Multimodal Perceptual Mechanisms for Unsupervised Self-Structuring Artificial Intelligence in Distributed Systems

Submitted by

Kanattege Madhura Chinthaka Jayaratne
M.Sc. 2016, UoM
B.Sc. Eng. (Hons) 2011, UoM

A thesis submitted in total fulfilment

of the requirements for the degree of

Doctor of Philosophy

Centre for Data Analytics and Cognition
La Trobe Business School
College of Arts, Social Sciences and Commerce
La Trobe University
Victoria, Australia

September 2020

To Amma and Thaththa

Contents

List of Tables

List of Figures

Abstract

This thesis aims to advance the knowledge on the development of multimodal perceptual mechanisms for artificial intelligent applications inspired by the findings of psychological, behavioural and neurobiological studies on human multimodal perception.

Related literature in this space has focused on modelling the multimodal dynamics of the human brain rather than on the application of the developed models to real-world problems, let alone the challenges posed by the vast amount of data generated in big data environments. Moreover, while most of the previous attempts have focused on the supervised paradigm, multimodal fusion techniques for unsupervised environments are still unresolved and an ongoing problem.

The proposed artificial perceptual model consists of a conceptual model, an architectural model and a computational model. It is grounded on the biological mechanisms of neocortex including the development of cortical patterns through neural self-organization, hierarchical organization of cortical areas generating progressively abstract representations and crossmodal connections facilitating different aspects captured via different modalities to interact/affect each other. The model is implemented with artificial cortical areas modelled by the growing self-organizing map (GSOM) algorithm (Alahakoon et al., 2000), multimodal interactions modelled with a multimodal distance metric and a clustering algorithm which facilitate the fusion of modalities based on their co-occurrence relationships. The implementation is demonstrated on multimodal datasets and evaluated in terms of the quality of the fused representation.

Most of the multimodal applications need to derive efficient representations to perceive the environment effectively. Moreover, the time taken to adapt/retrain the decision models in response to environmental changes needs to be reasonable. To this end, this thesis proposes a distributed architecture for improving the efficiency and scalability of multimodal fusion. The self-organizing mechanism of the architecture is realised through the Distributed GSOM algorithm, which is further adapted to three contemporary distributed computing platforms,

Apache Hadoop, Spark and Hama. Furthermore, the multimodal clustering algorithm is adapted to distributed computing, and a case study of the overall distributed implementation from the physical activity monitoring domain is presented.

Declaration

Except where reference is made in the text of the thesis, this thesis contains no material published elsewhere or extracted in whole or in part from a thesis accepted for the award of any other degree or diploma. No other person's work has been used without due acknowledgement in the main text of the thesis. This thesis has not been submitted for the award of any degree or diploma in any other tertiary institution.

This work was supported in part by an Australian Government Research Training Program Scholarship and in part by the Data to Decisions Cooperative Research Centre (D2D CRC).

Kanattege Madhura Chinthaka Jayaratne

26 August 2020

Acknowledgements

First and foremost, my deepest gratitude goes to my supervisor, Prof. Damminda Alahakoon, for his exceptional supervision with wisdom, sound advice and patience throughout my candidature. The inspiration, encouragement and support you provided were invaluable. The life lessons I learnt from you during this period would guide me through my professional and personal lives.

I would also like to thank my second supervisor, Dr. Daswin de Silva, for continuously engaging in the research and always encouraging me to do better. I am thankful to Prof. Paul Mather and Dr. Su Nguyen for their valuable support and guidance.

I am also thankful to my colleagues, Tharindu, Yujie, Rashmika, Achini, Quing, and Nilu for their friendship and making the PhD journey enjoyable. Our interesting discussions over lunch, both intellectual and casual, are something I cherish.

I am eternally grateful to my parents. *Amma* and *Thaththa*, thank you for your lifelong guidance and advice in every endeavour of my life. I am also thankful to my extended family for their kind words and encouragement.

My PhD wouldn't have been possible without my lovely wife, Dinithi, with whom I shared the PhD journey. Thank you for embarking on this adventure with me and being by my side through ups and downs. Finally, I would like to thank my lovely children, Thinuk and Akenya for all the joys they bring to my life.

Chapter 1
An Introduction

1.1 The Digital Environment and Rethinking AI

Compared to a decade or two ago, artificial intelligence applications operate in a very different landscape in terms of the variety, volume and volatility of data available for processing (Allam & Dhunny, 2019; O'Leary, 2013). In the past, available datasets were small, unimodal, isolated and infrequent. The AI applications had access only to a particular aspect of a situation or event at hand with only a small amount of data. Once the AI applications have processed these individual aspects of an event or situation, the humans in the loop were responsible for forming a holistic understanding of the situation to take appropriate actions (Carvalho et al., 2001; Dautenhahn, 1998; Falcone & Castelfranchi, 2001). However, the sensing and data capturing in the Big Data era have transformed the datasets being generated (Han et al., 2015; Ma et al., 2015; Rathore et al., 2015; Williams et al., 2017; Yin Zhang et al., 2014). Now the datasets are large, multimodal and multisource, dense and high frequent, starting to represent the natural environment more closely.

This phenomenon has created what we would like to call a *digital environment*, which is a closer representation of the natural environment than the one derived with smaller, unimodal, isolated and infrequent data. The examples include various Internet of Things (IoT) enabled environments such as smart home/office/city as well as applications such as autonomous robots/vehicles/drones. For instance, the latest autonomous robots carry state-of-the-art sensory devices such as multichannel microphones, various proximity sensors, high-resolution imagery sensors, force and tactile sensors (Noda et al., 2014), and each of these sensors captures a particular aspect of its environment. The key reason for placing such a vast array of sensors to capture multimodal sensory inputs in the first place is the inability of a single sensory modality to fully capture different aspects or features of the environment. Capabilities to fuse these multimodal sensory inputs are required to achieve a coherent and holistic understanding of the surrounding. Applications operating with limited information from single/limited sensory modalities pose risks to people and objects around them due to their restricted understanding of the surrounding. Autonomous applications that require real-time responses require computational means to fuse multimodal sensory inputs without any intervention from human operators.

The *digital environment* closely resembles how a human would perceive his environment. Humans do not sense their environment with disparate, unimodal, and limited sensory inputs. Instead, they perceive the surrounding in a continuous and holistic manner by analysing and fusing different sources of sensory excitations (Stein et al., 2009; Stein & Meredith, 1993). Multimodal nature of human sensation is key to the holistic perception as different sensory modalities represent different aspects of a given event or a situation. They carry complementary information, and when fused together, they support forming a more coherent picture of the underlying event or situation. Overall, the sensory modalities act as the mediums that transfer the features about the natural environment to the human, allowing him to perceive the environment as a continuum.

The artificial counterparts in such an environment are tasked with taking appropriate actions to maximise the likelihood of achieving their stated goals (Hanheide et al., 2017; Van den Berg et al., 2011). The key to success for them is the accurate sensing and perception of the external environment forming a coherent *impression* about it. A new breed of AI applications that operates in such data-intensive *digital environment* (Y. Pan, 2016) needs to form a holistic *impression* on the *digital environment*, similar to how humans would form a holistic *impression* on the natural environment from multimodal sensory excitations they receive (Bult et al., 2007; Edelman & Gally, 2013; Tononi et al., 1998). The findings on biological mechanisms enabling the accurate and holistic perception have played an inspirational role in developing artificial systems aimed at accurate sensing and perception, especially how multimodal, multisource sensor data could be integrated to improve the representation (Khacef et al., 2020; Velik, 2014). With environment sensing being performed across multiple modalities with high frequency, the *digital environment* represented by these data provides a more natural and realistic environment for artificial counterparts to interact and operate. The accurate perception of the external environment by fusing multimodal data sources is paramount for the success of new AI operating in the *digital environment* as it allows for autonomy and proactiveness compared to manual and human-driven AI applications in the past.

1.2 Motivation - in Brief

Sensory systems that capture different aspects of an event/object offer richer information about the same due to the fact that they jointly capture the same event/object in multiple modalities supplementing each other (Mareschal et al., 2012). For example, the vision of the speaker's face, especially the lips, greatly helps in the understanding what is being said (Schwartz et al., 2004). This positive effect of vision is greatly highlighted in instances where there is significant background noise, which is commonly referred to as the 'cocktail party' situation (Arons, 1992). Psychological, behavioural and neurophysiological studies studying this phenomenon have identified many cases of interactions between modalities, where the perception of one sensory modality is conditioned by the information simultaneously available to another (Choe

et al., 1975; Jack & Thurlow, 1973). Moreover, they carry complementary information which, when fused forms a more coherent picture of the underlying event/object. How humans effortlessly perform this fusion belies its complexity (Mareschal et al., 2012).

Psychological and behavioural studies have long examined the crossmodal effect between sensory modalities, including crossmodal influence and crossmodal calibrations (Harris, 1965; Radeau & Bertelson, 1974, 1977). These experiments have analysed the multimodal nature of human perception externally providing us with an abstract view of the process. Neurobiological studies, on the other hand, have revealed the regions in the brain and biological mechanisms which are responsible for the process. Recent advents of neuroimaging, which includes techniques such as functional magnetic resonance imaging (fMRI), has facilitated the study of neuronal activation at a single neuron level (Stein & Stanford, 2008). Such finer granularity was earlier restricted to non-human subjects such as primates and cats and has now been extended to study neuronal activations in the human brain (James & Stevenson, 2012). The knowledge accumulated about sensory fusion from these studies has encouraged computer scientists to model the dynamics of the brain in computational models. As highlighted earlier, the accurate perception of the external environment by fusing multimodal data sources is paramount for the success of new AI operating in the *digital environment* as it allows for autonomy and proactiveness compared to manual and human-driven AI applications in the past. Hence, the primary motivation of this thesis is the need for utilising psychological and neurobiological knowledge on multisensory fusion to facilitate coherent and holistic *impression* generation by artificial intelligence in the *digital environment*, and the current dearth of research on biologically inspired models of multisensory fusion.

While there have been limited attempts at building computational models of multisensory fusion, they have mostly focused on modelling the dynamics of the brain (Cuppini et al., 2010; Magosso et al., 2012; Rowland et al., 2007; Ursino et al., 2009) rather than on the application of the developed models on real-world problems, let alone the challenges posed by the vast amount of data generated in *digital environments.* In this work, we acknowledge the challenges

posed by the volume and variety of data generated and develop algorithms to perform multimodal data fusion at scale. Moreover, while most of the previous attempts have focused on the supervised paradigm, fusion techniques for unsupervised environments are still unresolved and an ongoing problem (Dasarathy, 2006). On the other hand, with the vast amount of data being generated, unsupervised learning mechanisms are important more than ever before due to the inability to label such large datasets. Motivated by this fact, the novel neural network algorithms presented in this thesis adhere to the unsupervised learning paradigm.

1.3 Research Objectives

Based on the above-stated motivation, the central goal of this research is to develop multimodal perceptual mechanisms for generating artificial *impressions* on *digital environments* inspired by the findings of psychological, behavioural and neurobiological studies on human multimodal perception. As stated above, we draw inspiration from the underlying innate mechanisms in the human brain that allows humans to perform fusion of multimodal sensory cues with seemingly no effort. The mechanisms include the development of cortical patterns through neural self-organization, hierarchical organization of cortical areas that process sensory cues generating progressively abstract representation as the hierarchy is traversed, and crossmodal links which facilitate different aspects of an event captured via different modalities to interact with/affect each other.

Based on the above goal, the main objectives of the research are as below.

- The key objective of this thesis is to design and develop multimodal perceptual mechanisms for artificial *impression* generation drawing upon the organization and functionality of the human brain to facilitate artificial counterparts to be autonomous and proactive by forming a holistic understanding of the *digital environment*.
- The second objective is to elevate these mechanisms and algorithms to support large volumes of data, which is a key characteristic of the *digital environment*, such that the above could be practicable in real-life situations.

- The third objective is to use and validate the developed algorithm on real-life and benchmark dataset and demonstrate their use in unsupervised learning settings.

1.4 Research Questions

Consistent with the above objectives, the main research question that is aimed to be addressed in this thesis is,

> *Inspired by sensing and perception mechanisms in the brain, how can unsupervised machine learning algorithms be developed for holistic data representation and fusion in digital environments?*

As the main research question is broad and abstract, sub research questions were drawn to identify different research areas that are discussed and addressed in this thesis. These research areas include neurobiology, psychology, unsupervised machine learning, parallel and distributed computing. The detailed research questions identified are as below.

1. How can sensing and perception mechanisms in the brain inspire data fusion for holistic representation in digital environments?
2. How can unsupervised machine learning be advanced to develop holistic multimodal data fusion algorithms?
3. How can the multimodal fusion algorithms in 2 be implemented for distributed computing paradigms to enable fusion at scale?
4. How can algorithms in 2 and 3 be validated using benchmark datasets and real-life environments?

Research question 1 is concerned with the investigation of sensing and perception mechanisms in the brain that allows for a seamless fusion of multimodal sensory data as inspiration for developing a holistic data representation mechanism in digital environments. Research along this line of inquiry is guided by questions such as,

- What psychological evidence of multimodal perception has been observed and what psychological models have been proposed?
- What theories on the organization of the human brain to support knowledge representation have been put forward?
- What theories on the dynamics of the human brain to support multimodal perception have been proposed?

Research question 2 is concerned with advancing unsupervised learning algorithms to develop multimodal fusion mechanisms inspired by the sensing and perception mechanisms in the brain for holistic representation in digital environments. Research along this line of inquiry is guided by questions such as,

- Are there any unsupervised learning algorithms that have been proposed for multimodal data representation and fusion; are there any limitations?
- Can the principles of self-organization be used to realise unsupervised learning for developing artificial cortical areas that represent information from individual modalities?
- How can co-occurrence of neuronal activations across modalities be used for developing a multimodal fusion mechanism?

With the understanding that generating efficient representations from large multimodal data sources is essential in most online application scenarios, research question 3 is concerned with implementing algorithms developed for research question 2 be adapted and implemented for distributed computing paradigms. Research endeavours targeted at this are in answer to questions such as,

- What is an appropriate distributed architecture for improving the efficiency and scalability of the multimodal fusion algorithm in order to provide results under acceptable computing times?

- How can self-organizing maps that are used to represent information from individual modalities be implemented for distributed computing paradigms, MapReduce (Dean & Ghemawat, 2008), Bulk Synchronous Parallel (BSP) (Valiant, 1990) and Resilient Distributed Dataset (RDD) (Zaharia et al., 2012)?

Research question 4 is concerned with the demonstration and evaluation of developed models and algorithms with benchmark datasets and real-life environments. This can be further elaborated with questions such as,

- How can the improvement in multimodal representation accuracy of the proposed multimodal fusion algorithm be evaluated with appropriate benchmark datasets?
- How can the efficiency gains attained by the distributed implementations be evaluated?

1.5 Contribution to Knowledge

The main contributions of this research can broadly be categorized under two categories; theoretical contributions and computational contributions.

1.5.1 Theoretical Contributions

We identify *multimodal fusion* to be the most critical feature in enabling artificial *impression* generation in *digital environments*. The fusion of multimodal inputs enables incorporating multiple aspects of a situation allowing for the formation of an unambiguous interpretation of the event from partial - and often ambiguous - information present in each modality. To this end, this thesis proposes an artificial model inspired by the human neocortex for the purpose of generating artificial *impressions* in *digital environments.* The proposed model consists of a conceptual model, an architectural model and a computational model. The conceptual model describes the abstract organization of multiple cortical layers and information flow among them. The architectural model consists of components that implement various sections of the conceptual model by generating the associated functionality while the computational model proposes the algorithmic means by which we propose to achieve this.

Most of the multimodal applications need to derive efficient representations from the multimodal sensory inputs to effectively perceive the environment. The efficient online fusion of data from multiple sensory modalities facilitates responding promptly when dealing with real-world situations. Moreover, the time taken to adapt/retrain the decision models in response to changes in the environment needs to be reasonable so that the decisions are not made with outdated models. To facilitate this, this thesis proposes a distributed architecture for improving the efficiency and scalability of the multimodal fusion algorithm in order to provide results under acceptable computing times.

1.5.2 Computational Contributions

This thesis makes three major computational contributions. The first is the implementation of the proposed architectural model for generating artificial *impressions* in *digital environments*. The proposed multi-layered architectural model is implemented with artificial cortical areas modelled by the GSOM algorithm and multimodal clustering algorithm allowing for the fusion of modalities based on their co-occurrence relationships. The multimodal clustering algorithm is based on the hypothesis that observations of an event recorded over multiple modalities should bear similarities across them due to natural regularities. The computational model is evaluated in terms of the quality of the fused representation.

The second major computational contribution is a data parallelised distributed SOM algorithm as an implementation of the distributed self-organizing components of the proposed architectural model. The algorithm is adapted to three distributed computing paradigms, MapReduce (Dean & Ghemawat, 2008), Bulk Synchronous Parallel (BSP) (Valiant, 1990) and Resilient Distributed Dataset (RDD) (Zaharia et al., 2012), and implemented on three contemporary platforms, Apache Hadoop, Hama and Spark. The empirical evaluations demonstrate super-linear speedup compared to the serial SOM using several benchmarking and real-life data sets.

The third major computational contribution is the distributed implementation of the multimodal clustering algorithm. Computationally heavy multimodal distance calculation and multimodal clustering algorithms have been adapted to use distributed computing to support the efficient online fusion of data from multiple sensory modalities. The distributed implementation on Apache Spark is evaluated with a dataset from the physical activity monitoring domain, which suffers from both large data volumes and multimodality.

1.6 Roadmap

This thesis presents the research contributions in detail in the chapters to follow. Figure 1.1 presents the organization of the rest of the thesis.

Chapter 2 provides a detailed introduction to the multimodal sensory perception in humans and a thorough review of the literature related to the psychological, neurobiological and computational aspects of the same.

In Chapter 3, we layout our premise on sensation and perception in humans. We discuss how humans construct the state of the external environment from the multimodal sensory inputs by forming what we call a coherent *impression* about the external world. We propose an artificial model, which consists of an architectural model as well as a computation model, for generating artificial *impressions* on *digital environments.*

The implementation of the above model is presented in Chapter 4. We demonstrate this artificial model on an audio-visual dataset and experiment with various parameters of the model. Results demonstrate that the multimodal representation achieves higher clustering accuracy compared to unimodal representation. Further, highlighting the necessity of generating efficient representations from multimodal data sources in most online application scenarios, we present a distributed architecture for online multimodal sensory fusion.

Chapter 5 discusses a distributed SOM algorithm as an implementation of the distributed self-organizing component of the proposed distributed architecture. The algorithm is adapted to

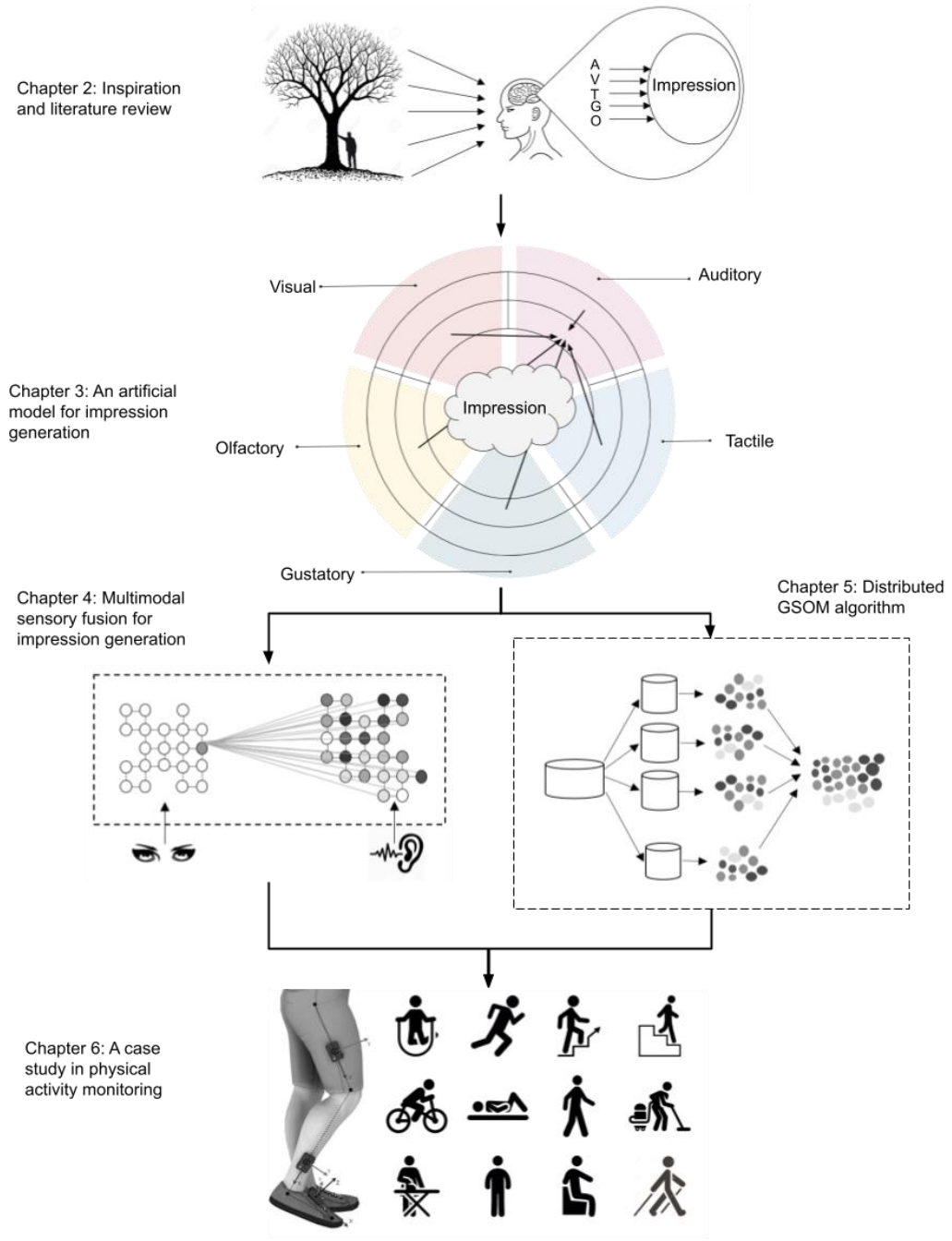

Figure 1.1 Thesis organization

three contemporary distributed computing platforms, Apache Hadoop, Spark and Hama and empirical evaluations, which demonstrate super-linear speedup compared to the serial SOM using several benchmarking and real-life data sets, are reported.

Chapter 6 presents the details of multimodal clustering component of the proposed architecture implemented for distributed computing. Moreover, we demonstrate the overall distributed

implementation using a case study from the physical activity monitoring domain, which suffers from large data volumes and multimodality.

Concluding in Chapter 7, the proposed models, architectures, implementations, results and contributions are discussed while outlining future research directions.

Chapter 2
Setting the Stage

The central goal of this research is to develop an artificial *impression* generation architecture for *digital environments* inspired by the findings of psychological, behavioural and neurobiological studies on human multimodal perception. Hence, it is necessary to have background knowledge of the neurobiology of the brain, psychological and neurobiological evidence of multimodal perception, and theories on how the human brain creates a unified conscious perceptual experience, known as the binding problem. Moreover, it is important to review the computational models of multisensory fusion proposed thus far to understand their strengths and limitations. This chapter provides a detailed introduction to the above subject areas while critically reviewing existing computational models.

The chapter starts with a brief introduction to the neurobiology of the brain, giving an overview of different cortical areas and their functionalities. Then the chapter reviews various academic literature related to the evidence of multimodal sensory perception from two aspects, 1) the psychological evidence and models of multimodal perception, 2) neuro-biological studies of multimodal perception and their findings. The behavioural and psychological studies have demonstrated fascinating crossmodal effects between sensory modalities in the form of 1)

crossmodal influence where the sensation in one modality influence the perception of a co-occurring sensation of another modality, 2) crossmodal recalibrations when artificially induced discrepancies in a single modality lead to the alteration of correspondence between the modalities and, 3) medical conditions demonstrating the crossmodal effect between sensory modalities. On the other hand, the neuro-biological studies discussed in the chapter include experiments carried out on non-human subjects to assess the multisensory integration in terms of the effectiveness of the crossmodal stimulus with respect to the strength of individual stimuli. Moreover, more recent neuroimaging studies that use techniques such as functional magnetic resonance imaging (fMRI) to study multisensory nature of different cortical areas of the brain are discussed.

Once the evidence of multimodal perception in humans has been established, the chapter discusses various theories on how humans perceive the information captured from different sensory modalities as a coherent event/object, known as the binding problem. Here we discuss theories such as feature integration theory, synchronisation theory and the theories on the role of attention in perceptual binding.

Then, the chapter discusses self-organization, which has long been viewed as a central mechanism of nature. Self-organization has been hypothesised as the mechanism by which the feature maps of the brain responsible for processing sensory modalities are organized. The chapter discusses a range of computation algorithms that uses self-organization as the central processing mechanism as possible candidates for implementing artificial multimodal processing.

Finally, the chapter critically reviews the computational models of multisensory fusion found in the literature. These computational models are broadly organized under biologically inspired models and models that view multisensory fusion as Bayesian inference.

2.1 Neurobiology of the Human Brain

The human brain is the most vital organ of the central nervous system, which facilitate perception, cognition, consciousness, reasoning, language understanding, motor control etc. A fundamental component of the brain that facilitate the above "higher-order" functions is the *neocortex*, the most recent part of the brain to be evolved. The neocortex is composed of six neuronal layers and forms the major part of the cerebral cortex, the outermost layer of neural tissue of the brain.

The *cortical areas* of the neocortex have specific functions. These include sensory, which is responsible for receiving and processing sensory information, association, which is responsible for combining multiple sensory stimuli to a meaningful perceptual experience of the world, abstract thinking and language functionalities, and motor, which is responsible for the control of voluntary movements (Yeo et al., 2011). The cortical areas that receive and process sensory information are further localized as the primary visual cortex, primary auditory cortex and primary somatosensory cortex which processes visual, auditory and touch sensations initiated at the corresponding organs. The primary cortical areas are organized as topographic maps which preserve topological relationships from the sensing areas onto the primary cortical areas (Goodhill & Xu, 2005). Moreover, the organization of these cortical areas is highly influenced by the exposure to the stimuli at a young age, highlighting the neuronal plasticity. This has been demonstrated with experiments that deprive such stimuli in kittens (Wiesel & Hubel, 1963).

2.2 Evidence of Multimodal Perception

Humans, like many other organisms, possess multiple sensory systems. These systems that capture different aspects of the environment offers richer information of the surrounding due to the fact that they jointly capture the same event or object in multiple modalities supplementing each other. A number of research disciplines studying this phenomenon have identified many cases of interactions between modalities, where the perception of one sensory modality is conditioned by the information simultaneously available to another. This section describes

research work examining this phenomenon in the field of behavioural and psychological studies as well as in the field of neurobiology, enabled by brain-imaging techniques such as functional magnetic resonance imaging (fMRI).

2.2.1 The Psychology of Multimodal Perception

A large body of research in behavioural and psychological fields has demonstrated the crossmodal effect between sensory modalities. These effects include crossmodal influences where the sensation in one modality influence the perception of a co-occurring sensation of another modality, crossmodal recalibrations when artificially induced discrepancies in a single modality lead to the alteration of correspondence between the modalities and in some cases medical conditions demonstrating the crossmodal effect between sensory modalities (see (Bertelson & De Gelder, 2004) for a comprehensive survey of literature).

2.2.1.1 Crossmodal Influence

A classic example of crossmodal influence is the *ventriloquism effect* (Howard & Templeton, 1966) where the audience experiences the voice as coming from the dummy when the performing ventriloquist moves the lips, eyes, and head of the dummy in synchrony with the voice produced by him. The observers are involuntarily tricked by their brain to perceive the dummy is speaking, demonstrating the intersensory bias of vision on audition in generating an impression on the point of origin.

Among a large number of early behavioural studies that evaluate the online reactions to the exposure of spatially conflicting multimodal inputs, Bertelson & De Gelder (2004) identify two main effects that have been studied; namely the spatial fusion of conflicting inputs and immediate crossmodal bias of spatial perception. The spatial fusion studies are predominantly based on the participants judging the origins of the conflicting inputs as same or different (for example: (Choe et al., 1975; Jack & Thurlow, 1973)). The immediate visual bias of proprioception has been demonstrated with experiments where the participant has to point with

one of his hand which is hidden under a cover towards the felt location of the other hand while the visual of the second hand is being displaced with the use of a prism (Hay et al., 1965).

The crossmodal influence between audition and vision has been studied in behavioural experiments extensively. A popular scenario that has been studied is the interaction between auditory and visual speech recognition. It is generally understood that the vision of the speaker's face, especially the lips, greatly helps in the understanding what is being said. This has been demonstrated experimentally, emphasising on how the visuals enable the listener "to hear better and hence to understand better" (Schwartz et al., 2004, p. B69). The positive effect of vision is greatly highlighted in instances where there is significant background noise, which is commonly referred to as the 'cocktail party' situation (Arons, 1992).

A seminal research paper by Mcgurk & Macdonald (1976), aptly named "Hearing lips and seeing voices", on visual influence on auditory describes what is today known as the McGurk effect. They identified two cases of influence; one where the incompatible visual and auditory cues being transformed into something new, different from both original visual and auditory cues ("fusion"), and the other where a composite comprising of unmodified elements from the two modalities ("combination"). For the experimentation, they mixed up incompatible audio and visual of lip movements of a woman uttering [ba-ba], [ga-ga], [pa-pa], [ka-ka] creating audio/video pairs of 1) [ba-ba] voice/[ga-ga] lips, 2) [ga-ga] voice/[ba-ba] lips, 3) [pa-pa] voice/[ka-ka] lips, and 4) [ka-ka] voice/[pa-pa] lips. The most spectacular case with very high support from the experiments was the "fusion" of [ba-ba] voice with [ga-ga] lips leading to hearing [da-da]. Similarly, about 2/3 of the participants reported hearing [ta-ta] when the audio of [pa-pa] was combined with visuals of [ka-ka]. They highlight how the auditory-based theories of speech perception fall short of completely explaining this phenomenon and the important role of vision in the perception of speech.

2.2.1.2 Crossmodal Recalibration

Another class of examples of crossmodal perception demonstrates the recalibration effect of sensory modalities due to artificially induced discrepancies in one of the modalities. The early

experiments involved wearing optical devices that displaced the retinal image, which leads to a discrepancy between the vision and proprioception. While this initially led the subjects to miscalculate the locations of objects, they quickly got adapted to the discrepancy allowing them to grasp objects despite optical distortion (Harris, 1965). Moreover, when the optical devices were taken off, the subjects were still adjusting for a while, making them miss the objects. Similarly, the recalibration effect has been demonstrated for both visual and auditory locations using concurrent light flashes and sound bursts originating at slightly different locations (Radeau & Bertelson, 1974, 1977). For example, in (Radeau & Bertelson, 1974) the subjects were exposed to concurrent light flashes and sound bursts with light emitted at an angle of 15^{o} on to the right relative to the sound, resulting in both post-exposure sound location and light location being shifted to left and right respectively. Based on similar perceptual adaptation experiments, Wallach (1968) presented a general view of *information discrepancy* as the basis for such perceptual adaptation.

2.2.1.3 Evidence from Medical Cases

A number of specialised medical conditions have been identified as a great source of evidence for multimodal dynamics in the human brain.

Synaesthesia is such an interesting medical condition where a stimulus in one modality induces sensation in another (Ramachandran & Hubbard, 2001). For example, seeing a particular number may always induce experiencing a specific colour (known as Grapheme-colour synaesthesia) or listening to everyday sounds/musical notes may trigger seeing colours (known as Chromesthesia). While the colour associated with the particular number or sound/note may differ from patient to patient, it is constant over time for a given patient. Other forms of synaesthesia include auditory-tactile synaesthesia where certain sounds induce touch sensations on the body, lexical-gustatory synaesthesia where hearing certain words induce particular unrelated tastes among others. Synaesthesia is automatic and does not require the attention of

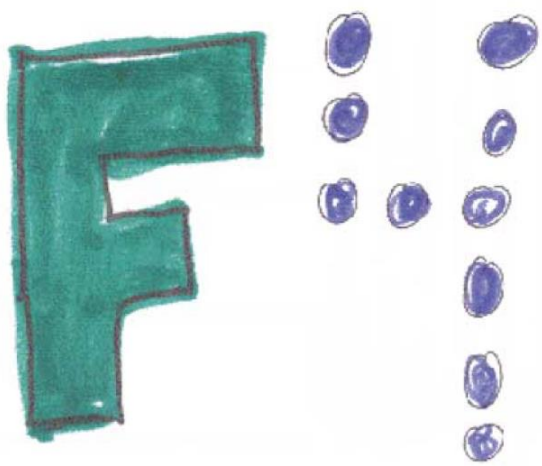

Figure 2.1 Synesthetic colouring. A synesthete placed the colours exactly as she perceived them after the shapes (black outline) drawn by the experimenter (Robertson, 2003).

the patient (Robertson, 2003). Moreover, the condition is persistent throughout life. Figure 2.1 contains a colouring done by a synesthete who were perceiving colour when presented with shapes.

Synaesthesia represents a case of abnormal feature binding. It has been widely hypothesized as being due to direct connections between cortical feature maps that had not been properly pruned during development (Baron-Cohen & Harrison, 1997). This hypothesis has been further supported by functional imaging studies of the brain where the activations are noted in the ventral pathway that registers sound, shape and colour as well as in parietal lobe where sensory information among various modalities are integrated (Nunn et al., 2002). Moreover, this is consistent with behavioural studies on synaesthesia, which suggest the synesthetic binding occurs before attention (Robertson, 2003).

Other medical cases include patients who have lost the ability to consciously perceive a particular modality due to brain damage, who nevertheless process the stimuli, which ultimately bias the perception of other modalities. Studying such patient cases provides a wealth of evidence on the crossmodal effect of perception and allows the researchers to examine such effects in great details.

Prosopagnosia is one such cognitive disorder of face perception where the patients are unable to recognize familiar faces, including their own, even though other visual tasks, such as object

discrimination, are unaffected. De Gelder et al. (2000) present a case study of a patient suffering from Prosopagnosia who "shows a complete loss of processing facial expressions in recognition as well as in matching tasks" (p. 425) along with her inability to recognize familiar faces. The researchers studied crossmodal bias of vision on audition by presenting incongruent visual and auditory stimuli such as a happy face with a fearful tone of voice and asking her to identify the emotional tone. The results show that her identification is biased by the face shown, even though she is unable to identify the emotion on the face consciously.

Bertelson et al. (2000) have studied the visual bias of auditory location with a patient having visual unilateral neglect, who is unable to detect any visual light flashes present in his left visual field. In the experiment (see Figure 2.2), when the patient was asked to point to the location of sound which was emitted centrally in synchrony with light flashes presented in the patient's visual left hemifield, his perception of the location of the sound was strongly biased towards the light flashes the patient was not consciously seeing. Bertelson and De Gelder (2004) highlights the importance of this result, demonstrating the automaticity of visual bias in auditory localisation, where it biases even without the awareness of its occurrence of the visual stimuli.

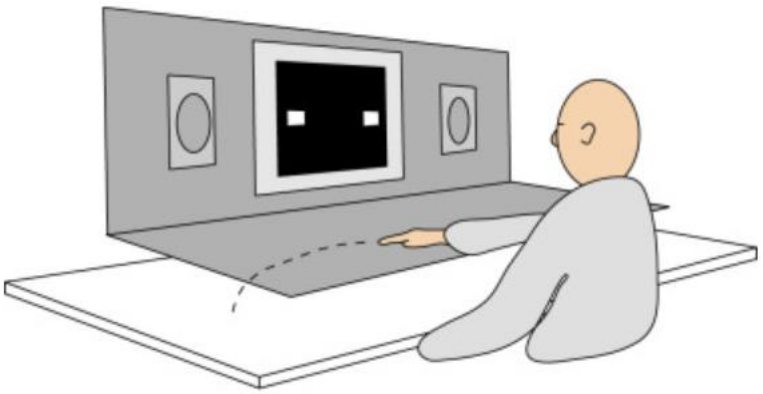

Figure 2.2: The experimental setup for examining the visual bias of auditory location. The patient points to the location of sound that was emitted centrally in synchrony with light flashes presented in left/right/both sides (De Gelder & Bertelson, 2003).

2.2.2 Neuro-biological Evidence of Multimodal Perception

Before the advent of neuroimaging techniques, the neuro-biological studies into the multisensory integration of the brain was limited to the experiments carried out on non-human subjects such as cat and primates. In these experiments, the multisensory integration is assessed by the effectiveness of the crossmodal stimulus with respect to the strength of individual stimuli. Stein and Stanford (2008) define the multisensory integration at the level of single neuron as "statistically significant difference between the number of impulses evoked by a crossmodal combination of stimuli and the number evoked by the most effective of these stimuli individually" (p. 255).

The experimental research into the multisensory phenomenon at the single neuronal level has identified a number of important characteristics of the multisensory interplay. As depicted in Figure 2.3, *multisensory enhancement* (or *depression*) of the integrated stimulus in the case of congruent (or incongruent) individual stimuli demonstrate response which is greater (lesser) than individual stimuli (Calvert et al., 2004; Gillmeister & Eimer, 2007). Of great interest is the case of superadditivity where the combined response is much greater than the summation of the individual response highlighting how the two senses combine to invoke a response from motor mechanism to generate a reaction. Moreover, it has been demonstrated that the multisensory integration can shorten the time between sensory reception the motor command generation (Bell et al., 2005).

Another important characteristic of the multimodal neurons is the *inverse effectiveness* (Meredith & Stein, 1986) of their combination, i.e. multisensory enhancement is inversely proportional to the strength of individual stimuli. This principle is important for effectively sensing the environment. The individual stimuli that are salient will anyway be detected, and the inverse effectiveness is the mechanism that generates sizable neuronal responses to weak multimodal cues. Moreover, the magnitude of the integrated response is affected by the time overlap of the individual stimuli. Even though the time disparities up to several hundreds of

milliseconds are tolerated, the combined response is maximized when the individual stimuli coincide.

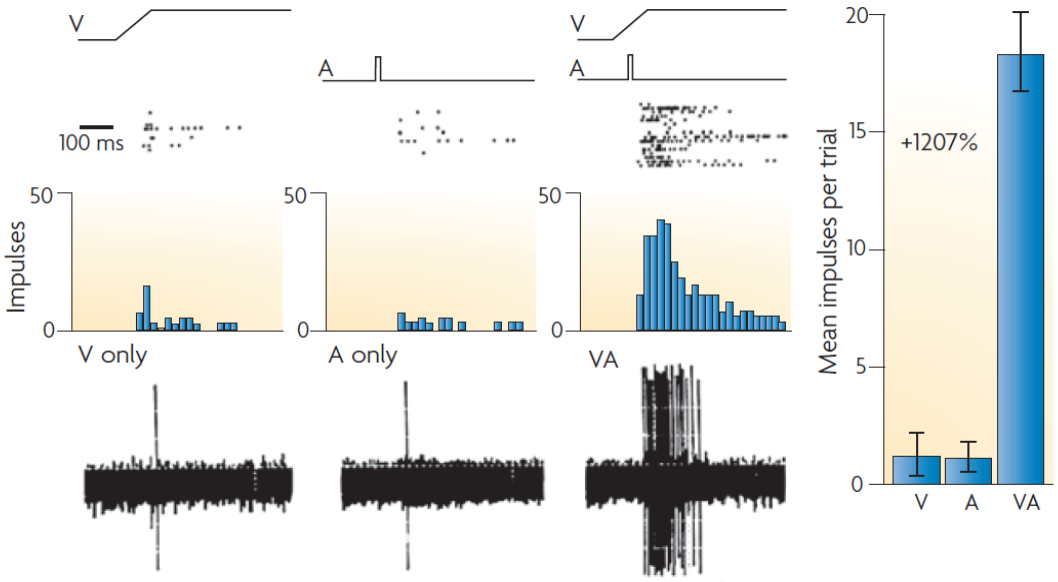

Figure 2.3: Multisensory enhancement. The neuronal response of a single multisensory neuron in superior colliculus to visual (V), to auditory (A) and combined (VA) stimuli (Stein & Stanford, 2008).

The experiments carried out with cats and primates have identified a number of regions that have multisensory neurons in abundance. The cat superior colliculus contains multisensory neuron having multiple receptive fields for each sensory modality (Stanford et al., 2005; M. T. Wallace & Stein, 1997). These receptive fields are in spatial overlap among each other, allowing for the multimodal stimuli arising from the same/close by physical location to be integrated. Similarly, the experiments on primates have focused on posterior parietal cortex where cues from multiple sensory modalities such as visual, auditory and tactile converge.

2.2.2.1 Evidence from Neuroimaging

With the advent of neuroimaging techniques such as functional magnetic resonance imaging (fMRI), the study of neuronal activation at the single neuron level, which was earlier restricted to non-human subjects such as primates and cats, has now been extended to study neuronal activations in the human brain (James & Stevenson, 2012). Insights from ground-breaking studies using such technology have identified various multisensory regions of the human brain

and broaden our understanding of the multimodal dynamics at the single neuronal level (Stein & Stanford, 2008). The fMRI uses the blood oxygenation level dependent (BOLD) contrast as the primary measure for neuronal activation as the cerebral blood flow and neuronal activation are closely linked. However, James and Stevenson (2012) point out the fundamental difference between what is measured by BOLD activation and single neuron activity as the BOLD activation is measured from vasculature supplying a population of neurons opposed to an individual neuron.

One of the first phenomena of multimodal dynamics of the human brain to be studied using fMRI was the superadditivity. Calvert et al. (2000), in their fMRI-based study, presented the human subjects with semantically congruent and incongruent audio-visual speech and to each modality in isolation. By analysing the BOLD activation, they identified an area in the left superior temporal sulcus that demonstrated superadditive response to congruent stimuli and subadditive response to incongruent stimuli. The use of stronger superadditive criterion, where multisensory response needs to be greater than the sum of unisensory response, compared to maximum criterion, where multisensory response needs to be greater than the maximum of unisensory response (Beauchamp, 2005), ensures that the region contains multisensory neurons and the BOLD activation is not due to individual activations from unisensory neurons.

Similar fMRI studies have been carried out to investigate the integration of auditory and visual information at the human superior temporal sulcus area (Beauchamp et al., 2004, 2010; Hertz & Amedi, 2010). Based on previous invasive experiments on macaque monkeys, superior temporal sulcus area contains neuron sensitive only to auditory stimuli, only to visual stimuli, or both to auditory and to visual stimuli. In their fMRI study, Beauchamp et al. (2004) identified similar patches of voxels in human superior temporal sulcus where 44% of voxels responded more to unimodal auditory than unimodal visual stimuli, 30% of voxels that responded vice versa and 26% of voxels were multisensory patches that responded equally to unimodal auditory and visual stimuli (see Figure 2.4). These regions are identified as containing

individual neurons that are receiving primarily auditory, primary visual inputs and multisensory auditory-visual neurons, respectively.

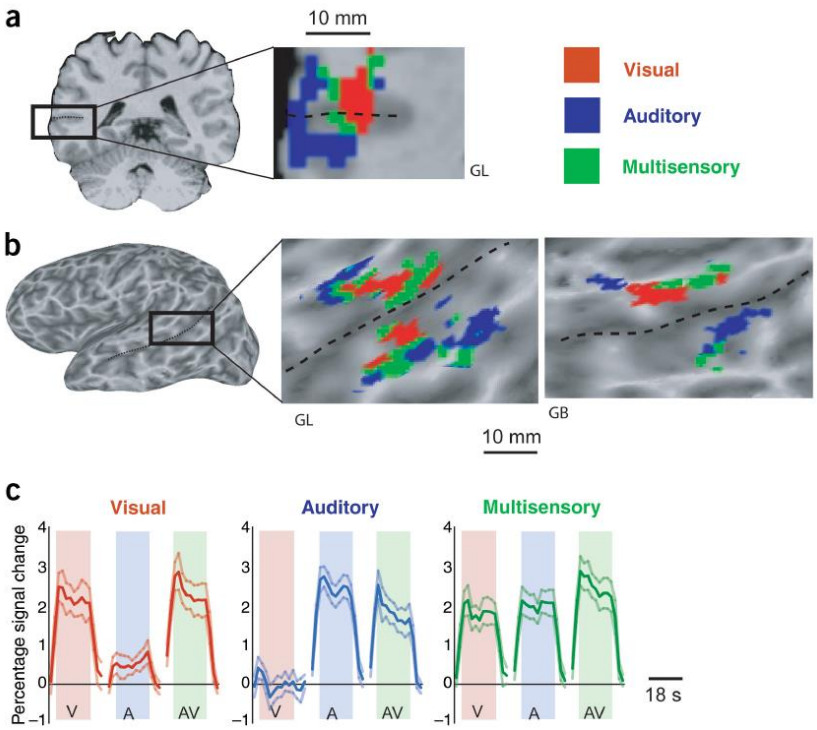

Figure 2.4: Patches of voxels in the human superior temporal sulcus. Coronal section (a) and lateral view of the left hemisphere (b) with enlargements showing of the superior temporal sulcus with visual patches (orange), auditory patches (blue) and multisensory patches (green). GL and GB are the identities of the two subjects. (c) Response in visual, auditory and multisensory patches to three stimulus types (pink shade – visual, blue shade – auditory, green shade - multisensory). (Beauchamp et al., 2004)

Hertz and Amedi (2010) using multifrequency fMRI spectral analysis identified audio-visual areal convergence in superior temporal sulcus supporting bottom-up processing of audio-visual signals where single modalities are processed in primary areas, and then multiple sensory streams are converged outside primary areas. Interestingly, they found a weak level of convergence of audio-visual signals in primary areas as well. The fMRI studies using audio-visual stimuli have also been extended to study the McGurk effect, a prominent example of auditory-visual multisensory influence. It has been demonstrated that by disrupting the multisensory areas of the superior temporal sulcus with single-pulse transcranial magnetic

stimulation while the subject is exposed to the incongruent audio-visual stimuli, the likelihood of the McGurk percept is significantly reduced (Beauchamp et al., 2010).

The fMRI-based studies have been utilised to study cortical areas of multimodal integration of different multimodal stimuli. Foxe et al. (2002) demonstrated how auditory and somatosensory inputs converge in the superior temporal gyrus, a subregion of the human auditory cortex. Further, they observed superadditivity of sensory signals demonstrating the multimodal integration in the convergence. Moreover, this finding suggests multisensory integration earlier in the cortical processing hierarchy than previously anticipated. Similarly, Gentile et al. (2010) identified regions of the brain that demonstrated nonlinear, superadditive responses to visual-tactile stimuli, including left anterior intraparietal sulcus, the insula and dorsal premotor cortex.

2.3 Binding Problem

The binding problem refers to how the human brain processes sensations from multiple stimuli generated by our surrounding to create a united conscious perceptual experience (Revonsuo & Newman, 1999). These multiple stimuli could be multimodal as well as relating to multiple constituents of the same modality. Binding problem, on the one hand, can be viewed as a segregation problem, addressing the questions 'How does the brain segregate different features relating to different objects from the sensory input and correctly assign them to distinct objects?' (Smythies, 1994). Contrary to the name, this view of the binding problem is more of discrimination of the stimuli. On the other hand, the binding problem can be viewed as a combination problem which explains the mechanisms by which physically separated neural signals processed at disparate processing areas of the brain are combined on to a single perception (Goldstein, 2009). Figure 2.5 highlights how the binding allows us to experience a stimulus such as a rolling ball as a coherent percept, while the location, form, depth, motion and colour activate neurons at various locations of the brain, as opposed to separated location, form, depth, motion and colour perception.

There has been a number of theories put forward to answer the segregation aspect of the binding problem both at psychological and physiological levels. Following are some of the theories in brief.

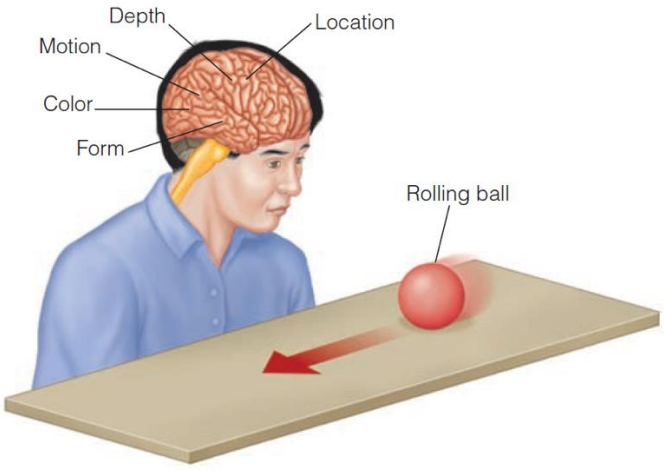

Figure 2.5: Binding allows us to perceive as a single experience the activations generated by a single stimulus at various locations of the brain (Goldstein, 2009)

2.3.1 Feature Integration Theory and Role of Attention

The feature integration theory (Treisman & Gelade, 1980) proposes that the object's location mediates the binding of the features such as the form, depth, motion and colour. The theory defines perception as a two-stage process. The first stage, *preattentive stage*, rapidly, parallelly and automatically register features such as form, depth, motion and colour without the explicit attention. The second state, *focused attention stage*, is responsible for the unified perception of an object where it combines individual features and is based upon the explicit attention to the location of the object registered in the "master map" of locations.

This is further explained with respect to the *what* and *where* streams of the cortex. The attention is the "glue" that combines the form and colour information of the *what* stream with the location and motion information of the *where* stream. The experiments described by Treisman and Schmidt (1982) provide experimental evidence for early registration of features in the *preattentive stage* where the subjects make *illusory conjunction*, combining features from two images flashed at a quick succession on the same location (see Figure 2.6).

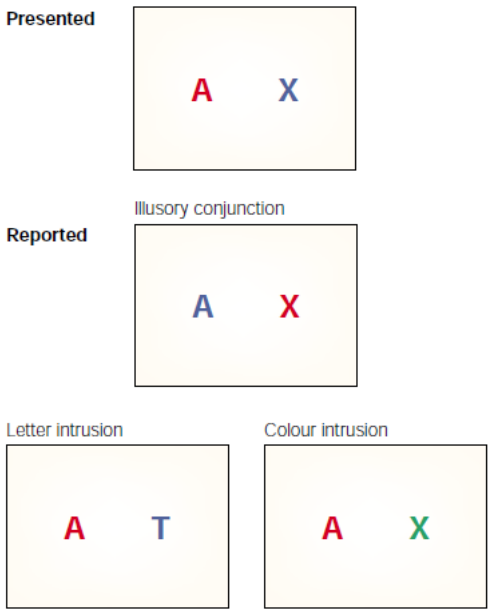

Figure 2.6: Illusory conjunctions. Flashing two images at a quick succession on the same location may lead to recombining features such as colour and shape. Letter intrusion and colour intrusion may be introduced to correct the results for guessing as intrusion are very rare. (Robertson, 2003).

Further support for feature integration theory is provided by the patients having Bálint's syndrome, who are unable to focus attention on individual objects due to parietal lobe damage. During the experiments, the patients having Bálint's syndrome often reported colours and letters mixed up such as ‘red T’ when they were presented with a ‘red O’ and a ‘blue T’ even for relatively long durations (Friedman-Hill et al., 1995; Robertson et al., 1997). Another implication of the two-stage process of the perception is the difference between time and attention required to perform a *feature search* and a *conjunction search* (Treisman, 1982; Treisman & Gelade, 1980). In a *feature search*, the search is performed based on a single characteristic such as colour, shape, movement or orientation and can be performed much faster than a *conjunction search* where the search needs to be performed based on two or more characteristics (see Figure 2.7). The conjunction searches require more effort with conscious attention and performed serially in the *focused attention stage*.

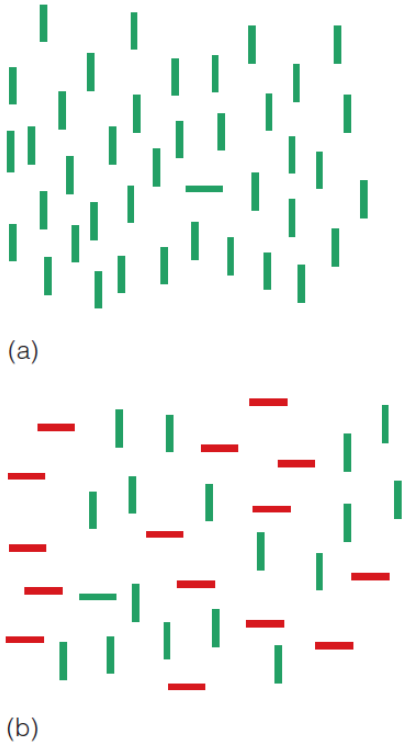

Figure 2.7: Feature search versus conjunction search. Try finding the green horizontal bar in section (a) and (b). The task is fast and effortless in (a) since it is a feature search easily discriminated by the orientation at the preattentive stage. Much time and concision attention at each location are required for (b) as it a conjunction search requiring discrimination by multiple features, colour and orientation in this case (Goldstein, 2009).

Nevertheless, the feature integration theory has been criticized for its vague presentation of the concept of attention (Tsal, 1989; Di Lollo, 2012). Attention is generally characterized as a limited resource for choosing a particular aspect of the visual field that is of interest or relevant at the moment. Di Lollo (2012) notes that even though attention has been described with various metaphors such as a spotlight, a filter, a zoom lens and more commonly glue that binds various stimuli relating to the same object, these descriptions lack explanations on any specific mechanism mediating the binding.

2.3.2 Synchronization Theory

A popular hypothesis on the underlying mechanism of binding of feature binding is identified as the synchronization theory (Engel et al., 1999; Varela, 1995). The synchronization theory suggests that the neuronal activation in various parts of the brain induced by the same object

are in synchrony and this synchronization is the basis of binding which leads to the perception of these objects, opposed to individual features (see Figure 2.8).

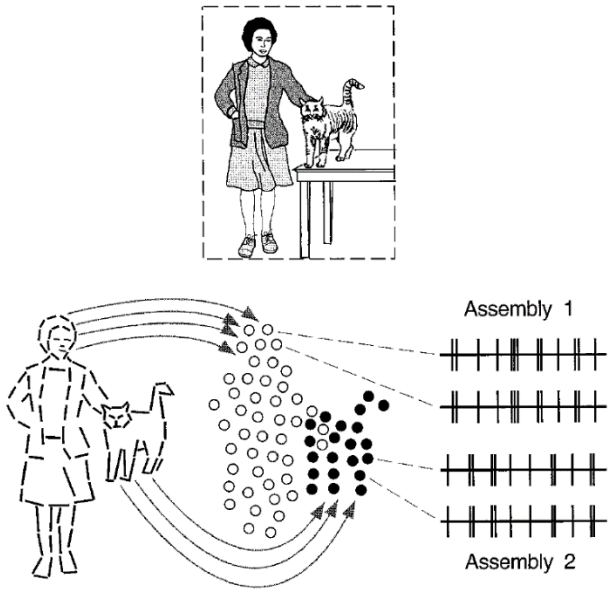

Figure 2.8: Coherent perception by synchrony in neuronal activity. The neuronal activities of neurons that get activated to various features of the same objects are synchronized in their firing (Engel et al., 1999).

The early results from the empirical studies of Gamma wave activity in the brain provided the impetus for this hypothesis (Gray et al., 1989). However, more recent experimental results have questioned the role of synchrony in binding. For instance, Dong et al. (2008) reported that the neurons activating for the contours of the same or different shapes, having no effect on the temporal synchrony. Moreover, Thiele and Stoner (2003) reported how the synchronization theory was not supported by examining the neuronal responses to moving plaid patterns.

2.4 Self-organization, a Biologically Central Process

Self-organization has long been viewed as a central mechanism of nature that organizes selected parts of a system so as to promote a specific function (Camazine et al., 2003). Isaeva (2012) defines self-organization as the "emergence of spatio-temporal order, during which the global pattern of systems is formed by local interactions of its elements" (p. 110). Contrary to external organization, where the system organization imposed by external factors, the self-organization facilitates the evolution of the system into an organized form in the absence of external

constraints. Another major characteristic of self-organized systems is the global order which emerges from the local interactions. Isaeva (2012) highlights how this is central to the formation of complex systems such as biological system where the whole system does not demonstrate the characteristics of its constituents, rather demonstrates a new arising, or emergent features. Self-organization is commonplace in biological systems (Camazine et al., 2003; Kelso, 1997; Kohonen, 1989). For example, in gene expression, connectivity graphs and the systems of proteins with autocatalyzing properties gain collective order by self-organization (Isaeva, 2012).

Self-organization has been hypothesised as the mechanism by which the features maps of the brain responsible for processing sensory modalities are developed. Undoubtedly, the main structure of the brain, which has been evolved over a long period of time, is fixed at the time of birth for a given person. However, experimental evidence suggests that sensory projections are conditioned by experience. This is due to the plasticity of the neurons and the self-organization mechanism that drives the conditioning. It has been demonstrated that after depriving sensory experience at a young age or after surgical removal of brain tissue (Berman & Hunt, 1975; Chu et al., 2000), some projections are not developed, and the remaining projections move on to occupy the corresponding area of the brain.

The self-organization process results in developing features maps in all sensory systems of the brain. These are known as topographic maps as they preserve topological relationships from the input signal onto the cortical areas (Goodhill & Xu, 2005). For example, in the human auditory system, tonotopic maps, which spatially maps sound frequencies in an orderly fashion onto regions of the cortex, are developed under the control of received information (Humphries et al., 2010; Petkov et al., 2006; Talavage et al., 2000). Similarly, in the human visual system, the ganglion cells of the retina are mapped to the lateral geniculate nucleus (LGN) in an orderly progression and to the primary visual cortex (V1) from there onwards in an orderly fashion (Lemke & Reber, 2005). Known as retinotopic maps, adjacent areas of the retina are represented by adjacent neuron in LGN and V1.

The elegance of self-organization process and the capability of topographic maps to produce a *reduced representation* of facts without loss of knowledge about their interrelationships (Kohonen, 1989) have inspired a range of computation algorithms for information processing and knowledge representation. These algorithms develop simplified models of the world at a level of abstraction in relations to the inputs from the observable world. Below we describe several such computation algorithms inspired by the self-organization process and *reduced representation* mechanism of the brain.

2.4.1 Adaptive Resonance Theory

Adaptive resonance theory (ART) (Grossberg, 1982, 2013) is a cognitive and neural theory on how the human brain learns to organize events and objects observed on a continuing basis. The unsupervised computational model built on top of this theory (Carpenter & Grossberg, 1987a, 1987b) places high emphasis on tackling the stability-plasticity dilemma. The model strives to achieve stability without rigidity and plasticity without chaos while continuing to perform learning and to preserve the learned patterns. The model exhibits self-organizing and self-stabilizing characteristics in its recognition prototypes once trained by the unsupervised competitive learning algorithm.

A central idea of ART is that the object categorization is based on the interaction between the sensory information that flows bottom-up and the "expectations" that flows top-down in the forms of memory templates or prototypes. In the computation model, this is implemented with two artificial neural layers known as comparison layer, F_1, and recognition layer, F_2. The comparison layer initially receives the input, $x \in \mathbb{R}^D$, which is passed on to the recognition field for selection of a winner, j^*, using (2.1), by multiplying with the forward weight matrix, $B = [b_{ij}]$. The winner selection is based on competitive learning, a form of unsupervised learning. This process represents the bottom-up flow of sensory stimuli.

$$j^* = \arg max_j \sum_i b_{ij} x_i \tag{2.1}$$

The top-down expectation flow or feedback from the recognition layer to the comparison layer is calculated using the feedback weight matrix, $T = [t_{ji}]$. This produces a prototype pattern on the comparison layer and reset module assesses the recognition match to the vigilance parameter, which controls the granularity of the recognition layer. If the comparison indicates a close enough match, training is carried out by adjusting b_{*j}, the weight vector of B corresponding to the winning neuron. Otherwise, the winning neuron is inhibited, and the search for a new winner is carried out in iterations until the vigilance criterion is met.

2.4.2 Self-Organizing Map Algorithm

The self-organizing map (SOM) algorithm (Kohonen, 1990), also known as self-organizing feature map, is a computational algorithm inspired by the characteristics and processing mechanisms of the human brain. The SOM structure consists of a set of neurons (also known as nodes or units) that resembles the structure of a cortical layer of the human brain, and the neurons are assembled in a lattice of usually two dimensions. The algorithm produces a discretised representation of the input space onto a lower-dimensional space, usually two-dimensional, facilitating dimensionality reduction while producing the reduced representation. The algorithm uses an unsupervised learning algorithm; in particular, competitive learning where the neurons compete for the incoming input data. Moreover, the training algorithm preserves the topological relationships from the input domain onto the lower-dimensional space inspired by the topographic maps in the human sensory system.

The SOM is arranged as a lattice of neurons each having a weight vector, $w_k(t) \in \mathbb{R}^D$, at iteration t, representing the input space and x, y coordinates in the lattice representing the output space. To train the SOM, each input vector, $x_i \in \mathbb{R}^D$, is presented to the SOM and the best matching unit (BMU), k^* is found using (2.2) based on the distance between the input and the weight vectors of the neurons using a suitable distance metric.

$$k^* = \arg min_k \ (\|w_k - x_i\|) \tag{2.2}$$

The weight vectors of the BMU and its neighbours are updated using (2.3), resulting in the weight vectors of the BMU and its neighbouring neurons moving towards the presented input.

$$w_k(t+1) = w_k(t) + \alpha(t)h_{BMU,k}(t)[x_i - w_k(t)] \quad (2.3)$$

Here, $w_k(t+1)$ is the updated weight vector of the k[th] neuron while $w_k(t)$ is the previous weight vector of the same neuron. Parameter α is a time-decreasing learning rate, $h_{BMU,k}(t)$ is a neighbourhood function which decreases the size of the neighbourhood of weight adjustment over time and x_i is the input presented. The use of neighbourhood function is responsible for the careful preservation of the topology, and the Gaussian function is a common choice for the neighbourhood function. Usually, the input presentation and weight update are carried out for a pre-defined number of iterations.

The main limitation of the SOM algorithm is its fixed network size and shape. Usually, the number of neurons and the arrangement of the neurons (the width and height of the map in case of two-dimensional SOM) need to be defined prior to the learning phase. However, any information that may be useful in defining the shape and size of the grid may not be available at the time of learning as SOM is widely used for exploratory tasks where the user has no or very little knowledge about the underlying structures of the input. Unsuitable map size and shape may lead to under-representation of certain areas of the input domain. Moreover, the rigid size and shape do not represent the dynamic structure adaptation of the cortical areas it is inspired by. There are several alternatives/extensions to the SOM algorithm proposed to overcome this limitation, which includes the growing self-organizing map (Alahakoon et al., 2000), growing neural gas (Fritzke, 1994), growing when required network (Marsland et al., 2002) and growing hierarchical self-organizing map (Dittenbach et al., 2000).

2.4.3 Growing Self-Organizing Map Algorithm

As a dynamic variant of the SOM algorithm, the growing self-organizing map (GSOM) algorithm (Alahakoon et al., 2000) addresses the limitation of fixed size and shape of the SOM map. The algorithm initializes the topographic map with just four neurons, and iteratively adds

neurons to the map during the training phase to form a better representation of the inputs space. Hence, the GSOM algorithm is able to identify the ideal map structure automatically without the user having to specify the same arbitrarily. The ability to grow the grid shape and size according to the input data results in GSOMs requiring a lesser number of nodes for representing a data set to a comparable SOM. Moreover, as the algorithm is initialized with a minimal number of neurons and neurons being added only when required, the algorithm has also been demonstrated to be faster than the standard SOM algorithm (Fonseka et al., 2011; Ganegedara & Alahakoon, 2011; Hsu & Halgamuge, 2003).

Due to the favourable characteristics of GSOM, it has been successfully applied in numerous research endeavours ranging from basic science (Hsu et al., 2003; Gunasinghe et al., 2014; Chan et al., 2008), engineering (De Silva et al., 2011; Guru et al., 2005) to e-commerce (Hsu et al., 2009; Nathawitharana et al., 2015), and adapted for data streams (Nallaperuma et al., 2017, 2018), text mining (Matharage et al., 2013), taxonomic classification (Weber et al., 2011), anomaly detection (Ippoliti & Zhou, 2012) and sequence mining (Gunasinghe & Alahakoon, 2013).

The GSOM algorithm consists of two phases, a growing phase which adjusts the structure of the map to represent the input data and a smoothing phase to fine-tune the weights of the neurons. In the growing phase, the GSOM network is initialized with four neurons arranged in a 2×2 lattice, each having a weight vector, $w_k(t) \in \mathbb{R}^D$. To train the GSOM, each input vector, $x_i \in \mathbb{R}^D$, from the dataset, S is presented to the map over a number of iterations and the winning neuron, known as the best matching unit (BMU), is identified similar to the SOM algorithm (see (2.2)). The distance between the weight vector and the BMU is called the quantization error $E(t)$, given by (2.4), is accumulated on the BMU.

$$E(t) = \|w_{BMU}(t) - x_i\| \tag{2.4}$$

If the accumulated quantization error of the neuron k, $AE_k(t)$ is greater than the growth threshold GT, as defined in (2.5), that neuron is said to under-represent the input space it represents, and if the neuron is on the boundary, the map is grown from the boundary by adding new neurons to the map. Otherwise, the error is spread among neighbouring neurons.

GT is determined by the number of dimensions D, and spread factor SF, which allows controlling the growth of the GSOM and is independent of the dimensionality of the dataset.

$$GT = -D \times \ln(SF) \tag{2.5}$$

A boundary node is a node which has at least one of its immediate neighbouring positions vacant. Since the GSOM algorithm utilizes a square grid structure of neuron positioning one to three positions of a boundary node can be vacant. When the neurons are added to the grid, all the vacant neighbouring positions are filled with new neurons and the weight vectors of them are initialized to match the weight vectors of its neighbours to maintain a smooth weight surface. We can identify three cases of weight initialization based on the location of the new node being added, as illustrated in Figure 2.9.

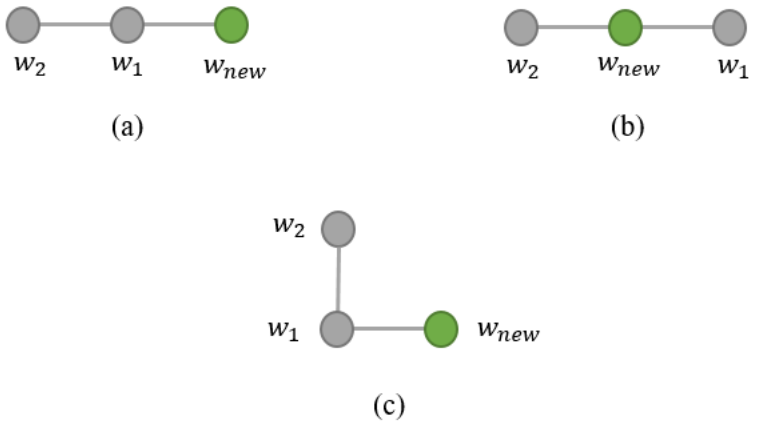

Figure 2.9: Weight initialization for a newly added node

With the weight vector of the node that initiated node growth, the weight vector of the neighbouring node and the weight vector of the newly added denoted as w_1, w_2 and w_{new} respectively, the weight initialization for the new node is carried out as follows.

Case (a): The new node has two consecutive old nodes on one of its sides.

$$w_{new} = \begin{cases} w_1 - (w_2 - w_1); & w_2 > w_1 \\ w_1 + (w_1 - w_2); & w_2 \leq w_1 \end{cases} \quad (2.6)$$

Case (b): The new node is in the middle of two older nodes.

$$w_{new} = \frac{w_1 + w_2}{2} \quad (2.7)$$

Case (c): The new node has only one older neighbour, and this neighbour has another node on its side which is not directly opposite to the new node.

$$w_{new} = \begin{cases} w_1 - (w_2 - w_1); & w_2 > w_1 \\ w_1 + (w_1 - w_2); & w_2 \leq w_1 \end{cases} \quad (2.8)$$

If both cases (a) and (c) are valid for a given node insertion, case (a) take precedence in weight initialization.

For a non-boundary node, k, the error is redistributed to its immediate neighbours when the accumulated quantization error, $AE_k(t)$, exceeds the growth threshold, GT. The effect of the error redistribution is to pass the high error value in the middle of the map towards the boundary leading to node addition along the boundary. With error redistribution, the accumulated quantization error of the node, k, that exceeds the growth threshold is set as follows.

$$AE_k(t+1) = GT/2 \quad (2.9)$$

The weight vectors of the immediate neighbour, $n_i; i = 1 \ldots 4$, are updated as follows.

$$AE_{n_i}(t+1) = (1+\gamma)\, AE_{n_i}(t) \quad (2.10)$$

Constant γ is called the *factor of distribution* (FD), which controls the error redistribution rate. This is usually set as $0 < \gamma < 1$.

Smoothing phase is the fine-tuning phase responsible for fine-tuning the weight vectors of the neurons. Similar to the growing phase, inputs are presented, and weights are adjusted, however,

with no new neuron growth. The purpose of this phase is to smooth out any existing quantization error.

2.4.4 Neural Gas and Variants

Neural gas (NG) (Martinetz & Schulten, 1991) is an artificial neural network which draws its inspiration from the vector quantization of self-organizing maps. Similar to SOM algorithm, the NG algorithm strives to achieve an optimal representation of the input space with a finite number of neurons (also referred to as nodes or units) each having a prototypes/feature vector. However, as the name suggests, NG does not have a pre-defined structure, and the neurons distribute themselves reminiscing of the dynamics of a gas filling the "input" space.

The NG algorithm trains the feature vectors by iteratively presenting them with samples drawn randomly from the input dataset. Given the input dataset S consisting of input vectors, $x_i \in \mathbb{R}^D$ and a finite number of feature vectors, $w_k(t) \in \mathbb{R}^D$, at each input presentation, the order of the neurons is identified by the distance between the input and the feature vector using a suitable distance measure. Next, all feature vectors are adapted using (2.11), where ε is the adaptation step size while λ is the neighbourhood range which determines the number of neural units significantly changing their weights. Both parameters are decreased over time to stabilize the training process

$$w_k(t+1) = w_k(t) + \varepsilon\, e^{-\frac{k}{\lambda}} (x_i - w_k(t)) \tag{2.11}$$

Similar to the SOM algorithm, the fixed number of neurons used in the NG algorithm has been identified as a major limitation. A number of alternatives have been proposed to eliminate this limitation, including growing neural gas algorithm (Fritzke, 1994) and growing when required network (Marsland et al., 2002).

Unlike the NG algorithm, the growing neural gas algorithm (GNG) (Fritzke, 1994), starts with a mere two neurons and adds additional neuron gradually to adapt to the distribution of the input data. Extending on the work by Martinetz (1993), the GNG algorithm performs the

learning by means of competitive Hebbian learning. Due to the dynamic neuronal addition, the GNG algorithm does not require parameters that change over time and continue to learn the input data topology by adding new neurons and neuronal connections until a convergence criterion is met. The GNG algorithm has been further extended with utility criterion to follow non-stationary distribution in (Fritzke, 1997). The new version allows for the removal of some of the neurons when the utility criterion falls below a certain threshold for the neuron. This allows for better representation of the input space allowing for incorporating dynamic changes observed in the input space.

Growing when required (GWR) network (Marsland et al., 2002) extends the neural gas algorithm to tackle the fixed size of the network of neurons used for training in order to approximate the input space more accurately, more parsimoniously. The underlying idea of the proposed algorithm is to accelerate the node growth compared to other dynamic neural network architectures. Marsland et al. (2002) argue that other growing networks only add neurons after a number of iterations as the previous iterations are required to accumulate the error at each node. To compensate this limitation, the GWR network adds new neurons whenever the best matching unit differs from the input vector by some (arbitrary) accuracy and the new neuron is initialized to match the current input vector. It is highlighted that GWR would work well with non-stationary distributions, adding new neurons to approximate new distribution once the distribution changes.

2.5 Computational Models of Multisensory Fusion

As more and more physiological evidence is made available about the neural mechanism that underly the multisensory interplay, a large number of computational theories have been put forward about the multisensory integration/fusion in the brain. The computation models implementing these theories can broadly be categorized as 1) biologically inspired artificial neural network models that implement known neurophysiological characteristics, 2) models that view multisensory fusion as Bayesian inference. The former category is inspired by the

fact that the multisensory integration is not present at birth but acquired during the course of interaction with multimodal stimuli during early stages thanks to the neural plasticity (Burr & Gori, 2012). Such models rely on Hebbian learning or principles of self-organization to implement the neuronal adaptation to perform the multisensory fusion. The latter category uses Bayes theorem and is based on conditional probabilities in achieving an optimal estimator of external multimodal stimuli.

2.5.1 Biologically Inspired Artificial Neural Network Models

A number of artificial neural models, both at the single neuronal level and neural network level, have been proposed to model the neurophysiological mechanism of the Superior Colliculus and the Cortex that is responsible for fusing multimodal stimuli. Some of these models implement the traditional neuroscience view of the multisensory fusion, which is often known as *unisensory before multisensory*. This view assumes that individual senses are first processed in their respective unisensory cortical areas of separated channels, and the fusion of the senses happens only at a later stage at higher-level cortical areas. Models implementing this view are usually organized in a hierarchical feedforward manner to resemble the hierarchical organization of individual sensory modalities and multisensory processing higher up at the hierarchy. While this view is still partially valid, more recent physiological evidence has demonstrated the role of primary cortical areas in the fusion process. They have been shown to receive inputs from other primary cortical areas and higher-order association areas, making them involved in an early stage fusion process. Subsequently, recent biologically inspired models of multisensory fusion have adapter this evidence with artificial synaptic links between primary areas and feedback links from higher-order areas.

2.5.1.1 Hierarchical Feedforward Models

Below we discuss a few noteworthy hierarchical feedforward artificial neural models of multisensory fusion.

Hierarchical Growing When Required Networks

Parisi et al. (2017) have proposed a self-organizing neural network hierarchy with four layers as depicted in Figure 2.10 for the unsupervised fusion of multimodal pose and motion inputs. The first two layers consist of GWR networks (Marsland et al., 2002) organized as a two-stream hierarchy for pose and motion modalities. The GWR algorithm trains a dynamic network of neurons to achieve an optimal data representation based on prototype vectors. The integration of the two streams are carried out in the third layer, (aptly named G^{STS} for superior temporal sulcus (STS) which shows a superadditive response to multimodal stimuli (Calvert et al., 2000)), which is modelled using another GWR network modified to account for movement dynamics in the joint vector space. The fourth layer is implemented with an extended GWR network called online semi-supervised GWR (OSS-GWR), which performs semi-supervised associative learning for learning action-word mapping and subsequently classifying multimodal inputs with action labels. Throughout the network, higher-order networks are trained with the activation trajectories of their immediately preceding layer as the input vectors.

However, a major limitation of the proposed approach is that it fuses the two unimodal representation at the 3rd layer by simply concatenating the two activation trajectories from layer 2. A number of studies have pointed out the limitation of such concatenation based approaches due to the fact that the representation, distribution and scale of the two dataset may vary (Ngiam et al., 2011; Srivastava & Salakhutdinov, 2014). Moreover, Zheng (2015) highlights three limitations of direct concatenation approach; 1) it causes over-fitting for small datasets, and specific statistical properties of each dataset are lost, 2) it makes it hard to learn highly non-linear relationships between the low-level features across the two modalities, and, 3) redundancies and dependencies that exist between the two modalities are overlooked.

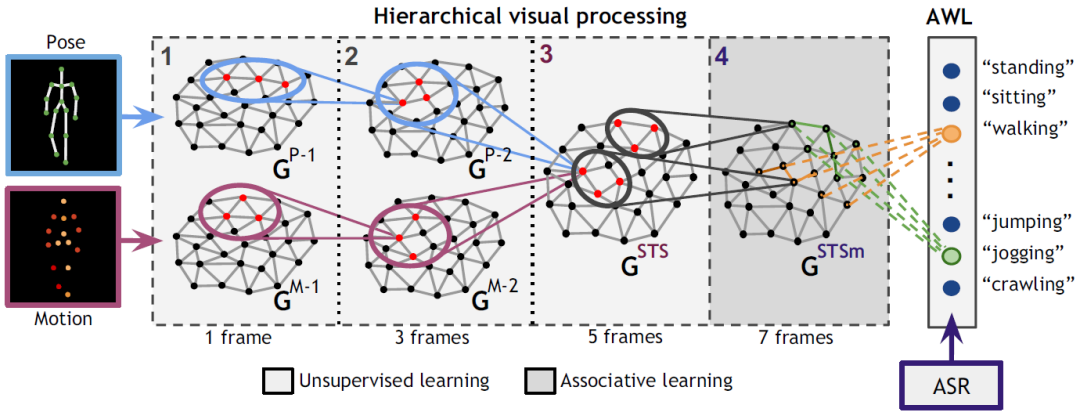

Figure 2.10: Hierarchical processing of pose and motion modalities with growing when required (GWR) networks (Parisi et al., 2017).

Hierarchical GSOM

Motivated by the hierarchical cortical modelling proposed by Hawkins and Blakeslee (2007), Fonseka (2012) presents a hierarchical model for processing multimodal input data. The architecture prescribes the low-level neuronal regions pertaining to multiple sensory modalities are connected indirectly through regions higher up in the hierarchy. Moreover, the simultaneous activation in different parts of the hierarchy and the integration of such activation for recognition is in agreement with the Ensemble Coding Hypothesis (Gazzaniga et al., 2013).

The hierarchy consists of multiple artificial neural layers modelled using the GSOM algorithm (Alahakoon et al., 2000). The hierarchy is composed of *primary cortical area*, *association area* and *higher-order association area* symbolizing the V1, V2 and V4 regions in the human visual processing hierarchy. The above areas may be composed of one or more neural layers and provide an increasingly abstract view of the inputs as the hierarchy is traversed from the bottom. All the layers use the activation trajectory of it immediately preceding layer as their input.

The *higher-order association area* is responsible for combining the sensations generated at each sensory channel. As the GSOM algorithm used for the implementation of this level uses competitive learning, the winning neuron is determined by the weighted errors of activation trajectories of two modalities of the lower level, $w_{A_k(v)}$ and $w_{A_{k+1}(v)}$, with their corresponding portions of the weight vector, $w^k_{O_{k,k+1}}$ and $w^{k+1}_{O_{k,k+1}}$.

$$e_{O_{k,k+1}} = \left[\alpha \times d_k\left(w_{A_k},\ w^k_{O_{k,k+1}}\right)\right] + \left[(1-\alpha) \times d_{k+1}\left(w_{A_{k+1}},\ w^{k+1}_{O_{k,k+1}}\right)\right] \quad (2.12)$$

Here the subscript A_k denotes the *association area* of the k^{th} modality while the subscript $O_{k,k+1}$ denotes the *higher-order association area* that receives inputs from modalities k and $k+1$. This allows for the use of different distance functions for each modality ($d_k(x)$ and $d_{k+1}(x)$) opposed to direct concatenation of vectors, and parameter α controls the extent of each modality's influence.

Fusion Adaptive Resonance Theory

Tan et al. (2007) proposed a generalization to the adaptive resonance theory (ART) (Carpenter & Grossberg, 1987a, 1987b), named fusion adaptive resonance theory (fusionART), extending the algorithm to multimodal pattern channels. Refer section 2.4.1 for an overview of ART. As shown in Figure 2.11, the comparison layer, F_1, which initially receive the inputs, is now divided into multiple layers to accommodate multiple pattern channels. One advantage of fusionART highlighted by Nguyen et al. (2008) is that it does not require input to be available to each channel. Such missing inputs are handled by initializing the input vector to all zeros.

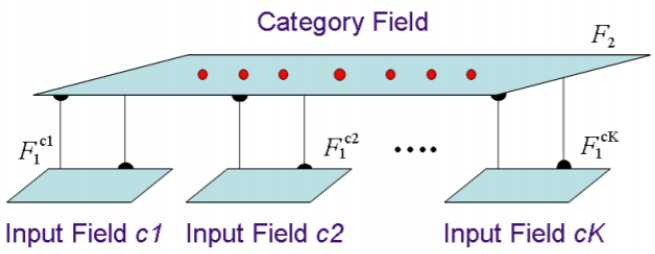

Figure 2.11: Architecture of fusionART (Tan et al., 2007)

The fusion of the multimodal inputs is performed by weighted combination the activations of layer F_1. With $I^{ck} = \left(I_1^{ck}, I_2^{ck}, \dots, I_n^{ck}\right)$ denoting the inputs to channel $ck;\ k = 1, \dots, K$ and w_j^{ck} denoting the weight vector associated with j^{th} node in F_2 and pattern channel F_1^{ck}, the code activation at node j, T_j, is calculated as,

$$T_j = \sum_{k=1}^{K} \gamma^{ck} \frac{\left|I^{ck} \wedge w_j^{ck}\right|}{\alpha^{ck} + \left|w_j^{ck}\right|} \tag{2.13}$$

The contribution parameter, $\gamma^{ck} \in [0,1]$, determines the extent to which each modality influence the dynamics of the system.

FusionART has been demonstrated with image and text data extracted from news articles on terrorist attacks (Nguyen et al., 2008). The evaluations are based on the clustering achieved with the fusionART on multimodal inputs, comparing it against the clustering achieved on unimodal inputs. The accuracy reported is around 40%, while the unimodal clusters report similar results. They attribute this less-than-satisfactory results to the small size of the training sample compared to the high dimensionality of the input vectors. However, experiments carried out with artificial data have archived better results.

Hierarchical Modelling of Superior Colliculus

Magosso et al. have presented an artificial neural network that models the multimodal dynamics of the superior colliculus (SC) in a series of papers (Magosso et al., 2008; Ursino et al., 2009). Their model consists of two layers, a layer of unisensory neurons having receptive fields for each modality and another layer of multisensory neurons feeding on the unisensory layer (see Figure 2.12).

The neurons in the unisensory layers exhibit non-linearity with the use of a sigmoid function for the activation of neurons while the neurons in the multisensory layer perform weighted sum calculation of the inputs from the unimodal layer. Due to the non-linearity, the multimodal neural network can demonstrate the inverse effectiveness property of biological neurons, switching between the superadditive enhancement and the subadditive enhancement based on

the crossmodal stimuli (Ursino et al., 2014). Moreover, the proposed model demonstrates the multisensory enhancement and multisensory suppression properties (Meredith & Stein, 1986) facilitated by the lateral connections having an inhibition profile in the shape of a Mexican hat.

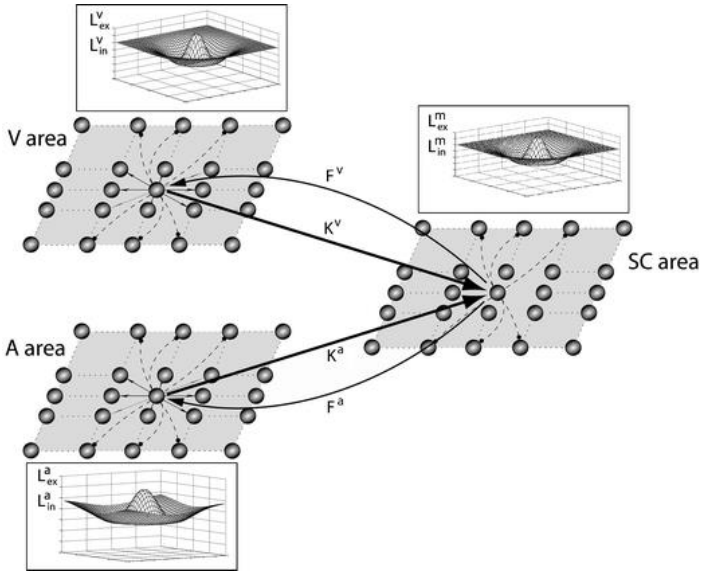

Figure 2.12: Architecture of artificial hierarchical model of the superior colliculus. Each neuronal area (A: auditory, V: visual and SC: superior colliculus) is laterally connected to the neuron in the same area, and their excitatory/inhibitory profile is shown in their respective charts. Inter-area feedforward excitatory connections send signals from unimodal neurons to the multimodal neurons in the SC area (arrow K) while the feedback connections send signals the other way around (arrow F). (Magosso et al., 2008)

2.5.1.2 Models with Inter-area Feedback

Recent physiological evidence has demonstrated the role of primary cortical areas in the fusion process. They have been shown to receive inputs from other primary cortical areas and higher-order association areas, making them involved in an early stage fusion process. Models inspired by this phenomenon contain lateral connections between unimodal areas of the multiple modalities. Among other demonstrations, they have been used to simulate audio-visual illusions among primary areas.

Magosso et al. (2012) proposed a multisensory model containing direct lateral connections between the artificial cortical areas of each modality. The model has been demonstrated for audio-visual inputs with a fine topological organization (higher spatial resolution) for the visual modality and a coarse topological organization (lower spatial resolution) for the auditory

modality. As shown in Figure 2.13, intra-layer excitatory and inhibitory synaptic connections connect the neurons within a single modality while inter-layer excitatory connections connect the two modalities.

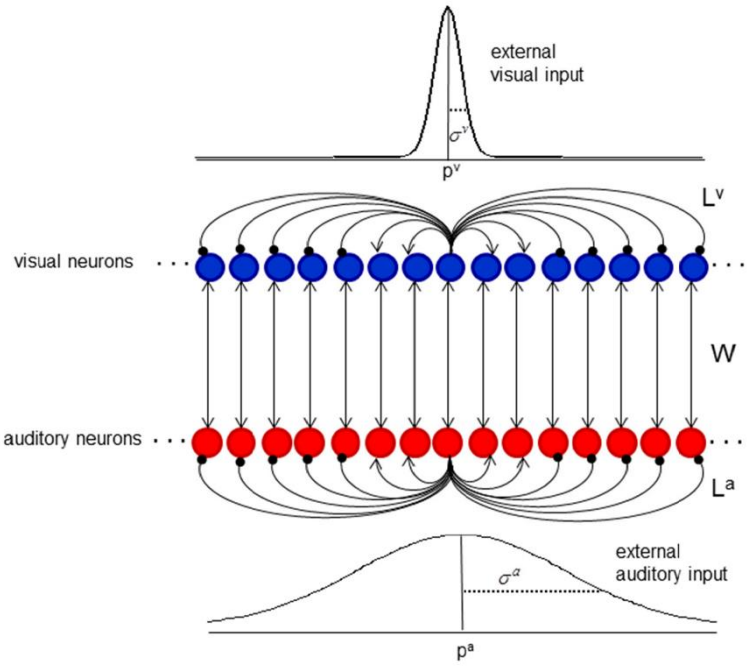

Figure 2.13: Schematic diagram of the model with intra-layer excitatory and inhibitory connections and inter-area excitatory connections. Arrowheads denote excitatory synaptic connections while dots denote inhibitory synaptic connections (Magosso et al., 2012).

This model has been successful in demonstrating the effects and aftereffects of ventriloquism (Refer section 2.2.1.1 for an introduction to ventriloquism) despite the absence of a dedicated convergent multimodal area. The effect of ventriloquism is ascribed to the excitatory feedback between the visual and auditory neuronal layers while the aftereffects are ascribed to the Hebbian training mechanism that modifies the intra-area lateral connection weights.

Another implementation of crossmodal interaction between multiple modalities has been presented in (Hoshino, 2011). This specifically models the direct interactions between the lower-order unimodal sensory areas and demonstrates the contribution of such interaction in the multisensory fusion of subthreshold stimuli that would not otherwise reach perceptual awareness in isolation. The model consists of two unimodal areas (X and Y) connected to each other via lateral connections and a higher-order multimodal area (M). The experiments

conducted with the implementation of the proposed model demonstrate how the interaction between the lower unisensory areas is essential to generate a suprathreshold response to congruent subthreshold multimodal stimuli.

2.5.2 Bayesian Models

A number of Bayesian models have also been proposed to model the multisensory fusion of the brain. The underlying idea behind using a Bayesian approach for multisensory fusion is that the human brain operates in an environment with uncertainty. The uncertainty arrives from the environmental noise, neural variability and neural structural constraints (Ursino et al., 2014), especially in the case of multimodal integration, and the brain needs to account for this uncertainty in its operations. Hence, Bayesian principles are proposed to model this uncertainty and to compute the posterior probability given the uncertain sensory information. These models mathematically formalize the fusion of multimodal sensory signals having different reliability levels.

Given an uncertain sensory cue c and the actual attribute, a, being transmitted by the cue, the Bayesian formulation calculates the posterior probability $p(a|c)$ using the Bayes theorem,

$$p(a|c) = p(c|a) \cdot p(a)/p(c) \tag{2.14}$$

Here, $p(c|a)$ is the likelihood of the cue occurring given attribute, taking into account the uncertainty associated with the cue and $p(a)$ brings in the prior knowledge about the attribute a. The simplest Bayesian formulation transformed into the multimodal domain would be,

$$p(a|c_A, c_V) = p(c_A, c_V|a) \cdot p(a)/p(c_A, c_V) \tag{2.15}$$

where c_A and c_V are the sensory cue in individual modalities, respectively. A similar Bayesian model has been proposed by Battaglia et al. (2003) to explain the localization of audio-visual signals. With the above formulation, a is the location of the event to be estimated while c_A and c_V are the auditory and visual locations reported by the subjects.

Further extensions to these models have been proposed to model the dynamics of incongruent stimuli, for example, when the auditory and visual are emitted from different locations. In these models, the attribute, for example, the location is allowed to have two distinct values for each modality. Starting with visual and auditory locations reported, c_A and c_V, the position is estimated by maximizing $p(a_A, a_V|c_A, c_V)$ where a_A and a_V are the auditory and visual locations respectively. The new formulation takes the form,

$$\begin{aligned} p(a_A, a_V|c_A, c_V) &= p(c_A, c_V\,|a_A, a_V) \cdot p(a_A, a_V)/p(c_A, c_V) \\ &= p(c_A|a_A) \cdot p(c_V|a_V) \cdot p(a_A, a_V)/p(c_A, c_V) \end{aligned} \tag{2.16}$$

Interaction prior, $p(a_A, a_V)$, is the joint prior probability of the auditory and visual locations, and this captures the interaction between the two modalities (Körding et al., 2007). When the two locations are independent, there is no interaction between the modalities. The interaction prior was a Gaussian ridge along the diagonal when the model was fitted with a cat's localization response, reflecting the frequent case of actual auditory and visual locations being the same.

2.6 Summary

Multimodal perception in humans offers richer information about the surrounding since sensory modalities jointly capture the same event or object supplementing each other. Psychology and neurobiology research disciplines have identified many cases of interactions between modalities, where the perception of one sensory modality is conditioned by the information simultaneously available to another (Bertelson & De Gelder, 2004). This conditioning has been demonstrated in the form of crossmodal influence (Mcgurk & Macdonald, 1976; Schwartz et al., 2004) and crossmodal recalibration (Radeau & Bertelson, 1974; Wallach, 1968) while special medical conditions provide further evidence of multimodal dynamics in the human brain (De Gelder & Bertelson, 2003; Ramachandran & Hubbard, 2001). Moreover, neurobiological experiments carried out on non-human subjects such as cat and primates (Stein & Stanford,

2008) and more recently, the findings from neuroimaging have reaffirmed the role of multimodality in human perception (Beauchamp et al., 2004, 2010).

There has been a number of theories put forward on how the human brain processes sensations from multimodal stimuli to create a united conscious perceptual experience. The feature integration theory (Treisman & Gelade, 1980) proposes that the object's location mediates the binding of the features such as the form, depth, motion and colour while the attention is proposed as the "glue" that combines this information. On the other hand, the synchronization theory (Engel et al., 1999) proposes that the neuronal activation in various parts of the brain induced by the same object are in synchrony and this synchronization is the basis of binding.

As more and more physiological evidence is made available about the neural mechanism that underly the multisensory interplay, a number of computational theories have been put forward about the multisensory fusion in the brain (Cuppini et al., 2010; Fonseka, 2012; Magosso et al., 2012; Parisi et al., 2017; Rowland et al., 2007; Ursino et al., 2009). A number of such computational models proposed are based on the principles of self-organization as cortical maps in the human brain develop early by means of self-organization mechanisms (Fonseka, 2012; Khacef et al., 2020; Parisi et al., 2017; Tan et al., 2007). The self-organization-based algorithms that have been used as the basis for such computational models include SOM (Kohonen, 1990), GSOM (Alahakoon et al., 2000), ART (Grossberg, 1982, 2013), and NG (Martinetz & Schulten, 1991).

While there have been a number of attempts at building computational models of multimodal fusion, they have mostly focused on modelling the dynamics of the brain (Cuppini et al., 2010; Magosso et al., 2012; Rowland et al., 2007; Ursino et al., 2009) rather than on the application of the developed models on real-world problems, let alone the challenges posed by the vast amount of data generated in *digital environments.* Hence, despite recent research effort in multimodal fusion for a holistic perception, the question remains open on how to better adapt biologically plausible algorithms to develop an artificial *impression* generation architecture for *digital environments*, especially for data-intensive environments.

Moreover, while most of the previous attempts have focused on the supervised paradigm, fusion techniques for unsupervised environments are still unresolved and an ongoing problem (Dasarathy, 2006). With the vast amount of data being generated, unsupervised learning mechanisms are important more than ever before due to the inability to label such large datasets.

Chapter 3

Theoretical Foundation - Computational Basis for Artificial Impression Generation

3.1 Introduction

Even before the advent of computing, humans envisioned developing intelligence artificially. Sensing and perceiving the surrounding by fusing different sensory modalities has been a core component of the AI from the inception of the concept. However, recent changes in the data and technology landscape have made a significant shift, requiring rethinking how this can be achieved in AI.

As highlighted in Chapter 1, compared to a decade or two ago, artificial intelligence applications operate in a much more different landscape in terms of the variety and volume of data available for processing. Then, with less automation and connectivity, the data collection was passive. Moreover, datasets collected were small, mostly unimodal, isolated and infrequent. With these datasets, the AI applications had access only to limited aspects of an

issue at hand with a small amount of data. By analysing these small datasets individually, the interactions among different modalities were mostly not accounted for while it remained with the human in the loop to analyse and synthesize outcomes of different analysis made by AI to form a holistic understanding of the situation and to make appropriate decisions.

However, nowadays, the sensing and capturing in Big Data era generates large, unlabelled, multimodal and multisource, connected and high frequent datasets starting to more closely represent the natural environment being captured. This phenomenon has created a *digital environment* which is a closer representation of the natural environment than the one derived with smaller, labelled, unimodal, isolated and infrequent data. The *digital environment* closely resembles how a human would perceive his environment where they perceive the surrounding in a holistic manner by analysing and fusing different sources of sensory excitations.

Humans' ability to create a holistic understanding of an event or a situation is well supported by the brain functionality, developing an *impression* of such an event/situation based on multiple input sources representing diverse aspects of the event/situation. The ability to exert the impact of an additional co-occurring sensory input on a particular input (or what is represented by a particular input) results in the contextualization of the particular input (or representation). Moreover, it is important to note that sensory inputs do not carry any labelling or annotation, and the *impression* arises solely from the impacting and fusion of the modalities in the context of past knowledge/memory.

We argue that AI for Big data era must be capable of imitating such ability and propose an innovative concept of a *digital impression* as the basis of achieving this functionality. That is, enabling AI applications to form a coherent and holistic *impression* on the *digital environment*, similar to how humans would form a coherent and holistic *impression* on the natural environment from multimodal sensory excitations they receive. We further argue the need and justification: (a) The need for more autonomous AI, i.e., a new breed of AI applications which is autonomous and proactive compared to manual and human-driven AI applications in the past.

(b) Availability of multiple and multimodal data sources with deeper granularity, captured with higher frequency and in large volumes.

Chapter 3 is organized as under two major sections. The first section posits our premise on human sensation and perception in forming a coherent *impression* about the external world and how this can be implemented on artificial counterparts. The second section proposes an artificial model of neocortex for the purpose of generating artificial *impressions* on *digital environments.* The artificial model consists of a conceptual model conceptualizing the organization of the artificial cortical layers, neuronal connections and information flow among them, an architectural model describing the components of the proposed artificial model and a computational model outlining the algorithmic means by which we propose to achieve this.

3.2 Our Premise

In this section, we layout our premise of sensation and perception in humans. We discuss how humans construct the state of the external environment from the sensory inputs by forming what we call a coherent *impression* about the external world. We draw on neurobiological and physiological research in developing our understanding and highlights the role of knowledge representation which provides the context for making sense of the sensory information.

It has been one of the major goals of the field of AI to equip AI algorithms with the ability to perceive as we humans do. Noting that the sensing and capturing in Big Data era generates large, multimodal, dense and high frequent datasets creating a *digital environment* which closely represent the natural environment being captured, we discuss how we can enable AI algorithms to generate artificial *impressions* on *digital environments,* and which biologically plausible algorithms could be utilised/developed to achieve this.

3.2.1 Coherent Impressions

Humans, similar to other living beings, consume a plethora of external information through their sensory system. They receive this information in the form of sensory excitations at their sensory organs, and the human brain is then responsible for processing these sensory excitations. It is the duty of the human brain to reconstruct the state of the external environment from multiple sensory cues to provide us with an appropriate interpretation of the surrounding.

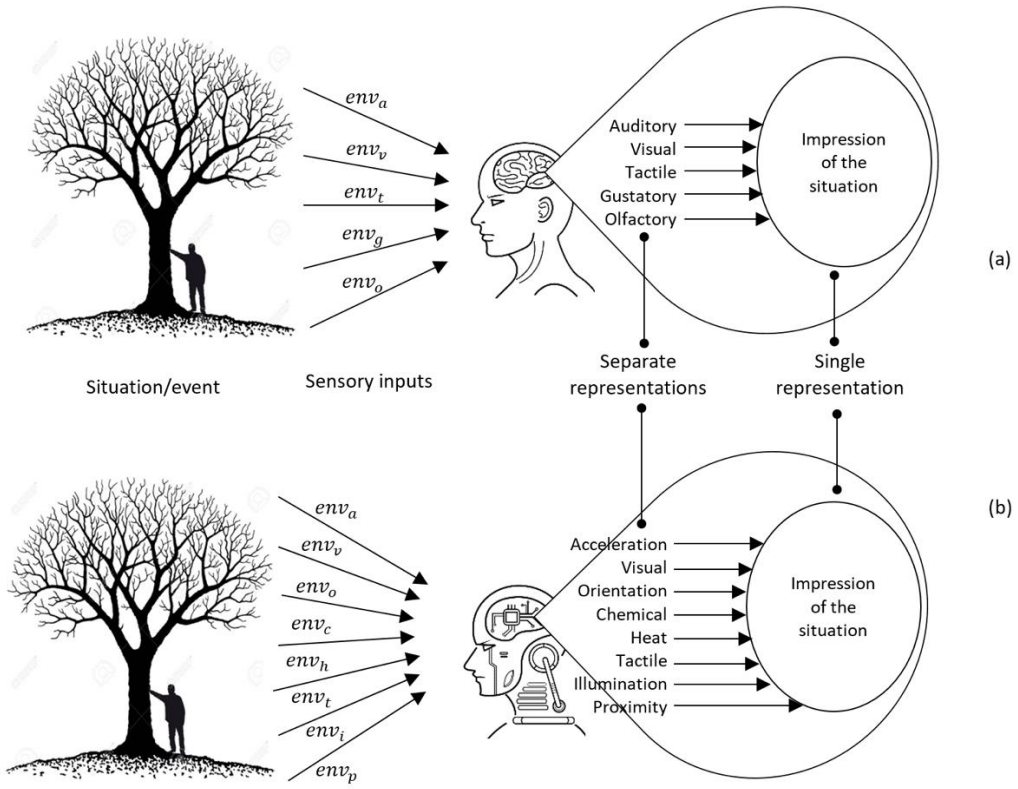

Figure 3.1 Generation of impressions on a situation/event by humans (a) and artificial counterparts (b)

The highly specialized neural mechanism in our brain processes sensory excitations allowing us to perceive the surrounding in the form visual, auditory, olfactory, gustatory and tactile perceptions. The process of analysing and fusing different sources of sensory information

generates what we call a coherent *impression* about the external world, which mediates our actions and reactions.

We have so far used the term *impression* without providing a formal definition. Given the centrality of this concept for this thesis, it is imperative that we provide a precise formal definition of what we mean by the term *impression*. In the context of this thesis, the term *impression* refers to,

> *Any form of interpretation derived from sensory data at hand, either captured naturally by a living being or artificially by a sensor, in the context of a knowledge representation derived from past encounters.*

As the above definition implies, an *impression* is a brain's interpretation of the surrounding/situation derived from the sensations. This *impression* in humans (as well as animals) is likely something developed over the course of evolution, allowing for them to be safe from dangers in nature such as predators and natural hazards.

The concept of *impression* is well aligned with the notion of sense data (Russell, 1914). Sense data refer to the mind-dependent objects that we are directly aware of in perception. They are the mind's interpretation of sensory inputs or mental images and have exactly the properties they appear to have. For example, looking at a ripened lemon, we form the image of the lemon in our mind. This image is yellow and round. Sometimes the notion of sense data is interpreted narrowly limited to things perception makes us directly aware of. However, we also perceive things that we are indirectly aware of, that is, being aware of something in a way that depends on the awareness of something else (Jackson, 1977). Elaborating on the previous example, we only see the surface of the lemon that is facing us. However, we count as seeing the lemon by virtue of seeing something else, namely, the facing surface of the lemon. The notion of *impression* we discuss here encompasses both direct and indirect perceptions.

Another aspect of *impression* is making sense of sensory data in the context of past knowledge/experience. It is, in fact, the past experience that gives meaning to what we perceive

right now. In the previous example, we would only count a yellow, round object instead of a lemon in the absence of past knowledge/experience. The *impression* generation is heavily dependent on the classification or clustering process that interprets the current perception in the context of past knowledge/experience. This process can also be viewed as a form of transfer learning, "extracting the knowledge from one or more *source tasks* and applying the knowledge to a *target task*" (S. J. Pan & Yang, 2010, p. 2), in the same domain in this case. Hence, the details of the knowledge representation mechanism which allows for classifying current perception to interpret it is of paramount interest.

With regards to the process of *impression* generation in humans, the process is aligned well with the mind's system 1 proposed by Kahneman and Egan (2011). They identified system 1 to be fast, instinctive, stereotypical and automatic. It is the seat of trained expertise. It carries unconscious biases coded into it from the past experience. Moreover, the system 1 does not spend time reasoning compared to system 2, which is deliberate and logical and is the seat of deduction and insight.

3.2.2 Structure of Neocortex to Support Impression Generation

Turning back to the natural counterpart – the brain – for inspiration, it is interesting to analyse how the neural mechanisms in the neocortex are organized to facilitate knowledge representation. There is a large body of research that demonstrates human knowledge is organized in a category-specific manner at both the cognitive and neural levels (Caramazza & Shelton, 1998; Tyler & Moss, 2001; Capitani et al., 2003; Mahon & Caramazza, 2011). Experiments conducted with brain-damaged patients having category-specific semantic impairments has been used as evidence of such category-specific organization of knowledge in the neocortex. These patients have conceptual level impairments that are specific to a particular category such as animals, plants, conspecifics or artefacts. It is hypothesised that different semantic categories are processed by distinct and dedicated neural regions. Domain-specific hypothesis (Caramazza & Shelton, 1998) specify that "there are innately dedicated neural

circuits for the efficient processing of a limited number of evolutionarily motivated domains of knowledge".

More recently, findings of such studies have been re-evaluated by neuroimaging studies using functional magnetic resonance imaging (fMRI) technology to map category-specific regions of the neocortex (Martin, 2007). Martin (2007) identifies a more intricate organization of the neural region, consistent with the categorical organization proposed in earlier studies. He notes that different aspects of an object - such as what it looks like, how it is used, and how it moves - are coded in different parts of the neural circuitry and object categories such as animals, plants and tools have a distributed, partially distinct sensory-based coding. Hence, the object concepts *emerge* from activity in aspect-based regions of the brain. However, he notes that aspect-based regions demonstrate categorical organization, thus providing evidence aligned with the category-based formulation.

This understanding of the organization of the neural mechanism in the neocortex is important in our quest to implement *impression* generation process for artificial counterparts. The aspect-based representation - such as what it looks like, how it is used, and how it moves - is analogous to different aspects of an object/event captured by different sensory modalities. This means that an artificial implementation would require a representation mechanism at modality level to capture different aspects of the object/event. Moreover, to facilitate the *emergence* of higher-level object concepts from activity in aspect-based representations, a higher-order fusion mechanism would need to be implemented.

Just as much as it is important to understand the organization of the neocortex supporting *impression* generation, it is important to look at the dynamics of the neural mechanism that facilitate it. Reentry in nervous systems has long been suggested as the mechanism that couples the functioning of multiple areas of the cerebral cortex and thalamus (Edelman & Mountcastle, 1978) and the experimental evidence on the phenomenon have since suggested that it is one of the most important mechanisms supporting multimodal integration in the mammalian brain (Edelman, 1993). Reentry is the "ongoing bidirectional exchange of signals along reciprocal

axonal fibres linking two or more brain areas" (Edelman & Gally, 2013, p. 1). It supports the coordination of neuronal activity in functionally and anatomically segregated areas in the brain. By these means, they bind crossmodal sensory features by synchronizing and integrating patterns of neural activity in different brain regions. Edelman and Gally (2013) go further to suggests that "by sustaining attention and short-term memory, reentry might even play a central role in generating conscious awareness" (p. 1).

As highlighted about the organization earlier, the neocortex has evolved to be a mosaic of functionally and anatomically segregated areas (Somogyi et al., 1998). Due to this, neurons responsive to various modalities or sub-modalities of a given multimodal sensory input are distributed across separate areas in the neocortex. As shown in Figure 3.2, the neurons belonging to different layers within a cortical area form a dense columnar array and neurons belonging to different cortical areas are reciprocally interconnected by reentrant networks of excitatory axons (Markov et al., 2014). These reentrant neurons are thought to develop very early by means of self-organization mechanisms during the embryonic development of the mammalian brain (Shatz, 1992). Stimulus evoked patterns of activity in newly developed neurons help develop the connectivity patterns both within and among cortical areas.

The key role played by reentrant neuronal circuitry is the integration or binding of the multimodal sensations into a coherent percept. Evidence suggests that synchronous exchanges of signals among neuronal groups in dispersed cortical areas correlate with, and bind together, the multiple but distinguishable features of unified, conscious scenes (Edelman & Gally, 2013). Reentry is thus thought to be critical for transformation of sensory neural activity into a stable, consciously reportable percept.

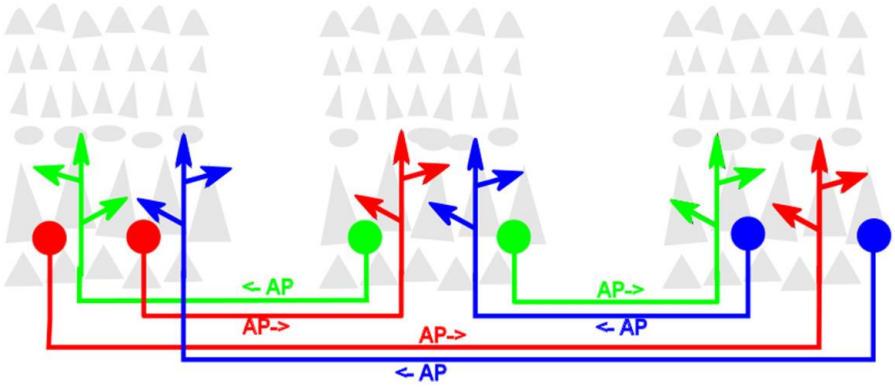

Figure 3.2 Schematic diagram of reentrant neuronal bundles linking segregated cortical areas (Edelman & Gally, 2013)

This understanding of reentry, the interaction mechanism among segregated areas in the neocortex that process different modalities or sub-modalities, is important in our quest to implement *impression* generation process for artificial counterparts since the *impression* is the *impact* multiple sensory inputs make on the mind as a whole (captured by cortical regions). As such, it is important to understand the mechanism which results in sensory inputs of one modality which influences cortical regions capturing inputs from other modalities. Such a mechanism should shape the design of artificial impression generation mechanism. In our conceptual model of artificial neocortex (Section 3.3.1), we propose to implement this mechanism with inter-modal associative connections.

3.2.3 Digital Environment

For the artificial counterparts, the key to success is the accurate sensing and perception of the external environment forming a coherent *impression* about it. Remember, our goal here is to enable digital counterparts to form a holistic *impression* on the *digital environment* (Figure 3.1 (b)), similar to how humans would form a holistic impression on the natural environment from multimodal sensory excitations they receive (Figure 3.1 (a)). With environment sensing being performed across multiple modalities with high frequency, the *digital environment* represented by these data provides a more complete and realistic environment for artificial counterparts to interact and operate. Hence, to be able to take advantage of this, it is essential that modern AI is geared to tackle volume and variety challenges posed by the *digital environment*.

To elaborate the above, let us consider an example use case. Consider a traffic monitoring and management system that helps monitor the traffic situation of a city in real-time to facilitate remedial actions. A holistic *impression* of the city-wide traffic situation is essential for such a system as individual situations across sites and areas of the city is highly interrelated. The sensory network of the system would consist of various types of sensors placed around the city to get an understanding about the traffic situations at specific sites, areas of the city as well as the whole city. Examples include Bluetooth sensors placed in road intersections uniquely capturing vehicle movements, motion sensors placed near pedestrian paths and crossings to judge the pedestrian traffic, and video cameras capturing vehicles and pedestrian movement. It is the data, that is captured across multiple modalities, multiple sources, multiple sites with high frequency, that enables the monitoring system to form a city-wide holistic *impression* of the traffic situation which is the essential first step for remedial actions. Such extensive coverage of the traffic situation of the city provides a comprehensive *digital environment* for the algorithm to operate on and generate a holistic *impression.*

However, with the lack of algorithmic capability for machines to form a holistic *impression,* it is currently created in the minds of people who operate these systems. While humans are good at forming a holistic *impression,* they lack in terms of the volume of information they can handle. For example, while a human can effectively integrate information from multiple sensory modalities to form a clear *impression* on his surrounding, a city-wide traffic situation involving hundreds of intersection, thousands of road segments and hundreds of thousands of vehicles and pedestrians would easily overwhelm even a team of human operators of a city-wide traffic monitoring and management system. Therefore, we would greatly benefit from enabling artificial counterparts algorithmically forming holistic impression from large volumes of multimodal data.

However, the question remains: how can we provide artificial counterparts with the ability to form *impressions* in *digital environments* they operate?

3.2.4 Computational Elements for the Simulation of Cortical Functions: SOMs and GSOMs

As highlighted earlier, it has been one of the major goals of the field of AI to equip AI with the ability to perceive as we humans do. As an important first step in allowing AI to form an artificial *impression* about the external world which it operates in, researchers have long been developing knowledge representation mechanisms (Brachman & Levesque, 1985).

The family of self-organizing map algorithms have been widely used to model the feature maps in the human brain. This is due to two major reasons; 1) the working mechanism of the algorithm has been inspired by the self-organization, which is a central mechanism of nature that organizes selected parts of a system so as to promote a specific function (Camazine et al., 2003), 2) the algorithm uses an unsupervised training mechanism which resembles how humans learn from sensory inputs which are not accompanied by any labelling.

Self-organization has been hypothesised as the mechanism by which the features maps of the brain responsible for processing sensory modalities are developed. The experimental evidence suggests that sensory projections are conditioned by experience, due to the plasticity of the neurons and the self-organization mechanism that drives the conditioning.

Nature does not label its data. Humans do not receive any labelling with the sensations they receive through their sensory systems. There are no data outside of the sensory inputs for humans to form an understanding or an impression of the sensory inputs. This unsupervised learning nature should be accounted for in the AI algorithm that mimics the natural counterpart.

The elegance of self-organization process, unsupervised nature of learning and the capability of topographic maps to produce a *reduced representation* of facts without loss of knowledge about their interrelationships (Kohonen, 1989) have inspired the family of self-organizing map algorithms for information processing and knowledge representation. These algorithms develop simplified models of the world at a level of abstraction in relations to the inputs from the

observable world and have been widely used in knowledge representation tasks in a wide array of fields.

As described in Section 2.4, the SOM algorithm is the most widely used algorithm of the family of self-organizing map algorithms. The algorithm produces a discretised representation of the input space onto a lower-dimensional space, facilitating dimensionality reduction while producing the reduced representation. The training algorithm preserves the topological relationships from the input domain onto the lower-dimensional space inspired by the topographic maps in the human sensory system. Moreover, the algorithm uses an unsupervised learning mechanism consistent with how a human would learn/organize his experiences.

To alleviate the limitation of fixed size and structure of SOM, the GSOM algorithm has been proposed. As highlighted in Section 2.4.3, the dynamic nature of structure allows for better representation of relationships in input data and closer resemblance of dynamic structure adaptation of the cortical areas it is inspired by.

3.2.5 Generating Digital Impressions

Let us briefly review where we stand at this point. We have analysed the process of fusing multimodal sensory data in humans to generate coherent and wholistic *impressions* about the external world and formally defined the term *impression*. We discussed the biological structure of human neocortex facilitating *impression* generation. Noting that the technological advancements in data capture (large volumes of multimodal data at high frequency) have created a *digital environment* which closely approximates the natural environment, we aspired to enable digital counterparts to form a similar holistic *impression* on the *digital environment.* Since the natural *impression* in the human mind is generated by the structure and functionality of the human cortex, we reviewed computational elements that can be used for the simulation of cortical functions.

Let us now delve into how we can enable digital counterparts to form a holistic *impression* on the *digital environment*.

We identify four key features of such an artificial *impression* generation system.

1. The ability to capture and represent multimodal data in a common representation.

   Such a system would be required to consume multimodal data pertaining to a given situation and a common representation mechanism and common format to represent information from different modalities would be beneficial in associating similar concepts across modalities. A common representation would facilitate implementing a unified way of interaction between modalities which is identified as the second key feature. Such a representation would allow extending the overall system to multiple modalities without much effort given that representation mechanism is independent of the modality-specific nuance.

   Moreover, a common representation is reflective of the biological organization of the sensory system in humans where primary cortical areas of different modalities are organized in a similar way. Due to this uniformity in representation mechanism at the lower level, when a person or an animal is deprived of one sensory modality, the other sensory modalities overtake the primary cortical region that is used to process that sensory modality. For example, humans who are congenitally deaf process visual information in areas that normally become the auditory region. Similarly, the congenitally blind humans use the rearmost section of the cortex, which usually processes visual signals to read braille with tactile sensations (Hawkins & Blakeslee, 2007).

2. A mechanism for different aspects of an event to interact with/impact each other.

   How we perceive a single modality is impacted by other co-occurring modalities. We perceive a given modality in the context of the other modalities, and they condition our perception of the given modality; thus, the representations of the same situation attain a level of contextualization.

The crossmodal effect of multimodal sensory stimuli is backed by biological evidence from experiments carried out on humans and animals. As highlighted in Section 2.2.1, numerous researchers in behavioural and psychological fields have demonstrated the crossmodal effect between sensory modalities. These effects include crossmodal influences where the sensation in one modality influence the perception of a co-occurring sensation of another modality and crossmodal recalibrations where artificially induced discrepancies in a single modality lead to the (temporary) alteration of correspondence between the modalities (Harris, 1965; Radeau & Bertelson, 1974, 1977).

Hence, this feature is aimed at capturing the interaction among modalities and enabling the system with such capability.

3. The ability to combine multiple aspects of an event/situation.

   We identify fusion to be one of the most important features in enabling artificial *impression* generation in *digital environments*. The fusion of multimodal inputs enables incorporating multiple aspects of a situation, allowing for forming an unambiguous interpretation of the event from partial - and often ambiguous - information present in each modality. The stimuli received through different sense organs at a given time are highly relatable. This is due to the fact that those multimodal stimuli originate from the same underlying event/object, hence represent the different elements of the same situation. The idea here is to utilize complementary information from co-occurring modalities to enhance the total information we know about the external even/object. This is the main rationale for fusing information from multiple modalities.

   As highlighted earlier in the chapter, the aspect-based representation - such as what it looks like, how it is used, and how it moves – in the human brain is analogous to different aspects of an object/event captured by different sensory modalities. Hence, the emergence of higher-level object concepts from activity in aspect-based

representations requires a higher-order fusion mechanism to achieve a final coherent impression based on the primary sense data. With this feature, we aim to empower the artificial system with a similar ability.

4. A mechanism to maintain memory/knowledge of the situation.

   This feature is for allowing for new events/situations to be evaluated based on the accumulated knowledge from the past.

   As highlighted earlier, an important aspect of *impression generation* is making sense of sensory data performed in the context of past knowledge/experience. It is, in fact, the past experience that gives meaning to what we perceive right now. Hence the representation mechanism needs to support memory/knowledge in order to classify the current perception and interpret it.

The perspective taken in this thesis is that this *impression generation* is a process based on the four key features identified above. It is imperative to think about how these four key features are implemented in an artificial system for *impression generation*.

We propose to implement the first key feature - the ability to capture and represent multimodal data in a common representation - by implementing artificial cortical areas modelled by topographic maps. As highlighted in Section 3.2.4, topographic maps produce a *reduced representation* of facts without loss of knowledge about their interrelationships (Kohonen, 1989) and have been used for information processing and knowledge representation due to this favourable characteristic. Topographic maps capture the interrelationships of data in its underlying structure, and the points on the map represent coherent concepts for a given modality. Hence, topographic maps are independent of the modality-specific details, and the proposal is to use such topographic maps in all modalities facilitating collocating similar concepts across modalities.

The use of topographic maps satisfies the fourth key feature - a mechanism to maintain memory/knowledge of the situation - as well. A topographic map is built with a set of training

data, and the structure of the topographic map represents the knowledge acquired and loose memories formed during the training. As the new data arrives, they are evaluated against the existing memory/knowledge. In a more algorithm specific details, the best matching unit is searched for in the topographic map, which represents loose matching of the new data to existing memory/knowledge. Hence, the use of topographic maps as the common representation supports maintaining memory/knowledge of the environment.

In this thesis, we propose implementing the second and third key features as an unsupervised multimodal clustering process which organizes artificial cortical areas into meaningful clusters. The crossmodal effect (second key feature) and multimodal fusion (third key feature) are proposed to be modelled as clustering of cortical areas of a particular modality while considering the presence and the context of co-occurring stimuli on other modalities, i.e. clustering each cortical area using a metric that also accounts for co-occurring stimuli. The clustering algorithm should take the following two factors into consideration; 1) the activation distribution within the cortical area of the modality under consideration, similar to a regular clustering algorithm, and 2) the co-activation distribution of cortical areas of other co-occurring modalities and own cortical area. Taking the co-activation across modalities into consideration accounts for the crossmodal effect while the multimodal clustering satisfies multimodal fusion, the second and third key features identified. This process results in mutual bootstrapping of each cortical area being clustered while taking co-occurrence into account.

This process is different from the traditional sense of multimodal integration where attributes pertaining to multiple modalities are joined together by a common factor (or a key) such as time, before being subjected to a clustering process. We find the proposed approach more biologically plausible given that sensations in individual modalities are processed by their respective modality-specific cortical hierarchies while inter-modal associative connections assist interpreting the sensation in the context of co-occurring stimuli.

Thinking about this from a psychological point of view, while we are able to construct a coherent *impression* from the biological fusion process, we are still able to distinguish sensation

in each modality. We perceive each modality in the context of other modalities. While the other co-occurring modalities have affected and influenced the complete fused perception, they have not made it to an indistinguishable mixture. We draw parallels to this with the proposed multimodal clustering approach where the co-occurring stimuli make an influence on the clustering of a modality, however still maintaining a level of separation between modalities.

3.3 An Artificial Model of Neocortex

In this section, we propose an artificial model of neocortex for the purpose of generating artificial *impressions* on *digital environments.* The artificial model consists of a conceptual model, an architectural model and a computational model. The conceptual model describes the high-level organization of the proposed artificial cortical layers and their functionality. The architectural model describes the components which generate associated functionality of the conceptual model and information flow among them. The computational model proposes the algorithmic means by which we propose to achieve this. There we describe how the artificial cortical layers and the dynamics of the information processing could be modelled algorithmically.

3.3.1 A Conceptual Model

In this section, we present our conceptual model of the neocortex for impression generation in digital environments. As depicted in Figure 3.3, the model consists of three layers, outer layer, inner layer and core. The outer layer is the external-facing layer of the model, which is exposed to external data pertaining to different modalities. The outer layer represents the primary cortical areas of the human brain, which process the incoming sensory stimuli captured through sensory organs and transmitted though the respective neurons. In our conceptual model, different regions in the outer layer represent these different modality-specific primary cortical areas. The outer layer satisfies the first and fourth key features identified in Section 3.2.5, the ability to capture and represent multimodal data in a common representation and a mechanism to maintain memory/knowledge of the situation.

The inner layer of our conceptual model is analogous to the association areas in the human sensory system. The association areas receive inputs from different sensory modalities and are responsible for the fusion of co-occurring stimuli from them. Similarly, our theoretical model contains multiple regions in its inner layer with each of them associated to a particular sensory modality. Inter-modal associative connections, which connects different regions in the outer layer onto regions in the inner layer, represent the neural mechanism that connects primary cortical areas to the association areas in the human brain. Each region in the inner layer is connected to outer layer regions of all other modalities via these inter-modal associative connections. Moreover, each region in the inner layer is connected to its corresponding region in the outer layer via a direct intra-modal connection. For clarity, only such connections to the auditory modality are depicted in Figure 3.3. The inner layer of the conceptual model satisfies the second feature identified in Section 3.2.5, a mechanism for different aspects of an event to

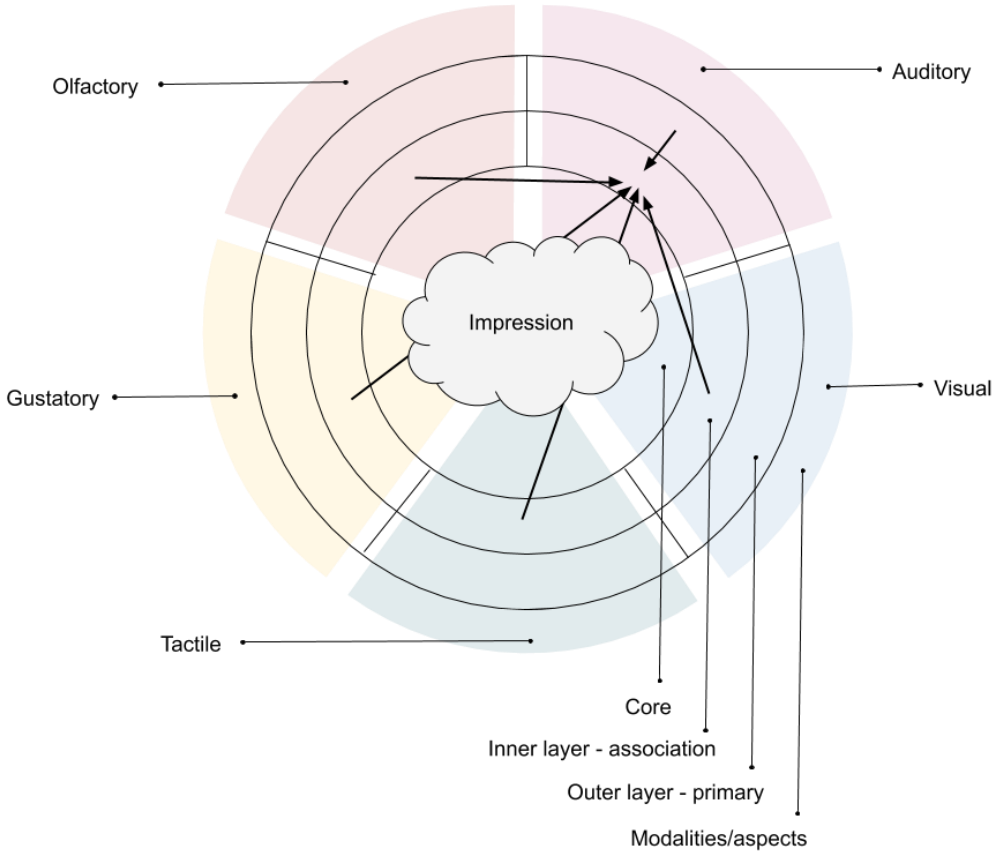

Figure 3.3 Conceptual model of neocortex for *impression* generation in digital environments

interact with/impact each other. This interaction and the ability to impact the perception of other modalities is facilitated by the inter-modal associative connections.

The innermost ‘core’ layer represents the *impression* generation mechanism facilitated by the previous two layers. This facilitates the third feature identified in Section 3.2.5, the ability to combine multiple aspects of an event/situation to form a coherent and wholistic *impression* about the situation.

3.3.2 An Architectural Model

In this section, we present the realization of the above conceptual modal into an architectural model. The architectural model consists of components that implement various sections of the conceptual model by generating the associated functionality. Figure 3.4 is an illustration of the proposed architectural model, presenting its components and how these components achieve the four key features of an artificial *impression* generation system identified earlier.

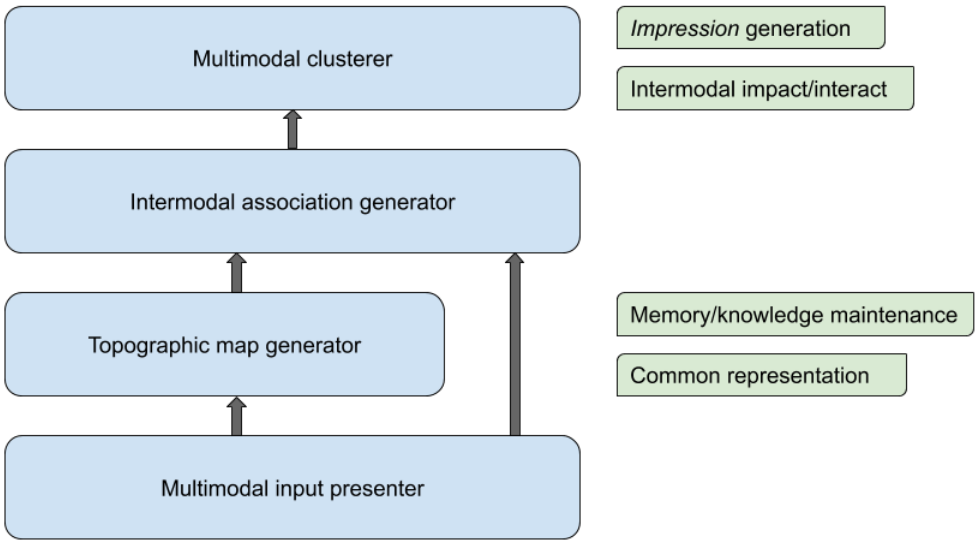

Figure 3.4 Proposed architecture for artificial impression generation

The multimodal input presentation module is responsible for the simultaneous presentation of co-occurring multimodal stimuli which is an input to both topographic map generation process as well as intermodal association mapping process. The topographic map generator generates the topographic maps which form the outer layer of the conceptual model. Irrespective of the

type of the modality (e.g., image, text, numerical) topographic maps are employed as the *common representation* of different modalities. Moreover, with topology preservation, the neuron and areas of topographic maps act as the mechanism to maintain knowledge/memory. Future inputs are evaluation against these neurons with a similarity function allowing to match and map with the stored knowledge/memory.

Using generated topographic maps and simultaneous multimodal input presentation, intermodal association generation module records the co-activation of neurons among modalities. Recorded co-activations are used to calculate the co-activation distributions of individual neurons.

Our goal is to organize cortical areas that process sensory inputs into meaningful clusters, and the inner layer of the conceptual model is the outcome of this clustering process. Multimodal clustering module, which performs clustering based on intermodal co-activations distributions, captures intermodal interactions and allow modalities to impact each other in forming the final multimodal clustering. The multimodal clusters formed over the cortical areas represent the organization of multimodal concepts which we consider as the basis of grounding sensory experiences in forming a coherent *impression* on them.

3.3.3 A Computational Model

As highlighted earlier, the family of SOM algorithms have been widely used to model the feature maps in the human brain. We proposed to use the topographic maps generated by the GSOM algorithm to model the outer layer of the conceptual model. The GSOM algorithm was chosen due to the dynamic structure adaptive nature of the algorithm to represent the input space being modelled. Based on the conceptual model, each region of the outer layer would be modelled by a topographic map using the GSOM algorithm. The GSOM algorithm receives modality-specific inputs which are used to iteratively build and condition the respective topographic map.

As the proposed architecture is to simulate the multimodal impression generation, input presentation is not carried out in isolation for individual modalities. Instead, the co-occurring multimodal inputs are presented to their respective regions in the outer layer in parallel. The inter-modal associative connections capture the co-activation of neuron pairs in different regions. These co-activation profiles can be used to unearth the intricate relationships between co-occurring modalities.

Our goal is to organize cortical areas that process sensory inputs into meaningful clusters, and the inner layer of the conceptual model is the outcome of this clustering process. With inter-modal associative connections that captured the co-activation relationships among neurons in different modalities, the clustering algorithm can utilize these relationships to enrich the clustering process. The co-activation relationships can be used to generate the probabilities of co-activation among the neurons in multiple modalities. If we consider two of the multiple modalities, modality X and modality Y, what this gives us is the probability distributions of activation on X and Y when a neuron or a set of neurons in the other modality is active. Our proposal is to incorporate these probability distributions into the clustering process for it to capture the relationships between the modalities.

The probability distributions can be used to identify similar neurons by analysing their similarity. That is, if neurons n_i and n_j in the topographic map, G_X of modality X have similar probability distributions of activation on the topographic map, G_Y of modality Y, we consider it as a signal that these two neurons are similar from the point of view of the modality Y. By factoring this information into the clustering process, we can effectively incorporate information from modality Y into the clustering of G_X. The similarity of probability distributions can be measured objectively, and a distance-based metric can be provided on how similar two probability distributions are. A distance-based metric on crossmodal similarity is particularly useful given that the connectivity-based clustering algorithms operate on distance-

based metrics. Figure 3.5 depicts the flow of the proposed computational model with all the major tasks.

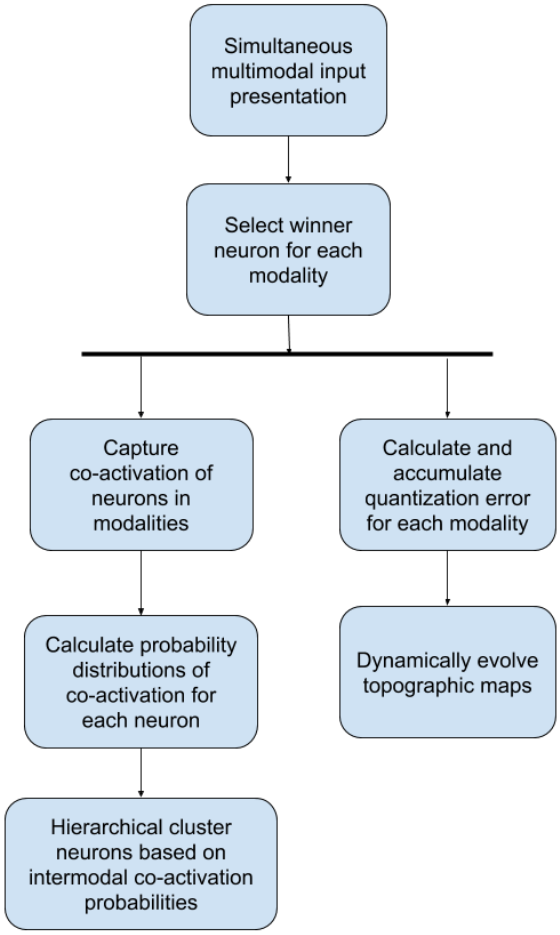

Figure 3.5 High-level flow diagram of the proposed computational model depicting the major tasks

3.4 Chapter Summary

In this chapter, we first laid out our premise of human sensation and perception. We discussed how humans form a coherent *impression* about the external world with the focus of how this can be implemented on artificial counterparts. In the process, we formally defined the term *impression* and drew upon neurobiological and physiological research in developing our understanding of the process.

Further, we closely examined the recent changes in data generation and collection brought in by Big Data revolution and how large, multimodal, multisource datasets create a *digital*

environment, a close approximation of the natural environment. We envisaged generating *digital impressions*, a human-like *impression* on the *digital environment*, and identified four key features of such a system.

Finally, we proposed an artificial model of neocortex for the *digital impression* generation consisting of a conceptual model conceptualizing the organization of the artificial cortical layers, neuronal connections, an architectural model describing the composition of the artificial model and a computational model outlining the algorithmic means of the model.

The next chapter presents the implementation and the validation of the model while an adaptation of the model for distributed computing, its implementation and demonstration on large datasets are presented in Chapter 6.

Chapter 4
Multimodal Sensory Fusion

In the previous chapter, we discussed how humans construct the state of the external environment from the sensory inputs by forming a coherent *impression* about the external world. Further, we discussed how AI algorithms could be facilitated to generate artificial *impressions* on *digital environments* and which biologically plausible algorithms could be utilised to achieve this. To this end, we proposed an artificial model, which consisted of a multi-layered conceptual model, an architectural model and a computational model, as theoretical contributions.

In this chapter, we discuss the implementation of the artificial model for *impression* generation. The outer cortical layer of the proposed conceptual model, which processes multimodal sensory inputs, is implemented with topographic maps generated by the GSOM algorithm. The multimodal interactions are modelled with a multimodal distance metric while a clustering algorithm based on it organizes cortical areas into meaningful clusters in the inner cortical layer. We demonstrate this artificial model for *impression* generation on an audio-visual dataset and experiment with various parameters of the model.

Further, highlighting the necessity of generating efficient representations from multimodal data sources in most online application scenarios, we present a distributed architecture for online multimodal sensory fusion.

Some of the work in this chapter has appeared in (Jayaratne et al., 2018).

4.1 Multimodal Sensory Fusion for Impression Generation

Categorization of sensory input into meaningful categories is of paramount importance for all animals in order to form awareness of their surroundings. Environment sensing and *impression* generation process have long been the subject of philosophical discussion, as highlighted in Chapter 3. The development of this capability in an unsupervised manner has amazed and puzzled most thinkers.

The perspective taken in this thesis is that environment sensing and *impression* generation process could be represented with a self-organization-based clustering process, which organizes cortical areas that processes sensory inputs into meaningful clusters. Since the same object/event is perceived by multiple sensory organs, this clustering process is incomplete without the fusion of sensations from multiple modalities. Hence, the *impression* generation is modelled as clustering of cortical areas of a particular modality while considering the influence of co-occurring stimuli on other modalities. Below, we present the algorithmic modelling of *impression* generation.

For algorithmic modelling, we define the *impression* generation as a multimodal clustering problem over the artificial cortical areas modelled by the GSOM algorithm in an unsupervised manner. The clustering should allow for meaningful organization of the inputs into categories without explicit knowledge such as the number of categories present in the input. The multimodal clustering algorithm is based on the hypothesis that observations of an event recorded over multiple modalities should bear similarities across them due to natural

regularities. We adapt the crossmodal clustering (Coen, 2005) to build upon and extend the topographic maps as a mechanism for fusing such information recorded across multiple modalities to achieve multimodal representation.

The proposed multimodal self-organizing neural architecture consists of topographic maps for each individual modality and inter-modality associative connections capturing the co-occurrence relationships among the modalities. The architecture falls under the post-perceptual binding paradigm as the individual modalities are processed first, and the fusion is carried out over these modality-specific topographic maps. Figure 4.1 outlines the overall layout of the proposed neural architecture.

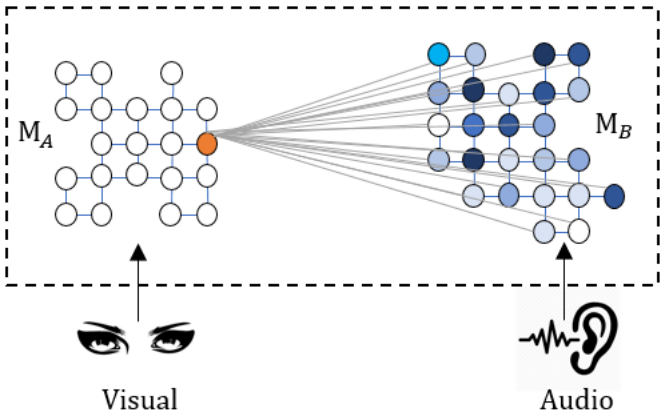

Figure 4.1: Proposed multisensory self-organizing neural architecture. M_A and M_B the modality-specific neuronal maps. Associative links running from a single neuron in M_A to the M_B are shown for clarity.

The modality-specific components are responsible for the processing of each individual modality and generating a topographic map using unsupervised machine learning. The individual topographic maps are trained using the GSOM (Alahakoon et al., 2000) algorithm and form representations of their respective modalities. For obtaining the multimodal representation over the individual topographic maps, we adapt the crossmodal clustering algorithm (Coen, 2005). The crossmodal clustering algorithm utilises the co-occurrence relationships captured on inter-modality associative connections to incorporate knowledge across modalities to cluster individual neural layers.

We utilise this algorithm to iteratively combine clusters of neurons until a certain stopping criterion is met. The initial clusters consist of single neurons, and the iterative process may end up clustering non-adjacent neurons based on the influence of other modalities. In essence, the clustering process forms a hierarchical clustering and allows us to inspect the hierarchy of clusters formed. Specifically, the clustering process is agglomerative or "bottom-up" where each neuron starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

4.1.1 A Multimodal Distance Metric

Without loss of generality, let us consider two topological maps, G_X and G_Y, trained using the GSOM algorithm pertaining to two co-occurring modalities, X and Y. The GSOMs consist of their modality neurons such that $G_X = \{x_1, x_2, x_3, \dots, x_m\}$ and $\mathrm{M_B} = \{\mathrm{b_1}, \mathrm{b_2}, \mathrm{b_3}, \dots, \mathrm{b_n}\}$. The co-occurrence relationship between the two modalities allows us to define the probabilities of co-activation among the neurons in the two modalities. We define the co-activation relationships using the notion of *Hebbian projection* of a cluster of neurons as below. Let $c \subseteq G_X$ be a cluster of neurons. We define the Hebbian projection of c onto G_Y, $H_X^Y(c)$, $H(c)$ for clarity, in (4.1) and this provide a spatial probability distribution of activation over G_Y whenever a neuron in c is active.

$$H(c) = [\Pr(y_1|c), \Pr(y_2|c), \dots, \Pr(y_n|c)] \tag{4.1}$$

$\Pr(y_i|c)$ is the probability of the neuron y_i being active while any neuron in cluster c is active, calculated as (4.2), where $h(c)$ is the number of times cluster c is active and $h(y_i, c)$ is the number of time cluster c and neuron y_i is active at the same time.

$$\Pr(y_i|c) = \frac{h(y_i, c)}{h(c)} \tag{4.2}$$

The weighted version of the Hebbian projection, $H_X^Y\omega(c)$, $H_\omega(c)$ for clarity, is defined in (4.3) with weights $\omega = [\omega_1, \omega_2, \omega_3, \dots, \omega_m]$ where $\sum \omega_j = 1$.

$$H_\omega(c) = [\Pr_\omega(y_1|c), \Pr_\omega(y_2|c), \dots, \Pr_\omega(y_n|c)] \tag{4.3}$$

where $\Pr_\omega(y_i|c) = h_\omega(y_i, c) \,/\, h_\omega(c) = \sum_{x \in c} \omega_x h(y_i, x) \,/\, \sum_{x \in c} \omega_x h(x)$.

Similar to how we define spatial probabilities of activation over G_Y we are able to define the same in the reverse direction, the Hebbian projection onto G_X. The notion of *reverse Hebbian projection* in (4.4) combines these two. The reverse Hebbian projection of cluster c onto Y, $\hat{\mathrm{H}}_X^Y(c)$, $\hat{\mathrm{H}}(c)$ for clarity, provides a probability distribution over G_X denoting which neurons in G_X are similar to neuronal cluster c from the point of view of modality Y.

$$\hat{\mathrm{H}}(c) = H_{H(c)}(G_Y) = [\Pr_{H(c)}(x_1|G_Y), \Pr_{H(c)}(x_2|G_Y), \dots, \Pr_{H(c)}(x_m|G_Y)] \tag{4.4}$$

The reverse Hebbian projection intuitively combines the activation probabilities of onward and backward directions. In essence, we utilise the co-activation relationships in the two neural layers to achieve information flow from modality Y to modality X.

The reverse Hebbian projection can be used to define a distant metric between clusters of neurons in G_X. As the reverse Hebbian Projections are n-dimensional probability distributions, we utilise the earth mover's distance as the distance metric. Consider two clusters $c_i, c_j \subseteq G_X$. The earth mover's distance between their reverse Hebbian projections are denoted as $d_{EMD}(\hat{\mathrm{H}}(c_i), \hat{\mathrm{H}}(c_j))$. The earth mover's distance accounts only for the distance from the view of the second modality. However, we would like to combine them both, the distance in its own modality with d_{EMD}. To measure the distance between c_i, c_j in their own modality, any suitable distance metric such as the Euclidian distance, d_{ED}, can be used. The combined multimodal distance, d_{MD} is defined in (4.5). The weighting parameter λ controls the relative weighting between the two distance metrics.

$$d_{MD}(c_i, c_j) = \sqrt{\lambda\left[d_{EMD}(\hat{\mathrm{H}}(c_i), \hat{\mathrm{H}}(c_j))\right]^2 + (1-\lambda)\left[d_{ED}(c_i, c_j)\right]^2} \tag{4.5}$$

The combined distance captures two perspectives on the distance between two clusters, one from the own modality and the other from the second co-occurring modality. Two clusters that

are close by in terms of distance in own modality, d_{ED}, maybe perceived distant apart by the second modality, d_{EMD}, and vice versa and the effect of this is intuitively captured by the combined distant metric, d_{MD}, which fuses information across modalities exploiting the co-occurrence relationships. We use the combined distances as a fused view to obtain a clustering.

Multimodal distance metric defined in 4.1 is based on the concept of co-activation of neurons in GSOMs pertaining to different modalities. This inherently brings in the assumption that all modalities are captured at the same frequency. When all modalities are not captured at the same frequency, some modalities need to be resampled (downsampled or upsampled) to match the others. Downsampling leads to some level of information loss, and synthetic upsampling would preserve most information. The common frequency for all modalities is very much application-specific, and the decision is best left with the users.

4.1.2 Multimodal Clustering

As highlighted earlier, our intention is to combine single neurons into larger cortical areas using a clustering process. The multimodal distance, d_{MD}, which intuitively calculates the distance between two clusters combining the perspective of the own modality as well as the perspectives of other co-occurring modalities, is our distance metric for the clustering process. This warrants for agglomerative hierarchical clustering where each neuron starts in its own cluster and pairs of clusters are merged as one moves up the hierarchy.

The agglomerative hierarchical clustering combines the closest cluster pair at each iteration and continues this iterative process till all the neurons are combined onto a single cluster. The multimodal distance, $d_{MD}(c_i, c_j)$, represents the distance between clusters c_i and c_j, $c_i, c_j \subseteq G_X$, from the perspective of the own modality as well as other modalities. At each iteration, clusters c_k and c_l, $c_k, c_l \subseteq G_X$ are merged if,

$$\forall i, j;\ d_{MD}(c_k, c_l) < d_{MD}(c_i, c_j);\ c_i, c_j, c_k, c_l \subseteq G_X \tag{4.6}$$

However, we are interested in stopping this process at an appropriate point, allowing for ideal clustering of cortical areas modelled by GSOM maps. The stopping criteria should ideally be based on d_{MD}. Such a criterion would allow for combining clusters c_i and c_j, $c_i, c_j \subseteq G_X$ if $d_{MD}(c_i, c_j)$ is sufficiently small or retaining them as separate clusters otherwise. However, the question remains, how small should $d_{MD}(c_i, c_j)$ be to be "sufficiently" small?

4.1.2.1 Self-distance

Coen (2005) defines the notion of *self-distance* as the threshold for being considered as sufficiently small for merging two clusters. The self-distance measures the internal crossmodal distance between data points inside a given cluster as opposed to measuring the distance between two clusters for which crossmodal distance was used thus far. This provides a measure of internal coherence of a given cluster and can be used as a threshold for merging two clusters by measuring the self-distance of the potentially merged cluster.

The self-distance of a cluster $c' \subseteq G_X$, $d_{self}(c')$, is defined as below.

$$d_{self}(c') = \frac{d_{MD}(\widehat{c_i}, \widehat{c_j})}{d_{MD}(c_i, c_j)} \tag{4.7}$$

Clusters $c_i, c_j \subseteq G_X$ are two clusters we consider for merging while $c' \subseteq G_X$ is the compound cluster created by the merging of c_i and c_j. Clusters $\widehat{c_i}, \widehat{c_j} \subseteq G_X$ are partitions of c' by fitting a linear orthogonal regression onto it.

The rationale of the self-distance is to identify whether we are merging two clusters representing the same concept or different concepts. When we consider merging c_i and c_j we have already calculated the multimodal distance between the two and found it to be the lowest among other potential cluster pairs to be merged; however, yet unsure as to whether they represent the same concept. Cluster c' is formed by the hypothetical merging of the two clusters. However, if c_i and c_j represent the same concept, we should observe similar multimodal distances between different partitions of c', despite the way we decide to partition them

subsequently. This is because they should co-occur in the same way across other modalities. On the other hand, if c_i and c_j represent the different concepts we should observe different multimodal distances between partitions of c'. To capitalise on the above, the formulation of self-distance uses the ratio between $d_{MD}(\widehat{c_i}, \widehat{c_j})$ and $d_{MD}(c_i, c_j)$ while using linear orthogonal regression for partitioning which minimises the multimodal distance between $\widehat{c_i}$ and $\widehat{c_j}$ if c_i and c_j represent different concepts.

Based on the above $d_{self}(c')$ would be close to 1 if c_i and c_j represent the same concept while it would be close to 0 if c_i and c_j represent the different concepts. We use $d_{self}(c') \geq 0.5$ as the threshold for merging two clusters.

4.1.2.2 Multimodal Clustering Algorithm

Below, we formalise the multimodal clustering algorithm, which is based on the agglomerative hierarchical clustering. The clustering process uses multimodal distance, d_{MD}, as the distance metric while an important distinction from agglomerative hierarchical clustering is the stopping criteria based on the self-distance, d_{self}. The algorithm takes GSOMs trained on different co-occurring modalities and the weighting parameter λ which controls the relative weighting between two distance components of d_{MD} as inputs and iteratively combines clusters in a greedy fashion until the stopping criterion is met. Below, algorithm 4.1 outlines the steps of the algorithm while we examine the steps in detail afterwards.

The multimodal clustering algorithm starts by creating a set of clusters, each containing a single neuron from the trained GSOMs (lines 1-5). The algorithm then iterates inside a loop (lines 6-27) merging clusters in a greedy fashion until no clusters were merged in a given iteration (lines 24-26).

Algorithm 4.1. Multimodal clustering algorithm

Input: G: GSOMs trained on different co-occurring modalities. λ: The weighting parameter which controls the relative weighting between d_{ED} and d_{EMD}.

1 **for** GSOM $G_x \in G$ **do**
2 **for** neuron $n_i \in G_x$ **do**
3 Create cluster $c_i \in G_x$ as $c_i = \{n_i\}$
4 **end for**
5 **end for**
6 **while** (**true**) **do**
7 merged ← **false**
8 Calculate d_{MD} for all pairs of clusters over all $G_x \in G$
9 **for** GSOM $G_x \in G$ **do**
10 Sort pairs of clusters $(c_i, c_j) \in G_x$ by $d_{MD}(c_i, c_j)$
11 **for** cluster pair $(c_i, c_j) \in G_x$ in sorted order **do**
12 **if** $d_{self}(c') \geq 0.5$ where $c' = c_i \cup c_j$ **then**
13 # Merge c_j with c_i
14 $c_i = c_i \cup c_j$
15 Remove c_j from G_x
16 merged ← **true**
17 **for** cluster $c_k \in G_x$ **do**
18 $d_{MD}(c_i, c_k) \leftarrow$ **min**$(d_{MD}(c_i, c_k), d_{MD}(c_j, c_k))$
19 **end for**
20 **break**
21 **end if**
22 **end for**
23 **end for**
24 **if** merged = **false then**
25 **break**
26 **end if**
27 **end while**

Each iteration starts by calculating the multimodal distance, d_{MD}, for all pairs of clusters within each modality $G_x \in G$ (line 8). The algorithm proceeds in a greedy manner, considering pairs of clusters of each modality sorted by d_{MD} for merging. If the cluster pair, (c_i, c_j), satisfies

$d_{self}(c') \geq 0.5$ where $c' = c_i \cup c_j$ then we proceed to merge the two clusters (lines 13-20). The merging involves transferring all neuron into a single cluster and removing the empty cluster. Moreover, we set the multimodal distance between the combined cluster and the rest of the clusters in the modality to the minimum of the multimodal distances of the original clusters. This is due to the fact the now we consider all the neurons in the combined cluster to represent the same concept, hence equivalent to each other. Now that the d_{MD} inside the modality under consideration have changed, we exit the inner loop (line 20) to start considering pairs of clusters for merging sorted by their d_{MD}. After each iteration, we recompute d_{MD} to propagate the effect of merges that took place so far.

4.2 Experiments

4.2.1 Dataset

We used Tulips1 audio-visual dataset (Movellan, 1995) for the experimentation to evaluate the accuracy of the proposed *impression* generation algorithm. The dataset consists of utterances from 12 speakers captured on both audio and video modalities. Each speaker utters the digits one through four twice, making the dataset consists of 96 instances in total. Figure 4.2 contains sample frames captured from one such utterance. For the audio modality, we extracted Mel-frequency cepstral coefficients (MFCCs) as their feature representation. The video consists of 100×75 pixels frames captured at a rate of 30 frames per second. We used the following six manually engineered features provided by Baldwin, Martin, and Saeed (1999) for each frame.

1. The width of outer corners of the mouth
2. The height of outer corners of the mouth
3. The width of inner corners of the opening of the mouth
4. The height of inner corners of the opening of the mouth
5. The height of the upper lip
6. The height of the lower lip

Since the utterances are of varying lengths, we used dynamic time warping (DTW) to optimally align sequences, so they are comparable. DTW retains the natural order of input to ensure the temporal dependencies within the sequence are maintained.

Figure 4.2: Sample frames captured from an utterance

Tulip1 dataset has been extensively used for evaluating multimodal information processing algorithms for multimodal representation and fusion in various application areas (Cheng et al., 2018; Galatas, 2014; Makkook, 2007; Movellan & Mineiro, 1998; Wysoski et al., 2010).

4.2.2 Experimental Plan

Below we outline the plan for evaluating the multimodal clustering algorithm. First, the experiments evaluate the quality of clustering generated by multimodal clustering compared that of unimodal clustering. Then we analyse the iterative cluster formation to understand the process better. Moreover, we focus on analysing the effect of parameters in the algorithm on the quality of fused representation.

4.2.2.1 Multimodal clustering vs Unimodal clustering

In this experiment, we are interested in quantifying the effect of the information brought in from other modalities by the multimodal clustering algorithm. To achieve this, we compared the quality of clusters generated by the multimodal clustering algorithm with those generated by a unimodal clustering algorithm. We implemented k-means clustering over the individual neuronal layers, which only utilised unimodal representations. While the multimodal clustering

did not have to know the number of cluster/classes in advance, this value had to be provided to the k-means algorithm with the parameter k being set to 4, the number of classes in the dataset.

4.2.2.2 Clustering process analysis

As highlighted above, the multimodal clustering process forms a hierarchical clustering and allows us to inspect the hierarchy of clusters formed. The clustering process is agglomerative or "bottom-up" where each neuron starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy. Dendrograms are used to illustrate the arrangement of the clusters produced by such hierarchical clustering processes. We use dendrograms to analyse the order of merging as well as the multimodal distance at which each merger takes place.

4.2.2.3 Coefficient of fusion

The parameter λ in the definition of $d_{MD}(c_i, c_j)$ in (4.5) controls the relative weighting between the earth mover's distance, $d_{EMD}(\hat{\mathrm{H}}(c_i), \hat{\mathrm{H}}(c_j))$, which accounts for the distance from the view of the second modality and the Euclidian distance, $d_{ED}(c_i, c_j)$, which accounts for the distance in their own modality for two clusters c_i and c_j. The parameter λ acts as the coefficient of fusion between the two components and has the range [0, 1]. At the extreme ends, when $\lambda = 0$ the multimodal distance only considers information from the second modality whereas when $\lambda = 1$ only the distance in their own modality is considered. While it is sensible to combine the information from both sources using λ in the range of (0, 1), it would be interesting to understand the effect of λ on the quality of multimodal clustering. In this experiment, we evaluate the quality of multimodal clustering with $\lambda = 0.0, 0.1, 0.2, \ldots, 1.0$.

4.2.3 Configurations

The following base parameters were used to conduct the above experiments except where we varied a particular parameter to understand the effect of the parameter. GSOM training in individual modalities was run for 100 growing phase iterations and 100 smoothing phase iterations allowing for sufficient growth and weight convergence. The starting learning rate, α_0, was set at 0.3 while the spread factor, *SF*, was set at 0.7. The starting neighbourhood radius

N_0 was set to 4 to include the immediate neighbour neurons. For the multimodal clustering, the relative weighting factor between d_{EMD} and d_{ED}, λ, was set at 0.5, giving both the distance in own modality and the distance based on the second modality similar importance. Experiments were carried out for five runs, and the averages of the metrics of these runs are presented.

4.3 Experimental Results

4.3.1 Evaluation Metrics

We have used a number of evaluation metrics to evaluate the quality of the multimodal clustering generated. Clustering quality evaluation metrics fall under two broad categories; internal and external. The internal metrics evaluate the clustering based on the data that were used to perform the clustering themselves. These measures usually reward high intra-cluster similarity and low inter-cluster similarity in order to generate coherent clusters. The external metrics, on the other hand, use class labels to evaluate the clustering generated. We note that the external metrics are suitable to evaluate the quality of clusters generated by multimodal clustering due to the following reasons. Multimodal clustering incorporates information from multiple modalities leading to cluster together two neurons in a given modality that are spatially apart given the other modalities perceive them as similar. Conversely, two neurons that are spatially close might be clustered separately if they are perceived distant from the view of other modalities. However, external cluster evaluation metrics measure the quality of clustering based on class labels which, in fact, is the desired ground truth.

External cluster evaluation metrics F1, which is the harmonic mean between precision and recall, Rand measure (Rand, 1971), Dice index (Dice, 1945), cluster purity, normalised mutual information (NMI) and internal cluster evaluation metric Davies–Bouldin index (DB-Index) (Davies & Bouldin, 1979) have been calculated to evaluate the quality of clustering.

The purity of a cluster is defined by assigning to the cluster, the most frequent class of inputs grouped to it and calculating the ratio between the number of instances of the most frequent class and the total instances assigned to the cluster. With C_i denoting the i^{th} cluster, $|C_i|$ its size

and $C_i^l \in C_i$ the number of instances of whose class is l, the purity of the cluster is C_i defined as,

$$\text{Purity}(C_i) = \frac{1}{|C_i|}\max_l(C_i^l) \tag{4.8}$$

Based on the above, the cluster purity measure is defined as the weighted average of the purities of each cluster, weighted by the number of instances in each cluster.

$$\text{Purity} = \frac{1}{N}\sum_{i=1}^{k}|C_i| \times \text{Purity}(C_i) \tag{4.9}$$

where k is the number of clusters and $N = \sum_{i=1}^{k}|C_i|$. Due to the lack of any penalisation for similar inputs being grouped into separate clusters, one of the limitations of the cluster purity measure is that a high purity measure can be obtained by the clustering algorithm being extra stringent leading to a high number of clusters. For instance, a perfect purity score can be achieved by clustering each input as a cluster.

The normalised mutual information (NMI) eliminates this limitation and examines both cluster quality and the number of clusters. Let $C = \{C_1, C_{2,\ldots,}C_k\}$ be the set of clusters and $L = \{L_1, L_{2,\ldots,}L_m\}$ be the set of class labels. The NMI is evaluated as follows,

$$NMI(C, L) = \frac{I(C, L)}{[H(C) + H(L)]/2} \tag{4.10}$$

where $I(C, L)$ is the mutual information calculated using,

$$I(C, L) = \sum_i \sum_j P\left(C_i \cap L_j\right) \log \frac{P(C_i \cap L_j)}{P(C_i)P(L_j)} \tag{4.11}$$

and H is the entropy calculated using,

$$H(C) = -\sum_i P(C_i) \log P(C_i) \tag{4.12}$$

$$H(L) = -\sum_{j} P(L_j) \log P(L_j) \tag{4.13}$$

Under the maximum likelihood estimates of the probabilities, the above probability terms can be evaluated using corresponding relative frequencies.

Evaluating clustering with accuracy measures such as precision, recall, and F1 takes the view of decision making in clustering. Two similar inputs grouped into the same cluster can be considered as an instance of true positive (TP) while a grouping of two dissimilar inputs into separate clusters can be considered as an instance of true negative (TN). Similarly, a false negative (FN) is the placement of similar inputs in separate clusters while a false positive (FP) is the placement of dissimilar inputs in the same cluster. Based on the above, cluster evaluation metrics precision (P), recall (R) and F1 measure can be defined as,

$$P = \frac{TP}{TP + FP} \tag{4.14}$$

$$R = \frac{TP}{TP + FN} \tag{4.15}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{4.16}$$

The Rand measure (Rand, 1971) is also based on the same view and can be considered as calculating the fraction of correct input assignments. The Rand measure is defined as,

$$RI = \frac{TN + TP}{TN + TP + FN + FP} \tag{4.17}$$

The Rand measure gives equal weights to true positives and true negatives. While this property might be undesirable for evaluating classification results, the same is not a problem for cluster quality evaluation. The Dice index (Dice, 1945) which doubles the weight of true positives and ignore true negatives are defined as,

$$DI = \frac{2 \times TP}{2 \times TP + FN + FP} \tag{4.18}$$

On the other hand, DB-Index (Davies & Bouldin, 1979), an internal cluster evaluation measure, is defined as a function of the ratio between the intra-cluster scatter and inter-cluster separation. That is, a lower value of the DB-Index is desired, and the metric rewards high cohesion within the clusters and high separation among clusters. DB-Index is defined as,

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i} \frac{S_i + S_j}{M_{i,j}} \tag{4.19}$$

where k is the number of clusters, S_i and S_j are measures of intra-cluster scatter of i^{th} and j^{th} clusters respectively and is $M_{i,j}$ a measure of inter-cluster separation between the same. S_i is defined as,

$$S_i = \left(\frac{1}{T_i}\sum_{j=1}^{T_i} \left|X_j - A_i\right|^p\right)^{1/p} \tag{4.20}$$

where given the i^{th} cluster, X_j is the j^{th} input mapped to it, T_i is its size and A_i is its centroid. Parameter p is usually set to 2, which makes the distance calculation Euclidean. $M_{i,j}$ is defined as,

$$M_{i,j} = \left(\sum_{k=1}^{n} \left|a_{k,i} - a_{k,j}\right|^p\right)^{1/p} \tag{4.21}$$

where n is the dimensionality and $a_{k,i}$ is the k^{th} element of A_i.

4.3.2 Results and Discussion

This section presents and discusses the results of the experiments outlined in the experimental plan. The results are presented in terms of the evaluation metrics identified in the previous section.

4.3.2.1 Multimodal clustering vs unimodal clustering

Table 4.1 presents the performance of the multimodal representation using the multimodal clustering compared to the clustering performance of unimodal representation using k-means clustering. It can be observed that precision has been significantly improved with multimodal clustering compared to k-means clustering for both audio and video modalities. Usually, a high gain in precision is at the expense of recall. However, in this case, the observed drop in the recall is marginal for both modalities. The F1-measure, the harmonic average of precision and recall, which accounts for both the measures, can be observed to have improved in multimodal clustering. Similarly, both cluster purity and NMI is higher for the multimodal clustering asserting the positive impact of incorporating the information from the second modality.

Table 4.1: Multimodal representation performance. Clustering quality of multimodal representation compared to that of unimodal representation

Metric	**Audio modality**		**Video modality**	
	Multimodal	**Unimodal**	**Multimodal**	**Unimodal**
Precision	**79.33%**	58.08%	**43.91%**	37.60%
Recall	91.79%	**95.71%**	56.99%	**58.68%**
F1	**84.12%**	71.59%	**49.33%**	45.30%
Purity	**86.67%**	68.54%	**62.08%**	52.29%
NMI	**0.84**	0.74	**0.41**	0.31
DB-Index	0.52	**0.45**	1.40	**1.18**

It can be observed that the DB-Index of multimodal clustering is higher than that of k-means clustering, while a lower value is desired. DB-Index is defined as a function of the ratio of the intra-cluster scatter and inter-cluster separation, favouring higher cluster cohesion and better cluster separation. In the case of multimodal clustering, the algorithm might add two neurons spatially apart in its own modality into a cluster if the neurons are perceived similar from the

view of the second modality. Conversely, two neurons that are spatially close might be separated into two clusters based on the view of the second modality. However, this is penalised in DB-Index as it is not designed for such a multimodal clustering scenario. On the other hand, metrics precision, recall, F1-measure, cluster purity and NMI are based on actual class labels and are able to effectively measure the clustering improvement brought in by the multimodal effect.

4.3.2.2 Clustering process analysis

Below, we use dendrograms to analyse the hierarchical nature of the multimodal clustering process. A dendrogram is a tree-like structure illustrating the arrangement of clusters produced by the hierarchical clustering process. The distance between merged clusters increases with the level of the merger. That is, the height of each node in the plot is proportional to the value of the intergroup dissimilarity, as measured by the multimodal distance, between its two children.

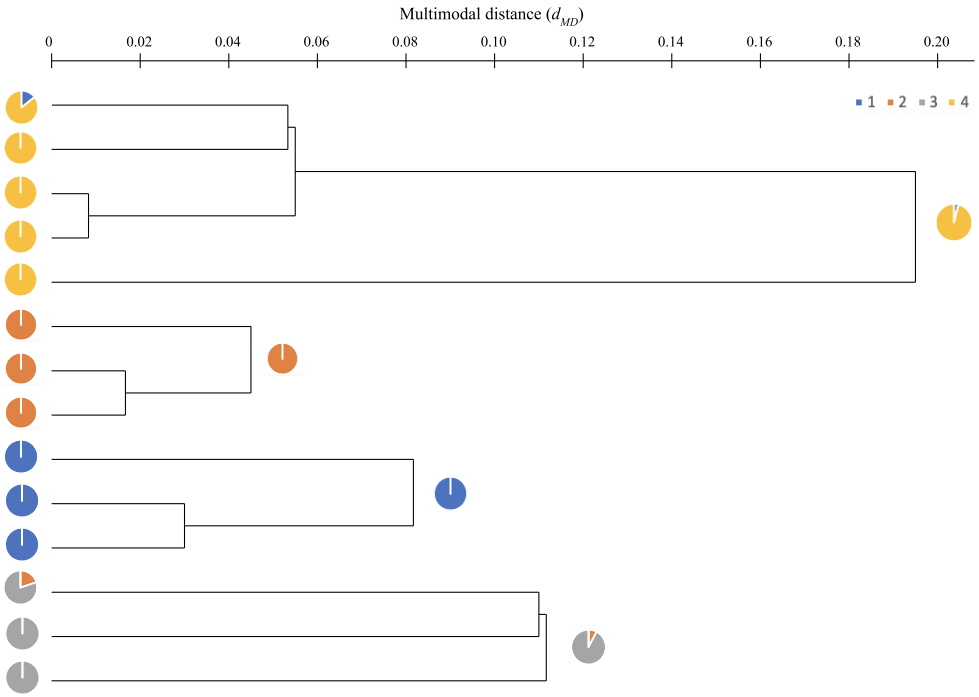

Figure 4.3 Hierarchical nature of the multimodal clustering process presented in dendrograms. The primary modality is audio while video is the secondary modality. The small pie charts illustrate the composition of clusters in terms of class labels at the beginning and end of the clustering process.

Usually, the hierarchical clustering process would continue until all the records are merged into a single cluster which would result in a single tree-like dendrogram. However, as the multimodal clustering impose an appropriate stopping criterion, the process results in multiple clusters. This corresponds to multiple dendrograms with one dendrogram for each final cluster.

Figure 4.3 illustrates the clustering process with audio as the primary modality with video modality as the secondary. As indicated by the small pie charts, two neurons at the initial level have records from multiple classes mapped to them. The multimodal clustering process seems to accurately combine clusters at each iteration with the clusters containing "impurities" combined last, only at higher multimodal distances.

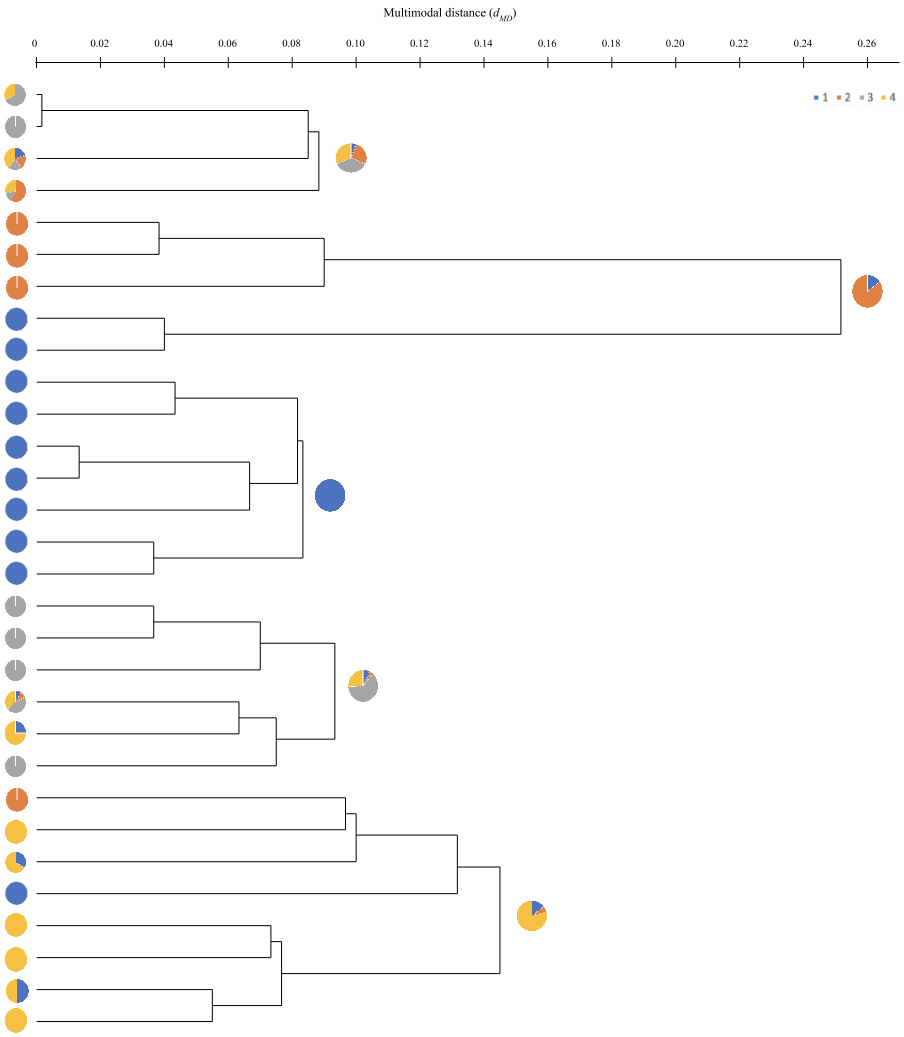

Figure 4.4 Hierarchical nature of the multimodal clustering process presented in dendrograms. The primary modality is video while the audio is the secondary modality.

Similarly, Figure 4.4 illustrates the clustering process with video modality as the primary modality while having audio modality as the secondary. The clustering process starts with the majority of the neurons "pure", having records pertaining to only one class mapped to them. Seven neurons have records from multiple classes mapped to them. Similar to the above, we can notice that "pure" clusters are merged at a lower multimodal distance while clusters with "impurities" are merged at a higher multimodal distance.

4.3.2.3 Coefficient of fusion

We varied the parameter λ in d_{MD} from 0 to 1 (inclusive of the extremes) by increments of 0.1 and evaluated its effect on the quality of multimodal clustering as measured by the evaluation metrics defined above. The audio modality was used as the main modality while the video modality was considered as the supplementary modality. Figure 4.5 illustrates the quality of multimodal clustering with varying values of λ.

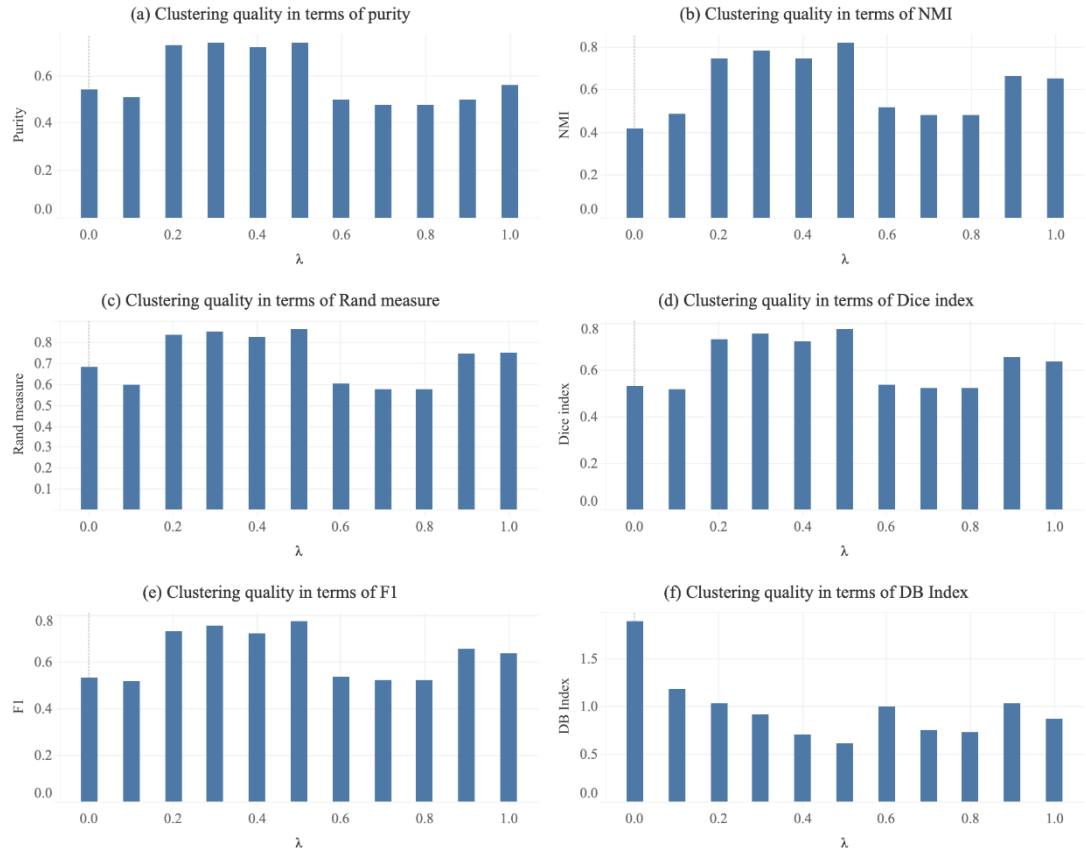

Figure 4.5 Multimodal clustering quality with varying values of λ

We observed highest clustering quality when $\lambda = 0.5$ for metrics NMI, Rand measure, Dice index and F1, where a higher value of the metric indicates higher clustering quality. For cluster purity, where a higher value of the metric indicates higher clustering quality, λ values 0.3 and 0.5 recorded the highest purity value of 0.74. On the other hand, DB-Index, where a lower value is desired, confirmed 0.5 as the optimal λ value with DB-Index value of 0.62.

All the cluster quality evaluation metrics suggest 0.5 as the optimal value for λ in this application. Parameter value $\lambda = 0.5$ provides equal relative weighting between the earth mover's distance, which accounts for the distance from the view of the second modality and the Euclidian distance, which accounts for the distance in their own modality in the distance measure. As with any hyperparameter, the value of λ is application-specific and would require tuning for the application at hand. However, $\lambda = 0.5$ would provide a good starting point for hyperparameter tuning.

4.4 A Distributed Architecture for Impression Generation

Most of the multimodal applications need to derive efficient representations from the multimodal sensory inputs to effectively perceive the environment. The efficient online fusion of data from multiple sensory modalities facilitates responding promptly when dealing with real-world situations. Moreover, the time taken to adapt/retrain the decision models in response to changes in the environment needs to be reasonable so that the decisions are not made with outdated models. For example, autonomous systems would need to be periodically retrained with latest training examples to maintain a degree of accuracy in dynamically changing environments. For such use cases, the efficiency and the scalability of the training algorithms are of paramount importance. Training algorithms that can be parallelised to take advantage of parallel and distributed computing are essential to training decision models with very large training datasets.

Below, we propose a distributed architecture for improving the efficiency and scalability of the *impression* generation algorithm in order to provide results under acceptable computing times. Chapter 5 presents a novel distributed GSOM algorithm as an implementation of the distributed self-organizing components of the proposed distributed architecture while chapter 6 presents the detail of multimodal clustering algorithm implemented for distributed computing.

4.4.1 Proposed Distributed Architecture

Figure 4.6 depicts the proposed distributed architecture for *impression* generation with major modules and the data flow among them. The proposed architecture consists of two major modules. The first module is the distributed GSOM training module implementing a distributed variant of the GSOM algorithm for training GSOM maps of individual modalities. The second module is the multimodal clustering module, which performs multimodal clustering over the modality specific GSOMs generated by the first module.

Distributed GSOM training module ingests multimodal training data and uses data parallelism to train modality specific GSOM maps. It should be noted that the proposal here is to parallelise GSOM training among multiple modalities as well as within individual modality. This would achieve greater parallelism unrestricted by the number of modalities compared to mere parallelism among multiple modalities. Moreover, the use of data parallelism (partitioning input data across processors) opposed to network parallelism (partitioning neurons or weights across processors) within each modality allows parallelising unrestricted by the GSOM map size. We identify three major tasks for the distributed GSOM training within each modality, 1) data partitioning, which partitions the dataset by a given criterion 2) distributed GSOM training, which trains a GSOM map on each partition parallelly and 3) merging GSOM maps, which generates a single topographic map for each modality from the individual GSOM maps.

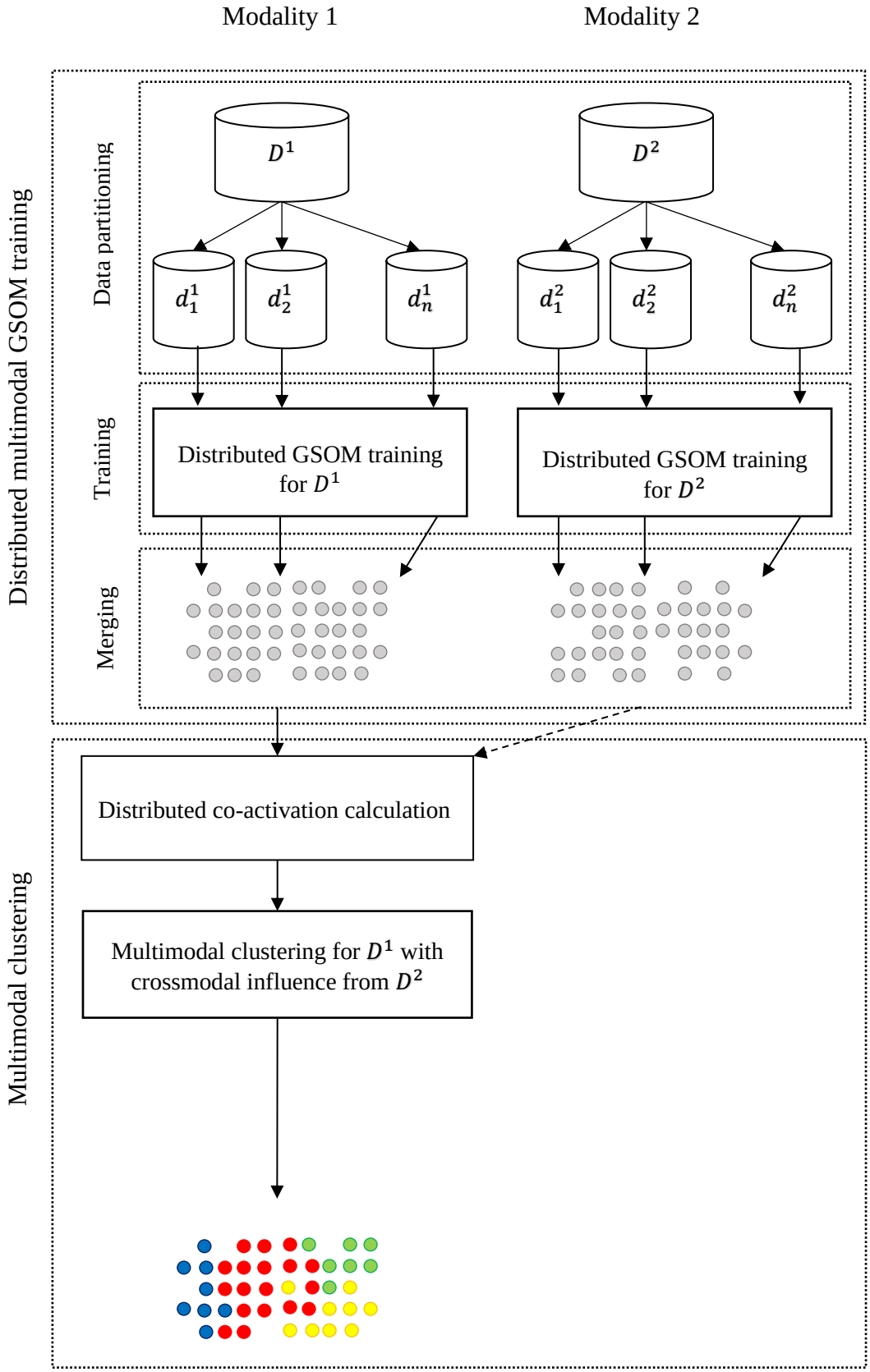

Figure 4.6: High-level architecture of the scalable fusion process, shown only for two modalities

The second major module is the multimodal clustering module, which performs multimodal clustering over the modality specific GSOMs generated by the first module. We identify two tasks for performing multimodal clustering, 1) co-activation distribution calculation, and 2) multimodal cluster generation. In the first task, multimodal data is presented to the GSOM maps in order to record co-activation among the neurons pertaining to different modalities which are used to calculate a co-activation distribution for each neuron. We propose to data parallelise co-activation calculation and implement for distributed computing. The second task generates multimodal clusters, and this is done by iteratively merging clusters based on the multimodal distance.

4.5 Chapter Summary

In this chapter, we presented the implementation details of the model for *impression* generation proposed in the previous chapter. The implementation adheres to a multi-layered conceptual model with the outer cortical layer, which receives multimodal inputs, modelled using topographic maps generated by the GSOM algorithm. The multimodal interactions are modelled using a multimodal distance metric which effectively combines information across modalities exploiting the co-occurrence relationships. The multimodal clustering algorithm, which uses multimodal distance, takes neurons of the outer cortical layer as input and organize them into meaningful categories without explicit knowledge such as the number of categories present in the input.

The proposed multimodal clustering algorithm was demonstrated with an audio-visual dataset. In the experimentation, we compared the quality of clustering generated by the multimodal clustering with that of a unimodal clustering to demonstrate cluster quality improvements. Hierarchical cluster merging was analysed to a better understanding of the process while further experiments were conducted to analyse the effect of hyperparameters.

Noting how most of the multimodal applications need to derive efficient representations from multimodal sensory inputs to perceive the environment effectively, we proposed a distributed

architecture for improving the efficiency and scalability of the *impression* generation algorithm in order to provide results under acceptable computing times. Chapter 5 continues on this quest, proposing a distributed GSOM algorithm and its implementations on multiple distributed computing paradigms to speedup computations.

Chapter 5
A Distributed GSOM Algorithm

The previous chapter concluded with a theoretical contribution, proposing a distributed architecture for *impression* generation. This chapter presents a novel distributed SOM algorithm as an implementation of the distributed self-organizing components of the proposed distributed architecture while the next chapter presents the detail of multimodal clustering algorithm implemented for distributed computing.

Current literature reports several studies towards scalable SOM algorithms to address the scalability issues of the SOM algorithm while being applied to Big Data environments. They can be classified into two groups, network parallelisation and data parallelisation. In network parallelisation, the number of nodes in the SOM becomes a constraint to scalability while data parallelisation requires a batch version of the SOM algorithm with a costly synchronisation step after each iteration. As a solution, this chapter describes a new data parallelised SOM algorithm which does not require the use of the batch SOM. The proposed algorithm processes data in parallel to generate multiple SOMs which are then projected together to a single mapping while preserving topological relationships of the original input data. We utilise the GSOM algorithm, a structure adapting variation of the SOM, which results in further processing improvement.

The new algorithm is adapted to three contemporary distributed computing platforms, Hadoop, Spark and Hama and empirical evaluations which demonstrate super-linear speedup compared to the serial SOM using several benchmarking and real-life data sets are reported in this chapter.

5.1 Introduction

With large data volumes, using the SOM algorithm for data processing become increasingly difficult due to the algorithm's inherent time complexity, $O(N)$, where N is the data size. With recent changes in social, technology and economic forces that created the Big Data phenomenon, the amount of data available has grown exponentially in all fields. In such instances, SOM computations on a single computing resource are resource-intensive, which has been a significant constraint in the application of the SOM in Big Data environments. This limitation can be addressed by adapting SOM learning and topology preservation into parallel and distributed computing environments where processors function independently without sharing or synchronisation. Many research endeavours targeting the development of scalable SOMs using parallel and distributed computing environments have been reported in the literature. Such research could be categorised in two main directions as network parallelisation-based and data parallelisation-based SOMs. Since network parallelisation is constrained by the number of nodes in the SOM, data parallelisation has been the more predominant and widely accepted direction. The SOM in its original form cannot be adapted to data parallelisation since weight adaptation across multiple data partitions needs to be synchronised and as such a batch processing version of the SOM is used in the data parallelised algorithms.

The proliferation of distributed computing platforms makes it convenient and economical to conduct distributed computations. Several of the proposed parallel SOM algorithms, both network and data parallelised, have been adapted to distributed computing platforms such as Apache Hadoop and Apache Spark. Although significant improvements in processing speed have been reported, the following key limitations constrain the use of distributed SOMs with large data volumes:

1. Network parallelised SOMs are restricted by maximum processors equal to the number of SOM nodes, resulting in limiting the processing improvement as well as under-utilisation of GPU capacity
2. Data parallelised SOMs require batch SOM where weight synchronisation after each iteration becomes a major bottleneck on speed
3. It has been reported that the batch version of the SOM results in the reduction of final map quality (Fort et al., 2002).

As such, there is still further advancement required on the existing work on parallelisation of the SOM, which could then be adapted into the distributed computing platforms. The research reported in this chapter describes a new distributed SOM algorithm to address the above limitations with adaptation into three well known distributed computing platforms for practical utilisation. A dynamic and adaptive version of the SOM called the Growing SOM (GSOM) is used due to its proven processing advantages over the original SOM. Data parallelisation was chosen as the more suitable direction for very large data volumes since this enables parallelisation unrestricted by map size. The chapter reports on design and development of the Distributed GSOM algorithm and its adaptation to three well-established distributed computing paradigms, MapReduce (Dean & Ghemawat, 2008), Bulk Synchronous Parallel (BSP) (Valiant, 1990) and Resilient Distributed Dataset (RDD) (Zaharia et al., 2012). Each adaptation is demonstrated on its respective platform, MapReduce on Apache Hadoop, BSP on Apache Hama and RDD on Apache Spark.

5.2 Background

This section describes distributed computing paradigms and related past research, which make up the foundation on which the proposed algorithm was developed. This discussion further highlights the gap in distributed SOM research and algorithms and justify the direction of work resulting in the proposed distributed GSOM algorithm.

5.2.1 Distributed Computing Paradigms

Below we present a brief account of three popular distributed computing paradigms. These paradigms have been utilised for implementing several of the existing distributed SOM algorithms, and the proposed Distributed GSOM has been implemented on all paradigms. Most of the known distributed adaptations of SOM have been utilised in the MapReduce (Dean & Ghemawat, 2008) computing paradigm which is designed for processing large datasets on a cluster of commodity hardware. MapReduce works by allowing the user to specify a map function and a reduce function, and the underlying system takes care of parallelising the processing to multiple computing nodes. Mapper nodes are responsible for processing a subset of the dataset and outputting intermediate results as key-value pairs. The MapReduce framework delivers these intermediate pairs to reducer nodes and guarantees that all the pairs with the same key are delivered to the same reducer. The reducer is responsible for aggregating the intermediate key-value pairs for a single key and outputs the aggregated value against the key as yet another key-value pair.

Apache Hadoop is the widely used open-source implementation of the MapReduce paradigm. Hadoop facilitates MapReduce computation by providing a number of essential services such as job scheduling, a distributed file system, and fault tolerance. It is designed to scale from one node to thousands of computing nodes offering local computing and storage. Hadoop's YARN module is responsible for job scheduling and cluster resource management while Hadoop Distributed File System (HDFS) module provides a distributed file system for high-throughput access to application data. Although widely used, accessing partial results stored in HDFS is a significant overhead in performance, especially when iterative steps are required. Bulk Synchronous Parallel (BSP) has been proposed as a solution for such situations.

Bulk Synchronous Parallel (BSP) is a generic distributed computing paradigm capable of operating on a cluster of computing nodes. Usually, a BSP algorithm consists of a series of *supersteps*, each having three components, concurrent computation, communication, and barrier synchronisation. In a *superstep,* each computing node may perform its computations

using data available to it locally and the nodes may exchange data among them by sending messages. The computation and communication are asynchronous and may overlap as well. Barrier synchronisation is used to synchronise all nodes such that a node would wait until other nodes have reached the barrier. Barrier synchronisation concludes a *superstep* and a BSP algorithm may consist of one or more such *supersteps*. Apache Hama is a popular implementation of the BSP paradigm and provides the necessary infrastructure for communicating with peer nodes and performing synchronisation. Hama is closely related to Hadoop in that Hama uses HDFS as the distributed storage.

Apache Spark implements the Resilient Distributed Dataset (RDD) (Zaharia et al., 2012), which is an immutable, fault-tolerant distributed dataset divided into logical partitions. Due to in-memory processing, Spark has reported significant improvement in processing speed compared to Hadoop MapReduce. RDDs can be created by deterministic operations on either data or other RDDs and computations on these partitions may be carried out on different nodes in the cluster. Spark provides a large number of operations such as `map`, `reduce`, `reduceByKey`, `collect`, and `filter` to manipulate and transform the RDDs. RDDs are fault-tolerant, such that, Spark uses the lineage of operations to regenerate an RDD in the case of a failure. Spark also uses HDFS as distributed storage and can be configured to use YARN for job scheduling and cluster management.

Although each platform has reported advantages in certain situations, no clear 'winner' for all data types and situations has been reported. Although Spark is said to have significant speed improvement due to in-memory processing, Hadoop is reported to perform better when the data set is larger than available memory. With Hama, the construction of iterative workloads is simpler as the same logic could be re-executed in a series of super steps. User-friendliness of the user interfaces as well as faster learning curves have been reported as criteria when comparing these platforms. Since each platform seems to have its advantages according to different scenarios, the proposed Distributed GSOM algorithm was adapted to all three as described in section 5.4.

5.2.2 Parallel and Distributed Models of Self-Organizing Maps

The parallel and distributed algorithms targeted at addressing scalability issues of the SOM learning can be categorised based on parallelism as network parallelised (partitioning neurons or weights across the processors) and data parallelised (partitioning the input data across processors). A number of such algorithms have been proposed and goes as far back as 1990 (Huntsberger & Ajjimarangsee, 1990). However, more recent attempts can be identified under two common themes and direction, (1) GPU based network parallelised models and (2) batch SOM algorithm-based data parallelised models.

Network parallelism is common in approaches targeted towards GPUs and specialised hardware such as Very-Large-Scale Integration (VLSI) chips. The parSOM algorithm (Rauber et al., 2000) is parallelised at the neuron level, i.e. each partition contains a subset of neurons of the complete map, which is further improved for distributed memory systems by Tomsich, Rauber, and Merkl (2000). More recent implementations of network parallel SOM algorithms focus on GPUs for parallelism. (Moraes et al., 2012) propose such an implementation based on CUDA. Moraes et al. used Single Instruction, Multiple Data (SIMD) paradigm for distance calculations and for finding the BMU using parallel reduction. Another GPU-based implementation of the SOM proposed by Zhongwen, Zhengping, and Xincai (2005) assigns a single neuron or a group of neurons to each processor of the GPU for parallel operation. The main limitation of network parallelised algorithms is that their speedup is limited by the number of neurons in the map if neurons are processed in parallel or the dimensionality of data if dimensions are processed in parallel. Due to the above limitation, these approaches do not scale well to large datasets or have limited applicability for specialised purposes such as handling high dimensional data.

Batch SOM algorithm based data parallelism is the most predominant variant used for parallelising the SOM algorithm. The original online learning algorithm for SOM cannot be parallelised in its original form since weights of neurons in the map have to be updated for each input vector (Garabato et al., 2015). In the modified batch variant of the SOM algorithm (Mulier & Cherkassky, 1995) weights are updated only once per each iteration, and most data

parallelisation attempts are based on this variant. In these attempts, the current map is published to all worker nodes, and the training data is distributed among them for each iteration. The worker nodes present each input vector from the training data assigned to them, record the winning neuron and calculate the necessary information for the weight update. At the end of the iteration, this information is transmitted to the master node, which performs the weight updates of the neurons in the map.

This workflow has been ported to the MapReduce paradigm and implemented for distributed computing frameworks Apache Hadoop (Garabato et al., 2015; Weichel, 2010) and in recent works, Apache Spark (Koutsoumpakis, 2014; Malondkar, 2015; Sarazin et al., 2014). Weichel (2010) used two MapReduce jobs, and two-fold emit in reduce phase to implement the calculation of an iteration. This was improved by Garabato et al. (2015) when they implement the same with only one MapReduce job per iteration. Another MapReduce-based implementation of SOM, which utilises the batch learning algorithm, is proposed in (Wittek & Darányi, 2012) and (Wittek & Darányi, 2013). The algorithm is implemented for GPUs in MR-MPI, which is a light-weight framework that allows to program GPUs with the MapReduce paradigm.

The main drawback of reported research on the batch variant of the SOM is the computationally expensive synchronisation step at the end of each iteration. Parallelisation is achieved only within iterations, and workers need to synchronise and communicate results to the master node at the end of each iteration. Moreover, it is noted in the literature that the quality of the map produced by the batch algorithm is inferior to that of the online algorithm (Fort et al., 2002). In Hadoop, each iteration is implemented as a MapReduce job, and the intermediate results are shared among jobs by writing them to an external stable storage system such as HDFS which is extremely slow compared to in-memory sharing. In addition, the number of outputs generated by the map phase of the MapReduce job is the mathematical product of the total number of input vectors and the total number of neurons in the map. In a distributed system, this can slow down calculation for large datasets. Sarazin et al. (2014) show that the ratio between the number

of training samples and computation time drops when the size of the training set exceeds 10^6. This is attributed to the large number of outputs the map phase generates, and thereby such approaches are not scalable beyond 10^6 training samples.

A different approach to data parallelism has been adopted in Scalable GSOM algorithm (Zhai et al., 2006). It uses a two-level growing phase, where the first level divides data among independent parallel processing units and the second level trains GSOMs on each data partition. However, using a GSOM training level to partition data may limit the benefits of distributed processing. The method also lacks a final combined map which is the hallmark feature of the SOM algorithm and an essential feature for the visualisation of neighbourhood relationships in the data. Gorgonio and Costa propose another approach, the partSOM (Gorgonio & Costa, 2008a, 2008b) which uses vertical data partitioning and assigns a subset of attributes to each node for SOM training and uses another SOM to combine individual SOMs. The main drawback is that the parallelisation of this approach is limited by the number of attributes in the dataset. Given the variety of approaches adopted for SOM parallelisation, it is useful to summarise all in terms of key features (Table 5.1).

The speedup of a network parallelised algorithm is restricted by the number of neurons or the dimensionality of data. Hence, GPU based network parallelised models would not scale well to process large datasets available in the Big Data era or have limited applicability for specialised purposes. On the contrary, the speedup of a data parallelised algorithm is proportional to the input data size and would be more suitable for large datasets. However, as discussed above, the online SOM algorithms cannot be parallelised in the original form and as such data parallelism has to be based on the batch variant which is inherently impacted by (a) the delayed synchronisation step in each iteration, (b) the large number of intermediate outputs emitted and (c) the inferiority of quality of the batch learning algorithm. It is pertinent to conclude this subsection by highlighting the need for a novel data parallelised SOM algorithm that demonstrates effective scalability on Big Data volumes.

Table 5.1: Parallel and distributed SOM algorithm comparison

Author(s), Year	Largest Dataset Size	Map Size	Parallelism		Memory		Processor		Variant	
			Data	Network	Distribute	Shared	CPU	GPU	Batch	Online
Lawrence et al., 1999 (Lawrence et al., 1999) Data parallelised batch SOM algorithm with calculations further improved for sparse data. Implemented in MPI paradigm.	128,282 rows (14 dims)	64 neurons	✓		✓		✓		✓	
Weichel, 2010 (Weichel, 2010) Data parallelised batch SOM algorithm ported to MapReduce paradigm and implemented with Hadoop. Uses two MapReduce jobs per iteration.	2,500 rows (20 dims)	440 neurons	✓		✓		✓		✓	
Sarazin et al., 2014 (Sarazin et al., 2014) Data parallelised batch SOM algorithm ported to MapReduce paradigm and implemented with Spark. Proposes a way to reduce the large number of outputs in the map phase.	100 million rows (2 dims)	100 neurons	✓		✓		✓		✓	
Koutsoumpakis, 2014 (Koutsoumpakis, 2014) Data parallelised batch SOM algorithm implemented with Spark.	1,244 rows (# dims not given)	25 neurons	✓		✓		✓		✓	
Garabato et al., 2015 (Garabato et al., 2015) Data parallelised batch SOM algorithm ported to MapReduce paradigm and implemented with Hadoop.	10.36 million rows (6 dims)	900 neurons	✓		✓		✓		✓	
Malondkar, 2015 (Malondkar, 2015) Data parallelised batch SOM algorithm ported to MapReduce paradigm and implemented with Spark for individual SOM training of Growing Hierarchical SOM algorithm.	8,124 rows (22 dims)	70 neurons	✓		✓		✓		✓	
Gorgonio and Costa, 2008 (Gorgonio & Costa, 2008b) Data parallelised (with vertical data partitioning) SOM algorithm which assigns a subset of attributes to each node for SOM training and uses another SOM to combine individual SOMs.	699 rows (10 dims)	132 neurons	✓		✓		✓			✓
Gorgonio and Costa, 2008 (Gorgonio & Costa, 2008a) Combines the above algorithm with k-mean clustering for cluster analysis in distributed databases.	699 rows (10 dims)	Not given	✓		✓		✓			✓
Wittek and Darányi, 2012 (Wittek & Darányi, 2012) Data parallelised batch SOM algorithm ported to MapReduce paradigm and implemented with MR-MPI for distributed GPUs.	84,283 rows (200 dims)	100 neurons	✓		✓			✓	✓	
Wittek and Darányi, 2013 (Wittek & Darányi, 2013) Improving the above algorithm for text mining tasks on distributed GPUs.	84,283 rows (100 dims)	100 neurons	✓		✓			✓	✓	

Takatsuka and Bui, 2010 (Takatsuka & Bui, 2010) Data parallelised batch SOM algorithm implemented with OpenCL for GPUs.	2,000 rows (3 dims)	92 neurons	✓			✓		✓	✓	
Rauber et al., 2000 (Rauber et al., 2000) Serial SOM algorithm parallelised using network partitioning where each partition contains a subset of neurons of the map.	420 rows (4012 dims)	150 neurons		✓		✓	✓			✓
Tomsich et al., 2000 (Tomsich et al., 2000) The above algorithm improved for distributed memory systems.	420 rows (4012 dims)	150 neurons		✓	✓		✓			✓
Zhongwen et al., 2005 (Zhongwen et al., 2005) Serial SOM algorithm parallelised using map partitioning targeting GPUs.	80 rows (# dims not given)	262,144 neurons		✓		✓		✓		✓
Moraes et al., 2012 (Moraes et al., 2012) Serial SOM algorithm with some of the subtasks parallelised using SIMD paradigm. Implemented using CUDA targeting GPUs.	# rows not given (1,000 dims)	16,384 neurons		✓		✓		✓		✓

5.3 Proposed Distributed GSOM

In this section, we present the Distributed GSOM algorithm, which addresses the limitations highlighted in the existing parallel and distributed SOM algorithms. The algorithm uses data parallelism to be able to process large datasets available in the Big Data era. Moreover, the algorithm is based on the online SOM algorithm.

5.3.1 Comparison with Batch SOM Based Approaches

Figure 5.1 presents the execution flow of the batch variant-based SOM parallelisation, and Figure 5.2 highlights how the proposed Distributed GSOM algorithm is different from the batch variants. The proposed algorithm (Figure 5.2) does not have to synchronise after each iteration and does not suffer from a large number of intermediate outputs. Moreover, the use of the online variant preserves the quality of the final map.

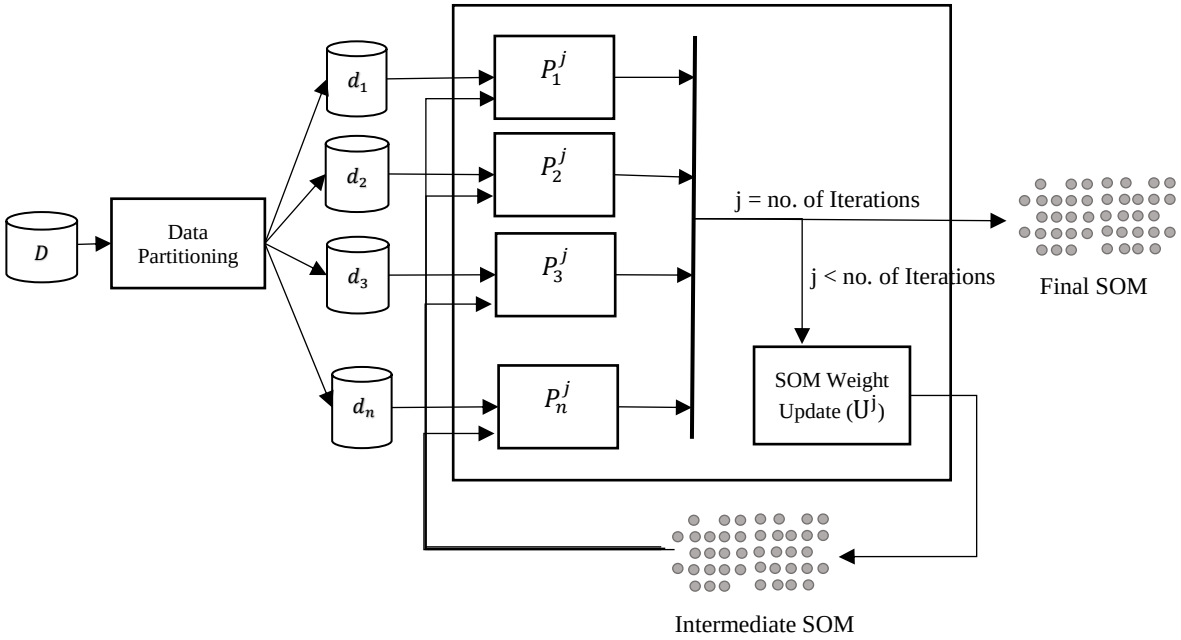

Figure 5.1: Common execution flow of batch variant based parallelisation efforts. d_i is the i[th] data partition, P_i^j is the processing on i[th] partition on j[th] iteration and U^j is the weight update after j[th] iteration.

5.3.2 Distributed GSOM Algorithm

A high-level outline of the Distributed GSOM algorithm is presented in Algorithm 5.1, and the flow is illustrated in Figure 5.2. Multiple partitioning techniques can be utilised for the first step, including random partitioning, class-based partitioning and high-level clustering-based partitioning. The number of partitions is a parameter to the algorithm and is usually decided based on the number of computing resource available. We have used random partitioning and ruled out class-based partitioning since the class may not be readily available in the dataset and high-level clustering due to the associated computing cost. Next, individual GSOMs are trained in parallel on each partition using the GSOM algorithm.

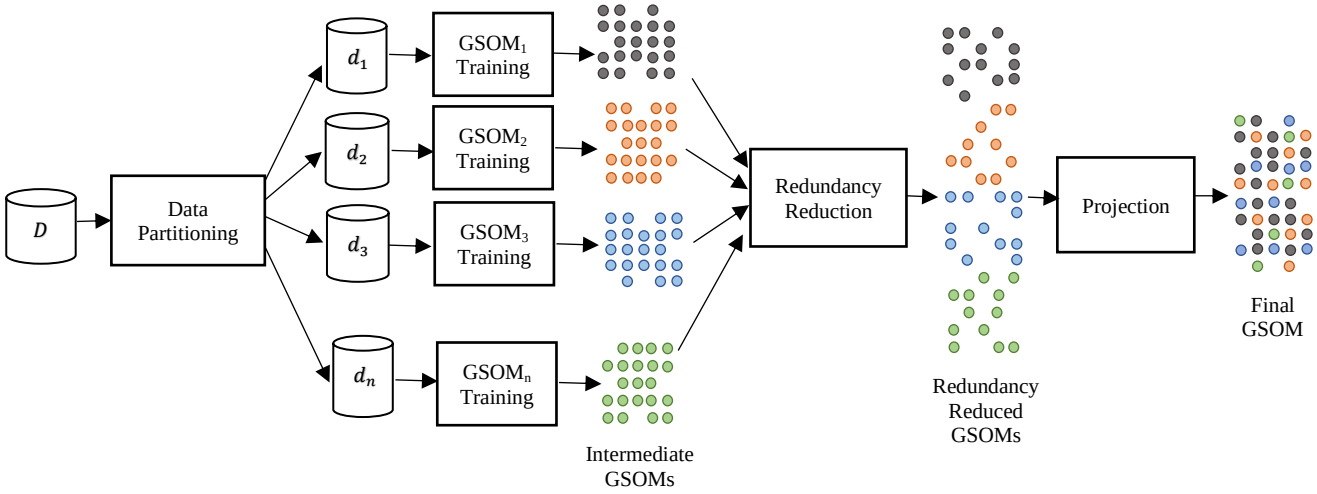

Figure 5.2: High-level outline of Distributed GSOM algorithm

Algorithm 5.1. The high-level outline of Distributed GSOM algorithm	
1	Split the dataset into partitions
2	Train a GSOM on each partition in parallel
3	Remove redundant neurons across partitions
4	Merge by performing Sammon's projection on neurons in trained GSOMs

5.3.2.1 Redundancy Reduction

One of the issues in partitioning the dataset and training GSOMs on each of them is that, when the actual groupings are spread among computing nodes, the individual GSOMs could contain similar neurons leading to redundant neurons in the merged map. This could lead to high processing times for the merging process, which employs Sammon's projection. Redundancy reduction step of the algorithm addresses this issue by removing such redundant neurons. We identify that the redundancy reduction can also be performed in a distributed manner where redundancies among smaller subsets of maps can be removed simultaneously. Specifically, the reduce operation of functional programming - which applies some function to pairs of objects of similar type and produces a single object of the same type as the output - can be applied for redundancy reduction. Here, the reduce operation will consume two maps at a time to output a

single map with redundancies among them removed. Such a reduction operation requires the reduce function to be commutative and associative.

Algorithm 5.2. The redundant neuron reduction algorithm

Input: P: GSOMs trained on training data partitions
$d \leftarrow 0, n \leftarrow 0$
for partition $P_i \in$ P **do**
  for neuron $n_{i,j} \in P_i$ **do**
    for partition $P_k \in$ P such that $i \neq k$ **do**
      $n_{k,l} \leftarrow$ GetBMU$(w_{i,j}, k)$
      $E_{k,l}^{i,j} \leftarrow \sum_{x=0}^{N_{i,j}} |w_{k,l} - I_{i,j}[x]|$
      $E_c \leftarrow E_{i,j} + E_{k,l}$ {Total current error}
      $E_1 \leftarrow E_{i,j} + E_{i,j}^{k,l}$ {Total $n_{i,j}$ error}
      $E_2 \leftarrow E_{k,l} + E_{k,l}^{i,j}$ {Total $n_{k,l}$ error}
      if $E_c > E_1$ AND $E_2 > E_1$ **then**
        remove $n_{k,l}$
        $d \leftarrow d + |w_{i,j} - w_{k,l}|$
        $n \leftarrow n + 1$
      else if $E_c > E_2$ **AND** $E_1 > E_2$ **then**
        remove $n_{i,j}$
        $d \leftarrow d + |w_{i,j} - w_{k,l}|$
        $n \leftarrow n + 1$
      end if
    end for
  end for
end for
$RI = e^{SF} \times d/n$
for partition $P_i \in$ P **do**
  for neuron $n_{i,j} \in P_i$ **do**
    if IsNonHitNeuron$(n_{i,j})$ **then**
      for partition $P_k \in$ P such that $i \neq k$ **do**
        for neuron $n_{k,l} \in P_k$ **do**
          if IsNonHitNeuron$(n_{k,l})$ **then**
            if $|w_{i,j} - w_{k,l}| \leq RI$ **then**
              remove $n_{k,l}$
            end if
          end if
        end for
      end for
    end if
  end for
end for

where,

$n_{i,j}$ The neuron j in partition i

$w_{i,j}$ The weight vector of $n_{i,j}$

$E_{i,j}$ The total quantisation error for $n_{i,j}$ for $I_{i,j}$

$E_{i,j}^{k,l}$ The total quantisation error for $n_{i,j}$ for $I_{k,l}$

RI Redundancy index

$\text{GetBMU}(w_{i,j}, k)$ returns the closest neuron to $w_{i,j}$ in partitioned network k.

$\text{IsNonHitNeuron}(n_{i,j})$ returns true if the neuron $n_{i,j}$ does not have any training data mapped.

5.3.2.2 Projection for Final GSOM

Finally, Sammon's projection (Sammon, 1969) is used to merge the individual maps due to its ability to preserve the topological ordering. The accuracy of using Sammon's projection in the merging phase in terms of topology preservation and clustering results have been demonstrated in earlier work (Ganegedara & Alahakoon, 2011). Sammon's projection algorithm is based upon point mapping of higher dimension vectors to lower dimensional space such that the inherent structure of the data is preserved. It does so by attempting to minimise Sammon's stress, E, in (5.1), over a number of iterations.

$$E = \frac{1}{\sum_{i<j}[d_{i,j}^*]} \sum_{i<j}^{N} \frac{[d_{i,j}^* - d_{i,j}]^2}{d_{i,j}^*} \tag{5.1}$$

where, $d_{i,j}^*$ is the distance between vectors X_i and X_j in the higher dimensional space and $d_{i,j}$ is the distance between the corresponding vectors Y_i and Y_j in the lower dimension space. Sammon's projection has the time complexity of $O(n^2)$, where n is the number of vectors. In the case of merging GSOMs trained in parallel, this would be the total number of neurons in individual maps. Hence, it is important that the redundancy reduction mechanism removes any redundant neurons in individual maps.

We adapt the algorithm to three distributed computing paradigms, MapReduce, Bulk Synchronous Parallel (BSP) and Resilient Distributed Dataset (RDD) and implement these adaptations on their respective platforms Apache Hadoop, Apache Hama and Apache Spark.

The Distributed GSOM algorithms presented here have their roots in earlier work, the spilt and merge GSOM algorithm (Ganegedara & Alahakoon, 2011, 2012). Ganegedara and Alahakoon (2011) assessed the topology preservation of Sammon's projection with smaller datasets. The results indicate that GSOM with Sammon's projection preserved topology better than standard GSOM algorithm and concluded that "maps generated using Sammon's projection have better topology preservation leading to better results in terms of accuracy" (p 199).

5.4 Adaptation and Implementation

This section describes the adaptations of the Distributed GSOM algorithm to 3 widely used distributed computing paradigms, MapReduce, BSP and RDD and their implementations on corresponding platforms Apache Hadoop, Apache Hama and Apache Spark respectively.

Apache Hadoop is one of the most popular distributed computing platforms implementing the MapReduce paradigm. It is well suited for batch-oriented processing; however, it suffers when multiple MapReduce jobs need to be employed for computation as the intermediate result sharing happens over the disk-based HDFS. This may affect the Hadoop based implementation of the Distributed GSOM algorithm as it uses two MapReduce jobs, as shown in section 5.4.1. Apache Hama and Apache Spark, on the other hand, avoid using HDFS and use communication among peer nodes and in-memory data sharing respectively to share intermediate results. This led us to adapt the Distributed GSOM algorithm to BSP and RDD paradigms and implement on platforms Apache Hama and Apache Spark, respectively. Moreover, Apache Spark's support for distributed reduce operation (with `treeReduce` function) is well suited for distributed redundancy reduction. Here, the reduce operation consumes two maps at a time and outputs a single map with redundant neurons among them removed.

5.4.1 Distributed GSOM on Hadoop MapReduce

We adapt the distributed GSOM algorithm to the MapReduce paradigm, and it is implemented using two MapReduce jobs. Below we describe these jobs for (1) data partitioning and (2) parallel GSOM training, redundancy reduction and Sammon's projection.

5.4.1.1 Job 1: Data Partitioning

We use random partitioning as the data partitioning strategy in our study. However, the main limitation of the random partitioning itself is the randomness of the assignment. If a mapper node training a GSOM on a data partition fails, it should be possible to access the same set of data for consistency. However, this would not be possible with the random partitioning. Hence, we use a separate MapReduce job that randomly partitions data and writes the results to HDFS, which is later used by the MapReduce job performing GSOM training and redundancy reduction.

Algorithm 5.3. Job 1: Data partitioning

map (*key*, *value*)

Input: Desired number of partitions, *P*, the file offset, *key*, the line composing of a record, *value*
Output: <*key'*, *value'*> pair, where the *key'* is the index of the partition
1 *key'* ← Get random integer in the range [1, P]
2 *value'* ← *value*
3 Output <*key'*, *value'*> pair

reduce (*key*, *V*)

Input: Index of the partition, *key*, all the record lines for the partition, *V*
Output: None
1 Write *V* to a file named *key*

Data partitioning MapReduce job uses standard `TextInputFormat` so that the mapping phase of the job would receive a line of the file containing data pertaining to a single record at a time. The mapper is responsible for assigning each of these records an integer in the range [1, P] where P is the desired number of partitions. This integer is used as the key of the output while the record is used as the value. A reducer, which receives all the records assigned to a single partition, would then write all the records on to a file on HDFS using the key as the file name.

5.4.1.2 Job 2: GSOM Training, Redundancy Reduction, and Projection

The Distributed GSOM algorithm is well aligned with the MapReduce computing paradigm such that parallel GSOM training step can be mapped to the map phase, the redundancy reduction phase can be mapped to the combine and reduce phases while merging can be mapped

to the reduce phase. Each mapper in MapReduce is responsible for training a GSOM on its data partition while each combiner is responsible for removing redundant neurons from a set of GSOMs assigned to a single mapper node. Finally, a single reducer is responsible for removing redundant neurons among GSOMs originating from multiple mapper nodes and finally performing Sammon's projection.

Algorithm 5.4. Job 2: GSOM training, redundant neuron removal, and projection

map (*key*, *value*)

Input: Name of the partition file, *key*, records of the partition, *value*
Output: <*key'*, *value'*> pair, where the *key'* is the output key (constant) and *value'* is the neurons of the trained GSOM map

1 *gsom* ← Train a GSOM on *value*
2 *value'* ← *gsom*.neurons
3 *key'* ← K (constant)
4 Output <*key'*, *value'*> pair

combine (*key*, *V*)

Input: GSOM neurons from the same node, *V*
Output: <*key'*, *value'*> pair, the *value'* is the non-redundant neurons of the *V*

1 *value'* ← Remove redundant neurons in *V*
2 *key'* ← *key*
3 Output <*key'*, *value'*> pair

reduce (*key*, *V*)

Input: Partially redundancy reduced GSOM neurons, *V*
Output: None

1 *V'* ← Remove redundant neurons in *V*
2 *P* ← Perform Sammon's projection on *V'*
3 Write *P* to a file

This MapReduce job uses a specialised input reader to read comma-separated values written by the data partitioning job. The input reader uses the file name as the key in making sure that all the records from a single partition are received by the same mapper. GSOMs are trained on each partition in parallel by the mappers using the standard GSOM algorithm and output them with the single key, so they are received by a single reducer.

A combiner is an optional step in Hadoop that operates between the mappers and reducers to reduce network traffic by summarising output records of the map phase with the same key. In

our implementation, the combiner removes redundant neurons among maps assigned to the same computing node. The redundancy reduction algorithm examines the input vectors mapped to each neuron to calculate its quantisation error. Hence, the records are stored with the best matching unit to avoid the combiners and the reducer having to access all the partitions in the HDFS. The single reducer performs further redundancy reduction among GSOMs generated in different computing node and perform Sammon's projection to generate the final GSOM map.

5.4.2 Distributed GSOM on Apache Hama

The Hadoop implementation of the MapReduce-based Distributed GSOM has inherent disadvantages. The main disadvantage arises due to how data is shared between jobs in Hadoop MapReduce. It does so by writing the intermediate results to HDFS, which is much slower than sharing it over memory. In the Distributed GSOM algorithm, this affects the data sharing between the two MapReduce jobs. The BSP paradigm is an alternative to overcome this limitation. The *supersteps* facilitate local computations, communications among peer nodes and barrier synchronisation, eliminating the need to use HDFS to communicate results among steps.

In this section, we investigate the BSP computing paradigm and its implementation Apache Hama as a candidate for the Distributed GSOM algorithm. The BSP based Distributed GSOM algorithm starts by choosing one node in the cluster as the master node in the setup phase. We use two *supersteps* in our algorithm, and they correspond closely with the two jobs in the MapReduce-based implementation.

The first *superstep* is responsible for partitioning the dataset into a predefined number of partitions using random partitioning strategy. Here we choose the number of computing nodes as the number of partitions as each node can work on each partition in the second *superstep*. We use a specialised input reader to read comma-separated values, and each node iteratively read each line - which corresponds to a single record - from its file split. It then draws a random integer in the range [1, P] where P is the desired number of partitions and based on the number, places the record in a temporary bin corresponding to the integer. When all the records are read

from the file split, each node sends its bins to respective nodes using the peer communication infrastructure available with Hama. There is a barrier synchronisation at the end of communication to ensure all nodes are synchronised at this point, concluding the first *superstep*.

The second *superstep* is responsible for parallel GSOM training, and it starts by reading the messages from peers to collect the set of records assigned to it. Once all the records are acquired the standard GSOM algorithm is used by the node to train a GSOM on the data. The trained GSOM is sent as a message using the peer messaging infrastructure to the master node chosen initially. Finally, a barrier synchronisation is applied to conclude the second *superstep* of the algorithm.

We use the third *superstep* for the aggregation of the maps. The final phase operates only on the master node, and it starts by collecting all the GSOMs sent to it by the peer nodes. Then the redundancy reduction algorithm removes redundant neurons among GSOMs, which facilitates faster merging. As the final step, Sammon's projection is used to generate the final GSOM.

Below we outline the algorithm using the following notation,

$\text{size}(P)$ returns the size of the list P, $\text{rand}(0, n)$ returns a random integer between 0 and n, $\text{add}(D, m)$ adds data item m to the list D, $\text{train}(D)$ trains a GSOM on the dataset D, $\text{reduceRedundancies}(M)$ performs redundant neuron removal among maps M and $\text{merge}(M)$ merges maps M using Sammon's projection.

Algorithm 5.5. Distributed GSOM BSP job

```
     Input: P: The list of all peers
1    bsp-setup
2      master ← peer P_1
3    superstep
4      B ← ∅
5      n ← size(P)
6      while d ← read() ≠ NIL do
7        i ← rand(0, n)
8        add(B_i, d)
9      end while
10     for peer P_j ∈ P do
11       send message B_j to P_j
12     end for
13   synchronise
14   superstep
15     D ← ∅
16     while m ← read message ≠ NIL do
17       add(D, m)
18     end while
19     N ← train(D)
20     send message N to master
21   synchronise
22   superstep
23     if self = master
24       M ← ∅
25       while m ← read message ≠ NIL do
26         add(M, m)
27       end while
28       reduceRedundancies(M)
29       Q ← merge(M)
30       write(Q)
31     end if
32   synchronise
```

By default, Hama divides the data file into a number of splits based on the block size and spawns a similar number of workers called peers. If the number of splits is low compared to the processing cores available with the cluster, it leads to underutilisation of resources. Even though Hama allows for specifying the number of BSP tasks, this value is not respected by the file splitter. We believe this is a bug in the platform or an implementation decision by the developers. To work around the above, we set the configuration `bsp.max.split.size`, which controls the split size, to appropriate lower values to generate a number of splits similar to the number of cores when carrying out the experiments. This allows for comparable running times with Hadoop and Spark.

5.4.3 Distributed GSOM on Apache Spark

The main disadvantage of implementations based on both Hadoop MapReduce and Hama is having to perform most of the redundancy reduction in a single reducer node. In this section, we investigate the RDD paradigm for distributed computing where its implementation Apache Spark has shown to outperform Hadoop MapReduce in distributed computing tasks.

Similar to Distributed GSOM on Hadoop MapReduce, we have a two-phase algorithm for Distributed GSOM on Spark. In this algorithm, we have used a number of data transformation operations defined on RDDs such as, `map`, which applies a provided function for each record, `reduceByKey`, which aggregates all the records for a particular key by applying a provided function, `collect`, which creates a local list from the records and `treeReduce` which aggregates all the records. While the former two generate new RDDs, the latter two generate local lists of records from the input RDDs.

Similar to job1 of the MapReduce algorithm, the first phase of the Spark algorithm is responsible for reading data from HDFS, parsing them and partitioning parsed records into a number of partitions. As outlined in Algorithm 5.6, the first phase starts by reading a file containing data from HDFS, which creates an RDD of lines of the file, each line containing a record entry. This RDD is then subjected to a `map` operation, which transforms the data by

parsing each line and returning an RDD of records. Then the resultant RDD is further subjected to another `map` operation, which assigns each of these records an integer in the range [1, P] where P is the desired number of partitions. The output of this operation is an RDD of key-value pairs with the integer as the key and the record as the value. The final step of this phase is a `reduceByKey` operation to aggregate all the records assigned to a particular partition with an RDD of key-value pairs as its output.

Algorithm 5.6. Data partitioning in Spark

Input: file
Output: partitions: PairRDD
1 lines ← readFile(file)
2 records ← lines.**map** {l → d}
3 pairs ← records.**map** {d → (i, d)}
4 partitions ← pairs.**reduceByKey** {(i, d) → (i, D)}

The second phase operates on the output of the first phase, similar to job2 of the MapReduce algorithm. However, the major difference is that while MapReduce writes the resulting data from the job1 to HDFS to share them with job 2, Spark allows in-memory sharing of data between jobs. RDD stores the state of memory as an object across the jobs, and the object is shareable between those jobs. The first transformation of the second phase is a `map` operation, which trains GSOMs on each data partition in parallel, which results in a distributed RDD of GSOMs.

Algorithm 5.7. GSOM training, redundancy reduction and projection in Spark

Input: partitions: PairRDD
Output: projection
1 maps ← partitions.**map** {(i, D) → M}
2 neurons ← maps.**treeReduce** {(M, M) → M}
3 final map ← merge(neurons)

For the next task, redundancy reduction, we have deviated from the MapReduce approach of having all the GSOMs in a single node and have used the `treeReduce` operation of Spark. We choose `treeReduce` operation over `reduce` operation due to the serial nature of the `reduce` operation's implementation, which prevents it from utilising the full potential of distributed computation capability. The custom function provided to the `treeReduce` operation receives two GSOMs at a time and removes redundant neurons in the two maps using the redundancy reduction algorithm and outputs the other neurons. The function provided to the `treeReduce` operation needs to be commutative and associative. Below we outline that the redundancy reduction algorithm operating on two GSOMs satisfies the above two requirements.

Denoting the redundancy reduction algorithm for two GSOMs as $RR(P_i, P_k)$ where P_i and P_k are the two GSOMs, the algorithm operates as follows: for each neuron $n_{i,j}$ of P_i, it checks whether there exists a neuron $n_{k,l}$ of P_k, which gives a lower total quantisation error if one of them is removed and vectors mapped to it is transferred to the other. The quantisation error check is performed for both options of removing either $n_{i,j}$ and $n_{k,l}$. Hence, irrespective of whether the neurons in P_i or neurons in P_k are iterated the same neuron will be removed. This ensures that the redundancy reduction algorithm is commutative.

Similarly, it can be shown that the redundancy reduction algorithm is associative. Let us consider three GSOMs, P_i, P_k and P_m and operations $RR(RR(P_i, P_k), P_m)$ and $RR(P_i, RR(P_k, P_m))$. For neuron $n_{k,l}$ of P_k to make it to the final map, it needs to ensure that there does not exist $n_{i,j}$ of P_i or $n_{m,n}$ of P_m which gives a lower total quantisation error if $n_{k,l}$ is removed and vectors mapped to it is transferred to the other. This does not depend on whether P_k is redundancy reduced with P_k or P_m first and only depends on the fact that $n_{k,l}$ has the best representation for vectors mapped to it among neurons in all three maps. Hence, only the neurons having the best representation for vectors mapped to it make it to the final map. This ensures that the redundancy reduction algorithm is associative as well.

The output of the `treeReduce` operation is a list of neurons pertaining to all the non-redundant neurons from multiple maps. These neurons are then merged with Sammon's projections to obtain the final map.

5.4.4 Algorithm Summary

Table 5.2 summarises the three algorithms with respect to the steps of the Distributed GSOM algorithm. Here we denote the two MapReduce jobs as `J1` and `J2`, three BSP *supersteps* as `SS1`, `SS2` and `SS3` and two RDD phases as `S1` and `S2`.

Table 5.2: Distributed GSOM steps in three algorithms

<table>
<tr><th>Step</th><th colspan="2">Hadoop</th><th>Hama</th><th>Spark</th></tr>
<tr><td>Data partitioning</td><td colspan="2">Job 1 (J1)</td><td>Superstep1 (SS1)</td><td>Phase 1 (S1)</td></tr>
<tr><td>GSOM training</td><td rowspan="3">Job 2 (J2)</td><td>Map</td><td>Superstep2 (SS2)</td><td rowspan="3">Phase 2 (S2)</td></tr>
<tr><td>Redundancy reduction</td><td>Combine</td><td rowspan="2">Superstep3 (SS3)</td></tr>
<tr><td>Merging</td><td>Reduce</td></tr>
</table>

5.5 Experiments and Results

Experiments were carried out to compare the efficiency of the Hadoop-, Hama- and Spark-based algorithms. Random partitioning was used as the partitioning technique for all the experiments.

5.5.1 Test Environment, Configurations and Experiment Plan

All our experiments were conducted on Amazon Elastic MapReduce (EMR) cloud platform for data processing and analysis. We employed 16 virtual machines, each with four 2.3 GHz processing cores and 16 GB of memory, in our cluster. Apache Hadoop version 2.7.3 with the Java version 8, Apache Hama version 0.7.1 with the Java version 8 and Apache Spark version 2.0.2 with Scala version 2.11.6 were used for MapReduce, BSP, and RDD based

implementations, respectively. Further, we implemented serial versions of GSOM and SOM algorithms which were run on a single machine with the same hardware specification to act as upper bounds for Hadoop-, Hama- and Spark-based implementations.

Total elapsed time is our primary performance measure and includes the time from job submission to the end of execution. This encompasses both CPU and non-CPU times such as file system IO and network IO times. Total elapsed time is a suitable metric since it is in fact what matters to the user, especially with cloud platforms charging based on the elapsed time.

The parameters for the algorithms are selected as follows. We set the spread factor to 0.0001, considering the size of the datasets and employ 100 growing iterations. The number of partitions, P, for Hadoop- and Spark-based implementations was set to 64 to match the total number of virtual processing cores available with the hardware setup. Since specifying the number of BSP tasks did not have any effect we set the configuration `bsp.max.split.size`, which controls the split size, to appropriate lower values to generate 64 partitions. This allows for comparable running times with Hadoop and Spark. For all three implementations, random partitioning is used as the partitioning strategy.

Our primary experiment was concerned with the speedups of the three distributed implementations over a serial implementation. We investigated this by comparing the elapsed time of the three distributed implementations on their respective platforms and the serial version for three datasets described in section 5.5.2. Since the parallelising mechanism is not bound to a particular variant of self-organizing maps, traditional SOM or any other variant of it can be used in place of GSOM. We also investigated the overall speedup, in terms of total elapsed time, when the SOM algorithm is used for the distributed processing. However, the major challenge of using SOM is its requirement to specify the size and the shape of the map in advance. For the comparison, we use square-shaped maps with the same number of neurons on each side. The size of the map is determined by the average number of neurons in each partition before the redundancy reduction phase. These values are available in Table 5.5. This generates

comparable maps with a similar number of neurons and facilitates the comparison of elapsed time between the use of SOM and GSOM for distributed processing.

As the three distributed implementations have comparable phases, as highlighted in Table 5.2, we analysed the relative performance of each of these phases in different implementations in terms of the elapsed time. Finally, we analysed the scaling out of the three implementations by varying the number of machines in the computer cluster from 1 to 16 (virtual cores from 4 to 64).

5.5.2 Datasets

The performance of our three algorithms was evaluated using three real-life datasets. Datasets are MIRFlickr dataset (Huiskes et al., 2010), Million Songs dataset (Bertin-Mahieux et al., 2011) and power consumption dataset (Dua & Graff, 2017). These datasets have been widely used to evaluate distributed algorithms (Hadgu et al., 2015; Huang et al., 2016; Kraska et al., 2013; Sozykin & Epanchintsev, 2015). In (Kraska et al., 2013) Million Songs dataset has been used to demonstrate the capabilities of the MLBase, the predecessor of the Apache Spark's own machine learning library, MLib. Similarly, it has been used to evaluate the distributed implementation of Adaptive Sub-gradient Descent (AdaGrad), a variant of Stochastic Gradient Descent (SGD), for large-scale machine learning tasks using Apache Spark (Hadgu et al., 2015). A distributed implementation of Earth Mover's Distance (EMD) on Apache Hadoop, Hama and Spark (Huang et al., 2016) has been evaluated on both MIRFlickr and Million Songs datasets while MIRFlickr dataset has been used to demonstrate distributed image processing using Apache Hadoop (Sozykin & Epanchintsev, 2015).

Statistics about datasets are presented in Table 5.3.

1) MIRFlickr dataset contains features related to images extracted from Flicker social media platform, and the features include MPEG-7 edge histogram descriptors and homogeneous texture descriptors. The dataset consists of one million records of

images, and each record in texture descriptor feature dataset has 43 attributes. We denote the texture descriptors feature dataset as A.

2) Million Songs dataset is a collection of audio features and metadata for a million contemporary popular music tracks. We use a subset of the dataset made available at the UCI machine learning repository (Dua & Graff, 2017) with timbre features extracted. Each record in the dataset has 90 features, 12 timbre average features, and 78 timbre covariance features. This feature dataset is denoted as B.
3) Power consumption dataset is a multivariate dataset with records of household electric power consumption data containing more than 2 million measurements gathered between December 2006 and November 2010. Each record consists of 9 attributes. After removing missing values, the dataset consists of 2,075,259 records. We denote this dataset as C.

Table 5.3: Summary of datasets used in the study

Dataset	Instances	Attributes
MIRFlickr texture descriptors (A)	1,000,000	43
Million songs (B)	515,345	90
Power consumption (C)	2,075,259	9

5.5.3 Results

Total elapsed time in seconds for three Distributed GSOM algorithm implementations and the serial implementations using GSOM and SOM algorithms for three datasets are shown in Table 5.4 and Figure 5.3. The very high elapsed time of the serial implementations and relatively low elapsed time of the distributed implementations highlights the need for distributed implementation for GSOM, and SOM in general, to be practical for large datasets.

Table 5.4: Total Elapsed Time in seconds

Dataset	Serial		Hadoop		Hama		Spark	
	GSOM	SOM	GSOM	SOM	GSOM	SOM	GSOM	SOM
A	75,719	78,305	1,026	1,312	806	1,103	953	1,312
B	17,994	23,144	275	356	78	158	64	167
C	141,561	145,103	912	974	845	829	831	856

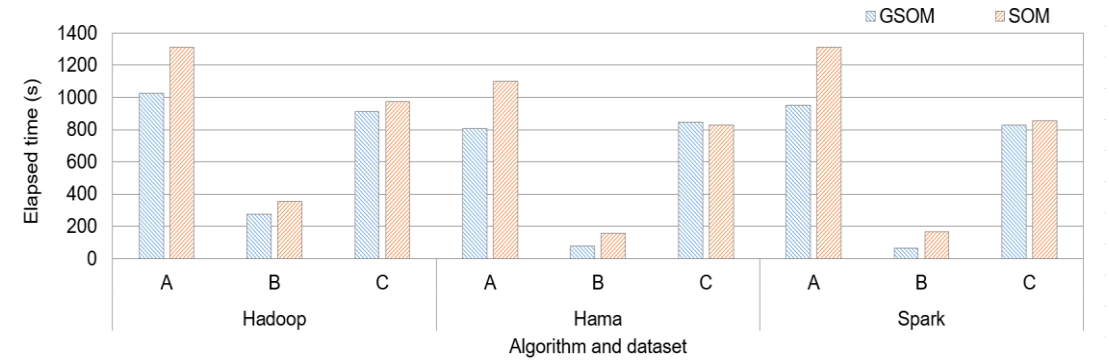

Figure 5.3: Total elapsed time for three Distributed GSOM implementations using GSOM and SOM as underlying algorithm

It can be seen that Hadoop-based implementation records highest elapsed times for all three datasets while there is no clear winner across all three datasets. Higher elapsed times of Hadoop is likely due to the high cost associated with the intermediate results sharing over HDFS. Moreover, it can be noticed that elapsed time for dataset B is significantly lower across three distributed platforms as well as on the serial implementation. The lower processing time is due to the lower number of neurons in both intermediate maps and the final map as can be seen in Table 5.5 and Table 5.6.

Table 5.5: Average number of neurons in individual maps

Dataset	Hadoop	Hama	Spark
A	480.56	478.08	476.47
B	152.40	151.28	151.06
C	1,107.39	1,098.56	1,050.36

It can be seen that using the GSOM algorithm over the SOM algorithm for the distributed processing speeds up the computations. GSOM based implementations have recorded faster running times for all datasets for all distributed and serial implementations except for dataset C on Apache Hama. The faster processing times of GSOM is linked to the fact that the map is initialised with mere four neurons and additional neurons are added later as required.

Given that 64 processing cores are available in our testing environment, Hadoop-, Hama-, and Spark-based implementations using the GSOM algorithm have achieved super-linear speedups of 73.80, 93.94 and 79.45 respectively on dataset A compared to the serial implementation using the GSOM algorithm. Similarly, high speedups are observed on dataset B with speedups of 65.43, 230.69 and 281.16 for Hadoop, Hama, and Spark respectively. Speedups for datasets C in the same order are 155.22, 167.53 and 170.35. Super-linear speedups compared to the serial implementation are because a processing core processes a portion of data as well as works only with a subset of nodes of the final map. Even though the distributed GSOM training leads to some redundant neurons being introduced in individual maps, the number of neurons in an individual map is still far less compared to the final map.

Table 5.6: Total number of neurons after redundancy reduction

Dataset	**Hadoop**	**Hama**	**Spark**
A	16,060	14,568	15,215
B	1,394	1,316	1,457
C	12,992	12,972	13,398

5.5.4 Performance of Different Phases

It is interesting to analyse the cost of different phases of the three algorithms. The MapReduce-based algorithm consists of two phases (jobs) responsible for data partitioning and running GSOM algorithm. Similarly, the RDD based algorithm consists of two phases corresponding to the same tasks. In the BSP based algorithm, the first *superstep* corresponds to the first phase of the other two algorithms while the combination of the second *superstep* and cleanup phase accounts for the second phase of the others. Hence, their comparison can unveil the strength (and weakness) of each platform for a particular phase.

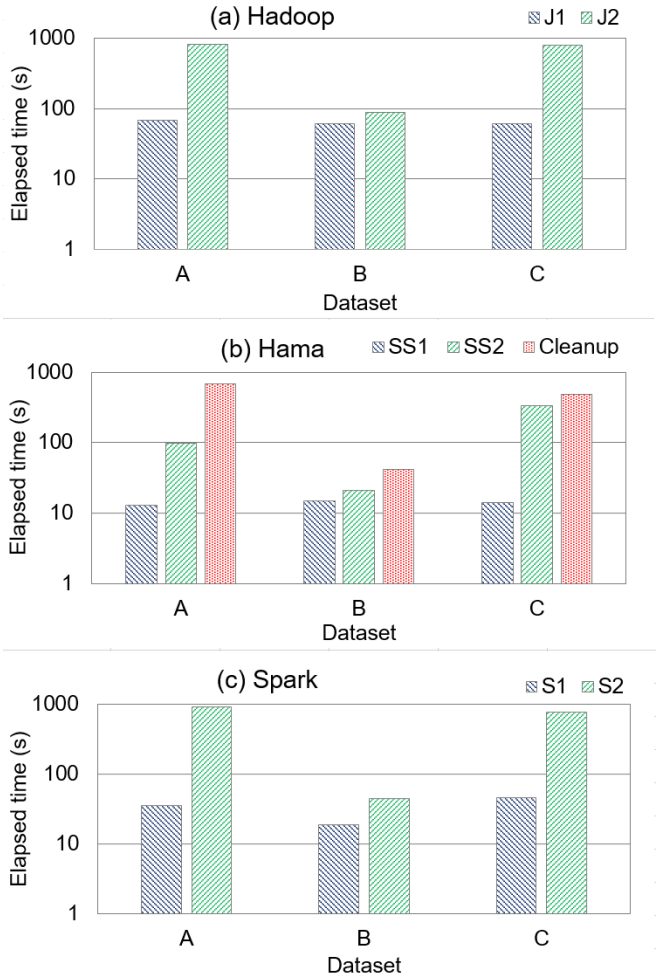

Figure 5.4: Cost of each phase in different implementations

We present the cost of different phases of the three implementations of Distributed GSOM algorithm in this section. Figure 5.4 illustrates the cost of each phase of the three implementations on datasets A, B, and C (Note the log scale of the Y-axis). `J1` and `J2`

correspond to the two MapReduce jobs of the Hadoop implementation. The cost of `J2` dominates the overall cost, and for dataset A and C, it is about an order of magnitude expensive than `J1` while the dominance is less for dataset B. This is due to the inherent serial nature of Sammon's projection algorithm used for merging parallel trained GSOMs. The relatively lower number of neurons in the final map for dataset B leads to faster Sammon's projection. For the Hama BSP implementation, `SS1`, `SS2,` and `Cleanup` correspond to the *supersteps* 1, 2 and cleanup phases respectively. `Cleanup` is expensive than `SS2` (about an order of magnitude for dataset A) while `SS2` is more expensive than `SS1`. Similar to Hadoop, the dominance of phases `SS2` and `Cleanup` over `SS1` is less for dataset B. Similarly, the data partitioning phase of the Spark implementation, denoted as `S1`, is much cheaper compared to the GSOM training phase. Parallel GSOM training, redundant neuron removal and merging with Sammon's projection, which constitutes the second phase of the Spark implementation denoted as `S2`. It can be seen that `S1` is more than an order of magnitude cheaper than `S2` for both datasets A and C. Overall, merging dominates the overall cost, highlighting the requirement of redundant neuron removal.

The merging phases of all three algorithms dominate the total elapsed time due to the serial nature of Sammon's projection being used for merging. The time complexity of Sammon's projection is $O(n^2)$, where n is the number of vectors, which broadens its effect on total elapsed time. Due to this, we can see that the total number of neurons after redundancy reduction in Table 5.6 is highly correlated with the total elapsed times in Table 5.4.

5.5.5 Effect of Scaling Out

We experimented with the scaling out of the three algorithms by varying the number of machines in the cluster. The experiments were carried out using the dataset A, and the cluster size is varied from 1 machine to 16 machines, which in turn increased the total number of virtual cores in the cluster from 4 to 64. The results in terms of speedup and the total elapsed time for the implementations on three paradigms are presented in Table 5.7 and Figure 5.5, respectively.

Table 5.7: Speedups when scaling out

Platform	Number of Nodes				
	1	**4**	**8**	**12**	**16**
Hadoop	1.00	2.86	4.84	6.11	6.77
Hama	1.00	3.17	5.27	6.62	7.84
Spark	1.00	2.78	5.04	6.25	7.44
Average	1.00	2.94	5.05	6.33	7.35

It can be seen that the total elapsed time decreases as the number of machines are increased from 1 to 16. The Distributed GSOM achieves speedups up to 7.84 on Hama with comparable speedups on Hadoop and Spark. Moreover, the speedups are sub-linear for all the platforms. We believe that a full linear speedup is not achieved due to two reasons, the overhead inevitably introduced by having more workers and the linear components in the algorithms such as merging of trained maps. Further, it can be seen that for dataset A, Hama slightly outperforms both Hadoop and Spark. While both Hadoop and Spark have similar run times, Spark slightly outperforms Hadoop as the number of nodes is increased.

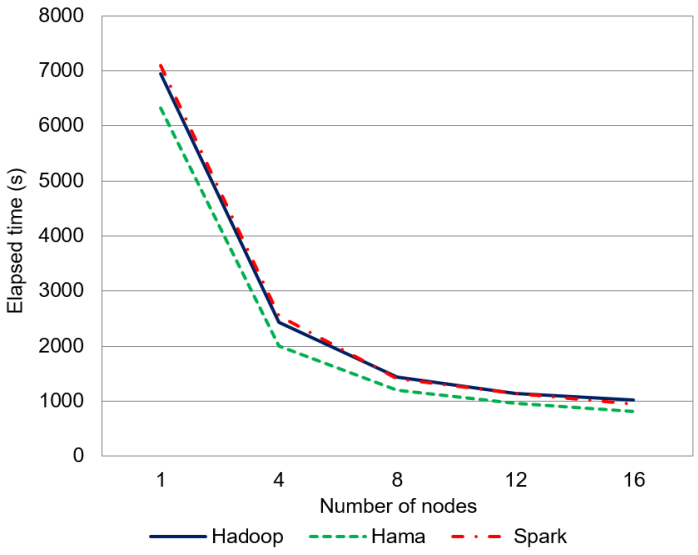

Figure 5.5: Total elapsed time when scaling out

5.6 Chapter Summary

This chapter proposed Distributed GSOM, a novel data parallelised distributed SOM algorithm to speed up SOM calculations. The proposed algorithm overcomes the limitations of previous parallelisation attempts such as the need for synchronisation operation after each iteration by using a single projection operation and the inferior map quality of batch SOM by using online SOM algorithm.

The proposed algorithm was adapted for MapReduce, BSP and RDD paradigms and implemented on well-established distributed computing platforms Apache Hadoop, Hama and Spark respectively. We conducted a number of experiments on real-life datasets to demonstrate the super-linear speedup achieved by the distributed version compared to the serial counterpart. The previous best result is the linear speedup achieved in (Lawrence et al., 1999) with data partitioned batch SOM algorithm. We compared the three implementations in terms of the total elapsed time and investigated the cost of each phase of the algorithms. Moreover, the scalability of the algorithm was demonstrated by varying the number of virtual computing cores in the cluster from 4 to 64. Finally, we demonstrated that using GSOM algorithm in place of traditional SOM for distributed processing improves the overall running time.

With a distributed algorithm proposed for the self-organizing component of the distributed architecture, the next chapter presents the detail of multimodal clustering component implemented for distributed computing. Further, the overall distributed implementation is demonstrated using a case study from the physical activity monitoring domain, which suffers from large data volumes and multimodality.

Chapter 6
A Case Study

Chapter 4 highlighted the advantages of using distributed computing to scale and speed up artificial *impression* generation and proposed a suitable distributed computing architecture for *impression* generation mechanism. The first component of the distributed architecture, the Distributed SOM algorithm, was presented, adapted to multiple distributed computing platforms, and demonstrated on large heterogenous datasets in Chapter 5.

This chapter addresses the second component of the distributed architecture by proposing the use of distributed computing for the multimodal clustering algorithm. Moreover, we evaluate the overall system implemented on Apache Spark with a dataset from physical activity monitoring domain, which suffers from large data volumes due to the high-frequent capture of sensor reading from body-worn sensors and multimodality due to different sensor modalities capturing the physical movements of the subject.

Some of the work in this chapter has appeared in (Jayaratne et al., 2017) and (Jayaratne et al., 2019)

6.1 Distributed Multimodal Clustering

In Chapter 5, we demonstrated the advantages of distributed computing in terms of computational speedup for processing large datasets. The DGSOM algorithm demonstrated super-linear speedup while generating topographic maps on underlying data allowing us to reduce processing times from days to minutes. In this section, we investigate possible ways of using distributed computing for generating a multimodal clustering over the unimodal GSOMs. The speedups achieved using distributed computing for both phases facilitate artificial *impression* generation for large multimodal datasets under acceptable computational times to facilitate real-time applications.

6.1.1 Apache Spark as Distributed Computing Platform

Due to the versatile nature and the performance demonstrated with Distributed GSOM algorithm, in this section, we extend the use of Apache Spark for implementation of the multimodal clustering algorithm. Three well-established distributed computing platforms, namely Apache Spark, Apache Hama and Apache Hadoop, were considered for the implementation. As established distributed computing platforms, all three platforms considered provide essential functionalities such as scheduling, fault recovery, job monitoring and interacting with distributed file systems. Below, we outline the reasons for our choice of Apache Spark as the preferred distributed computing platform for the overall implementation of the Multimodal Clustering algorithm over the others.

Apache Spark provides a large number of operations such as `map`, `reduce`, `reduceByKey`, `collect`, `filter` to manipulate and transform RDDs, which are Spark's immutable, fault-tolerant distributed dataset divided into logical partitions. These high-level operations distribute computation among worker nodes in the cluster under the hood with the programmer being provided with an easy-to-use abstraction from the low-level workings. Having to only worry about the logic and flow of the distributed algorithm, with Apache Spark, the programmer is

freed from the need to understand the nitty-gritty of the low-level distribution of computational tasks.

In contrast, Apache Hama merely provides low-level distributed computing constructs such as barrier synchronisation and peer-to-peer communication. Even though this provides the freedom to the programmer to implement a variety of distributed computing operations using these low-level constructs, the programmer is left to implement these operations ground up. Moreover, system-wide functionalities such as distributed caching are non-existent, whereas they are provided built-in in Apache Spark and Apache Hadoop.

Apache Hadoop, on the other hand, provides a higher-level abstraction and system-wide functionalities compared to Apache Hama, however, falls short of the easy-to-use operations of Apache Spark. With Apache Hadoop, all the distributed computation steps need to be transformed into the MapReduce paradigm, which could be challenging. On the other hand, the only way to transfer results of one MapReduce job as input to another MapReduce jobs is through the distributed file system which is much slower compared to in-memory data sharing of Apache Spark.

The above advantages of Apache Spark over Apache Hadoop and Apache Hama led to the selection of it for the overall implementation of the multimodal clustering algorithm.

6.1.2 Multimodal Distance Calculation on Apache Spark

Multimodal distance calculation is an expensive computation step that could take advantage of distributed computing. As part of multimodal distance calculation, it is required to calculate the activation probabilities for each neuron pair belonging to different modalities in order to derive the probability distribution of activation for each neuron. Activation probabilities are calculated using activation counts, and activation counts are recorded by presenting each multimodal input to the trained modality specific GSOMs and recording winner neuron pairs. This provides an ideal situation to data parallelise the processing by distributing multimodal inputs among worker nodes.

Moreover, once the activation frequencies are calculated for each neuron pair belonging to different modalities, distributed computing could be used for further processing these intermediate results. Given that the number of possible combinations is $N = n_1 \times n_2 \times n_3 \ldots \times n_m$ where $n_i = |G_i|$, the size of GSOM of i^{th} modality, N could run into a sizable number. Hence, it makes sense to use distributed computing for further process these frequencies into calculating required probability distributions. Similarly, these intermediate results could be data parallelised to calculate the probability distribution of activation for each neuron.

Algorithm 6.1 outlines the distributed multimodal distance (d_{MD}) calculation on Apache Spark. Each multimodal input, i, is presented to the trained modality-specific GSOMs using a `map` operation and winner pair, $w = (w_x, w_y)$, $w_x \in G_x, w_y \in G_y$, is recorded. The frequency of activation, n, for each pair is calculated with a `mapToPair` operation and a subsequent `reduceByKey` operation. To calculate the probability distribution for each neuron, we then subject this frequency information to a `mapToPair` operation and another `reduceByKey` operation. The `mapToPair` operation outputs the first neuron of the winner pair, w_x, as the key and a co-activation distribution (CoAD) object having co-activation frequency of the winner pair marked against the second neuron, w_y. The `reduceByKey` operation collects the co-activation frequencies of different second neurons for the same first neuron by aggregating them into a single CoAD object. The output of these operations is a map of CoAD objects for each neuron with the neuron as the key of the map. The multimodal distance calculator, d, is provided with the GSOMs as well as the map of co-activation distributions, which are required to calculate d_{ED} and d_{EMD} components of d_{MD} respectively.

Algorithm 6.1. Multimodal distance calculation on Apache Spark

Input: GSOM maps trained on each modality, *GSOMs*, multimodal input data, *input*
Returns: Multimodal distance object, d

winners ← input.**map** $\{i \rightarrow w\}$
winnerCounts ← winners.**mapToPair** $\{w \rightarrow (w, 1)\}$
frequencies ← winnerCounts.**reduceByKey** $\{(w, 1) \rightarrow (w, n)\}$
CoADs ← frequencies.**mapToPair** $\{(w, n) \rightarrow (w_x,\ CoAD(w_y, n)\}$
CoADs ← CoDs.**reduceByKey** $\{(w_x, CoAD) \rightarrow (w_x, CoAD)\}$
CoADs ← CoADs.**collectAsMap**
d ← MultiModalDistance(CoADs, GSOMs)

6.1.3 Multimodal Clustering Implementation on Apache Spark

Similarly, we implement the multimodal clustering using Apache Spark. The multimodal clustering algorithm is responsible for creating a hyper clustering on the topological map by defining clusters on each neuron containing only the particular neuron, and iteratively merging them based on the multimodal distance, d_{MD}.

While the main iterative process of the multimodal clustering cannot be parallelised due to the sequential nature imposed by the cluster merging process, the intermediate calculations can benefit from distributed computing. For example, to minimise the cost of having to calculate d_{MD} for all the cluster pairs in each iteration, we have utilised a caching mechanism where we calculate these distances upfront. Given that the number of possible cluster pairs could run into a sizable number, such pre-calculation benefits from distributed computing.

The details of the Apache Spark-based implementation are outlined in Algorithm 6.2.

Algorithm 6.2. Multimodal clustering on Apache Spark

Input: GSOM maps trained on each modality, *GSOMs*, multimodal distance object,*d*
Returns: Multimodal clusters, *C*

```
1  for each G_X ∈ GSOMs
2    for each neuron, x_i ∈ G_X
3      create cluster c_i in G_X containing x_i
4    end for
5  end for
6  distanceCache ← {}
7  for each G_X ∈ GSOMs
8    for each G_Y ∈ GSOMs and G_Y ≠ G_X
9      C ← clusters in G_X
10     pairC ← C.cartesian {c_i → (c_i, c_j), i ≠ j}
11     distances ← pairC.map {(c_i, c_j) → ((c_i, c_j), d_MD)}
12     distanceCache [X, Y] ← distances.sort # by d_MD
13   end for
14 end for
15 while (true)
16   for each G_X ∈ GSOMs
17     for each G_Y ∈ GSOMs and G_Y ≠ G_X
18       distances ← distanceCache [X, Y]
19       for each pair ((c_i, c_j), d_MD) ∈ distances
20         SD ← calculate self-distance for (c_i, c_j)
21         if SD > 0.5
22           merge (c_i, c_j)
23           update distanceCache [X, Y]
24           break
25         end if
26       end for
27     end for
28   end for
29   if no clusters were merged
30     exit
31   end if
32 end while
```

In Algorithm 6.2, lines 1-5 are concerned with generating a hyper clustering with each neuron as a cluster. To minimise the cost of having to calculate d_{MD} for all the cluster pairs in each iteration, we have utilised a caching mechanism. Distance d_{MD} is pre-calculated for each pair of clusters, (c_i, c_j) in each modality, G_X with respect to other modality, G_Y. Pairs of clusters are obtained with the `cartesian` operation of Apache Spark, while a `map` operation is used for the distributed calculation of corresponding multimodal distances. Moreover, the distributed `sort` operation of Apache Spark is used to sort these multimodal distances in descending order to identify clusters to merge quickly.

During the cluster merging phase (lines 15-32), each cluster pair, (c_i, c_j), sorted descending by multimodal distance, d_{MD}, is considered for merging. The cluster pair is merged if the self-distance of the combined cluster, (c_i, c_j) is less than 0.5. The self-distance of a cluster measures the multimodal distance, d_{MD} between points within the cluster. The value of self-distance is utilised to determine whether two potential clusters for merging represent the same concept or different concepts. Once two clusters are merged, distance cache is updated to invalidate all the distance values relating to the old clusters and to calculate distance values for the new combined cluster. Moreover, the inner loop is exited to combine closest clusters by d_{MD} which may have been updated due to cache updates. This iterative merging is carried out until no clusters are merged in the latest iteration.

6.2 Evaluation in a Physical Activity Monitoring Application

In Chapter 4, we utilised a dataset from speech recognition domain to demonstrate the multimodal clustering algorithm and in Chapter 5 datasets from image understanding and audio classification domains to demonstrate the distributed GSOM algorithm. In order to demonstrate the overall system, which performs distributed multimodal clustering for large datasets, we draw an experiment from the physical activity monitoring domain (Cornacchia et al., 2017). In

particular, this dataset presents the challenge of multimodality due to several modalities capturing the physical movements of the subjects and large data volumes due to the high-frequent capture of sensor reading from body-worn Inertial Measurement Units (IMUs).

6.2.1 Physical Activity Monitoring

Physical activity is any movement in the human body produced by the contraction of muscles resulting in some displacement and energy expenditure (C.-C. Yang & Hsu, 2010). Monitoring human physical activity has been an area of interest for several research fields including exercise physiology (Copeland & Esliger, 2009), sports physiology (Aughey, 2011), aged care research (Bagalà et al., 2012), and epidemiological research (C.-C. Yang & Hsu, 2010). Body-worn sensors have been widely accepted as a useful and practical method to assess and measure physical activity among research subjects. IMUs containing a wide array of sensors have been used to study subjects under free-living conditions in longitudinal studies (Berlin et al., 2006; Ruch et al., 2011). Raw sensor data from such experiments have been used to identify and calculate high-level constructs such as movement classification (Ugulino et al., 2012), energy expenditure calculation (Van Hees et al., 2011), and fall detection (Bagalà et al., 2012).

Moreover, with the diminishing cost of Internet of Things (IoT) devices, there has been a surge in interest on body-worn sensor-based devices that monitor activity levels among the public in general (Price et al., 2017). Many products from vendors such as Fitbit[1], Garmin[2] and Misfit[3] has been released to the market and gained popularity among health-conscious consumers.

Traditionally, a variety of subjective methods such as self-reported diaries, logs, questionnaires and surveys have been utilised to assess the levels of physical activity for research purposes (Prince et al., 2008). However, data collected with such methodology is heavily dependent on individual observation and subjective interpretation and pose a reliability issue for the research dependant on them (Tudor-Locke & Myers, 2001). On the other hand, more recent research

[1] https://www.fitbit.com
[2] https://www.garmin.com
[3] https://misfit.com/

requiring physical activity monitoring has opted-in for more objective methods such as the use of body-worn motion sensor such as accelerometers, magnetometers, gyroscopes and pedometers (Attal et al., 2015; Doherty et al., 2017; Pedišić & Bauman, 2015).

Accelerometers measure the acceleration of the subject along reference axes and are useful in estimating the intensity of the human activity. Acceleration data can be used to calculate velocity by calculating time integrals of acceleration and displacement by calculating time integral of velocity. Moreover, acceleration can be used to calculate acceleration-jerk, the time derivative for acceleration. Gyroscopes measure the angular velocity and similarly can be used to calculate additional measurements such as its time derivative, angular acceleration. Pedometers, one of the simplest body-worn sensors, count the number of steps which can be used to estimate the energy expenditure and the distance walked. An IMU combines several such sensors and provides an easy-to-wear wearable device for physical activity monitoring.

6.2.2 The Dataset

We utilised PAMAP2 (Reiss & Stricker, 2012), a multimodal physical activity monitoring dataset to evaluate the implementation of our overall system. The PAMAP2 dataset consists of sensor readings captured from body-worn IMUs on nine volunteers while they are performing 12 different activities (such as *lying, sitting, standing, rope jumping* and *ascending stairs*). The 3 IMUs are worn on the chest, the dominant arm's wrist and the ankle on the dominant side. The IMUs capture tri-axial acceleration at two scales (in ms^{-2}), tri-axial angular speed (in rad/s) and tri-axial magnetometer data (in μT) at the sampling frequency of 100Hz. The dataset contains a total of 2,872,533 readings. The PAMAP2 dataset has been widely used to demonstrate various algorithms that perform multimodal sensor fusion for human activity recognition (Guo et al., 2016; Kasnesis et al., 2019; Münzner et al., 2017; Wang et al., 2019; Z. Yang et al., 2018).

We have used the acceleration and angular speed recordings (with additional features engineered on them) as different modalities that capture the volunteers' activities. While the

distributed multimodal sensor fusion is demonstrated here with two modalities, it is generalizable and can be extended to more than two modalities, effectively fusing all the required sensor inputs to obtain a robust multimodal representation. The dataset was pre-processed to remove records with missing values and records related to transitioning between two activities. Moreover, of the two scales, acceleration captured at ±6g scale was discarded as the signal gets saturated during high impact activities while the acceleration captured at ±16g scale was retained. Further, we removed orientation data as it was indicated to be invalid for this data collection. The metadata of the dataset contains resting heart rate for each participant and based on that we calculated the average heart rate increase for each activity, as shown in Table 6.1.

Table 6.1: Average heart rate increase for each activity and intensity categorisation

Activity	**HR increase (bps)**	**Intensity**
Lying	9.02	Low
Sitting	13.21	
Standing	22.42	
Ironing	24.34	
Vacuum cleaning	37.69	
Walking	46.46	Moderate
Nordic walking	58.21	
Cycling	58.85	
Descending stairs	62.94	
Ascending stairs	63.01	
Running	81.62	High
Rope jumping	90.16	

We can observe that the activities include a mix of low-intensity activities such as *lying*, *sitting, standing, ironing, vacuum cleaning*, moderated intensity activities such as *walking*, *Nordic walking*, *cycling* and high-intensity activities such as *running* and *rope jumping*.

Similar to (Anguita et al., 2013), we engineered additional features from the raw signal to improve the accuracy of classification. For the acceleration modality, we engineered the tri-axial acceleration jerk (*da/dt*), and acceleration magnitude from the tri-axial acceleration components. Similarly, for the angular speed modality, we engineered the tri-axial angular acceleration and angular speed magnitude. The complete list of features for the two modalities is listed in Table 6.2.

Table 6.2: Original and engineered features for the two modalities

Modality 1	**Modality 2**
Acceleration – x axis (ms^{-2})	Angular speed – x axis ($rad \cdot s^{-1}$)
Acceleration – y axis (ms^{-2})	Angular speed – y axis ($rad \cdot s^{-1}$)
Acceleration – z axis (ms^{-2})	Angular speed – z axis ($rad \cdot s^{-1}$)
Acceleration jerk – x axis (ms^{-3})	Angular acceleration – x axis ($rad \cdot s^{-2}$)
Acceleration jerk – y axis (ms^{-3})	Angular acceleration – y axis ($rad \cdot s^{-2}$)
Acceleration jerk – z axis (ms^{-3})	Angular acceleration – z axis ($rad \cdot s^{-2}$)
Acceleration magnitude (ms^{-2})	Angular speed magnitude ($rad \cdot s^{-1}$)

6.2.3 Evaluation Metrics

Our artificial *impression* generation mechanism operates over the unimodal representations generated by the GSOM maps generating a clustering in each GSOM map. The clustering incorporates information from the other modalities by exploiting co-occurrence relationships among them. Similar to the evaluations in section 5.5.3, the primary metric used to evaluate the performance speedup due to distributed self-organizing calculations is the total elapsed time as

it encompasses both computation-based as well as non-computation-based times. To evaluate the quality of the multimodal clustering generated, we have used a number of evaluation metrics. Clustering quality evaluation metrics fall under two broad categories; internal and external. The internal metrics evaluate the clustering based on the data that were used to perform the clustering themselves. These measures usually reward high intra-cluster similarity and low inter-cluster similarity in order to generate coherent clusters. The external metrics, on the other hand, use class labels to evaluate the clustering generated. We note that the external metrics are suitable to evaluate the quality of clusters generated by multimodal clustering due to the following reasons. Multimodal clustering incorporates information from multiple modalities leading to cluster together two neurons in a given modality that are spatially apart, given the other modalities perceive them as similar. Conversely, two spatially close neurons might be clustered separately if they are perceived distant from the view of other modalities. Moreover, external cluster evaluation metrics measure the quality of clustering based on class labels, which, in fact, is the desired clustering.

External cluster evaluation metrics F1, which is the harmonic mean between precision and recall, Rand measure (Rand, 1971), Dice index (Dice, 1945), cluster purity, normalised mutual information (NMI) and internal cluster evaluation metric Davies–Bouldin index (DB-Index) (Davies & Bouldin, 1979) were calculated to evaluate the quality of clustering. A lower DB-index value represents a better clustering, and higher values represent better clustering for all other metrics used.

6.2.4 Test Environment and Configurations

The experiments were carried out on a commercial cloud computing platform employing 8 virtual machines each having 8 computing cores each with the clock speed of 2.3GHz and 16GBs of memory. The Spark-based algorithm is implemented using Apache Spark 2.0.0 with Scala 2.11.8. In order to compare the speedup of the DGSOM implementations, a serial version of the GSOM algorithm was implemented, and computations were carried out employing a single virtual machine with the same hardware specifications.

For the GSOM training, we employed 100 training iterations and 100 smoothing iterations with the parameters of the GSOM algorithm set as below. The spread factor which controls the spread of the map, $SF = 0.01$, the starting learning rate, $\alpha(0) = 0.3$ and starting neighbourhood radius, which is used in the neighbourhood calculation, $N(0) = 4$.

6.2.5 Evaluation Methods and Results

To demonstrate the quality of clusters generated with multimodal clustering, we compare them with those from the k-means clustering performed over the unimodal neuronal layers. While the number of clusters needs not to be provided as a parameter to the multimodal clustering algorithm, the k-means algorithm requires this to be provided as the parameter k beforehand. A common method to eliminate this requirement is to perform multiple rounds of clustering while varying k from 2 to $\sqrt{n}$, where n is the number of items to be clustered and choosing k such that an internal metric is optimal. We utilised this mechanism to choose the optimal value of k for k-means clustering while using DB-Index as the internal cluster evaluation metric. For the crossmodal clustering, the relative weighting factor between d_{ED} and d_{EMD}, λ, was set at 0.5, giving both the distance in own modality and the distance based on the second modality similar importance.

Table 6.3: Performance of Spark-based DGSOM implementation compared to the serial implementation

GSOM implementation	**Total elapsed time (s)**
Serial	24,139.212
DGSOM on Spark	329.941

Significant speedups could be observed for Spark-based adaptation compared to the serial implementations, as highlighted in Table 6.3. Compared to the serial implementation, Apache Spark-based implementations has achieved a 73.16-fold improvement in running time. While the distributed implementation utilised 64 virtual computation cores, the speedup of Apache Spark is even greater than that. This is due to the fact that while each computation core

processes 1/64 of the data, it only operates on a portion of neurons in the final map. This results in super-linear speedups compared to the serial implementation.

Table 6.4: Multimodal representation performance. Evaluation of the quality of multimodal clustering compared to the unimodal (k-means) clustering

Metric	Modality 1		Modality 1	
	Multimodal	**Unimodal**	**Multimodal**	**Unimodal**
F1	**22.75%**	18.94%	20.04%	**20.74%**
Rand measure	**0.7751**	0.4780	**0.7930**	0.5706
Dice index	**0.2275**	0.1894	0.2004	**0.2075**
Cluster purity	**26.68%**	16.25%	**23. 41%**	20.19%
NMI	**0.2151**	0.0888	**0.2096**	0.2078
DB-Index	6.6779	**2.2414**	6.6137	**2.1907**

Table 6.4 presents the performance of the multimodal representation using the multimodal clustering compared to the clustering performance of unimodal representation using k-means clustering. Modality 1 is the tri-axial acceleration and features derived from them while modality 2 is the tri-axial angular speed and their engineered features. It can be observed that modality 1 has benefited from the complementary information with all the external cluster evaluation metrics of multimodal clustering exceeding those of unimodal (k-means) clustering. Multimodal clustering for modality 2 has performed better in terms of Rand measure, cluster purity and NMI while marginally underperformed in terms of F1 and Dice index compared to k-means clustering.

It can be also be observed that DB-Index is higher for multimodal clustering for both modalities while usually a lower value is preferred. DB-Index, an internal cluster evaluation measure, is defined as a function of the ratio of the intra-cluster scatter and inter-cluster separation favouring higher cluster cohesion and better cluster separation. Designed for a unimodal clustering, DB-Index is unsuitable for multimodal clustering scenario as described earlier and

included only as a reference. On the other hand, metrics F1-measure, Rand measure, Dice index, cluster purity and NMI are based on actual class labels and effectively measure the clustering improvement brought in by the crossmodal effect.

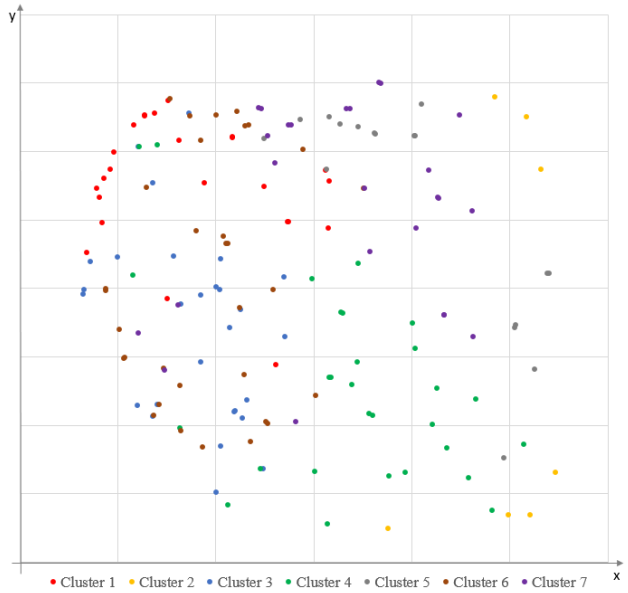

Figure 6.1: Multimodal clustering for modality 1. The number of clusters automatically selected by the multimodal clustering algorithm.

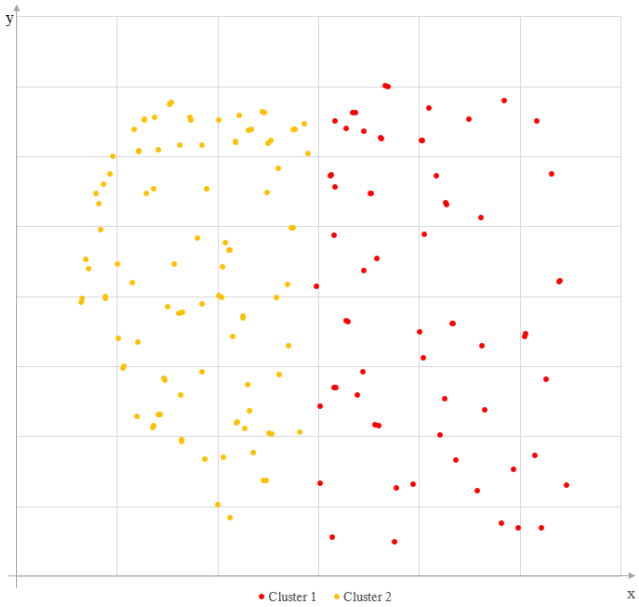

Figure 6.2: k-means algorithm based unimodal clustering for modality 1. Parameter k set to 2 for two clusters based on optimal DB-Index value.

Figure 6.2 displays the k-means algorithm based unimodal clustering of the neuron in modality 1 GSOM map. The parameter *k* was set to 2 for two clusters based on the optimal value of DB-index. Figure 6.1 displays the multimodal clustering achieved by the algorithm with seven clusters and the number of clusters has been selected by the stopping criteria of the iterative merging phase. While the algorithm has clustered spatially close neurons in modality 1 together most of the times, the clustering of neurons of spatially apart from each other into the same cluster is due to the co-occurrence influence of the second modality. Similarly, Figure 6.3 and Figure 6.4 displays the k-means algorithm based unimodal clustering and multimodal clustering for modality 2, respectively.

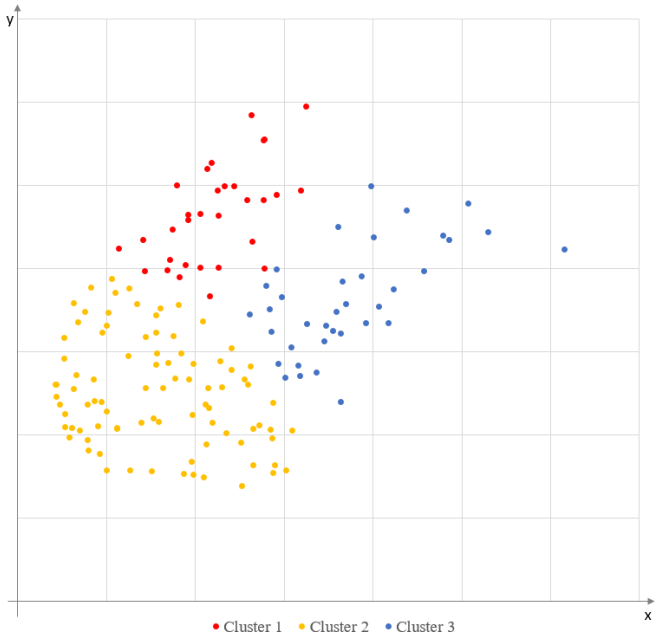

Figure 6.3: k-means algorithm based unimodal clustering for modality 2. Parameter k set to 3 for three clusters based on optimal DB-Index value.

6.3 Further Application Areas

Due to the speedups and scalability achieved using distributed computing, proposed multimodal clustering is well suited for application requiring real-time processing of multimodal data. The possible application areas span any scenario where the environment is comprehensively captured by multiple sensory modalities facilitating a *digital environment*.

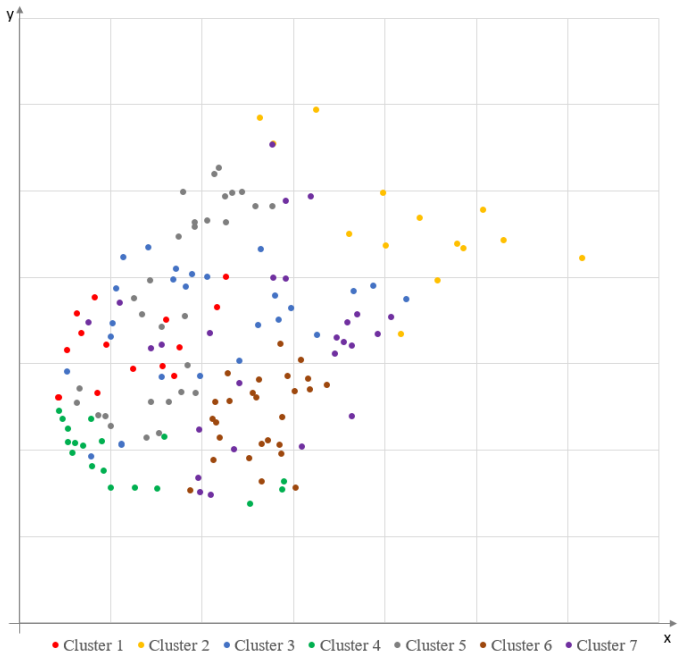

Figure 6.4 Multimodal clustering for modality 2. The number of clusters automatically selected by the multimodal clustering algorithm.

Human gesture recognition is one such application area that can benefit from multimodal perceptual mechanisms. Human gesture recognition is of interest for different fields of research including virtual, augment and mixed reality. Devices such as Microsoft Kinect, which carry depth-sensors in addition to traditional 2D sensors (e.g. RGB cameras), have become affordable and commonplace during the last decade. These sensors allow capturing the human gestures more robustly under varying lighting conditions and provide a closer representation of the environment (Parisi et al., 2017). Proposed multimodal clustering algorithm could be applied for such environments facilitating modelling the environment holistically using multiple modalities.

As discussed in Chapter 3, the sensor network of a city-wide traffic management system would consist of various types of sensors placed around the city to understand the traffic situations at various levels. Examples include Bluetooth sensors placed in road intersections, motion sensors placed near pedestrian paths, and video cameras capturing vehicles and pedestrian movement. Data captured across multiple modalities, multiple sources, and multiple sites at high frequency provides extensive coverage of the traffic situation of the city and creates a comprehensive

digital environment. Such a *digital environment* would be ideal for the proposed algorithm to operate on and generate a holistic *impression* of the city-wide traffic situation.

In (Jayaratne et al., 2018) and (Jayaratne et al., 2019), two of the publications arising from this thesis, we identified active perception in autonomous robotic systems as one the major application domain for the proposed algorithm. Active perception is a research area concerned with the accurate environment perception for robotic systems (including humanoid robots, unmanned aerial vehicle, autonomous cars) and determines their ability to successfully operate in the environment. The perceptions and actions of the robotic systems work in a reinforcement cycle where perception supports making necessary adjustments which in turn enables the more accurate perception of the surrounding. A large body of research under active perception is concerned with improving this cycle allowing improved autonomy for robotic systems (Chen et al., 2011; Eidenberger & Scharinger, 2010; Martinez-Hernandez et al., 2017, 2013). As emphasised in (Ferreira et al., 2013; Liu et al., 2017; Yongmian Zhang & Ji, 2006), active perception systems are complex systems characterised by 1) multimodality: multiple heterogeneous sensors with different degrees of reliability are used to sense the environment, 2) efficiency: the need for shorter processing times to make decisions in real-time, 3) evolution: the world situation is changing over time requiring adapting/retraining the decision models.

The fusion of different sensory modalities is key to the accurate perception in creating a holistic representation of the surrounding (Parisi et al., 2017). The fusion of multiple sensor modalities helps with noise reduction, disambiguating any ambiguities in sensor data and incorporating complementary information carried by individual modalities in generating a robust *impression* about the surrounding (Atrey et al., 2010; Suk et al., 2014). For example, an autonomous navigational system that requires identification of known or unknown objects/entities in the environment would utilise sensors such as imaging sensors, acoustic sensors and proximity sensors. The key characteristics of a target would be provided by different sensors such as imaging sensors for shape and size and sonar for speed.

Autonomous robotic systems need to derive efficient representations of the surrounding from the sensory inputs for them to effectively perceive the environment (Mnih et al., 2015). The efficient online fusion of data from multiple sensory modalities and forming a robust *impression* about the surrounding facilitates responding promptly when dealing with real-world situations (Luo et al., 1988). Moreover, the time taken to adapt/retrain the decision models in response to changes in the environment needs to be reasonable so that the decisions are not made with outdated models. Training algorithms that can be parallelised to take advantage of parallel and distributed computing are essential to training decision models with vast training datasets.

Further, given the challenges in obtaining labelling for high-volume and high-velocity sensory data, it is essential that such *impression* generation mechanisms use unsupervised machine learning techniques for it to be practically useful (Najafabadi et al., 2015). Among a large body of literature on multimodal perception, only a handful of research focuses on unsupervised machine learning and the mechanism proposed in this thesis is based on unsupervised machine learning for it be practical for real-life scenarios. Multimodal clustering uses natural regularities in patterns observed across multiple modalities and enriches each modality with information from other co-occurring modalities to achieve a multimodal clustering. The clusters can be used for online prediction and offline action planning of the robot by matching the current sensory inputs. The algorithm uses unsupervised machine learning, hence can be used with any multimodal dataset without being restricted by the labelling of the dataset. These characteristics make the proposed scalable *impression* generation algorithm suitable for perception tasks in autonomous robotic systems.

6.4 Chapter Summary

This chapter proposed the distributed implementation of the multimodal clustering algorithm, the final piece of the jigsaw puzzle. Implementing the distributed computing architecture for *impression* generation proposed in Chapter 4, the distributed multimodal clustering algorithm proposed in this chapter, along with the DGSOM algorithms proposed in Chapter 5 constitutes

the overall distributed implementation. The overall system implemented on Apache Spark was demonstrated on a physical activity monitoring application to tackle the multimodality arising from the use of various sensor modalities and large volumes of data resulting in frequent capture of sensor readings. The next chapter concludes the research endeavour documented in this thesis.

Chapter 7
Conclusion

This thesis explored the development of multimodal perceptual mechanisms for artificial intelligent applications inspired by the findings of psychological, behavioural and neurobiological studies on human multimodal perception. This investigation was conducted with the objective of designing and developing an artificial *impression* generation mechanism drawing upon the organization and functionality of the human brain to facilitate artificial counterparts to be autonomous and proactive by forming a holistic understanding of the *digital environment*.

With a brief introduction to the research area, Chapter 1 outlined the motivation for undertaking the research presented in this thesis. The main research question was formalised while identifying sub-research questions in neurobiology, psychology, unsupervised machine learning, parallel and distributed computing. A comprehensive survey of related literature in multiple research disciplines was presented in Chapter 2. Chapter 3 laid out our premise on sensation and perception in humans while proposing an artificial model for generating artificial *impressions* on *digital environments.* The implementation details of the proposed artificial model were presented in Chapter 4, along with the empirical evaluation of the model on an

audio-visual dataset. Chapter 4 concluded by highlighting the necessity of generating efficient representations from multimodal data sources in most online application scenarios and presenting a distributed architecture for online multimodal sensory fusion. Chapter 5 introduced a distributed SOM algorithm, its adaptation to three distributed computing paradigms, implementation on respective platforms, and empirical evaluation of performance while Chapter 6 presented the detail of multimodal clustering component of the proposed architecture implemented for distributed computing. Moreover, the overall distributed implementation was demonstrated using a case study from the physical activity monitoring domain.

Finally, this chapter concludes this thesis by discussing the answers to the research questions formulated in Chapter 1. The main research question addressed in this thesis is,

> *Inspired by sensing and perception mechanisms in the brain, how can unsupervised machine learning algorithms be developed for holistic data representation and fusion in digital environments?*

Further, a summary of research contributions is discussed while presenting future research directions in this chapter.

7.1 Summary of Research Contributions

The overall contribution of this research is the advancement of knowledge on the development of multimodal perceptual mechanisms for artificial intelligent applications inspired by findings of psychological, behavioural and neurobiological studies on human multimodal perception. This allows artificial intelligent applications to be autonomous and proactive by forming a holistic understanding of the *digital environment*.

Summaries of the research contributions presented in each chapter are outlined below.

Chapter 2 developed a comprehensive base of literature relating to multimodal sensory perception, including evidence of multimodal perception, theories on perceptual binding, artificial models of multimodal fusion and applications of such proposed models, by

systematically analysing past research. With respect to the natural systems, evidence of multimodal sensory perception in humans were reviewed from two aspects, 1) the psychological evidence and models of multimodal perception, 2) neuro-biological studies of multimodal perception and their findings. The behavioural and psychological studies have demonstrated fascinating crossmodal effects between sensory modalities in the form of crossmodal influence, crossmodal recalibrations and medical conditions demonstrating the crossmodal interplay between sensory modalities. The chapter discussed various theories on how humans perceive the information captured from different sensory modalities as a coherent event/object including feature integration theory, synchronisation theory and the theories on the role of attention in perceptual binding. With respect to the artificial systems, we reviewed models implementing multimodal sensory processing categorised under 1) biologically inspired artificial neural network models that implement known neurophysiological characteristics, 2) models that view multisensory fusion as Bayesian inference.

Chapter 3 presented a novel artificial model consisting of conceptual, architectural and computational components as the basis for generating artificial impressions on digital environments. We laid out our premise on sensation and perception in humans and discussed how humans construct the state of the external environment from the multimodal sensory inputs by forming what we called a coherent *impression* about the external world. This culminated with a theoretical contribution of the artificial model inspired by the human neocortex for generating artificial impression on digital environments consisting of a conceptual model, an architectural model as well as a computational model.

The proposed artificial model was implemented and empirically evaluated in Chapter 4. Proposed multi-layered architectural model was implemented with artificial cortical areas modelled by the GSOM algorithm and multimodal clustering algorithm allowing for the fusion of modalities based on their co-occurrence relationships. The model was empirically evaluated with a multimodal (audio-visual) dataset measuring the accuracy gains attained by multimodal fusion. Further, the process of hierarchical cluster formation was analysed for a better

understanding of the process while the optimal parameter values for multimodal interaction were demonstrated. Highlighting that most of the multimodal applications are required to derive efficient representations from the multimodal sensory inputs to perceive the environment effectively, *Chapter 4 concluded with a theoretical contribution: a distributed architecture for improving the efficiency and scalability of the multimodal fusion algorithm in order to provide results under acceptable computing times*.

A distributed SOM algorithm was proposed, adapted to three contemporary distributed computing paradigms, implemented on their respective platforms and empirically evaluated in Chapter 5. The adaptations are for MapReduce, BSP and RDD paradigms while implementations are on Apache Hadoop, Apache Hama and Apache Spark, respectively. This fulfils the distributed self-organizing component of the proposed distributed architecture. The empirical evaluation on three benchmarks, real-life datasets from various domains (image, audio and power consumption) demonstrate super-linear speedup compared to the serial SOM, highlighting applicability irrespective of the data modality.

Multimodal clustering algorithm was implemented to use distributed computing in Chapter 6. Computationally heavy multimodal distance calculation and multimodal clustering algorithm were adapted to use distributed computing to support the efficient online fusion of data from multiple sensory modalities. *Moreover, the overall distributed implementation was demonstrated using a case study from the physical activity monitoring domain, which suffers from large data volumes and multimodality*.

7.2 Addressing the Research Questions

The main research question composed of four sub research questions: questions on biological inspiration for multimodal data fusion, questions on development of unsupervised machine learning algorithms for multimodal data fusion, questions on adapting the developed algorithms for distributed computing to work efficiently and at scale, and validation of algorithms on real-life datasets.

7.2.1 Research Questions on Biological Inspiration for Multimodal Data Fusion

1. *What psychological evidence of multimodal perception has been observed and what psychological theories have been proposed?*

   There is a large body of psychological evidence of multimodal perception in humans. The evidence was analysed under three boards categories, 1) evidence of crossmodal influence, 2) evidence of crossmodal calibration, and 3) evidence from medical cases. Evidence of crossmodal influences includes the visual influence of proprioception and the visual influence on audition, which have been demonstrated with the ventriloquism effect (Howard & Templeton, 1966) and McGurk effect (Mcgurk & Macdonald, 1976). Temporary crossmodal recalibrations have been demonstrated on visual-proprioception and visual-audition modality combinations when exposed to incongruent multimodal stimuli. Furthermore, several special medical conditions have been identified as a source of evidence for multimodal dynamics in the human brain, including Synaesthesia (Ramachandran & Hubbard, 2001) and Prosopagnosia (De Gelder et al., 2000).

   There are two major theories on how humans perceive multimodal sensory information as a coherent event, commonly known as the binding problem. Feature integration theory (Treisman & Gelade, 1980) suggests that object's location mediates the binding of the features, and the attention is the "glue" that combines these features. On the other hand, the synchronisation theory (Engel et al., 1999; Varela, 1995) suggests that the neuronal activation in various parts of the brain induced by the same object are in synchrony and this synchronisation is the basis of binding which leads to the perception of these objects, opposed to individual features.

2. *What theories on the organization of the human brain to support knowledge representation have been put forward?*

Domain-specific hypothesis (Caramazza & Shelton, 1998), which suggests that human knowledge is organized in a category-specific manner at the cognitive level, has been proposed based on evidence from patients who have conceptual level impairments that are specific to a particular category such as animals, plants, conspecifics or artefacts (Tyler & Moss, 2001; Capitani et al., 2003; Mahon & Caramazza, 2011). More recently, neuroimaging studies have re-evaluated the category-specific organization of the neocortex (Martin, 2007) and concluded a more intricate organization where different aspects of an object - such as what it looks like, how it is used, and how it moves - are coded in different parts of the neural circuitry and object categories such as animals, plants and tools have a distributed, partially distinct sensory-based coding. Hence, the object concepts *emerge* from activity in aspect-based regions of the brain.

The aspect-based coding and category-based organization provided the inspiration for developing the structure/organization of the conceptual model. In our conceptual model, regions in the outer layer represent different modality-specific primary cortical areas where different aspects of an object received via different modalities are mapped. The innermost layer of our conceptual model consists of categories formed by the organization of these aspect-based coding into meaningful categories – which is analogous to the emergence of object concepts – using the multimodal clustering algorithm.

3. *What theories on the dynamics of the human brain to support multimodal perception have been proposed?*

   The experimental evidence on the reentry mechanism suggests that it is one of the essential mechanisms supporting multimodal integration in the mammalian brain (Edelman & Gally, 2013). The neurons belonging to different layers within a cortical area form a dense columnar array and neurons belonging to different cortical areas are reciprocally interconnected by reentrant networks of excitatory axons (Markov et al., 2014). Evidence suggests that synchronous exchanges of signals among neuronal

groups in dispersed cortical areas correlate with, and bind together, the multiple but distinguishable features of unified, conscious scenes (Edelman & Gally, 2013).

Reentry mechanism shaped the dynamics of our artificial impression generation mechanism. In our conceptual model, reentry mechanism inspired the inter-modal associative connections which connect different regions in the outer layer onto regions in the inner layer. They capture the co-activation of neuron pairs in different regions, and these co-activation profiles are used to unearth the intricate relationships between co-occurring modalities.

7.2.2 Research Questions on Development of Unsupervised Machine Learning Algorithms for Multimodal Data Fusion

1. *Are there any unsupervised learning algorithms that have been proposed for multimodal data representation and fusion; are there any limitations?*

   There are several artificial neural models, both at the single neuronal level and neural network level, which have been proposed to model the neurophysiological mechanism of the cortex that is responsible for fusing multimodal stimuli. A number of noteworthy models, hierarchical feedforward models such as hierarchical GWR (Parisi et al., 2017), hierarchical GSOM (Fonseka, 2012), fusionART (Tan et al., 2007), hierarchical model of superior colliculus (Magosso et al., 2008; Ursino et al., 2009) and models with inter-area feedback proposed by Magosso et al. (2012) and Hoshino (2011), were analysed. However, most of these models focus only on modelling the multimodal dynamics of the human brain rather than on the application of the developed models to real-world problems, let alone the challenges posed by the vast amount of data generated in *digital environments*. Moreover, some are primarily focused on the supervised paradigm; multimodal fusion for unsupervised environments are still unresolved and an ongoing problem (Dasarathy, 2006).

   These limitations were addressed in this thesis. Multimodal representation and fusion mechanisms presented in this thesis rely on unsupervised algorithms for both

representation and fusion aspects. Moreover, it was proposed with real-world applications in mind and later adapted to distributed computing to support the vast amount of data generated in *digital environments*. Proposed algorithms were adapted to multiple contemporary distributed computing paradigms and implemented on respective distributed computing platforms. Further, the implementation was demonstrated on a real-world dataset from physical activity monitoring domain.

2. *Can the principles of self-organization be used to realise unsupervised learning for developing artificial cortical areas that represent information from individual modalities?*

   Self-organization has long been viewed as a central mechanism of nature that organizes selected parts of a system to promote a specific function (Camazine et al., 2003). The self-organization-based algorithms that have been used as the basis for multimodal fusion models include SOM (Kohonen, 1990), GSOM (Alahakoon et al., 2000), ART (Grossberg, 1982, 2013), and NG (Martinetz & Schulten, 1991). In the proposed multisensory self-organizing neural architecture, individual topographic maps are trained using the GSOM algorithm and form representations of their respective modality. The GSOM algorithm was chosen for modelling the individual modalities due to the self-organizing characteristic and the dynamic structure adaptive nature of the algorithm.

3. *How can co-occurrence of neuronal activations across modalities be used for developing a multimodal fusion mechanism?*

   Sensory data from different modalities, mediated by the time dimension, carry the information about the same underlying event or situation. Inter-modal associative links between modality-specific topographic maps capture the co-occurrence relationships between the modalities. The co-occurrence of neuronal activations is the basis of multimodal fusion proposed. Co-occurrence is accounted for in the multimodal distance metric proposed, and it is in turn used as the distance metric of the multimodal

clustering algorithm which organizes neurons of topographic maps into meaningful clusters.

7.2.3 Research Questions on Adapting the Developed Algorithms for Distributed Computing for Efficiently and Scale

1. *What is an appropriate distributed architecture for improving the efficiency and scalability of the multimodal fusion algorithm in order to provide results under acceptable computing times?*

   Proposed distributed architecture for improving the efficiency and scalability of the multimodal fusion algorithm comprises of two major modules. The first module is the distributed GSOM training module implementing a distributed variant of the GSOM algorithm for training GSOM maps of individual modalities. The second module is the multimodal clustering module, which performs multimodal clustering over the modality-specific GSOMs generated by the first module.

2. *How can self-organizing maps that are used to represent information from individual modalities be implemented for distributed computing paradigms, MapReduce, BSP and RDD?*

   Proposed DGSOM algorithm uses data parallelism to train modality-specific GSOM maps. The algorithm consists of three major tasks for the distributed GSOM training, 1) data partitioning 2) distributed GSOM training, and 3) merging GSOM maps. The workflow of the DGSOM algorithm is adapted for MapReduce, BSP and RDD distribute computing paradigms and implemented on well-established distributed computing platforms Apache Hadoop, Hama and Spark, respectively. Empirical evaluations using several benchmarking and real-life data sets saw all three adaptations achieve super-linear speedup compared to the serial GSOM implementation. These speedups facilitate generating efficient representations from large multimodal data sources which is essential in most online application scenarios.

7.2.4 Research Questions on Validation of Algorithms on Real-life Datasets

1. *How can the improvement in multimodal representation accuracy of the proposed multimodal fusion algorithm be evaluated with appropriate benchmark datasets?*

   Clustering quality was identified as an appropriate proxy for the quality/accuracy of multimodal representation generated by the multimodal fusion algorithm. More specifically, external cluster evaluation metrics are appropriate given they rely on class labels for the accuracy. External cluster evaluation measures, F1, Rand measure (Rand, 1971), Dice index (Dice, 1945), cluster purity and NMI, were the metrics used.

   Tulips1 audio-visual dataset (Movellan, 1995) and PAMAP2 physical activity monitoring dataset (Reiss & Stricker, 2012) were used as the multimodal benchmark datasets. These experiments demonstrated the superiority of multimodal representation (achieved with multimodal clustering) compared to the baseline unimodal representation (achieved with k-means clustering) in terms of the above cluster quality evaluation measures.

2. *How can the efficiency gains attained by the distributed implementations be evaluated?*

   Total elapsed time is the primary performance measure used as it includes the time from job submission to the end of execution. DGSOM implementations on Apache Hadoop, Hama and Spark were evaluated in terms of the elapsed time and compared with the elapsed time of the same process running serially. Three implementations achieved super-linear speedups compared to the serial GSOM algorithm using several benchmarking and real-life data sets.

7.3 Future Directions

Further directions to the proposed *impression* generation mechanism include algorithmic improvements as well as applications.

While the distributed GSOM algorithm was presented for supporting multimodal *impression* generation for large datasets, it is not limited to that and can be applied in applications such as visual analytics and segmentation tasks where traditionally a SOM/GSOM would be used. Currently, in the implementation of the distributed GSOM, Sammon's projection algorithm is used for the merging of the individual maps. While the algorithm achieves the desired merging task by projecting neurons while preserving topology, the algorithm does not place neurons in a grid-like structure of SOM/GSOM. For applications such as visual analytics, the grid-like structure of SOM/GSOM is desired and one future direction would be to develop a novel heuristic-based merging algorithm to place neurons in grid-like structure while preserving the topology. A further limitation of Sammon's projection algorithm is that it operates serially. Hence, another future direction would be to target the novel heuristic-based merging algorithm for parallel programming to generate the final merged GSOM map concurrently. This would improve the efficiency of the DGSOM algorithm by using distributed computing throughout all stages.

While the multisensory neural architecture is generalisable to any number of modalities, it was implemented and demonstrated with two modalities. Multisensory neural implementation can be extended to more than two modalities effectively fusing all the required sensor inputs to obtain a robust multimodal representation.

Autonomous robotic systems need to derive efficient representations of the surrounding from the sensor inputs for them to perceive the environment effectively. The efficient online fusion of data from multiple sensory modalities facilitates responding promptly when dealing with real-world situations. Multimodal perception and efficient fusion supported by distributed computing make the research proposed in this thesis ideal for autonomous robotics applications.

While the multimodal neural architecture was demonstrated on benchmark datasets, we would like to embed it on a robotic platform allowing efficient representations of the surrounding from the sensor data.

Vita

Publications arising from this thesis include,

Jayaratne, M., Alahakoon, D., De Silva, D., & Yu, X. (2017). Apache spark based distributed self-organizing map algorithm for sensor data analysis. IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society, 8343–8349. https://doi.org/10.1109/IECON.2017.8217465

Jayaratne, M., Alahakoon, D., De Silva, D., & Yu, X. (2018). Bio-Inspired Multisensory Fusion for Autonomous Robots. IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society, 3090–3095. https://doi.org/10.1109/IECON.2018.8592809

Jayaratne, M., De Silva, D., & Alahakoon, D. (2019). Unsupervised Machine Learning Based Scalable Fusion for Active Perception. IEEE Transactions on Automation Science and Engineering, 16(4), 1653–1663. https://doi.org/10.1109/TASE.2019.2910508

Jayaratne, M., Alahakoon, D., & De Silva, D. (conditionally accepted). Unsupervised Skill Transfer Learning for Autonomous Robots using Distributed Growing Self Organizing Maps. Robotics and Autonomous Systems

References

Alahakoon, D., Halgamuge, S., & Srinivasan, B. (2000). Dynamic self-organizing maps with controlled growth for knowledge discovery. *IEEE Transactions on Neural Networks*, *11*(3), 601–614. https://doi.org/10.1109/72.846732

Allam, Z., & Dhunny, Z. A. (2019). On big data, artificial intelligence and smart cities. *Cities*, *89*, 80–91. https://doi.org/10.1016/j.cities.2019.01.032

Anguita, D., Ghio, A., Oneto, L., Parra Perez, X., Ortiz, R., & Luis, J. (2013). A public domain dataset for human activity recognition using smartphones. *Proceedings of the 21st International European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, *3*, 437–442.

Arons, B. (1992). A review of the cocktail party effect. *Journal of the American Voice I/O Society*, *12*(7), 35–50.

Atrey, P. K., Hossain, M. A., El Saddik, A., & Kankanhalli, M. S. (2010). Multimodal fusion for multimedia analysis: A survey. *Multimedia Systems*, *16*(6), 345–379. https://doi.org/10.1007/s00530-010-0182-0

Attal, F., Mohammed, S., Dedabrishvili, M., Chamroukhi, F., Oukhellou, L., & Amirat, Y. (2015). Physical Human Activity Recognition Using Wearable Sensors. *Sensors*, *15*(12), 31314–31338. https://doi.org/10.3390/s151229858

Aughey, R. J. (2011). Applications of GPS Technologies to Field Sports. *International Journal of Sports Physiology and Performance*, *6*(3), 295–310. https://doi.org/10.1123/ijspp.6.3.295

Bagalà, F., Becker, C., Cappello, A., Chiari, L., Aminian, K., Hausdorff, J. M., Zijlstra, W., & Klenk, J. (2012). Evaluation of Accelerometer-Based Fall Detection Algorithms on Real-World Falls. *PLOS ONE*, *7*(5), e37062. https://doi.org/10.1371/journal.pone.0037062

Baldwin, J. F., Martin, T. P., & Saeed, M. (1999). Automatic Computer Lip-Reading Using Fuzzy Set Theory. *AVSP'99-International Conference on Auditory-Visual Speech Processing*.

Baron-Cohen, S., & Harrison, J. E. (Eds.). (1997). *Synaesthesia: Classic and contemporary readings*. Blackwell Publishing.

Battaglia, P. W., Jacobs, R. A., & Aslin, R. N. (2003). Bayesian integration of visual and auditory signals for spatial localization. *Journal of the Optical Society of America. A*, *20*(7), 1391–1397. https://doi.org/10.1364/JOSAA.20.001391

Beauchamp, M. S. (2005). Statistical criteria in fMRI studies of multisensory integration. *Neuroinformatics*, *3*(2), 93–113. https://doi.org/10.1385/NI:3:2:093

Beauchamp, M. S., Argall, B. D., Bodurka, J., Duyn, J. H., & Martin, A. (2004). Unraveling multisensory integration: Patchy organization within human STS multisensory cortex. *Nature Neuroscience*, *7*(11), 1190–1192. https://doi.org/10.1038/nn1333

Beauchamp, M. S., Nath, A. R., & Pasalar, S. (2010). FMRI-Guided Transcranial Magnetic Stimulation Reveals That the Superior Temporal Sulcus Is a Cortical Locus of the McGurk Effect. *Journal of Neuroscience*, *30*(7), 2414–2417. https://doi.org/10.1523/JNEUROSCI.4865-09.2010

Bell, A. H., Meredith, M. A., Van Opstal, A. J., & Munoz, D. P. (2005). Crossmodal integration in the primate superior colliculus underlying the preparation and initiation of saccadic eye movements. *Journal of Neurophysiology*, *93*(6), 3659–3673. https://doi.org/10.1152/jn.01214.2004

Berlin, J. E., Storti, K. L., & Brach, J. S. (2006). Using Activity Monitors to Measure Physical Activity in Free-Living Conditions. *Physical Therapy*, *86*(8), 1137–1145. https://doi.org/10.1093/ptj/86.8.1137

Berman, N., & Hunt, R. K. (1975). Visual projections to the optic tecta in Xenopus after partial extirpation of the embryonic eye. *Journal of Comparative Neurology*, *162*(1), 23–41. https://doi.org/10.1002/cne.901620104

Bertelson, P., & De Gelder, B. (2004). The psychology of multimodal perception. In C. Spence & J. Driver (Eds.), *Crossmodal space and crossmodal attention* (pp. 141–177). Oxford University Press.

Bertelson, P., Pavani, F., Ladavas, E., Vroomen, J., & de Gelder, B. (2000). Ventriloquism in patients with unilateral visual neglect. *Neuropsychologia*, *38*(12), 1634–1642. https://doi.org/10.1016/S0028-3932(00)00067-1

Bertin-Mahieux, T., Ellis, D. P. W., Whitman, B., & Lamere, P. (2011). The Million Song Dataset. *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 591–596. https://doi.org/10.7916/D8NZ8J07

Brachman, R. J., & Levesque, H. J. (1985). *Readings in Knowledge Representation*. Morgan Kaufmann Publishers Inc.

Bult, J. H. F., de Wijk, R. A., & Hummel, T. (2007). Investigations on multimodal sensory integration: Texture, taste, and ortho- and retronasal olfactory stimuli in concert. *Neuroscience Letters*, *411*(1), 6–10. https://doi.org/10.1016/j.neulet.2006.09.036

Burr, D., & Gori, M. (2012). Multisensory Integration Develops Late in Humans. In M. Wallace & M. Murray (Eds.), *The Neural Bases of Multisensory Processes*. CRC Press/Taylor & Francis.

Calvert, G. A., Campbell, R., & Brammer, M. J. (2000). Evidence from functional magnetic resonance imaging of crossmodal binding in the human heteromodal cortex. *Current Biology*, *10*(11), 649–657. https://doi.org/10.1016/S0960-9822(00)00513-3

Calvert, G. A., Spence, C., & Stein, B. E. (2004). *The handbook of multisensory processes*. MIT Press.

Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Bonabeau, E., & Theraula, G. (2003). *Self-organization in biological systems*. Princeton University Press.

Capitani, E., Laiacona, M., Mahon, B., & Caramazza, A. (2003). What are the facts of semantic category-specific deficits? A critical review of the clinical evidence. *Cognitive Neuropsychology*, *20*(3), 213–261. https://doi.org/10.1080/02643290244000266

Caramazza, A., & Shelton, J. R. (1998). Domain-specific knowledge systems in the brain the animate-inanimate distinction. *Journal of Cognitive Neuroscience*, *10*(1), 1–34. https://doi.org/10.1162/089892998563752

Carpenter, G. A., & Grossberg, S. (1987a). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, *37*(1), 54–115. https://doi.org/10.1016/S0734-189X(87)80014-2

Carpenter, G. A., & Grossberg, S. (1987b). ART 2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, *26*(23), 4919–4930. https://doi.org/10.1364/AO.26.004919

Carvalho, M. M., Raj, A., & Drakunov, S. V. (2001). *Hierarchical Human-in-the loop Control Systems: An Application for Tactile Interfaces and Adjustable Autonomy*. (SAE Technical Paper No. 2001-01–3854). SAE International. https://doi.org/10.4271/2001-01-3854

Chan, C.-K. K., Hsu, A. L., Tang, S.-L., & Halgamuge, S. K. (2008). Using growing self-organising maps to improve the binning process in environmental whole-genome shotgun sequencing. *Journal of Biomedicine & Biotechnology*, *2008*, Article 513701. https://doi.org/10.1155/2008/513701

Chen, S., Li, Y., & Kwok, N. M. (2011). Active Vision in Robotic Systems: A Survey of Recent Developments. *The International Journal of Robotic Research*, *30*(11), 1343–1377. https://doi.org/10.1177/0278364911410755

Cheng, F., Wang, S.-L., & Liew, A. W.-C. (2018). Visual speaker authentication with random prompt texts by a dual-task CNN framework. *Pattern Recognition*, *83*, 340–352. https://doi.org/10.1016/j.patcog.2018.06.005

Choe, C. S., Welch, R. B., Gilford, R. M., & Juola, J. F. (1975). The "ventriloquist effect": Visual dominance or response bias? *Perception & Psychophysics*, *18*(1), 55–60. https://doi.org/10.3758/BF03199367

Chu, D., Huttenlocher, P. R., Levin, D. N., & Towle, V. L. (2000). Reorganization of the Hand Somatosensory Cortex Following Perinatal Unilateral Brain Injury. *Neuropediatrics*, *31*(2), 63–69. https://doi.org/10.1055/s-2000-7475

Coen, M. H. (2005). Cross-modal clustering. *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05)*, *2*, 932–937.

Copeland, J. L., & Esliger, D. W. (2009). Accelerometer Assessment of Physical Activity in Active, Healthy Older Adults. *Journal of Aging and Physical Activity*, *17*(1), 17–30. https://doi.org/10.1123/japa.17.1.17

Cornacchia, M., Ozcan, K., Zheng, Y., & Velipasalar, S. (2017). A Survey on Activity Detection and Classification Using Wearable Sensors. *IEEE Sensors Journal*, *17*(2), 386–403. https://doi.org/10.1109/JSEN.2016.2628346

Cuppini, C., Ursino, M., Magosso, E., Rowland, B. A., & Stein, B. E. (2010). An emergent model of multisensory integration in superior colliculus neurons. *Frontiers in Integrative Neuroscience*, *4*, 1–15. https://doi.org/10.3389/fnint.2010.00006

Dasarathy, B. V. (2006). Identity fusion in unsupervised environments. *Information Fusion*, *7*(2), 157–160. https://doi.org/10.1016/j.inffus.2006.01.003

Dautenhahn, K. (1998). The Art of Designing Socially Intelligent Agents: Science, Fiction, and the Human in the Loop. *Applied Artificial Intelligence*, *12*(7–8), 573–617. https://doi.org/10.1080/088395198117550

Davies, D. L., & Bouldin, D. W. (1979). A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-1*(2), 224–227. https://doi.org/10.1109/TPAMI.1979.4766909

De Gelder, B., & Bertelson, P. (2003). Multisensory integration, perception and ecological validity. *Trends in Cognitive Sciences*, *7*(10), 460–467. https://doi.org/10.1016/j.tics.2003.08.014

De Gelder, B., Pourtois, G., Vroomen, J., & Bachoud-Lévi, A.-C. (2000). Covert Processing of Faces in Prosopagnosia Is Restricted to Facial Expressions: Evidence from Cross-

Modal Bias. *Brain and Cognition*, *44*(3), 425–444. https://doi.org/10.1006/brcg.1999.1203

De Silva, D., Yu, X., Alahakoon, D., & Holmes, G. (2011). A Data Mining Framework for Electricity Consumption Analysis From Meter Data. *IEEE Transactions on Industrial Informatics*, *7*(3), 399–407. https://doi.org/10.1109/TII.2011.2158844

Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified Data Processing on Large Clusters. *Communications of the ACM*, *51*(1), 107–113. https://doi.org/10.1145/1327452.1327492

Di Lollo, V. (2012). The feature-binding problem is an ill-posed problem. *Trends in Cognitive Sciences*, *16*(6), 317–321. https://doi.org/10.1016/j.tics.2012.04.007

Dice, L. R. (1945). Measures of the Amount of Ecologic Association Between Species. *Ecology*, *26*(3), 297–302. https://doi.org/10.2307/1932409

Dittenbach, M., Merkl, D., & Rauber, A. (2000). The growing hierarchical self-organizing map. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, *6*, 15–19. https://doi.org/10.1109/IJCNN.2000.859366

Doherty, A., Jackson, D., Hammerla, N., Plötz, T., Olivier, P., Granat, M. H., White, T., Hees, V. T. van, Trenell, M. I., Owen, C. G., Preece, S. J., Gillions, R., Sheard, S., Peakman, T., Brage, S., & Wareham, N. J. (2017). Large Scale Population Assessment of Physical Activity Using Wrist Worn Accelerometers: The UK Biobank Study. *PLOS ONE*, *12*(2), e0169649. https://doi.org/10.1371/journal.pone.0169649

Dong, Y., Mihalas, S., Qiu, F., Von der Heydt, R., & Niebur, E. (2008). Synchrony and the binding problem in macaque visual cortex. *Journal of Vision*, *8*(7), 30–30. https://doi.org/10.1167/8.7.30

Dua, D., & Graff, C. (2017). *UCI Machine Learning Repository*. http://archive.ics.uci.edu/ml

Edelman, G. M. (1993). Neural Darwinism: Selection and reentrant signaling in higher brain function. *Neuron*, *10*(2), 115–125. https://doi.org/10.1016/0896-6273(93)90304-A

Edelman, G. M., & Gally, J. A. (2013). Reentry: A key mechanism for integration of brain function. *Frontiers in Integrative Neuroscience*, *7*, Article 63. https://doi.org/10.3389/fnint.2013.00063

Edelman, G. M., & Mountcastle, V. B. (1978). *The mindful brain: Cortical organization and the group-selective theory of higher brain function*. Massachusetts Institute of Technology Press.

Eidenberger, R., & Scharinger, J. (2010). Active perception and scene modeling by planning with probabilistic 6D object poses. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1036–1043. https://doi.org/10.1109/IROS.2010.5651927

Engel, A. K., Fries, P., König, P., Brecht, M., & Singer, W. (1999). Temporal Binding, Binocular Rivalry, and Consciousness. *Consciousness and Cognition*, *8*(2), 128–151. https://doi.org/10.1006/ccog.1999.0389

Falcone, R., & Castelfranchi, C. (2001). The human in the loop of a delegated agent: The theory of adjustable social autonomy. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *31*(5), 406–418. https://doi.org/10.1109/3468.952715

Ferreira, J., Lobo, J., Bessiere, P., Castelo-Branco, M., & Dias, J. (2013). A Bayesian framework for active artificial perception. *IEEE Transactions on Cybernetics*, *43*(2), 699–711. https://doi.org/10.1109/TSMCB.2012.2214477

Fonseka, A. (2012). *Bio-inspired approach for information fusion.* [Thesis, Monash University]. http://arrow.monash.edu.au/vital/access/manager/Repository/monash:120134

Fonseka, A., Alahakoon, D., & Bedingfield, S. (2011). GSOM sequence: An unsupervised dynamic approach for knowledge discovery in temporal data. *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, 232–238. https://doi.org/10.1109/CIDM.2011.5949456

Fort, J.-C., Letremy, P., & Cottrell, M. (2002). Advantages and drawbacks of the Batch Kohonen algorithm. In M. Verleysen (Ed.), *Proceedings of the 10th European Symposium on Neural Networks* (pp. 223–230).

Foxe, J. J., Wylie, G. R., Martinez, A., Schroeder, C. E., Javitt, D. C., Guilfoyle, D., Ritter, W., & Murray, M. M. (2002). Auditory-Somatosensory Multisensory Processing in Auditory Association Cortex: An fMRI Study. *Journal of Neurophysiology*, *88*(1), 540–543. https://doi.org/10.1152/jn.2002.88.1.540

Friedman-Hill, Robertson, L. C., & Treisman, A. (1995). Parietal contributions to visual feature binding: Evidence from a patient with bilateral lesions. *Science*, *269*(5225), 853–855. https://doi.org/10.1126/science.7638604

Fritzke, B. (1994). A Growing Neural Gas Network Learns Topologies. *Proceedings of the 7th International Conference on Neural Information Processing Systems*, 625–632.

Fritzke, B. (1997). A Self-Organizing Network that Can Follow Non-stationary Distributions. *International Conference on Artificial Neural Networks*, 613–618. https://doi.org/10.1007/BFb0020222

Galatas, G. (2014). *Multimodal Interaction In Ambient Intelligence Environments Using Speech, Localization And Robotics* [PhD thesis, University of Texas Arlington]. https://rc.library.uta.edu/uta-ir/handle/10106/24776

Ganegedara, H., & Alahakoon, D. (2012). Redundancy reduction in self-organising map merging for scalable data clustering. *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 1–8. https://doi.org/10.1109/IJCNN.2012.6252722

Ganegedara, H., & Alahakoon, D. (2011). Scalable Data Clustering: A Sammon's Projection Based Technique for Merging GSOMs. *Neural Information Processing*, 193–202. https://doi.org/10.1007/978-3-642-24958-7_23

Garabato, D., Dafonte, C., Manteiga, M., Fustes, D., Álvarez, M. A., & Arcay, B. (2015). A distributed learning algorithm for Self-Organizing Maps intended for outlier analysis in the GAIA–ESA mission. *2015 Conference of the International Fuzzy Systems*

Association and the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT-15), 895–901. https://doi.org/10.2991/ifsa-eusflat-15.2015.126

Gazzaniga, M., Ivry, R., & Mangun, G. (2013). *Cognitive Neuroscience: The Biology of the Mind* (4th ed.). W.W.Norton.

Gentile, G., Petkova, V. I., & Ehrsson, H. H. (2010). Integration of Visual and Tactile Signals From the Hand in the Human Brain: An fMRI Study. *Journal of Neurophysiology*, *105*(2), 910–922. https://doi.org/10.1152/jn.00840.2010

Gillmeister, H., & Eimer, M. (2007). Tactile enhancement of auditory detection and perceived loudness. *Brain Research*, *1160*, 58–68. https://doi.org/10.1016/j.brainres.2007.03.041

Goldstein, E. B. (2009). *Sensation and Perception*. Cengage Learning.

Goodhill, G. J., & Xu, J. (2005). The development of retinotectal maps: A review of models based on molecular gradients. *Network: Computation in Neural Systems*, *16*(1), 5–34. https://doi.org/10.1080/09548980500254654

Gorgonio, F. L., & Costa, J. A. F. (2008a). Combining Parallel Self-Organizing Maps and K-Means to Cluster Distributed Data. *11th IEEE International Conference on Computational Science and Engineering Workshops, 2008. CSEWORKSHOPS '08*, 53–58. https://doi.org/10.1109/CSEW.2008.65

Gorgonio, F. L., & Costa, J. A. F. (2008b). Parallel self-organizing maps with application in clustering distributed data. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 3276–3283. https://doi.org/10.1109/IJCNN.2008.4634263

Gray, C. M., König, P., Engel, A. K., & Singer, W. (1989). Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature*, *338*(6213), 334–337. https://doi.org/10.1038/338334a0

Grossberg, S. (1982). How Does a Brain Build a Cognitive Code? In S. Grossberg (Ed.), *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control* (pp. 1–52). Springer, Dordrecht. https://doi.org/10.1007/978-94-009-7758-7_1

Grossberg, S. (2013). Adaptive Resonance Theory: How a brain learns to consciously attend, learn, and recognize a changing world. *Neural Networks*, *37*, 1–47. https://doi.org/10.1016/j.neunet.2012.09.017

Gunasinghe, U., & Alahakoon, D. (2013). The adaptive suffix tree: A space efficient sequence learning algorithm. *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 1–8. https://doi.org/10.1109/IJCNN.2013.6707052

Gunasinghe, U., Alahakoon, D., & Bedingfield, S. (2014). Extraction of high quality k-words for alignment-free sequence comparison. *Journal of Theoretical Biology*, *358*, 31–51. https://doi.org/10.1016/j.jtbi.2014.05.016

Guo, H., Chen, L., Peng, L., & Chen, G. (2016). Wearable Sensor Based Multimodal Human Activity Recognition Exploiting the Diversity of Classifier Ensemble. *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 1112–1123. https://doi.org/10.1145/2971648.2971708

Guru, S. M., Hsu, A., Halgamuge, S., & Fernando, S. (2005). An Extended Growing Self-Organizing Map for Selection of Clusters in Sensor Networks. *International Journal of Distributed Sensor Networks*, *1*(2), 227–243. https://doi.org/10.1080/15501320590966477

Hadgu, A. T., Nigam, A., & Diaz-Aviles, E. (2015). Large-scale learning with AdaGrad on Spark. *2015 IEEE International Conference on Big Data (Big Data)*, 2828–2830. https://doi.org/10.1109/BigData.2015.7364091

Han, Q., Liang, S., & Zhang, H. (2015). Mobile cloud sensing, big data, and 5G networks make an intelligent and smart world. *IEEE Network*, *29*(2), 40–45. https://doi.org/10.1109/MNET.2015.7064901

Hanheide, M., Göbelbecker, M., Horn, G. S., Pronobis, A., Sjöö, K., Aydemir, A., Jensfelt, P., Gretton, C., Dearden, R., Janicek, M., Zender, H., Kruijff, G.-J., Hawes, N., & Wyatt, J. L. (2017). Robot task planning and explanation in open and uncertain worlds. *Artificial Intelligence*, *247*, 119–150. https://doi.org/10.1016/j.artint.2015.08.008

Harris, C. S. (1965). Perceptual adaptation to inverted, reversed, and displaced vision. *Psychological Review*, *72*(6), 419–444. https://doi.org/10.1037/h0022616

Hawkins, J., & Blakeslee, S. (2007). *On Intelligence*. Macmillan.

Hay, J. C., Pick, H. L., & Ikeda, K. (1965). Visual capture produced by prism spectacles. *Psychonomic Science*, *2*(1), 215–216. https://doi.org/10.3758/BF03343413

Hertz, U., & Amedi, A. (2010). Disentangling unisensory and multisensory components in audiovisual integration using a novel multifrequency fMRI spectral analysis. *NeuroImage*, *52*(2), 617–632. https://doi.org/10.1016/j.neuroimage.2010.04.186

Hoshino, O. (2011). Neuronal Responses Below Firing Threshold for Subthreshold Cross-modal Enhancement. *Neural Computation*, *23*(4), 958–983. https://doi.org/10.1162/NECO_a_00096

Howard, I. P., & Templeton, W. B. (1966). *Human Spatial Orientation*. John Wiley & Sons Ltd.

Hsu, A. L., & Halgamuge, S. K. (2003). Enhancement of topology preservation and hierarchical dynamic self-organising maps for data visualisation. *International Journal of Approximate Reasoning*, *32*(2), 259–279. https://doi.org/10.1016/S0888-613X(02)00086-5

Hsu, A. L., Saeed, I., & Halgamuge, S. K. (2009). Dynamic Self-Organising Maps: Theory, Methods and Applications. In A.-E. Hassanien, A. Abraham, A. V. Vasilakos, & W. Pedrycz (Eds.), *Foundations of Computational, Intelligence Volume 1* (pp. 363–379). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-01082-8_14

Hsu, A. L., Tang, S.-L., & Halgamuge, S. K. (2003). An unsupervised hierarchical dynamic self-organizing approach to cancer class discovery and marker gene identification in microarray data. *Bioinformatics*, *19*(16), 2131–2140. https://doi.org/10.1093/bioinformatics/btg296

Huang, J., Zhang, R., Buyya, R., Chen, J., & Wu, Y. (2016). Heads-Join: Efficient Earth Mover's Distance Similarity Joins on Hadoop. *IEEE Transactions on Parallel and Distributed Systems*, *27*(6), 1660–1673. https://doi.org/10.1109/TPDS.2015.2462354

Huiskes, M. J., Thomee, B., & Lew, M. S. (2010). New Trends and Ideas in Visual Concept Detection: The MIR Flickr Retrieval Evaluation Initiative. *Proceedings of the International Conference on Multimedia Information Retrieval*, 527–536. https://doi.org/10.1145/1743384.1743475

Humphries, C., Liebenthal, E., & Binder, J. R. (2010). Tonotopic organization of human auditory cortex. *NeuroImage*, *50*(3), 1202–1211. https://doi.org/10.1016/j.neuroimage.2010.01.046

Huntsberger, T. L., & Ajjimarangsee, P. (1990). Parallel Self-Organizing Feature Maps for Unsupervised Pattern Recognition. *International Journal of General Systems*, *16*(4), 357–372. https://doi.org/10.1080/03081079008935088

Ippoliti, D., & Zhou, X. (2012). A-GHSOM: An adaptive growing hierarchical self organizing map for network anomaly detection. *Journal of Parallel and Distributed Computing*, *72*(12), 1576–1590. https://doi.org/10.1016/j.jpdc.2012.09.004

Isaeva, V. V. (2012). Self-organization in biological systems. *Biology Bulletin*, *39*(2), 110–118. https://doi.org/10.1134/S1062359012020069

Jack, C. E., & Thurlow, W. R. (1973). Effects of Degree of Visual Association and Angle of Displacement on the "Ventriloquism" Effect. *Perceptual and Motor Skills*, *37*(3), 967–979. https://doi.org/10.1177/003151257303700360

Jackson, F. (1977). *Perception: A Representative Theory*. Cambridge University Press.

James, T. W., & Stevenson, R. A. (2012). The Use of fMRI to Assess Multisensory Integration. In M. M. Murray & M. T. Wallace (Eds.), *The Neural Bases of Multisensory Processes*. CRC Press/Taylor & Francis.

Jayaratne, M., Alahakoon, D., De Silva, D., & Yu, X. (2017). Apache spark based distributed self-organizing map algorithm for sensor data analysis. *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 8343–8349. https://doi.org/10.1109/IECON.2017.8217465

Jayaratne, M., Alahakoon, D., De Silva, D., & Yu, X. (2018). Bio-Inspired Multisensory Fusion for Autonomous Robots. *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 3090–3095. https://doi.org/10.1109/IECON.2018.8592809

Jayaratne, M., De Silva, D., & Alahakoon, D. (2019). Unsupervised Machine Learning Based Scalable Fusion for Active Perception. *IEEE Transactions on Automation Science and Engineering*, *16*(4), 1653–1663. https://doi.org/10.1109/TASE.2019.2910508

Kahneman, D. (2011). *Thinking, fast and slow*. Farrar, Straus and Giroux.

Kasnesis, P., Patrikakis, C. Z., & Venieris, I. S. (2019). PerceptionNet: A Deep Convolutional Neural Network for Late Sensor Fusion. *Intelligent Systems and Applications*, 101–119. https://doi.org/10.1007/978-3-030-01054-6_7

Kelso, J. A. S. (1997). *Dynamic Patterns: The Self-organization of Brain and Behavior*. MIT Press.

Khacef, L., Rodriguez, L., & Miramond, B. (2020). Brain-inspired self-organization with cellular neuromorphic computing for multimodal unsupervised learning. *ArXiv:2004.05488 [Cs, q-Bio]*. http://arxiv.org/abs/2004.05488

Kohonen, T. (1989). Self-Organizing Feature Maps. In T. Kohonen (Ed.), *Self-Organization and Associative Memory* (pp. 119–157). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-88163-3_5

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*(9), 1464–1480. https://doi.org/10.1109/5.58325

Körding, K. P., Beierholm, U., Ma, W. J., Quartz, S., Tenenbaum, J. B., & Shams, L. (2007). Causal Inference in Multisensory Perception. *PLOS ONE*, *2*(9), e943. https://doi.org/10.1371/journal.pone.0000943

Koutsoumpakis, G. (2014). *Spark-based Application for Abnormal Log Detection* [Master's thesis, Uppsala University]. http://uu.diva-portal.org/smash/get/diva2:751988/FULLTEXT01.pdf

Kraska, T., Talwalkar, A., Duchi, J. C., Griffith, R., Franklin, M. J., & Jordan, M. I. (2013). MLbase: A Distributed Machine-learning System. *Proceedings of 6th Biennial Conference on Innovative Data Systems Research (CIDR)*, *1*, 1–7.

Lawrence, R. D., Almasi, G. S., & Rushmeier, H. E. (1999). A Scalable Parallel Algorithm for Self-Organizing Maps with Applications to Sparse Data Mining Problems. *Data Mining and Knowledge Discovery*, *3*(2), 171–195. https://doi.org/10.1023/A:1009817804059

Lemke, G., & Reber, M. (2005). Retinotectal mapping: New Insights from Molecular Genetics. *Annual Review of Cell and Developmental Biology*, *21*(1), 551–580. https://doi.org/10.1146/annurev.cellbio.20.022403.093702

Liu, H., Yu, Y., Sun, F., & Gu, J. (2017). Visual–Tactile Fusion for Object Recognition. *IEEE Transactions on Automation Science and Engineering*, *14*(2), 996–1008. https://doi.org/10.1109/TASE.2016.2549552

Luo, R. C., Lin, M.-, & Scherp, R. S. (1988). Dynamic multi-sensor data fusion system for intelligent robots. *IEEE Journal on Robotics and Automation*, *4*(4), 386–396. https://doi.org/10.1109/56.802

Ma, Y., Wu, H., Wang, L., Huang, B., Ranjan, R., Zomaya, A., & Jie, W. (2015). Remote sensing big data computing: Challenges and opportunities. *Future Generation Computer Systems*, *51*, 47–60. https://doi.org/10.1016/j.future.2014.10.029

Magosso, E., Cuppini, C., Serino, A., Di Pellegrino, G., & Ursino, M. (2008). A theoretical study of multisensory integration in the superior colliculus by a neural network model. *Neural Networks*, *21*(6), 817–829. https://doi.org/10.1016/j.neunet.2008.06.003

Magosso, E., Cuppini, C., & Ursino, M. (2012). A Neural Network Model of Ventriloquism Effect and Aftereffect. *PLOS ONE*, *7*(8), e42503. https://doi.org/10.1371/journal.pone.0042503

Mahon, B. Z., & Caramazza, A. (2011). What drives the organization of object knowledge in the brain? *Trends in Cognitive Sciences*, *15*(3), 97–103. https://doi.org/10.1016/j.tics.2011.01.004

Makkook, M. (2007). *A Multimodal Sensor Fusion Architecture for Audio-Visual Speech Recognition* [Master's thesis, University of Waterloo]. https://uwspace.uwaterloo.ca/handle/10012/3065

Malondkar, A. M. (2015). *Extending the Growing Hierarchical Self Organizing Maps for a Large Mixed-Attribute Dataset Using Spark MapReduce* [PhD thesis, University of Ottawa]. http://www.ruor.uottawa.ca/handle/10393/33385

Mareschal, D., Westermman, D., & Althaus, N. (2012). In search of the developmental mechanisms of multi-sensory integration. In A. J. Bremner, D. Lewkowicz, & C. Spence (Eds.), *Multisensory Development* (pp. 342–359). Oxford University Press.

Markov, N. T., Ercsey-Ravasz, M. M., Ribeiro Gomes, A. R., Lamy, C., Magrou, L., Vezoli, J., Misery, P., Falchier, A., Quilodran, R., Gariel, M. A., Sallet, J., Gamanut, R., Huissoud, C., Clavagnier, S., Giroud, P., Sappey-Marinier, D., Barone, P., Dehay, C., Toroczkai, Z., … Kennedy, H. (2014). A weighted and directed interareal connectivity matrix for macaque cerebral cortex. *Cerebral Cortex*, *24*(1), 17–36. https://doi.org/10.1093/cercor/bhs270

Marsland, S., Shapiro, J., & Nehmzow, U. (2002). A self-organising network that grows when required. *Neural Networks*, *15*(8), 1041–1058. https://doi.org/10.1016/S0893-6080(02)00078-3

Martin, A. (2007). The representation of object concepts in the brain. *Annual Review of Psychology*, *58*, 25–45. https://doi.org/10.1146/annurev.psych.57.102904.190143

Martinetz, T. (1993). Competitive Hebbian Learning Rule Forms Perfectly Topology Preserving Maps. *International Conference on Artificial Neural Networks*, 427–434. https://doi.org/10.1007/978-1-4471-2063-6_104

Martinetz, T., & Schulten, K. (1991). A "neural gas" network learns topologies. *Proceedings of the International Conference on Artificial Neural Networks*, 397–402.

Martinez-Hernandez, U., Dodd, T. J., Evans, M. H., Prescott, T. J., & Lepora, N. F. (2017). Active sensorimotor control for tactile exploration. *Robotics and Autonomous Systems*, *87*, 15–27. https://doi.org/10.1016/j.robot.2016.09.014

Martinez-Hernandez, U., Metta, G., Dodd, T. J., Prescott, T. J., Natale, L., & Lepora, N. F. (2013). Active contour following to explore object shape with robot touch. *2013 World Haptics Conference (WHC)*, 341–346. https://doi.org/10.1109/WHC.2013.6548432

Matharage, S., Ganegedara, H., & Alahakoon, D. (2013). A scalable and dynamic self-organizing map for clustering large volumes of text data. *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 1–8. https://doi.org/10.1109/IJCNN.2013.6706733

Mcgurk, H., & Macdonald, J. (1976). Hearing lips and seeing voices. *Nature*, *264*(5588), 746–748. https://doi.org/10.1038/264746a0

Meredith, M. A., & Stein, B. E. (1986). Visual, auditory, and somatosensory convergence on cells in superior colliculus results in multisensory integration. *Journal of Neurophysiology*, *56*(3), 640–662. https://doi.org/10.1152/jn.1986.56.3.640

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236

Moraes, F. C., Botelho, S. C., Filho, N. D., & Gaya, J. F. O. (2012). Parallel High Dimensional Self Organizing Maps Using CUDA. *Robotics Symposium and Latin American Robotics Symposium (SBR-LARS)*, 302–306. https://doi.org/10.1109/SBR-LARS.2012.56

Movellan, J. R. (1995). Visual Speech Recognition with Stochastic Networks. *Advances in Neural Information Processing Systems 7*, 851–858.

Movellan, J. R., & Mineiro, P. (1998). Robust Sensor Fusion: Analysis and Application to Audio Visual Speech Recognition. *Machine Learning*, *32*(2), 85–100. https://doi.org/10.1023/A:1007468413059

Mulier, F., & Cherkassky, V. (1995). Self-Organization as an Iterative Kernel Smoothing Process. *Neural Computation*, *7*(6), 1165–1177. https://doi.org/10.1162/neco.1995.7.6.1165

Münzner, S., Schmidt, P., Reiss, A., Hanselmann, M., Stiefelhagen, R., & Dürichen, R. (2017). CNN-based Sensor Fusion Techniques for Multimodal Human Activity Recognition. *Proceedings of the 2017 ACM International Symposium on Wearable Computers*, 158–165. https://doi.org/10.1145/3123021.3123046

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, *2*(1), 1–21. https://doi.org/10.1186/s40537-014-0007-7

Nallaperuma, D., Silva, D. D., Alahakoon, D., & Yu, X. (2017). A cognitive data stream mining technique for context-aware IoT systems. *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 4777–4782. https://doi.org/10.1109/IECON.2017.8216824

Nallaperuma, D., Silva, D. D., Alahakoon, D., & Yu, X. (2018). Intelligent Detection of Driver Behavior Changes for Effective Coordination Between Autonomous and Human Driven Vehicles. *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 3120–3125. https://doi.org/10.1109/IECON.2018.8591357

Nathawitharana, N., Alahakoon, D., & Matharage, S. (2015). Improving the Decision Value of Hierarchical Text Clustering Using Term Overlap Detection. *Australasian Journal of Information Systems*, *19*(0). https://doi.org/10.3127/ajis.v19i0.1180

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal deep learning. *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 689–696.

Nguyen, L.-D., Woon, K.-Y., & Tan, A.-H. (2008). A self-organizing neural model for multimedia information fusion. *2008 11th International Conference on Information Fusion*, 1–7.

Noda, K., Arie, H., Suga, Y., & Ogata, T. (2014). Multimodal integration learning of robot behavior using deep neural networks. *Robotics and Autonomous Systems*, *62*(6), 721–736. https://doi.org/10.1016/j.robot.2014.03.003

Nunn, J. A., Gregory, L. J., Brammer, M., Williams, S. C. R., Parslow, D. M., Morgan, M. J., Morris, R. G., Bullmore, E. T., Baron-Cohen, S., & Gray, J. A. (2002). Functional magnetic resonance imaging of synesthesia: Activation of V4/V8 by spoken words. *Nature Neuroscience*, *5*(4), 371–375. https://doi.org/10.1038/nn818

O'Leary, D. E. (2013). Artificial Intelligence and Big Data. *IEEE Intelligent Systems*, *28*(2), 96–99. https://doi.org/10.1109/MIS.2013.39

Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

Pan, Y. (2016). Heading toward Artificial Intelligence 2.0. *Engineering*, *2*(4), 409–413. https://doi.org/10.1016/J.ENG.2016.04.018

Parisi, G. I., Tani, J., Weber, C., & Wermter, S. (2017). Emergence of multimodal action representations from neural network self-organization. *Cognitive Systems Research*, *43*, 208–221. https://doi.org/10.1016/j.cogsys.2016.08.002

Pedišić, Ž., & Bauman, A. (2015). Accelerometer-based measures in physical activity surveillance: Current practices and issues. *British Journal of Sports Medicine*, *49*(4), 219–223. https://doi.org/10.1136/bjsports-2013-093407

Petkov, C. I., Kayser, C., Augath, M., & Logothetis, N. K. (2006). Functional Imaging Reveals Numerous Fields in the Monkey Auditory Cortex. *PLOS Biology*, *4*(7), e215. https://doi.org/10.1371/journal.pbio.0040215

Price, K., Bird, S. R., Lythgo, N., Raj, I. S., Wong, J. Y. L., & Lynch, C. (2017). Validation of the Fitbit One, Garmin Vivofit and Jawbone UP activity tracker in estimation of energy expenditure during treadmill walking and running. *Journal of Medical Engineering & Technology*, *41*(3), 208–215. https://doi.org/10.1080/03091902.2016.1253795

Prince, S. A., Adamo, K. B., Hamel, M. E., Hardt, J., Gorber, S. C., & Tremblay, M. (2008). A comparison of direct versus self-report measures for assessing physical activity in adults: A systematic review. *International Journal of Behavioral Nutrition and Physical Activity*, *5*(1), 56. https://doi.org/10.1186/1479-5868-5-56

Radeau, M., & Bertelson, P. (1974). The after-effects of ventriloquism. *Quarterly Journal of Experimental Psychology*, *26*(1), 63–71. https://doi.org/10.1080/14640747408400388

Radeau, M., & Bertelson, P. (1977). Adaptation to auditory-visual discordance and ventriloquism in semirealistic situations. *Perception & Psychophysics*, *22*(2), 137–146. https://doi.org/10.3758/BF03198746

Ramachandran, V. S., & Hubbard, E. M. (2001). Synaesthesia: A Window Into Perception, Thought and Language. *Journal of Consciousness Studies*, *8*(12), 3–34.

Rand, W. M. (1971). Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, *66*(336), 846–850. https://doi.org/10.2307/2284239

Rathore, M. M. U., Paul, A., Ahmad, A., Chen, B.-W., Huang, B., & Ji, W. (2015). Real-Time Big Data Analytical Architecture for Remote Sensing Application. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *8*(10), 4610–4621. https://doi.org/10.1109/JSTARS.2015.2424683

Rauber, A., Tomsich, P., & Merkl, D. (2000). parSOM: a parallel implementation of the self-organizing map exploiting cache effects: Making the SOM fit for interactive high-performance data analysis. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, *6*, 177–182. https://doi.org/10.1109/IJCNN.2000.859393

Reiss, A., & Stricker, D. (2012). Introducing a New Benchmarked Dataset for Activity Monitoring. *16th International Symposium on Wearable Computers*, 108–109. https://doi.org/10.1109/ISWC.2012.13

Revonsuo, A., & Newman, J. (1999). Binding and Consciousness. *Consciousness and Cognition*, *8*(2), 123–127. https://doi.org/10.1006/ccog.1999.0393

Robertson, L. (2003). Binding, spatial attention and perceptual awareness. *Nature Reviews Neuroscience*, *4*(2), 93–102. https://doi.org/10.1038/nrn1030

Robertson, L., Treisman, A., Friedman-Hill, S., & Grabowecky, M. (1997). The Interaction of Spatial and Object Pathways: Evidence from Balint's Syndrome. *Journal of Cognitive Neuroscience*, *9*(3), 295–317. https://doi.org/10.1162/jocn.1997.9.3.295

Rowland, B., Stanford, T., & Stein, B. (2007). A Bayesian model unifies multisensory spatial localization with the physiological properties of the superior colliculus. *Experimental Brain Research*, *180*(1), 153–161. https://doi.org/10.1007/s00221-006-0847-2

Ruch, N., Rumo, M., & Mäder, U. (2011). Recognition of activities in children by two uniaxial accelerometers in free-living conditions. *European Journal of Applied Physiology*, *111*(8), 1917–1927. https://doi.org/10.1007/s00421-011-1828-0

Russell, B. (1914). The relation of sense-data to physics. In *Mysticism and Logic* (pp. 108–131). George Allen & Unwin.

Sammon, J. W. (1969). A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, *100*(5), 401–409. https://doi.org/10.1109/T-C.1969.222678

Sarazin, T., Azzag, H., & Lebbah, M. (2014). SOM Clustering Using Spark-MapReduce. *2014 IEEE International Parallel & Distributed Processing Symposium Workshops*, 1727–1734. https://doi.org/10.1109/IPDPSW.2014.192

Schwartz, J.-L., Berthommier, F., & Savariaux, C. (2004). Seeing to hear better: Evidence for early audio-visual interactions in speech identification. *Cognition*, *93*(2), B69–B78. https://doi.org/10.1016/j.cognition.2004.01.006

Shatz, C. J. (1992). How are specific connections formed between thalamus and cortex? *Current Opinion in Neurobiology*, *2*(1), 78–82. https://doi.org/10.1016/0959-4388(92)90166-i

Smythies, J. R. (1994). *The walls of Plato's cave: The science and philosophy of (brain, consciousness, and perception)*. Avebury. https://trove.nla.gov.au/version/36234094

Somogyi, P., Tamás, G., Lujan, R., & Buhl, E. H. (1998). Salient features of synaptic organisation in the cerebral cortex. *Brain Research. Brain Research Reviews*, *26*(2–3), 113–135. https://doi.org/10.1016/s0165-0173(97)00061-1

Sozykin, A., & Epanchintsev, T. (2015). MIPr—A framework for distributed image processing using Hadoop. *2015 9th International Conference on Application of Information and Communication Technologies (AICT)*, 35–39. https://doi.org/10.1109/ICAICT.2015.7338511

Srivastava, N., & Salakhutdinov, R. R. (2014). Multimodal Learning with Deep Boltzmann Machines. *The Journal of Machine Learning Research*, *15*(1), 2949–2980.

Stanford, T. R., Quessy, S., & Stein, B. E. (2005). Evaluating the Operations Underlying Multisensory Integration in the Cat Superior Colliculus. *Journal of Neuroscience*, *25*(28), 6499–6508. https://doi.org/10.1523/JNEUROSCI.5095-04.2005

Stein, B. E., & Meredith, M. A. (1993). *The Merging of the Senses*. MIT Press.

Stein, B. E., & Stanford, T. R. (2008). Multisensory integration: Current issues from the perspective of the single neuron. *Nature Reviews. Neuroscience*, *9*(4), 255–266. https://doi.org/10.1038/nrn2331

Stein, B. E., Stanford, T. R., & Rowland, B. A. (2009). The neural basis of multisensory integration in the midbrain: Its organization and maturation. *Hearing Research*, *258*(1), 4–15. https://doi.org/10.1016/j.heares.2009.03.012

Suk, H.-I., Lee, S.-W., & Shen, D. (2014). Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis. *NeuroImage*, *101*, 569–582. https://doi.org/10.1016/j.neuroimage.2014.06.077

Takatsuka, M., & Bui, M. (2010). Parallel Batch Training of the Self-Organizing Map Using OpenCL. In K. W. Wong, B. S. U. Mendis, & A. Bouzerdoum (Eds.), *Neural Information Processing. Models and Applications* (pp. 470–476). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-17534-3_58

Talavage, T. M., Ledden, P. J., Benson, R. R., Rosen, B. R., & Melcher, J. R. (2000). Frequency-dependent responses exhibited by multiple regions in human auditory

cortex. *Hearing Research*, *150*(1), 225–244. https://doi.org/10.1016/S0378-5955(00)00203-3

Tan, A.-H., Carpenter, G. A., & Grossberg, S. (2007). Intelligence Through Interaction: Towards a Unified Theory for Learning. *International Symposium on Neural Networks*, 1094–1103. https://doi.org/10.1007/978-3-540-72383-7_128

Thiele, A., & Stoner, G. (2003). Neuronal synchrony does not correlate with motion coherence in cortical area MT. *Nature*, *421*(6921), 366–370. https://doi.org/10.1038/nature01285

Tomsich, P., Rauber, A., & Merkl, D. (2000). Optimizing the parSOM neural network implementation for data mining with distributed memory systems and cluster computing. *11th International Workshop on Database and Expert Systems Applications, 2000. Proceedings*, 661–665. https://doi.org/10.1109/DEXA.2000.875094

Tononi, G., Edelman, G. M., & Sporns, O. (1998). Complexity and coherency: Integrating information in the brain. *Trends in Cognitive Sciences*, *2*(12), 474–484. https://doi.org/10.1016/S1364-6613(98)01259-5

Treisman, A. (1982). Perceptual grouping and attention in visual search for features and for objects. *Journal of Experimental Psychology: Human Perception and Performance*, *8*(2), 194–214. https://doi.org/10.1037/0096-1523.8.2.194

Treisman, A., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology*, *12*(1), 97–136. https://doi.org/10.1016/0010-0285(80)90005-5

Treisman, A., & Schmidt, H. (1982). Illusory conjunctions in the perception of objects. *Cognitive Psychology*, *14*(1), 107–141. https://doi.org/10.1016/0010-0285(82)90006-8

Tsal, Y. (1989). Do illusory conjunctions support the feature integration theory? A critical review of theory and findings. *Journal of Experimental Psychology: Human Perception and Performance*, *15*(2), 394–400. https://doi.org/10.1037/0096-1523.15.2.394

Tudor-Locke, C. E., & Myers, A. M. (2001). Challenges and Opportunities for Measuring Physical Activity in Sedentary Adults. *Sports Medicine*, *31*(2), 91–100. https://doi.org/10.2165/00007256-200131020-00002

Tyler, L. K., & Moss, H. E. (2001). Towards a distributed account of conceptual knowledge. *Trends in Cognitive Sciences*, *5*(6), 244–252. https://doi.org/10.1016/S1364-6613(00)01651-X

Ugulino, W., Cardador, D., Vega, K., Velloso, E., Milidiú, R., & Fuks, H. (2012). Wearable Computing: Accelerometers' Data Classification of Body Postures and Movements. *Advances in Artificial Intelligence - SBIA 2012*, 52–61. https://doi.org/10.1007/978-3-642-34459-6_6

Ursino, M., Cuppini, C., & Magosso, E. (2014). Neurocomputational approaches to modelling multisensory integration in the brain: A review. *Neural Networks*, *60*, 141–165. https://doi.org/10.1016/j.neunet.2014.08.003

Ursino, M., Cuppini, C., Magosso, E., Serino, A., & di Pellegrino, G. (2009). Multisensory integration in the superior colliculus: A neural network model. *Journal of Computational Neuroscience*, *26*(1), 55–73. https://doi.org/10.1007/s10827-008-0096-4

Valiant, L. G. (1990). A Bridging Model for Parallel Computation. *Communications of the ACM*, *33*(8), 103–111. https://doi.org/10.1145/79173.79181

Van den Berg, J., Abbeel, P., & Goldberg, K. (2011). LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. *The International Journal of Robotics Research*, *30*(7), 895–913. https://doi.org/10.1177/0278364911406562

Van Hees, V. T., Renström, F., Wright, A., Gradmark, A., Catt, M., Chen, K. Y., Löf, M., Bluck, L., Pomeroy, J., Wareham, N. J., Ekelund, U., Brage, S., & Franks, P. W. (2011). Estimation of Daily Energy Expenditure in Pregnant and Non-Pregnant Women Using a Wrist-Worn Tri-Axial Accelerometer. *PLOS ONE*, *6*(7), e22922. https://doi.org/10.1371/journal.pone.0022922

Varela, F. J. (1995). Resonant cell assemblies: A new approach to cognitive functions and neuronal synchrony. *Biological Research*, *28*(1), 81–95.

Velik, R. (2014). A brain-inspired multimodal data mining approach for human activity recognition in elderly homes. *Journal of Ambient Intelligence and Smart Environments*, *6*(4), 447–468. https://doi.org/10.3233/AIS-140266

Wallace, M. T., & Stein, B. E. (1997). Development of Multisensory Neurons and Multisensory Integration in Cat Superior Colliculus. *Journal of Neuroscience*, *17*(7), 2429–2444. https://doi.org/10.1523/JNEUROSCI.17-07-02429.1997

Wallach, H. (1968). Informational discrepancy as a basis of perceptual adaptation. *The Neuropsychology of Spatially Oriented Behaviour*, 209–230.

Wang, N., Zhang, Z., Li, T., Xiao, J., & Cui, L. (2019). SGSF: A Small Groups Based Serial Fusion Method. *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 97–108.

Weber, M., Teeling, H., Huang, S., Waldmann, J., Kassabgy, M., Fuchs, B. M., Klindworth, A., Klockow, C., Wichels, A., Gerdts, G., Amann, R., & Glöckner, F. O. (2011). Practical application of self-organizing maps to interrelate biodiversity and functional data in NGS-based metagenomics. *The ISME Journal*, *5*(5), 918–928. https://doi.org/10.1038/ismej.2010.180

Weichel, C. (2010). Adapting Self-Organizing Maps to the MapReduce Programming Paradigm. *Proceedings of the STeP*, 119–131.

Wiesel, T. N., & Hubel, D. H. (1963). Single-cell responses in striate cortex of kittens deprived of vision in one eye. *Journal of Neurophysiology*, *26*(6), 1003–1017. https://doi.org/10.1152/jn.1963.26.6.1003

Williams, M. L., Burnap, P., & Sloan, L. (2017). Crime Sensing With Big Data: The Affordances and Limitations of Using Open-source Communications to Estimate Crime Patterns. *The British Journal of Criminology*, *57*(2), 320–340. https://doi.org/10.1093/bjc/azw031

Wittek, P., & Darányi, S. (2012). A GPU-Accelerated Algorithm for Self-Organizing Maps in a Distributed Environment. *20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*.

Wittek, P., & Darányi, S. (2013). Accelerating text mining workloads in a MapReduce-based distributed GPU environment. *Journal of Parallel and Distributed Computing*, *73*(2), 198–206. https://doi.org/10.1016/j.jpdc.2012.10.001

Wysoski, S. G., Benuskova, L., & Kasabov, N. (2010). Evolving spiking neural networks for audiovisual information processing. *Neural Networks*, *23*(7), 819–835. https://doi.org/10.1016/j.neunet.2010.04.009

Yang, C.-C., & Hsu, Y.-L. (2010). A Review of Accelerometry-Based Wearable Motion Detectors for Physical Activity Monitoring. *Sensors*, *10*(8), 7772–7788. https://doi.org/10.3390/s100807772

Yang, Z., Raymond, O. I., Zhang, C., Wan, Y., & Long, J. (2018). DFTerNet: Towards 2-bit Dynamic Fusion Networks for Accurate Human Activity Recognition. *IEEE Access*, *6*, 56750–56764. https://doi.org/10.1109/ACCESS.2018.2873315

Yeo, B. T. T., Krienen, F. M., Sepulcre, J., Sabuncu, M. R., Lashkari, D., Hollinshead, M., Roffman, J. L., Smoller, J. W., Zöllei, L., Polimeni, J. R., Fischl, B., Liu, H., & Buckner, R. L. (2011). The organization of the human cerebral cortex estimated by intrinsic functional connectivity. *Journal of Neurophysiology*, *106*(3), 1125–1165. https://doi.org/10.1152/jn.00338.2011

Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M. J., Shenker, S., & Stoica, I. (2012). Resilient Distributed Datasets: A Fault-tolerant Abstraction for In-memory Cluster Computing. *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, 15–28.

Zhai, Y. Z., Hsu, A., & Halgamuge, S. K. (2006). Scalable Dynamic Self-Organising Maps for Mining Massive Textual Data. *International Conference on Neural Information Processing*, 260–267. https://doi.org/10.1007/11893295_30

Zhang, Yin, Chen, M., Mao, S., Hu, L., & Leung, V. C. M. (2014). CAP: Community activity prediction based on big data analysis. *IEEE Network*, *28*(4), 52–57. https://doi.org/10.1109/MNET.2014.6863132

Zhang, Yongmian, & Ji, Q. (2006). Active and dynamic information fusion for multisensor systems with dynamic bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *36*(2), 467–472. https://doi.org/10.1109/TSMCB.2005.859081

Zheng, Y. (2015). Methodologies for Cross-Domain Data Fusion: An Overview. *IEEE Transactions on Big Data*, *1*(1), 16–34. https://doi.org/10.1109/TBDATA.2015.2465959

Zhongwen, L., Zhengping, Y., & Xincai, W. (2005). Self-Organizing Maps Computing on Graphic Process Unit. *Neural Networks*.