjects, especially for a young child, this

Figure 8.4: Example of some images take for image gallery

approach is unlikely to work satisfactorily.

Another important characteristic of the feature matching algorithm is that the relative positions between objects in the original scene should not change from one image to another. For example, if only the corners of a door were used as features, they would work regardless of the door's position. However, if features located on a deformable object, this would typically not work. Practically, we found that the method does not work effectively with the child's body's features since the object image does not satisfy the aforementioned constraint.

8.8.2 Human re-identification

Both matching algorithms were implemented to support the tracking performance, however the tracking algorithm still cannot handle well in the case that the child changes his/her pose and direction, since in those cases the image looks quite different from the original one (i.e. the stored template images). It is understandable since the template matching method is a pixel-by-pixel approach while the feature-based matching is not able to differentiate a person and another person, or a face with another face since two images have very similar edge/feature points. Thus, both aforementioned approaches are observed to be not really helpful to improve the tracking performance.

Recent interesting research in human re-identification studies [42, 43, 44, 45] inspired us to investigate the use of some of the proposed algorithms to support the tracking per-

formance. There are several approaches to human re-identification as listed below. The traditional method still used the feature marching[42], another one was based on part-based mixture of model [43]. Recent state-of-the-art methods include deep convolutional neural network such as CNN models for person re-identification [44, 45] at the time when we studied this problem. These methods train a deep convolutional neural network from scratch and use publicly available human re-identification datasets such as the CUHK01, 02, 03 datasets [153],Market-1501 datasets [154].The deep CNN models work on deformable objects, such as humans, because they use the advantages of deep CNN models to form a good set of features, which summarizes the image based on the image rather than objects in the image, rather than on pre-assumed characteristics of the object, like it must be well defined sharp edges, or that it is non deformable.

The learning metric of this type of CNN models is distance metric function (i.e triplet loss function) [155].Specifically, the model learns to minimize the distance metric between images of the same person taken at different time, or circumstances, and maximize that of different subjects.Recently, this strategy has been applied to human re-identification and results are better than the hand-crafted feature based methods such as using The scale-invariant feature transform (SIFT) or histogram of oriented gradient (HOG) features [44, 45].

In this section, we made use of the model presented in [46], which is a Resnet [35] model with an extended layer before the output layer for learning the distance metric function [46].The Resnet model was pre-trained with Imagenet data and then re-trained the whole extended model with the combinations of human re-identification datasets as listed above. It should be noted that the output of this human re-identification model is a descriptive feature vector. In other words, each input image to the human re-identification model will result in a descriptive vector. In order to identify who is in the input image, we need an image gallery of each person. The descriptive vectors were produced before hand for each image in gallery. Then when considering a new image, the descriptive vector of this new

image will be compared with all descriptive vectors of the gallery. The person appeared in the nearest image (in terms of Eucledian distance) in the gallery will be the person appeared in the new image.

Experimental comparison: The human re-identification CNN model was used to compare with the template matching method using images shown in Figure 8.4 as template images and shown in Table 8.1 as the testing ones. The matching results presented are the average of one test image with all template images. For template matching, the greater is the matching result, the better is the testing image matched with the image gallery (or template images). For human re-identification, on the other hand, the smaller is the matching result, the closer is the testing image compared with the image gallery.

It can be observed that while the template matching algorithm is not able to identify the same subject with lowest scores (lower score signifies a better match), the human re-identification algorithm can. For template matching, the image of the lady instructor is shown best matched with the template images of the child, and the image on the first row of the child is associated with quite a low matching result, which is unexpected. As can be expected, the human re-identification method produces correct identification results for both of those two images of the child on the first two rows. The first row is associated with the lowest matching result which is anticipated since visually we can see that the second image of the child is little interfered with the image of a male instructor. The results shown in Table 8.1 are a good indication that the human re-identification approach can significantly improve the tracking performance.

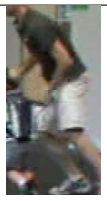
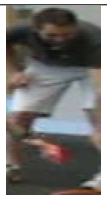
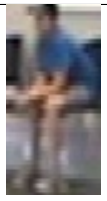
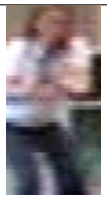
Image to test	Template matching	Human re-identification
	**30**	**0.60**
	45	**0.67**
	32	0.72
	35	0.73
	36	0.83
	44	0.75
	50	0.78

Table 8.1: Comparing matching results

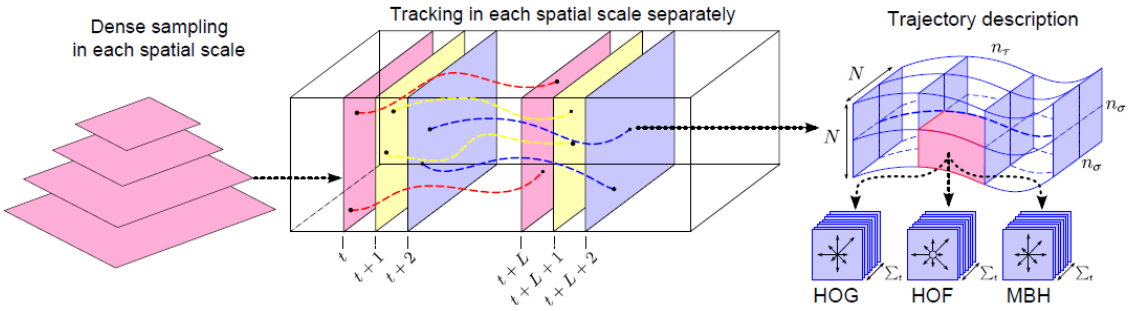


Figure 8.5: Illustration of the dense trajectories

8.9 Dense-trajectory feature extraction based on bounding box series

The Dense trajectory method for human action recognition has been used extensively in the literature [53, 52] before the advent of the deep CNN methods of feature extraction and classification methods were introduced in 2013 [156] which includes spacial-temporal based features such as HOG, histogram of oriented optical flow (HOF) and motion boundary histograms (MBH). The features computation is quite fast, as they are considered as handcrafted ones, as contrary to those obtained by the deep CNN ones [156, 157] so that the method has been applied in real-time human action recognition [54, 55]. The calculation of these features are based on a number of consecutive frames. In addition, a dense trajectories feature for long term evolution of the human action is also found [53]. This is called dense because the trajectories are evaluated at dense interest points in the image, and track across the sequence of images in the video. The dense trajectory feature is essentially the tube formed by these dense trajectories across the video sequences, and thus this gives an idea of the long term evolution of the those trajectories [52]. An example of this method is shown in Figure 8.6.

All these features are computed in local cuboids obtained by spatial-temporal interesting points (STIP) detectors [158] or dense sampling schemes [52]. There are some options to extract dense trajectories such as trajectory length based on the start and end frames or the

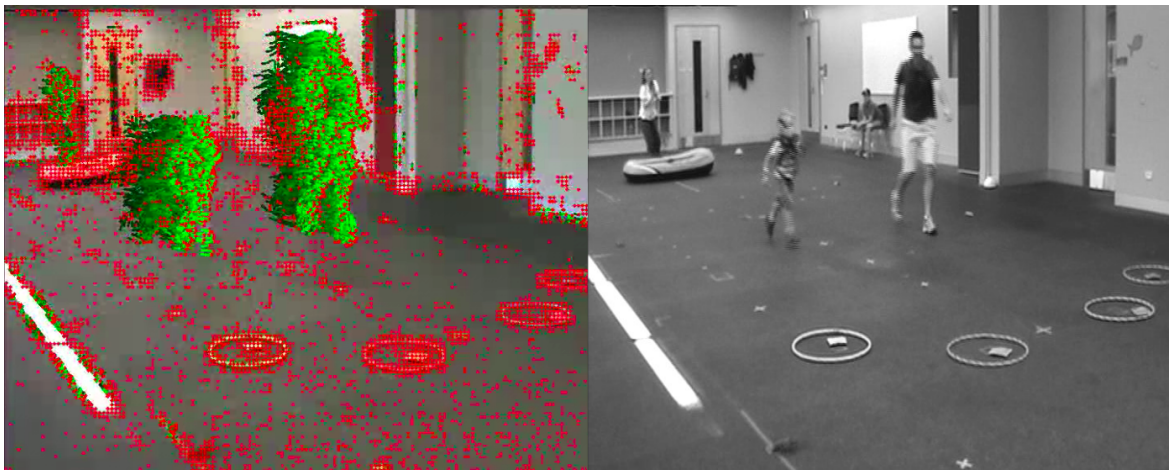


Figure 8.6: Example of the dense trajectories

size of spatial and temporal cells in the STIP scheme [158] which is used to extract the feature. The detailed work flow is illustrated in Figure 8.5 which was shown in [53].

Given a video sequence, one can extract features from the dense trajectories method [52]. In our experiments, we used suggested best parameters of choice for the feature extraction as in [52]. Several separated features include:

1. Mean-x: the mean value of the x coordinates of the trajectory
2. Mean-y: the mean value of the y coordinates of the trajectory
3. Var-x: the variance of the x coordinates of the trajectory
4. Var-y: the variance of the y coordinates of the trajectory
5. Length: the length of the trajectory
6. Scale: the trajectory is computed on which scale
7. X-pos: the normalized x position w.r.t. the video, for spatio-temporal pyramid
8. Y-pos: the normalized y position w.r.t. the video, for spatio-temporal pyramid
9. T-pos: the normalized t position w.r.t. the video, for spatio-temporal pyramid

The followings are five sets of description feature:

1. Trajectory: 2x[trajectory length]
2. HOG: 8x[spatial cells]x[spatial cells]x[temporal cells]
3. HOF: 9x[spatial cells]x[spatial cells]x[temporal cells]
4. MBHx: 8x[spatial cells]x[spatial cells]x[temporal cells]
5. MBHy: 8x[spatial cells]x[spatial cells]x[temporal cells]

Since in the video sequence, there are other people such as instructors present rather than the child, different from the accelerometer recordings, which only have the ones related to the child, and not those who were in the room at the time.. Dense trajectory technique for extraction of features, was applied directly on the original video sequence; this should result in a set of local features together with global features which describe the time evolution of the trjaectories of the bounding bozes, however in order to recognize the child's physical activities, we need to extract the description feature of the child only, not all other subjects present in the scene. Hence, the bounding boxes pertaining only to the child should be detected and tracked first. Then the feature extraction method would be applied. Given the bounding boxes of the child, one can restrict the calculation area of dense trajectory feature extraction method, so that only features within the child's bounding box is taken into account.

8.10 Experimental results

Referred to Section 8.4, the experimental setting, the following Table 8.2 presents the recognition accuracy given different experimental settings and using LSTM classifier.

It can be seen that by using CPM to detect human skeleton and then we extract feature as coordinates of these skeleton's points located on various body part, the classification ac-

Table 8.2: LSTM's recognition performance

Data processing	Accuracy	Precision	Recall	F-measure
Skeleton based	0.703	0.684	0.703	0.692
Setting 1	0.712	0.749	0.702	0.724
Setting 2	0.734	0.773	0.735	0.753
Setting 3	0.815	0.819	0.815	0.816

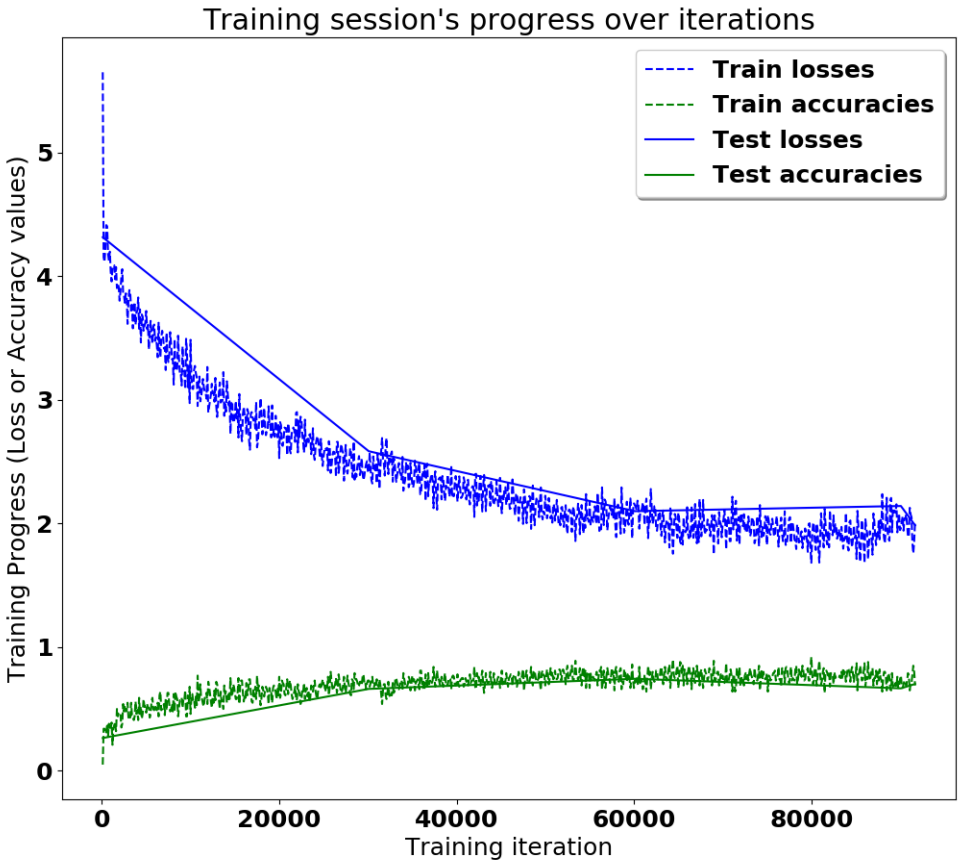


Figure 8.7: Training and testing performance or LSTM

curacy is about the poorest results which is only 70.3% (see Table 8.2). These are used as baseline results. Applying feature extraction after Yolo 2 for child detection (i.e. only selecting the smallest bounding box and assuming that is the child's bounding box) would be associated with the LSTM accuracy being 71.2% as listed as setting 1 in the table 8.2.For setting 2, we added tracking algorithm after the detection step, so that the recognition ac-

Table 8.3: LSTM result - confusion matrix for setting 3

Sedentary	**0.9360**	0.0400	0.0240	0.0000	0.0000
Light activities	0.0280	**0.8252**	0.1049	0.0000	0.0420
MV activities	0.0167	0.1667	**0.7250**	0.0750	0.0167
Walking	0.0000	0.0333	0.1667	**0.6333**	0.1667
Running	0.0000	0.0000	0.0286	0.1714	**0.8000**

Table 8.4: skeleton feature - Confusion matrix

Sedentary	**0.9240**	0.0367	0.0290	0.0070	0.0033
Light activities	0.1908	**0.5664**	0.1721	0.0216	0.0492
MV activities	0.1760	0.2624	**0.4658**	0.0390	0.0569
Walking	0.1360	0.2237	0.2412	**0.2763**	0.1228
Running	0.1444	0.2535	0.2359	0.1514	**0.2148**

curacy was boosted to 73.4% which by 2% better than the case without using the tracking approach. Finally, for setting 3, by applying human re-identification to enhance the tracking quality, the recognition outcome was associated with the best experimental result which is 81.5% in accuracy.

If compared with the base line that we used skeleton feature extraction, the final classification accuracy was improved more than 10% which is significant since this approach is non-intrusive and is not required any devices and sensors being attached to the subject's body parts.

Figure 8.7 shows the training and testing performance of LSTM given the time series feature extracted in the setting 3. The network starts to converge from around 50k training iterations, since from this point onward we do not observe any testing accuracy improvement. The confusion matrix of this experiment is given in Table 8.3. Table 8.4 presents the confusion matrix result of LSTM for the skeleton feature input space. It can be observed that skeleton feature contains very much of noise which should be the confusion between the skeletons detected of the kids and those of other people appeared in the scene. For this reason, the only Sedentary class was associated with relatively good recognition performance as in Table 8.4. For the case of the best accuracy shown in Table 8.3, the walking input

samples are most difficult to classified since this might be the action being most confused with other actions which also contains more or less amount of walking action.

8.11 Conclusion

In this chapter, small children physical activity recognition problem has been addressed given a number of video sequences. Steps of pre-processing video to label the segments of video corresponding with each of five activity types has been conducted. Several experimental settings and associated results have been drawn, which include human skeleton feature extraction, object detection and tracking, human re-identification for tracking enhancement and dense trajectory feature extraction for video sequences. The experiments indicated that the carefully feature preparation approach i.e. the method using all techniques like detecting human, tracking with the use of human re-identification and dense trajectory based feature extraction, produced the best recognition accuracy which is 81.5% and is more than 10% better than the base line which used the skeleton feature extraction approach.

Closely related to human action recognition in stored video sequences, the automatic video annotation is a kind of algorithm that allows naming what type of activities occur in the video clips using unsupervised algorithms or feature-based comparison approach. This automated procedure would help to improve the video quality and to result in more accurate annotation when labeling the video segments. In addition, the real-time activity recognition on video like in [159] and in [160] is also very interesting subject that is worth exploring. In term of real-time processing, the algorithm is required to process video sequence and to response quickly to the actual action occurring in reality. Due to the time limitation, we would wish to investigate this in the future research.

Chapter 9

Comparisons and Discussions

This Chapter compares the results of different prediction models in this thesis. The state-of-the-art accuracy performance for each dataset will be used to compare with our approach. While accelerometry is the methodology of choice for capturing and assessing PA and sedentary behaviour in young children [161, 162] , this thesis also considered a video sequence dataset for validation and comparisons of results.

The best results from the various methods in this Thesis are compared and discussed. Three accelerometry datasets were available for the work in this thesis. For the comparisons we will differentiate between accelerometry subsets as follows:

Dataset	**Subsets**
PA2012	1. Hip mounted accelerometry
PA2014	1. Hip mounted accelerometry 2. Hip and wrist mounted accelerometry

The PA2016 dataset became available to us in late 2017 and after the work on the SSEN was done. Thus, for the SSEN we have obtained results for the PA2012, PA2014 Hip only, and PA2014 Hip + Wrist datasets. In contrast, for the transfer learning models we found the PA2012 unsuitable as a background domain and hence we obtained results by using the PA2014 Hip + Wrist and the PA2016 GeneActive and ActivGraph datasets.

Table 9.1: Comparing results of different data modelling methods.

Dataset Name	**Models**	**Accuracy**
PA2012 data	Regression-based or cut-point method [5]	0.5900
	Traditional ANN [128]	0.8840
	SSEN model	**0.9111**
PA2014 Hip data	Traditional ANN [129]	0.8000
	SVM model [129]	0.8400
	SSEN model	**0.8990**
PA2014 Hip+Wrist data	Traditional ANN [129]	0.8100
	SVM model [129]	0.8550
	SSEN model	**0.9211**

The HRSOM was deployed to obtain a general overview of the data. A number of common learning systems such as the standard feed-forward MLP [128, 129], and SVM [129] have been used to verify published results or to explore suitability for the given PA prediction tasks. This thesis also introduced novel methods such as the SSEN (Chapter 6) and a transfer learning model (Chapter 7) in an attempt to address relevant shortcomings of the standard models. In this process we discovered that it is beneficial to incorporate the HRSOM with MLP in an ensemble manner (Chapter 5).

We found that traditional ANN/MLPs perform much better than regression models which were, prior to this thesis, commonly considered for PA prediction problems [5]. We also found that the performance of the MLP is hampered when dealing with the limited number of samples and relative high dimensional input space in the available datasets. Due to this sparsity in the available accelerometer data, an SSEN model was designed. The SSEN increases the data density along decision boundaries between classes thus addressing the lack of coverage of the feature space by the training input samples. The SSEN model is robust because it uses the best parameters selected within our three proposed sampling approaches, namely core-point group sampling, same-class orientation sampling and range-based sampling. A major benefit of the SSEN is that the algorithm is data driven.

It is seen in Table 9.1 that for the PA2012 dataset, the SSEN outperforms the current best

accuracy results of [128] by a significant margin (2.7%) to reach an accuracy 91.1%. At the time of the experiments, the SSEN produced a new state-of-the-art result.

For the PA2014 dataset we considered two subsets (i.e. PA2014 Hip only and PA2014 Hip + Wrist accelerometry) where the latter is simply concatenating the two sensor streams. The consideration of these subsets is made to make a consistent comparison with other work shown in [129]. The previous best result for the PA2014 Hip only data was 84% in accuracy. The result was produced by using an SVM prediction model [129]. The SVM result bettered the result of an MLP approach by 4% using the same evaluation method [129]. It can be seen in Table 9.1 that the SSEN boosts the accuracy performance significantly to 89.9%. An improvement of 5.9% over the previous state-of-the-art result. Similarly, given the PA2014 Hip + Wrist dataset, the SSEN model achieved an accuracy of 92.1%. An improvement by 6.6% over the previous best result that was produced by the SVM, and by 11.1% better than an MLP [129]. The result confirms that the standard methods are affected by the data sparsity in these datasets and that the SSEN is effective in addressing the problem. The observed improvements in results are consistently significant.

This thesis then investigated, developed and tested transfer learning approaches. An aim was to investigate whether it is possible and beneficial to transfer modelled knowledge from a background (source) domain to a target domain. Differences in the categorization of samples between the two domains presented a main challenge that this thesis overcame. Investigations and experiments were conducted by using the PA2014 hip + wrist dataset and the PA2016 Actigraph (or GeneActive) subset as the source to build a background model. Then to create a transfer learning model by using the PA2016 GeneActive (or ActiGraph) subset as the target domain as illustrated in the following Table:

Source Domain	**Target Domain**
{PA2014, PA2016 ActiGraph} accelerometry	PA2016 GeneActive accelerometry
{PA2014, PA2016 GeneActive} accelerometry	PA2016 ActiGraph accelerometry

We investigated three types of transfer models: (1) Unchanged architecture in which the

Table 9.2: Comparing results of different models

Dataset Name	**Models**	**Accuracy**
PA2016 GeneActiv dataset	MLP	0.7734
	SVM	0.7857
	FRPN	0.7635
	ETL-model-A	0.8258
	STL-model-A	**0.8516**
PA2016 ActiGraph dataset	MLP	0.7560
	SVM	0.7895
	FRPN	0.7784
	ETL-Model-B	0.7954
	STL-model-B	**0.8316**

architecture of the background model is kept unchanged. The last hidden layer of which is then updated on the data from the target domain; (2) Fully connected hidden layers are added in front of the output layer to the background model. The parameters of the newly added hidden layers are then trained on the data in the target domain. We called this model the Extended Transfer Learning (ETL) model; (3) A FRPN is added to the background model in front of the output layer. The parameters of the FRPN are trained by using the data in the target domain. This model was called the Stacked Transfer Learning (STL) model.

The results in Table 7.7 and Table 9.2 revealed that PA prediction can benefit from transfer learning if the background domain is related to the target domain. The type and architecture of the transfer learning model plays an important role in the effectiveness of the transfer learning methodology. Of the three investigated types the STL model has shown to work best by far. One explanation for this result is that FRPN simulates a deep neural network which however can adjust the depth according to each training sample. This is achieved while using only a few hidden neurons. The FRPN thus features the benefits of deep learning architectures while the number of adjustable parameters remains small. The model is thus less likely to overfit. These properties would allow the FRPN to learn effectively despite the sparsity of the training space, and the complexity of the learning problem which could benefit from a deep learning framework. The results in Table 9.2 show that for the

PA2016 GeneActiv target domain data, the best transfer learning approach boosts the prediction accuracy to 85.16%. This is an improvement by almost 8% over the baseline model. For the PA2016 ActiGraph target domain data, the best transfer learning model produced a prediction accuracy of 83.16%. Again, an improvement by almost 8% over the baseline. The FRPN model training by itself on the target domain data is not much better the standard MLP since the FRPN behaves similar to MLP especially in cases of small and sparse datasets.

To investigate whether the data quality of the PA accelerometer data played a role that prevented us from obtaining even better result we then investigated the use of videos as an alternative data source. This is the first time that videos were used for PA prediction of young children. While the videos were recorded for inspection and protocol monitoring purposes here in this thesis we use these videos for the PA recognition purpose. By doing so, this thesis overcame difficulties such as multiple people appearing at the scene at the same time, or educators occlude the child of interest. Multiple video preprocessing steps were conducted to label the segments of video corresponding to the five activity types being conducted. Several experimental settings and associated results have been obtained. The thesis has shown that the human skeleton feature extraction approach led to a relatively low prediction accuracy (70.3%). In contrast, the most effective approach is much more complex and involves the combination of object detection and tracking, human re-identification for tracking enhancement and dense trajectory feature extraction for video sequences. The experiments revealed that this approach accurately recognizes activity classes 81.5% of the time, an improvement by over 10% over the baseline. The video-based prediction results thus show that video based PA prediction is a viable alternative to accelerometry based PA prediction. In practice however, video capturing devices are more expensive and much less versatile in following a mobile actor when compared to accelerometers. The results would not support a motion to substitute the currently accepted approach based on accelerome-

ters with a video based approach. Nevertheless, the video based results are close to those obtained by using accelerometer data. Given that the quality of the video data is affected by i.e occlusion or actor leaving the scene, and given that the quality of the accelerometer data is affected by episodes of activities that do not correspond to the target label, and given that the two sets of results are comparable we can thus conclude that both , the video based approaches and the accelerometer based approaches, are affected by the data quality. This implies that it may not be possible to achieve 100% accurate predictions when using the available datasets. In this light, our results do not only mark the state-of-the-art on PA prediction of young children but that our results would also be close to a maximum achievable results.

In summary, the methods proposed in this thesis (SSEN in particular) produced better accuracy results than any other published work applied to the same problems. A main finding is that the sparsity of the datasets are a main factor which inhibits the performance of prediction models. While the problem could be addressed by collecting more data (a costly exercise) this thesis has shown that the data driven SSEN can achieve significant improvements in results by using available data.

Chapter 10

Conclusion

This thesis studied the research question on how to accurately classify physical activities of young children from accelerometer data. The research was motivated by the importance of physical activities to early childhood development and by the lack of studies on developing suitable data modelling approaches. To address this research question this thesis developed and evaluated several machine learning approaches. The work resulted in novel machine learning methods which are capable of processing sparse, noisy, high-dimensional accelerometer data for classifying physical activity classes of young children.

There are two sets of experiments conducted on different cohorts of young pre-school children over two different time span: 2014 and 2016 respectively. Multiple accelerometers were attached to different parts of the subject's body, e.g., hip, left wrist, right wrist performing various assigned physical activities. The list of activities performed in the 2014 cohort is different both in number and type to those performed in 2016. Moreover for probity reasons, "evidence" videos of the corresponding sessions were recorded. These two data collections became the main task in this thesis: to "make sense" of such data recorded pertaining to the possibility of classifying the accelerometer recordings into one of 5 categories: sedentary, light, medium, walking, and running.

At the data preprocessing level, a minor task: to align the labels of the 2016 dataset,

into 5 categories, which the 2014 dataset is categorized. This was conducted using a K-mean clustering on the points in the display space of the HRSOM into 5 clusters. This was successful in that the in-class correlation is reasonably high, while the cross class misclassifications were at acceptable levels. A review of relevant literature revealed (1) a lack of adequately powered studies of activity classification in pre-school children, (2) a lack of investigations in suitably designed data modelling techniques, (3) a lack of investigation into the robustness and scalability of suitable data driven models, and (4) a lack of investigations on whether accelerometry captures sufficient information to allow accurate PA classifications. This thesis addressed these deficiencies by first introducing a scalable and capable data visualization technique called a High Resolution Self-Organizing Map (HRSOM). The HRSOM can be trained very efficiently to reveal intricate patterns which consequently assisted the understanding of the data collected. These insights led to an understanding of why standard data modelling techniques, like a multilayer perceptron with a single hidden layer, perform not too well in this context, and led to the development of a data sampling and modelling technique; a technique for generating more data where they are needed by inspecting the HRSOM display space, to improve the robustness of the modelling approach. The corresponding algorithm, the Synthetic Sampling Ensemble Network (SSEN) was shown to significantly enhance classification accuracies (by an improvement of about 10%) while maintaining a high F1 measure of about 90% (which indicates that the high generalization capability is not achieved at the expense of high false positives) when compared with a baseline model which is the classic multilayer perceptron with a single hidden layer on the PA2014 dataset. It was shown that the method is robust and scalable which could be applied to other problems with similar characteristics, like having a small dataset. Obviously there will be limits to the effectiveness of additional data in improving the generalization accuracy, and robustness of the SSEN, and such limitations would be left as problems for further research.

The thesis then investigated an alternative approach via a novel transfer learning approach, which encapsulates the knowledge in a source model, which consists of a multilayer perceptron with one or two hidden layers, and a target model which appends one or two hidden layers further on the "frozen" source model, and found that such an approach can be effective in improving model robustness when applied to a source domain consisting of the 2014 dataset and a subset of the 2016 dataset, and the target domain consisting of the other subset of the 2016 dataset. It is found that if the one or two hidden layers in the target model is replaced by a fully recursive perceptron network (a fully connected recursive layer which is essentially an expanded deep multi-hidden layer with data dependent depth), the generalization accuracy is further improved. This shows that the proposed transfer learning approach may be deployed which may transfer the knowledge gained from the source domain be retained to improve the classification accuracies in the target domain. Obviously there will be limits on which the generalization accuracies may be improved using such an approach, and this is left as a problem for future research.

The deployment of methods described in this thesis resulted in state-of-the-art results though it still yield a classification error of around 9%. This accuracy probably could not be improved further due to the inherent noise in the accelerometer recordings: despite what the young child was doing during the episode, the corresponding accelerometer recordings was labelled by the assigned activity type. Therefore, there could be significance of what the assigned label should be when compared with the actual activities which the young child was performing at the recording session. This inherent source of labelling "noise" would place an upper limit to the generalization accuracy using either the SSEN or the transfer learning generalization accuracies. The impacts of such "noisy" labels on the SSEN and the proposed transfer learning approach was not investigated in this thesis, but instead would be left as a problem for future research.

As indicated, there are the "evidence" videos which are available and they were manu-

ally labelled. This thesis investigated two main approaches to extracting features from the videos, one using the dense trajectory approach (the so-called classical approach which uses hand-crafted features, e.g., histogram of oriented gradients, histogram of oriented optical flows, motion boundary histograms, evaluated on a set of dense interest points in each frame of the video, together with the trajectory tube which is "carved" by the dense set of trajectories on those dense sets of interest points in each frame and the classifier is using normalized Fisher vector on a chi-square kernel machine), and deep CNN on the trajectory "carved" out by the bounding boxes enclosing the intended subject, reconstructed using re-identification techniques if they were occluded in view). The results of the deep CNN with re-identified reconstruction of the bounding boxes approach showed that videos are a viable alternative of information for the PA classification task as they yielded comparable results to those obtained using the SSEN. An analysis of the results also provided indications which confirm that the "noisy" labelling of the accelerometer data s a contributing factor to the observed classification errors.

The following section summarizes the major contributions and findings of this thesis. Research limitations and future research directions will be presented later in this chapter.

10.1 Summary of major contributions and findings

1. An unsupervised visualization technique, the high resolution Self-Organizing Map (HRSOM), has been introduced for the purpose of data analysis. The model helps to expose the intricate characteristics of the input feature space thus assisting domain understanding and the analysis of results. The model was evaluated on a number of benchmark datasets, both artificial and real-world data. The experimental results showed that the HRSOM is not only useful as a data visualization tool, but also that the method is useful as a way to augment the original input feature vectors by their corresponding co-ordinate locations on the HRSOM display space, as augmented in-

puts to a supervised training algorithm, e.g., a multilayer perceptron.

2. It was found that the accelerometer data is relatively sparse, particularly with the choices of large window sizes during the feature extraction process, and that it is possible to augment such data by using a supervised DBSCAN technique, which labels each point in the HRSOM display space with a class label, and thus permit the visual identification of where more data might be required, and the corresponding additional generated point is generated by linear interpolation of the two closest existing points to the location of the generated point. Such synthetically generated features can be used to augment the lack of training data in a perceived cluster. The original inputs together with their corresponding co-ordinates on the HRSOM display space served as the augmented training dataset in the training of a multilayer perceptron. This scheme is called a synthetic Sampling Ensemble Network (SSEN). The SSEN is effective in overcoming the lack of training samples in the training dataset. The model has been experimentally shown to be better than other well-known sampling techniques, e.g., oversampling, undersampling.

3. The idea of transfer learning has been explored. A new transfer learning method is introduced which consists of a source model, a multilayer perceptron with one or two hidden layers, having its weights frozen after it has been trained on the source domain data, and the target model consists of additional one or two hidden layers appended to the frozen source model. It was shown that such a transfer learning approach is able to improve the generalization accuracies on the target domain. Moreover, if the appended one or two hidden layers of the target model is replaced by a fully recursive layer, i.e., a direct feedback connected output to the input of the feedforward hidden layer, further improvements in the generalization accuracies may be observed.

4. The "evidence" videos were processed using two different approaches: one using the

body keypoint detection approach, while the second approach uses a deep CNN to extract features, from the bounding boxes enclosing the intended subject, these bounding boxes can be reconstructed when occluded using re-identification techniques, and the classifier is the deep temporal LSTM. When applied to the physical activity videos, it is found that the deep CNN, with human re-identification approach produces better generalization accuracies than the one using the body keypoint detection approach. From such experiments, it is confirmed that the "noisy" labels in the accelerometer recordings has considerable impact on its generalization accuracies using either the SSEN or transfer learning techniques.

10.2 Research limitations

Despite the aforementioned contributions and findings, the work presented in this thesis was often not straight forward. Some of the issues that arose can lead to future research and are given as follows:

- There is no clear theoretical statement to prove that the HRSOM output in the display space is useful and interpretable. This may be related to the nature in which the HRSOM is deployed in this thesis: mainly as a visualization tool. It is difficult to show theoretically the properties or patterns of what our human eyes and mind can perceive. Moreover, the HRSOM, is serving as an unsupervised learning device in an ensemble system. The learning mechanism is not online. This means that the individual methods in the ensemble system are trained in a hierarchical manner, one at a time. Thus, for example, after training a HRSOM, the resulting activation map (co-ordinates of the location of the point in the display space) is then used for the training of a supervised learning model e.g., a multilayer perceptron. Hence, there exists a research opportunity on the unification of the learning algorithms such that the individual models can be trained towards the same goal of classifying an unknown

test sample as accurately as possible. Such research could result in a novel deep learning algorithm which can be expected to perform better than the method used in this thesis.

The lack of interpret-ability, in terms of why the classifier cannot classify a particular unknown testing sample correctly, is also observed with the SSEN algorithm. The capabilities of the SSEN were not formally shown. The proposed synthetic sampling approach leaves much room for further exploration such as, for example using a deep learning framework instead of the MLPs, or engaging the concept of generative adversarial networks in a controlled fashion, guided by the clustering results of the supervised DBSCAN, to generate artificial samples by using either core or border points (as defined in the supervised DBSCAN technique) as the input space.

- There are several types of data sources. Accelerometer data of various sampling rates, accelerometer measurements taken at different locations (wrists, hip), accelerometer data captured by different devices, video sequences, data captured in different sets of activity trials. The thesis has shown that the combination of data sources (i.e. combining hip and wrist data) can enhance the generalization abilities of a classifier. It was also shown that the various data sources offer complementary information. It should thus be interesting to take a broader approach to data augmentation by e.g. combining information captured by the accelerometers with information captured by the camcorders. This may result in a more comprehensive and informative input space. One approach to achieve this would be to have data augmentation done on the algorithm level where the outputs of a set of models that have been trained on the various different data sources is combined to create a final result.

- The number of participants involved in the data collection process is small (relative to the size of datasets normally used in machine learning). The input space would become even smaller if we had selected larger window sizes during the feature extrac-

tion process. In addition, the dimensionality of the input samples would increase with the window size. On the other hand, smaller windows sizes would increase the number of training samples and reduce the dimensionality of the data space. However, if the window size is too small then the data points do no adequately cover the pattern of the physical activities and thus a classification model could not predict well the type of activity if based only on quite a short bit of time series information. The windows size used in this thesis is a compromise between data dimensionality and number of data samples.

10.3 Future Research Directions

This thesis introduced several approaches to solve the physical activity recognition in young children. This research topic is still quite new and the problem domain is more challenging than the better studied area of activity prediction of older subjects. There are several open questions and problems that could be addressed in future research:

- Further studies on the theoretical properties of the HRSOM learning model in particular with respect to the effectiveness when used as a filter in an ensemble architecture could be conducted. An integrated training algorithm that trains the components of such an ensemble in unison is another aspect worth taking into consideration. For example, the learning algorithm could consider the HRSOM to be an internal unsupervised layer within a deep learning network with subsequent layer could be convolutional instead of fully connected layers. The HRSOM's neuron weights could be updated periodically while the convolutional of fully connected layers are being trained. The approach could find an inspiration from deep learning architectures where the first preprocessing layer is trained without utilizing a class label.

- The experiments on video based activity recognition requires more exploration in

terms of automatic video annotation, feature extraction and classification models. Firstly, the video segmentation or video segment division from the long and continuous video sequence has been done manually. The labelling process was tedious and time consuming. The automated algorithm should be taken into account for the video annotation part. Secondly, there are various types of feature extraction methods based on video sequence input. These methods then would be compared to find the best approach for this particular problem.

- Alternative approaches such as those based on Long Short Term Memory, deep Convolution Neural Networks and Graph Neural Networks would be worth exploring. It would also be interesting to study these methods in a learning ensemble. This has not yet been considered in PA prediction of young children and would thus be worth a consideration for future research. Since there exist several types of time series data such as raw acceleration signal and video sequences, using augmentation methods for both the data level and the model level could result in further improvements in the recognition accuracy.

- The transfer learning methods explored in this thesis did not take the unbalanced nature of the problem domain into account. Incorporating a sampling technique would likely lead to enhanced results. A good and interesting approach would be to substitute the MLP in the proposed Synthetic Sampling Ensemble Network by the proposed stacked transfer learning model.

- There has been an increasing requirement for real-time recognition tasks, such as realizing the human activities based on streamlined information extracted from video, camera, wearable sensors. The main purpose is to establish a recognition system that is able to respond quickly to met the human requirement in particular for decision making tasks. Such real-time system needs to be optimized to become fit for

deployment. Studies on the possibilities of implementation for clinical uses, and on the implementation for real-time mobile assessments would significantly enhance the translational value of work on PA prediction of young children.

References

[1] Australian Government Department of Health. (2017) National physical activity recommendations for children 0-5 years. [Online]. Available: http://www.health.gov.au/internet/main/publishing.nsf/Content/npra-0-5yrs-brochure

[2] A. P. Hills, N. A. King, and T. P. Armstrong, "The contribution of physical activity and sedentary behaviours to the growth and development of children and adolescents," *Sports medicine*, vol. 37, no. 6, pp. 533–545, 2007.

[3] S. G. Trost, L. Kerr, D. S. Ward, and R. R. Pate, "Physical activity and determinants of physical activity in obese and non-obese children," *International journal of obesity*, vol. 25, no. 6, pp. 822–829, 2001.

[4] Australian Government Department of Health, "Australia's physical activity and sedentary behaviour guidelines," *Department of Health Website*, 2014.

[5] S. G. Trost, W.-K. Wong, K. A. Pfeiffer, and Y. Zheng, "Artificial neural networks to predict activity type and energy expenditure in youth," *Medicine and science in sports and exercise*, vol. 44, no. 9, pp. 1801–1809, 2012.

[6] J. Staudenmayer, D. Pober, S. Crouter, D. Bassett, and P. Freedson, "An artificial neural network to estimate physical activity energy expenditure and identify physical activity type from an accelerometer," *Journal of Applied Physiology*, vol. 107, no. 4, pp. 1300–1307, 2009.

[7] S. C. Mukhopadhyay, "Wearable sensors for human activity monitoring: A review," *IEEE sensors journal*, vol. 15, no. 3, pp. 1321–1330, 2015.

[8] S. G. Trost, D. Cliff, M. Ahmadi, N. Van Tuc, and M. Hagenbuchner, "Sensor-enabled activity class recognition in preschoolers: Hip versus wrist data," *Medicine and science in sports and exercise*, vol. 50, no. 3, pp. 634–641, 2018.

[9] T. Shany, S. J. Redmond, M. R. Narayanan, and N. H. Lovell, "Sensors-based wearable systems for monitoring of human movement and falls," *IEEE Sensors Journal*, vol. 12, no. 3, pp. 658–670, 2012.

[10] Y.-C. Kan and C.-K. Chen, "A wearable inertial sensor node for body motion analysis," *IEEE Sensors Journal*, vol. 12, no. 3, pp. 651–657, 2012.

[11] D. Fuentes, L. Gonzalez-Abril, C. Angulo, and J. A. Ortega, "Online motion recognition using an accelerometer in a mobile device," *Expert systems with applications*, vol. 39, no. 3, pp. 2461–2465, 2012.

[12] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S.-M. Makela, and H. Ailisto, "Identifying users of portable devices from gait pattern with accelerometers," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2. IEEE, 2005, pp. ii–973.

[13] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657–675, 2009.

[14] J. Wu, G. Pan, D. Zhang, G. Qi, and S. Li, "Gesture recognition with a 3-d accelerometer," in *Proceedings of the International Conference on Ubiquitous Intelligence and Computing*. Springer, 2009, pp. 25–38.

[15] H. Zhong, J. Shi, and M. Visontai, "Detecting unusual activity in video," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2004.

[16] C. Dhiman and D. K. Vishwakarma, "A review of state-of-the-art techniques for abnormal human activity recognition," *Engineering Applications of Artificial Intelligence*, vol. 77, pp. 21–45, 2019.

[17] K. Pawar and V. Attar, "Deep learning approaches for video-based anomalous activity detection," *World Wide Web*, vol. 22, no. 2, pp. 571–601, 2019.

[18] V. Vishwakarma, C. Mandal, and S. Sural, "Automatic detection of human fall in video," in *Proceedings of the International conference on pattern recognition and machine intelligence*. Springer, 2007, pp. 616–623.

[19] U. Asif, B. Mashford, S. Von Cavallar, S. Yohanandan, S. Roy, J. Tang, and S. Harrer, "Privacy preserving human fall detection using video data," in *Proceedings of the Machine Learning for Health Workshop*, 2020, pp. 39–51.

[20] N. Robertson and I. Reid, "A general method for human activity recognition in video," *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 232–248, 2006.

[21] L. Bruckschen, S. Amft, J. Tanke, J. Gall, and M. Bennewitz, "Detection of generic human-object interactions in video streams," in *Proceedings of the International Conference on Social Robotics*. Springer, 2019, pp. 108–118.

[22] M. Subedar, R. Krishnan, P. L. Meyer, O. Tickoo, and J. Huang, "Uncertainty-aware audiovisual activity recognition using deep bayesian variational inference," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6301–6310.

[23] J. M. Chaquet, E. J. Carmona, and A. Fernández-Caballero, "A survey of video datasets for human action and activity recognition," *Computer Vision and Image Understanding*, vol. 117, no. 6, pp. 633–659, 2013.

[24] T. Singh and D. K. Vishwakarma, "Human activity recognition in video benchmarks: A survey," in *Advances in Signal Processing and Communication*. Springer, 2019, pp. 247–259.

[25] M. Hagenbuchner, D. P. Cliff, S. G. Trost, N. V. Tuc, and G. E. Peoples, "Prediction of activity type in preschool children using machine learning techniques," *Journal of Science and Medicine in Sport*, vol. 18, no. 4, pp. 426–431, 2015.

[26] S. E. Crouter, J. R. Churilla, and D. R. Bassett Jr, "Estimating energy expenditure using accelerometers," *European journal of applied physiology*, vol. 98, no. 6, pp. 601–612, 2006.

[27] S. E. Crouter, K. G. Clowers, and D. R. Bassett Jr, "A novel method for using accelerometer data to predict energy expenditure," *Journal of applied physiology*, vol. 100, no. 4, pp. 1324–1331, 2006.

[28] P. S. Freedson, K. Lyden, S. Kozey-Keadle, and J. Staudenmayer, "Evaluation of artificial neural network algorithms for predicting mets and activity type from accelerometer data: validation on an independent sample," *Journal of Applied Physiology*, vol. 111, no. 6, pp. 1804–1812, 2011.

[29] A. N. Akansu and R. A. Haddad, *Multiresolution signal decomposition: transforms, subbands, and wavelets*. Academic Press, 2001.

[30] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proceedings of the Computer Vision and Pattern Recognition*, 2016, pp. 4724–4732.

[31] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1145–1153.

[32] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1302–1310.

[33] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

[34] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proceedings of the Advances in neural information processing systems*, 2015, pp. 91–99.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[36] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proceedings of the European conference on computer vision*. Springer, 2016, pp. 21–37.

[37] S. Shantaiya, K. Verma, and K. Mehta, "Multiple object tracking using kalman filter and optical flow," *European Journal of Advances in Engineering and Technology*, vol. 2, no. 2, pp. 34–39, 2015.

[38] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, 2004.

[39] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," *Intelligence*, vol. 23, pp. 4902–4912, 2015.

[40] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 58–66.

[41] L. Zhu, L. Yang, D. Zhang, and L. Zhang, "Learning a real-time generic tracker using convolutional neural networks," in *Proceedings of the IEEE International Conference on Multimedia and Expo*. IEEE, 2017, pp. 1219–1224.

[42] Y. Xu, L. Lin, W.-S. Zheng, and X. Liu, "Human re-identification by matching compositional template with cluster sampling," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2013, pp. 3152–3159.

[43] F. M. Khan and F. Bremond, "Multi-shot person re-identification using part appearance mixture," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2017, pp. 605–614.

[44] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3908–3916.

[45] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[46] Z. Zheng, L. Zheng, and Y. Yang, "A discriminatively learned cnn embedding for person re-identification," *The ACM Transactions on Multimedia Computing, Communications, and Applications*, 2017.

[47] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proceedings of the 15th ACM international conference on Multimedia*. ACM, 2007, pp. 357–360.

[48] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proceedings of the European conference on computer vision*. Springer, 2008, pp. 650–663.

[49] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *Proceedings of the 19th British Machine Vision Conference*. British Machine Vision Association, 2008.

[50] L. Yeffet and L. Wolf, "Local trinary patterns for human action recognition," in *Proceedings of the IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 492–497.

[51] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action Recognition by Dense Trajectories," in *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition*, Colorado Springs, United States, Jun 2011, pp. 3169–3176. [Online]. Available: http://hal.inria.fr/inria-00583818/en

[52] H. Wang, D. Oneata, J. Verbeek, and C. Schmid, "A robust and efficient video representation for action recognition," *International Journal of Computer Vision*, vol. 119, no. 3, pp. 219–238, Sep 2016.

[53] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE International Conference on Computer Vision*, Sydney, Australia, 2013. [Online]. Available: http://hal.inria.fr/hal-00873267

[54] J. R. Uijlings, I. C. Duta, N. Rostamzadeh, and N. Sebe, "Realtime video classification using dense hof/hog," in *Proceedings of the International Conference on Multimedia Retrieval*. ACM, 2014, p. 145.

[55] I. C. Duta, J. R. Uijlings, T. A. Nguyen, K. Aizawa, A. G. Hauptmann, B. Ionescu, and N. Sebe, "Histograms of motion gradients for real-time video classification," in *Proceedings of the International Workshop on Content-Based Multimedia Indexing (CBMI)*. IEEE, 2016, pp. 1–6.

[56] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[57] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[58] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 11, pp. 625–660, February 2010.

[59] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[60] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Kdd*, vol. 96, 1996, pp. 226–231.

[61] M. H. Hassoun, *Fundamentals of Artificial Neural Networks*. MIT Press, 1995.

[62] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed., M. Horton, Ed. Prentice Hall, 1999.

[63] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, pp. 179–211, 1990.

[64] C. Goller and A. Küchler, "Learning task-dependent distributed representations by backpropagation through structure," in *Proceedings of the International Conference on Neural Networks*, 1996, pp. 347–352.

[65] A. Sperduti, D. Majidi, and A. Starita, "Extended cascade-correlation for syntactic and structural pattern recognition," in *Advances in structural and syntactical pattern recognition*, P. W. P. Perner and A. Rosenfeld, Eds. Springer-Verlag, Berlin, 1996, vol. 1121, pp. 90–99.

[66] A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 714–735, May 1997.

[67] M. Gori, M. Maggini, and L. Sarti, "A recursive neural network model for processing directed acyclic graphs with labeled edges," in *Proceedings of the International Conference on Neural Networks*, vol. 2, July 2003, pp. 1351–1355.

[68] F. Scarselli, A. C. Tsoi, M. Gori, and M. Hagenbuchner, "Graphical-based learning environments for pattern recognition," in *Structural, Syntactic, and Statistical Pattern Recognition*. Springer Berlin / Heidelberg, 2004, vol. 3138, pp. 42–56.

[69] F. Scarselli, M. Gori, A. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, January 2009.

[70] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "Computational capabilities of graph neural networks," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 81–102, January 2009.

[71] S. Yong, M. Hagenbuchner, A. C. Tsoi, F. Scarselli, and M. Gori, "Document mining using graph neural network," in *Comparative Evaluation of XML Information Retrieval Systems*, N. Fuhr, M. Lalmas, and A. Trotman, Eds. Springer Berlin / Heidelberg, 2007, vol. 4518, pp. 458–472.

[72] D. Muratore, M. Hagenbuchner, F. Scarselli, and A. C. Tsoi, "Sentence extraction by graph neural networks," in *Proceedings of the 20th International conference on Artificial neural networks: Part III*, K. Diamantaras, W. Duch, and L. Iliadis, Eds. Springer Berlin / Heidelberg, 2010, vol. 6354, pp. 237–246.

[73] F. Scarselli, S. Yong, M. Gori, M. Hagenbuchner, A. C. Tsoi, and M. Maggini, "Graph neural networks for ranking web pages," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, September 2005, pp. 666–672.

[74] A. C. Tsoi, F. Scarselli, M. Gori, M. Hagenbuchner, and S. Yong, "A neural network approach to web graph processing," in *Web Technologies Research and Development*, Y. Zhang, K. Tanaka, J. Yu, S. Wang, and M. Li, Eds. Springer Berlin / Heidelberg, 2005, vol. 3399, pp. 2–38.

[75] L. Lu, R. Safavi-Naini, M. Hagenbuchner, W. Susilo, J. Horton, S. Yong, and A. C. Tsoi, "Ranking attack graphs with graph neural networks," in *Information Security Practice and Experience*, F. Bao, H. Li, and G. Wang, Eds. Springer Berlin / Heidelberg, 2009, vol. 5451, pp. 34–359.

[76] A. Pucci, M. Gori, M. Hagenbuchner, F. Scarselli1, and A. C. Tsoi, "Applications of graph neural networks to large–scale recommender systems some results," in *Proceedings of the International Multiconference on Computer Science and Information Technology*, 2006, pp. 189–195.

[77] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[78] S. Das, M. Koperski, F. Bremond, and G. Francesca, "Deep-temporal lstm for daily living action recognition," in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2018, pp. 1–6.

[79] M. Hagenbuchner, A. C. Tsoi, F. Scarselli, and S. J. Zhang, "A fully recursive perceptron network architecture," in *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2017, pp. 1–8.

[80] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[81] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[82] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.

[83] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[84] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, p. 153, 2007.

[85] H. Lee, C. Ekanadham, and A. Ng, "Sparse deep belief net model for visual area v2," in *Proceedings of the Advances in neural information processing systems*, 2007, pp. 873–880.

[86] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," *The Journal of Machine Learning Research*, vol. 10, pp. 1–40, 2009.

[87] Y. Bengio, "Learning deep architectures for ai," *Foundations and Trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[88] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, vol. 9, May 2010, pp. 249–256.

[89] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *arXiv preprint arXiv:1809.02165*, 2018.

[90] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 1, pp. 142–158, 2016.

[91] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[92] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[93] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[94] C.-C. Yang and Y.-L. Hsu, "A review of accelerometry-based wearable motion detectors for physical activity monitoring," *Sensors*, vol. 10, no. 8, pp. 7772–7788, 2010.

[95] M. Hagenbuchner, M. Gori, H. Bunke, A. C. Tsoi, and C. Irniger, "Using attributed plex grammars for the generation of image and graph databases," *Pattern Recognition Letters*, vol. 24, no. 8, pp. 1081–1087, 2003.

[96] C. Castillo, D. Donato, L. Becchetti, P. Boldi, S. Leonardi, M. Santini, and S. Vigna, "A reference collection for web spam," *SIGIR Forum*, vol. 40, no. 2, pp. 11–24, December 2006.

[97] Y. Research. (2012, June) Web spam collections. http://chato.cl/webspam/datasets/ crawled by the laboratory of web algorithmics, university of milan. [Online]. Available: http://law.di.unimi.it/

[98] S. D. Bay, D. F. Kibler, M. J. Pazzani, and P. Smyth, "The uci kdd archive of large data sets for data mining research and experimentation," *SIGKDD Explorations*, vol. 2, p. 81, 2000.

[99] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set," *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 18–31, 2016.

[100] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[101] A. Saraswati, V. T. Nguyen, M. Hagenbuchner, and A. C. Tsoi, "High-resolution self-organizing maps for advanced visualization and dimension reduction," *Neural Networks*, vol. 105, pp. 166–184, 2018.

[102] A. Forti and G. L. Foresti, "Growing hierarchical tree som: An unsupervised neural network with dynamic topology," *Neural networks*, vol. 19, no. 10, pp. 1568–1580, 2006.

[103] A. Skupin and A. Esperbé, "Towards high-resolution self-organizing maps of geographic features," *Geographic visualization: Concepts, tools and applications*, pp. 159–181, 2008.

[104] H.-U. Bauer and T. Villmann, "Growing a hypercubical output space in a self-organizing feature map," *IEEE Transactions on Neural Networks*, vol. 8, no. 2, pp. 218–226, 1997.

[105] T. Villmann and H.-U. Bauer, "Applications of the growing self-organizing map," *Neurocomputing*, vol. 21, no. 1, pp. 91–100, 1998.

[106] V. Sauvage, "The t-som (tree-som)," in *Proceedings of the 10th Australian Joint Conference on Artificial Intelligence: Advanced Topics in Artificial Intelligence*, ser. AI '97. London, UK, UK: Springer-Verlag, 1997, pp. 389–397.

[107] E. V. Samsonova, J. N. Kok, and A. P. IJzerman, "Treesom: Cluster analysis in the self-organizing map," *Neural Networks*, vol. 19, no. 6, pp. 935–949, 2006.

[108] S. Mathew and P. Joy, "Ultra fast som using cuda," *NeST-NVIDIA Center for GPU computing, hpc@ nestgroup. net*, 2010.

[109] Y. Xiao, C. S. Leung, T.-Y. Ho, and P.-M. Lam, "A gpu implementation for lbg and som training," *Neural Computing and Applications*, vol. 20, no. 7, pp. 1035–1042, 2011.

[110] P. Wittek and S. Darányi, "A gpu-accelerated algorithm for self-organizing maps in a distributed environment." in *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2012.

[111] S. McConnell, R. Sturgeon, G. Henry, A. Mayne, and R. Hurley, "Scalability of self-organizing maps on a gpu cluster using opencl and cuda," in *Proceedings of the Journal of Physics: Conference Series*, vol. 341, no. 1. IOP Publishing, 2012, pp. 012–018.

[112] S. Q. Khan and M. A. Ismail, "Design and implementation of parallel som model on gpgpu," in *Proceedings of the International Conference on Computer Science and Information Technology*. IEEE, 2013, pp. 233–237.

[113] T. G. Dietterich, "Ensemble methods in machine learning," in *Proceedings of the International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[114] X. Liu, G. Wang, Z. Cai, and H. Zhang, "Ensemble inductive transfer learning," *Journal of Fiber Bioengineering and Informatics*, vol. 8, no. 1, pp. 105–115, 2015.

[115] M. Galar, A. Fernández, E. B. Tartas, H. B. Sola, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches." *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 42, no. 4, pp. 463–484, 2012.

[116] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority oversampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[117] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *Proceedings of the 14th International Conference on Machine Learning*, D. H. Fisher, Ed. Morgan Kaufmann, 1997, pp. 179–186.

[118] G.-G. Geng, C.-H. Wang, Q.-D. Li, L. Xu, and X.-B. Jin, "Boosting the performance of web spam detection with ensemble under-sampling classification," in *Proceedings of the 4th International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 4. IEEE, 2007, pp. 583–587.

[119] S. Hido, H. Kashima, and Y. Takahashi, "Roughly balanced bagging for imbalanced data." *Statistical Analysis and Data Mining*, vol. 2, no. 5-6, pp. 412–426, 2009.

[120] Y. Zheng, W.-K. Wong, X. Guan, and S. Trost, "Physical activity recognition from accelerometer data using a multi-scale ensemble method," in *Proceedings of the IAAI*, 2013.

[121] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[122] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority oversampling technique." *Journal of Artificial Intelligence and Research*, vol. 16, pp. 321–357, 2002.

[123] Z. Chen, J. Wang, H. He, and X. Huang, "A fast deep learning system using gpu," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, 2014, pp. 1552–1555.

[124] S. Zheng, Y. Song, T. Leung, and I. Goodfellow, "Improving the robustness of deep neural networks via stability training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4480–4488.

[125] K. M. Rashid and J. Louis, "Times-series data augmentation and deep learning for construction equipment activity recognition," *Advanced Engineering Informatics*, vol. 42, pp. 1–12, 2019.

[126] L. Yang, L. Tao, X. Chen, and X. Gu, "Multi-scale semantic feature fusion and data augmentation for acoustic scene classification," *Applied Acoustics*, vol. 163, pp. 11 617–11 637, 2020.

[127] M. Kubat, "Neural networks: a comprehensive foundation by simon haykin, macmillan, 1994, isbn 0-02-352781-7," *The Knowledge Engineering Review*, vol. 13, no. 4, pp. 409–412, 1999.

[128] S. G. Trost, W.-K. Wong, K. A. Pfeiffer, and Y. Zheng, "Artificial neural networks to predict activity type and energy expenditure in youth," *Medicine and science in sports and exercise*, vol. 44, no. 9, pp. 1801–1809, 2012.

[129] S. G. Trost, D. Cliff, and M. Hagenbuchner, "Sensor-enabled activity recognition in preschool children: Hip versus wrist data," *Medicine and science in sports and exercise*, vol. 48, no. 5 Suppl 1, p. 313, 2016.

[130] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[131] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the rprop algorithm," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 1, 1993, pp. 586–591.

[132] S.-i. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4-5, pp. 185–196, 1993.

[133] L. D. Raedt and H. Blockeel, "Using logical decision trees for clustering." in *Proceedings of the 9th international workshop on inductive logic programming, ILP*, ser. Lecture Notes in Computer Science, vol. 1297. Springer, 1997, pp. 133–140.

[134] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[135] N. Moustafa and J. Slay, "The significant features of the unsw-nb15 and the kdd99 data sets for network intrusion detection systems," in *Proceedings of the 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS)*. IEEE, 2015, pp. 25–31.

[136] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[137] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 193–200.

[138] G. M. Y. Dauphin, X. Glorot, S. Rifai, Y. Bengio, I. Goodfellow, E. Lavoie, X. Muller, G. Desjardins, D. Warde-Farley, P. Vincent *et al.*, "Unsupervised and transfer learning challenge: a deep learning approach," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, 2012, pp. 97–110.

[139] Y. Bengio *et al.*, "Deep learning of representations for unsupervised and transfer learning." *ICML Unsupervised and Transfer Learning*, vol. 27, pp. 17–36, 2012.

[140] W. Li, L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 6, pp. 1134–1148, 2014.

[141] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, "Cross-domain sentiment classification via spectral feature alignment," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 751–760.

[142] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[143] S.-R. Ke, H. L. U. Thuc, Y.-J. Lee, J.-N. Hwang, J.-H. Yoo, and K.-H. Choi, "A review on video-based human activity recognition," *Computers*, vol. 2, no. 2, pp. 88–131, 2013.

[144] J. P. Lewis, "Fast template matching," in *Proceedings of the Vision interface*, vol. 95, no. 120123, 1995, pp. 15–19.

[145] K. Kawakami, "Supervised sequence labelling with recurrent neural networks," *PhD dissertation*, 2008.

[146] G. de Haan and E. B. Bellers, "De-interlacing of video data," *IEEE Transactions on Consumer Electronics*, vol. 43, no. 3, pp. 819–825, 1997.

[147] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of the European conference on computer vision*. Springer, 2014, pp. 740–755.

[148] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[149] X. Li, K. Wang, W. Wang, and Y. Li, "A multiple object tracking method using kalman filter," in *Proceedings of the IEEE International Conference on Information and Automation*, June 2010, pp. 1862–1866.

[150] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," *CoRR*, vol. abs/1602.00763, 2016.

[151] X. Zhang, G.-S. Xia, Q. Lu, W. Shen, and L. Zhang, "Visual object tracking by correlation filters and online learning," *Journal of Photogrammetry and Remote Sensing*, vol. 140, pp. 77 – 89, 2018.

[152] F. Alhwarin, D. Ristić-Durrant, and A. Gräser, "Vf-sift: very fast sift feature matching," in *Proceedings of the Joint Pattern Recognition Symposium*. Springer, 2010, pp. 222–231.

[153] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 152–159.

[154] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1116–1124.

[155] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based cnn with improved triplet loss function," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1335–1344.

[156] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in neural information processing systems*, 2012, pp. 1097–1105.

[157] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[158] L. Cao, Y. Tian, Z. Liu, B. Yao, Z. Zhang, and T. S. Huang, "Action detection using multiple spatial-temporal interest point features," in *Proceedings of the IEEE International Conference on Multimedia and Expo*. IEEE, 2010, pp. 340–345.

[159] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, and J. Liu, "Online human action detection using joint classification-regression recurrent neural networks," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 203–220.

[160] L. Meng, C. Miao, and C. Leung, "Towards online and personalized daily activity recognition, habit modeling, and anomaly detection for the solitary elderly through unobtrusive sensing," *Multimedia Tools and Applications*, vol. 76, no. 8, pp. 10 779–10 799, 2017.

[161] D. P. Cliff, J. J. Reilly, and A. D. Okely, "Methodological considerations in using accelerometers to assess habitual physical activity in children aged 0–5 years," *Journal of Science and Medicine in Sport*, vol. 12, no. 5, pp. 557–567, 2009.

[162] S. G. Trost, "State of the art reviews: measurement of physical activity in children and adolescents," *American Journal of lifestyle medicine*, vol. 1, no. 4, pp. 299–314, 2007.