CHALMERS

Machine Learning Methods for Image Analysis in Medical Applications

From Alzheimer's Disease, Brain Tumors, to Assisted Living

CHENJIE GE

Computer Vision and Medical Image Analysis Group
Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2020

THESIS FOR THE DEGREE OF DOKTOR OF PHILOSOPHY

Machine Learning Methods for Image Analysis in Medical Applications

From Alzheimer's Disease, Brain Tumors, to Assisted Living

CHENJIE GE

CHALMERS
UNIVERSITY OF TECHNOLOGY

Computer Vision and Medical Image Analysis Group
Department of Electrical Engineering
Chalmers University of Technology
Göteborg, Sweden, 2020

Machine Learning Methods for Image Analysis in Medical Applications
From Alzheimer's Disease, Brain Tumors, to Assisted Living
Chenjie Ge
ISBN 978-91-7905-322-2

Doktorsavhandlingar vid Chalmers Tekniska Högskola
Ny serie nr 4789
ISSN 0346-718X

This thesis has been prepared using LaTeX.

Computer Vision and Medical Image Analysis Group
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
Phone: +46 (0)31 772 1000
www.chalmers.se

Printed by Chalmers Reproservice
Göteborg, Sweden, June 2020

Abstract

Healthcare has progressed greatly nowadays owing to technological advances, where machine learning plays an important role in processing and analyzing a large amount of medical data. This thesis investigates four healthcare-related issues (Alzheimer's disease detection, glioma classification, human fall detection, and obstacle avoidance in prosthetic vision), where the underlying methodologies are associated with machine learning and computer vision. For Alzheimer's disease (AD) diagnosis, apart from symptoms of patients, Magnetic Resonance Images (MRIs) also play an important role. Inspired by the success of deep learning, a new multi-stream multi-scale Convolutional Neural Network (CNN) architecture is proposed for AD detection from MRIs, where AD features are characterized in both the tissue level and the scale level for improved feature learning. Good classification performance is obtained for AD/NC (normal control) classification with test accuracy 94.74%. In glioma subtype classification, biopsies are usually needed for determining different molecular-based glioma subtypes. We investigate non-invasive glioma subtype prediction from MRIs by using deep learning. A 2D multi-stream CNN architecture is used to learn the features of gliomas from multi-modal MRIs, where the training dataset is enlarged with synthetic brain MRIs generated by pairwise Generative Adversarial Networks (GANs). Test accuracy 88.82% has been achieved for IDH mutation (a molecular-based subtype) prediction. A new deep semi-supervised learning method is also proposed to tackle the problem of missing molecular-related labels in training datasets for improving the performance of glioma classification. In other two applications, we also address video-based human fall detection by using co-saliency-enhanced Recurrent Convolutional Networks (RCNs), as well as obstacle avoidance in prosthetic vision by characterizing obstacle-related video features using a Spiking Neural Network (SNN). These investigations can benefit future research, where artificial intelligence/deep learning may open a new way for real medical applications.

Keywords: Alzheimer's disease detection, glioma subtype classification, fall detection, visual prosthesis, machine learning, deep learning, convolutional neural networks, generative adversarial networks, semi-supervised learning, recurrent convolutional networks, spiking neural networks.

List of Publications

This thesis is based on the following publications:

Alzheimer's disease detection

Paper 1: Chenjie Ge, Qixun Qu, Irene Y.H. Gu, and Asgeir S. Jakola, "Multi-Stream Multi-Scale Deep Convolutional Networks for Alzheimer's Disease Detection using MR Images", *Neurocomputing*, vol. 350, pp. 60-69, 2019.

Glioma classification

Paper 2: Chenjie Ge, Irene Y.H. Gu, Asgeir S. Jakola, and Jie Yang, "Enlarged Training Dataset by Pairwise GANs for Molecular-Based Brain Tumor Classification", *IEEE Access*, vol. 8, pp. 22560-22570, 2020.

Paper 3: Chenjie Ge, Irene Y.H. Gu, Asgeir S. Jakola, and Jie Yang, "Deep Learning and Multi-Sensor Fusion for Glioma Classification using Multistream 2D Convolutional Networks", in *40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5894-5897, 2018.

Paper 4: Chenjie Ge, Irene Y.H. Gu, Asgeir S. Jakola, and Jie Yang, "Deep Semi-Supervised Learning for Brain Tumor Classification" (submitted to journal).

Assisted living/health care (fall detection, obstacle avoidance)

Paper 5: Chenjie Ge, Irene Y.H. Gu, and Jie Yang, "Human Fall Detection using Segment-Level CNN Features and Sparse Dictionary Learning", in *IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1-6, 2017.

Paper 6: Chenjie Ge, Irene Y.H. Gu, and Jie Yang, "Human Fall Detection using Co-Saliency-Enhanced Deep Recurrent Convolutional Neural Networks", *International Research Journal of Engineering and Technology*, vol. 6, no. 9, pp. 993-1000, 2019.

Paper 7: Chenjie Ge, Keren Fu, Fanghui Liu, Li Bai, and Jie Yang, "Co-Saliency Detection via Inter and Intra Saliency Propagation", *Signal Processing: Image Communication*, vol. 44, pp. 69-83, 2016.

Paper 8: Chenjie Ge, Nikola Kasabov, Zhi Liu, and Jie Yang, "A Spiking Neural Network Model for Obstacle Avoidance in Simulated Prosthetic Vision", *Information Sciences*, vol. 399, pp. 30-42, 2017.

Other publications by the author:

- Chenjie Ge, Qixun Qu, Irene Y.H. Gu, and Asgeir S. Jakola, "Multiscale Deep Convolutional Networks for Characterization and Detection of Alzheimer's Disease Using MR images", in *IEEE International Conference on Image Processing (ICIP)*, pp. 789-793, 2019.

- Chenjie Ge, Irene Y.H. Gu, Asgeir S. Jakola, and Jie Yang, "Cross-Modality Augmentation of Brain MR Images using a Novel Pairwise Generative Adversarial Network for Enhanced Glioma Classification", in *IEEE International Conference on Image Processing (ICIP)*, pp. 559-563, 2019.

- Chenjie Ge, Irene Y.H. Gu, and Jie Yang, "Co-Saliency-Enhanced Deep Recurrent Convolutional Networks for Human Fall Detection in E-Healthcare", in *40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1572-1575, 2018.

- Chenjie Ge, Qixun Qu, Irene Y.H. Gu, and Asgeir S. Jakola, "3D Multi-Scale Convolutional Networks for Glioma Grading using MR Images", in *IEEE International Conference on Image Processing (ICIP)*, pp. 141-145, 2018.

- Chenjie Ge, Keren Fu, Yijun Li, Jie Yang, Pengfei Shi, and Li Bai, "Co-Saliency Detection via Similarity-Based Saliency Propagation", in *IEEE International Conference on Image Processing (ICIP)*, pp. 1845-1849, 2015.

- Chenjie Ge, Roger A.D. Oliveira, Irene Y.H. Gu, and Math H.J. Bollen, "Deep Feature Clustering for Seeking Patterns in Daily Harmonic Variations" (submitted to journal).

- Chenjie Ge, Roger A.D. Oliveira, Irene Y.H. Gu, and Math H.J. Bollen, "Unsupervised Deep Learning and Analysis of Harmonic Variation Patterns using Big Data from Multiple Locations" (submitted to journal).

Acknowledgment

First and foremost, I would like to express my deepest gratitude to my main supervisor Prof. Irene Yu-Hua Gu for her immense help and support on my research at Chalmers. Her enthusiasm, patience and encouragement helped me grow as a better researcher. I would also like to thank my co-supervisor Prof. Jie Yang at Shanghai Jiao Tong University for his valuable suggestions. Gratitudes also go to my co-supervisor Neurosurgeon and Assoc.Prof. Asgeir Store Jakola at Sahlgrenska University Hospital for his suggestions and help from the medical perspective.

I would like to thank Chalmers University of Technology, Sweden and Shanghai Jiao Tong University, China for providing me the chance and scholarship to conduct my research and study as a double-degree student. I would also like to acknowledge China Scholarship Council (CSC) and STINT Joint Swedish-China Mobility Programme in Sweden for their financial support in part of this thesis work.

Further, I am grateful to Department of Electrical Engineering for accepting me as a PhD student. I would like to thank my colleagues Muhaddisa Barat Ali and Qixun Qu for the help and the stimulating discussions in my research. Many thanks to the administrative staff especially Ann-Christine Lindbom for the kind help. Specially, I would like to thank all my Chinese friends in this department for the good time we have had together.

Last but not least, I would like to thank my family for their love and encouragement during my PhD journey. I wish to share this happiness with them.

Chenjie Ge
Göteborg, June 2020

Acronyms

AD:	Alzheimer's Disease
AI:	Artificial Intelligence
CNN:	Convolutional Neural Network
CSF:	Cerebrospinal Fluid
CT:	Computerized Tomography
FC:	Fully-Connected
FLAIR:	T2-Weighted-Fluid-Attenuated Inversion Recovery
GAN:	Generative Adversarial Network
GM:	Grey Matter
HGG:	High-Grade Glioma
IDH:	Isocitrate Dehydrogenase
kNN:	K-Nearest Neighbours
LGG:	Low-Grade Glioma
LSTM:	Long Short Term Memory
MCI:	Mild Cognitive Impairment
MLP:	Multilayer Perceptron
MR(I):	Magnetic Resonance (Image)
NC:	Normal Control
PET:	Positron Emission Tomography
PR:	Precision-Recall
RCN:	Recurrent Convolutional Network
ReLU:	Rectified Linear Unit
RNN:	Recurrent Neural Network
ST:	Spatio-Temporal

SVM:	Support Vector Machine
T1:	T1-Weighted
T1ce:	Post-Contrast-Enhanced T1-Weighted
T2:	T2-Weighted
WM:	White Matter

Contents

Part I

Introductory chapters

CHAPTER 1

Introduction

With the rapid development of artificial intelligence (AI), healthcare has greatly progressed as AI can do what humans do and even surpass performance of humans in the areas such as early detection of disease, diagnosis and assisted-living. Motivated by the success of AI, this thesis investigates four different health-related issues using machine learning techniques: Alzheimer's disease (AD) detection, brain tumor (glioma) classification, human fall detection and obstacle avoidance in prosthetic vision.

The importance of this thesis is also to emphasize the underlying methodologies associated with machine learning and computer vision problems, which makes the thesis work interdisciplinary research related to AI, machine learning, pattern recognition, computer vision, E-healthcare and assisted living. In the following sections, four applications including AD detection, glioma classification, fall detection and obstacle avoidance will be introduced respectively.

1.1 Alzheimer's Disease Detection

AD is the most common type of dementia. It is reported that over 46 million patients suffer from AD worldwide, and the population of the AD patients will grow to 131.5 million by 2050 [1]. AD is a progressive neurodegenerative brain disease that affects people in various ways. Patients with AD suffer from memory loss, deterioration in abilities of thinking, speaking, and eventually fail to carry out activities of daily life. They also frequently show behavioral and psychological problems, leading to additional distress in both patients and caregivers [2]. At the early stage of AD, one common symptom is

memory loss, especially the loss of short-term memory. As AD progresses, motor skills such as walking and even swallowing are gradually deteriorated. Currently, the reasons for the abnormal changes in the brain are still not clear. Advancing age and family history are two known risk factors. Lifestyle, head injury, depression and environmental factors are also believed to affect the brain over time and possibly cause AD. The survival time of AD varies in different age groups. Patients who are diagnosed as AD at around 65 years old have an average survival time of 8.3 years. In comparison, patients who are diagnosed as AD at around 90 years old can only survive 3.4 years on average [3]. AD is caused by abnormal deposits of protein in the brain, leading to the destruction of brain cells. A brain with AD has a thinner cortical grey matter (GM) but larger ventricles filled with cerebrospinal fluid (CSF) [4] compared to a normal brain, as illustrated in Figure 1.1. It is observed that the hippocampus and white matter (WM) also atrophy, leading to the atrophy of the whole brain tissue. Sadly, there is no cure for AD for the time being and all medications and treatments can only help relieve the symptoms of AD [5]. However, early diagnosis of AD provides opportunities for early intervention and thus plays an important role in extending the survival of patients [6].

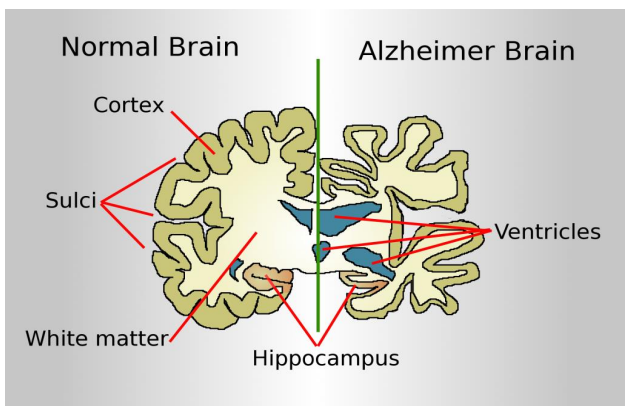

Figure 1.1: Comparison of a normal brain and a brain with AD. The image is taken from [7].

Related work

Diagnostic methods for AD can be categorized into several groups, for example,

- symptom-based methods,
- clinical test-based methods.

AD can be diagnosed from symptoms [8]. For instance, patients may suffer from memory impairment and sometimes have difficulty remembering work/social events and finishing daily tasks. It can also be observed from language problems that their vocabulary is reduced in speech or writing. Abilities to concentrate, plan, make decisions and

solve problems can be affected as well. In addition, patients suffer from a poor sense of location, time, sight and even mental changes in their mood, behavior or personality.

Clinical techniques for medical assessment of AD include physical and neuropsychological exams as well as lab tests. At first, medical doctors would inquire about the medical history, diet and overall health status of patients. The physical exams, including motor skills, muscle tone and strength, reflexes, the ability of balancing and coordination are taken to assess the overall neurological health. In addition, neuropsychological exams are often conducted to evaluate the brain function related to memory and thinking skills. Lab tests such as urine and blood tests are also used to rule out other diseases that may also cause memory loss.

Clinical tests for AD also include imaging of brain structures using techniques such as magnetic resonance (MR) imaging, computerized tomography (CT) and positron emission tomography (PET), among which MR imaging is widely used in AD detection. An example of an MR imaging scanner is shown in Figure 1.2. To illustrate the difference between AD and normal control (NC) groups in MRIs, Figure 1.3 shows the magnetic resonance image (MRI) examples of these two groups. Observing Figure 1.3, it is still challenging to diagnose AD from MRIs, as the visual changes of brain tissues may not be easily captured especially at the early stage of AD. Inspecting MR scans for signs of AD requires the expertise of medical doctors.

Figure 1.2: Example of an MR imaging scanner. The image is taken from [9].

With the development of AI technologies, machine learning can be used to assist medical doctors in AD diagnosis. Many efforts have been made using hand-crafted features from MRIs for AD classification, where feature extraction is based on the knowledge of human researchers. Yang et al. [10] studied potential AD-related MR image features based on independent component analysis. A support vector machine (SVM) was then

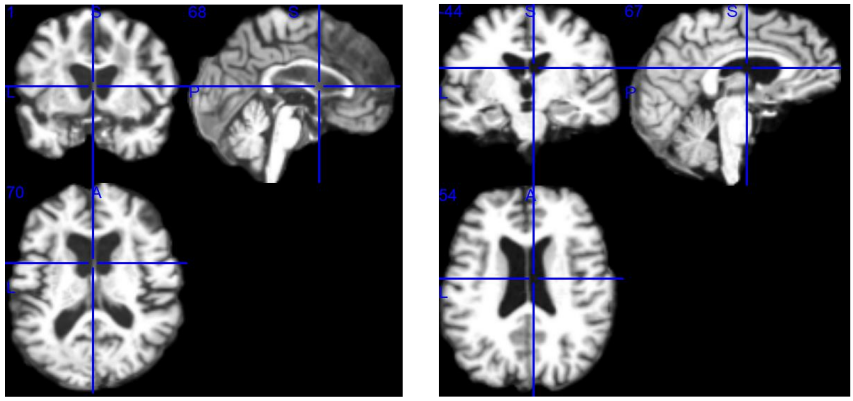

Figure 1.3: Examples of skull removed MRIs in axial, coronal and sagittal views. Left: AD, right: NC.

used for classifying AD and NC subjects. Tong et al. [11] utilized the strategy of multiple instance learning to classify dementia, where features were extracted using bags of MRI voxel patches and graph mapping. Arvesen et al. [12] studied methods of dimensional reduction and variations in the learning task to analyze structural MRI data, where a model of decision trees with principal component analysis-based dimensional reduction has achieved good performance for AD detection. Liu et al. [13] proposed to extract multi-view features using selected templates. Encoded features were then obtained by clustering subjects in each view space, followed by an ensemble of SVMs to classify the subjects. The recent development of deep learning methods for AD detection has drawn significant attention since features are learned automatically. Brosche et al. [14] proposed to learn the manifold of brain images using a deep brief network model, where patterns in image groups were used to distinguish AD from NC subjects. Sarraf et al. [15] employed the Convolutional Neural Network (CNN) architectures LeNet and GoogleNet to detect Alzheimer's disease using MR brain scans. GoogleNet achieved good performance using imbalanced training data where the ratio of AD and NC scans is 5:1. Bäckström et al. [16] proposed an efficient and simple 3D CNN architecture, where good results were achieved for AD detection on a dataset containing 340 subjects. Auto-encoders (AEs) were shown to be another effective method for learning unsupervised generic features, followed by fine-tuned task-specific layers for final classification. Suk et al. [17] used stacked AEs to extract features from MRI, PET image regions and CSF biomarkers. A multi-kernel SVM was then used for the classification. Gupta et al. [18] extracted the slice-wise feature of MR images using 2D CNNs. Pre-trained sparse AEs were proposed for further performance enhancement. Hosseini-Asl et al. [19] used a pre-trained 3D convolutional AE to learn generic features, followed by a 3D CNN for refined training. Although some promising results have been achieved for AD/NC classification, much re-

search is still needed to further improve the ability to characterize AD brains with deep learning.

1.2 Glioma Classification

A brain tumor is a central nervous system disease observed as a mass or growth of abnormal cells in a brain. Gliomas are the most common tumors originating from the brain [20], and make up 80% of all malignant brain tumors [21]. Gliomas can affect various basic brain functions and even be life-threatening in a short time.

Grading
World Health Organization (WHO) grades gliomas into four classes (grades I-IV) according to their aggressiveness. The diffuse gliomas are conventionally divided into low-grade gliomas (LGG, WHO grade II) and high-grade gliomas (HGG, WHO grade III and IV). Grade I gliomas (pilocytic astrocytomas) are noninvasive with a slow rate of growth, and patients can be cured through surgeries. Grade II and Grade III gliomas can be astrocytoma or oligodendroglioma. Oligodendrogliomas are usually slow-growing tumors but astrocytomas in Grade II or III often progress to gliomas of higher grades. Grade IV gliomas are also known as glioblastoma (GBM), and patients have the shortest survival time among all the gliomas. Table 1.1 summarizes gliomas in different grades and their corresponding 5-year survival rate according to [21]. Pre-surgical assessment or prediction of glioma grade is important for clinical decision making and planning, as it can help to predict how the tumor would progress over time and hence plan future treatment, making it possible to extend the survival time of patients. Normally, experienced radiologists can tell the grade of glioma by observing the brain MRI of a patient.

Table 1.1: Gliomas in different grades and their corresponding 5-year survival rate according to [21].

Glioma grade	Glioma type	5-year survival rate (%)
I	Astrocytoma	Can be cured
II	Astrocytoma	50%
II	Oligodendroglioma	80%
III	Astrocytoma	30%
III	Oligodendroglioma	80%
IV	Glioblastoma	5%

Molecular-based subtype classification
Different from the grade of glioma that can be observed from the brain MRI of a patient, some molecular biomarkers that are crucial for the diagnosis of gliomas are not visible from MRIs. Table 1.2 lists some different molecular biomarkers based on which gliomas

can be categorized into different subtypes.

Table 1.2: Gliomas and their molecular biomarkers according to [22]. IDH: isocitrate dehydrogenase, TP53: tumor protein p53, ATRX: alpha-thalassemia/mental retardation syndrome X-linked, MGMT: O^6-methylguanine-DNA methyltransferase, H3 K27M: substitution to methionine at 27 position in histone variant H3.3.

Glioma subtype	Molecular biomarkers
Astrocytoma	IDH1/2, TP53, ATRX
Oligodendroglioma	IDH1/2, 1p/19q codeletion, TERT
Glioblastoma	IDH1/2, TERT, MGMT methylation
Diffuse midline glioma	H3 K27M, ATRX, TP53

Codeletion of 1p/19q defines the oligodendrogliomas, and it is a strong prognostic molecular marker associated with the longer survival [23], [24]. Since oligodendrogliomas are more sensitive to chemotherapy, the role of surgery is controversial [25], [26]. Accurate non-invasive classification would ease the diagnostic process since determining 1p/19q status today requires at least a surgical biopsy.

The mutations of isocitrate dehydrogenase (IDH) are observed in 12% of glioblastomas [27], and 50% to 80% of LGG [28]. Patients with IDH mutated gliomas have a significant increase in overall survival rate than those with IDH wild-type gliomas [29]–[31]. Hence, IDH mutation information is important for diagnosis, prognosis and guidance in clinical decisions. The identification of IDH mutation is challenging, and it usually requires tissue diagnosis from an invasive procedure (e.g. biopsy or resection) that involves some risks to patients.

Related work

Imaging tests and biopsies can be used to diagnose gliomas. A biopsy is a procedure to extract a sample of tissue from the brain followed by further analysis in a laboratory. It is a more direct and definitive method for diagnosis. Tissues can be extracted during surgeries for removing tumors or in a pre-operative biopsy, which is usually guided by CT or MR scanning. However, the safety of biopsy is questioned because it can lead to potential complications and even threat to life. It is necessary to seek other effective non-invasive brain tumor diagnostic tools for assisting medical doctors. Imaging approaches such as MR imaging has been widely used to show the anatomical structures for medical purposes as a non-invasive method. There are four main MRI modalities, known as T1-weighted (T1), T2-weighted (T2), post-contrast-enhanced T1-weighted (T1ce) and T2-weighted-fluid-attenuated inversion recovery (FLAIR). They emphasize different tissues in the brain by different densities as shown in Figure 1.4.

However, it is difficult to tell the molecular-based glioma subtypes by just inspecting MRIs even for experienced medical doctors, as the signs related to molecular-based glioma subtypes can hardly be seen with human eyes. With the help of AI technologies, machine

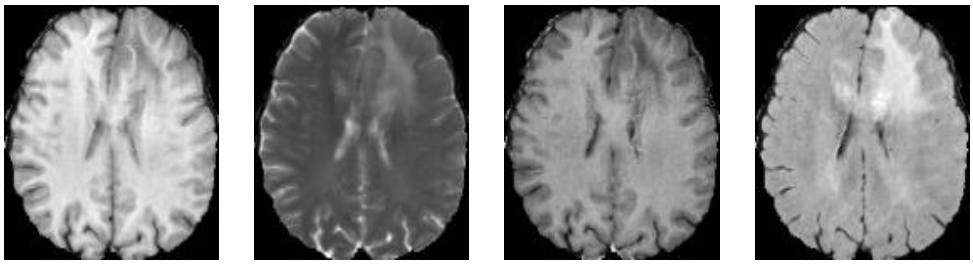

Figure 1.4: Examples of brain MRIs in four different modalities. From left to right: T1, T2, T1ce, FLAIR.

learning can be used to assist medical doctors in the diagnosis of gliomas. Such methods for characterizing gliomas can be roughly divided into two paradigms: those using hand-crafted features (i.e. features defined by human experts), and those using deep learning methods for automatically learning the features. Kang et al. [32] analyzed histograms of apparent diffusion coefficient maps based on the entire tumor volume for grading gliomas. Carrillo et al. [33] used features from MRIs such as tumor size, frontal lobe localization, presence of cysts and satellite lesions to classify glioma patients between IDH mutation and wild-type. Qi et al. [34] studied MRI features such as the pattern of growth, tumor margins, signal density and contrast enhancement to predict IDH mutation. Yu et al. [35] extracted features such as location, intensity, shape, texture and wavelet features for grade II glioma classification. Zhang et al. [36] used texture, histogram and Visually Accessible Rembrandt Images (VASARI) features with an SVM classifier to detect IDH and TP53 mutations. Shofty et al. [37] extracted features like size, location and texture of gliomas from images in three modalities, and 17 machine learning classifiers were tested for LGG classification with and without 1p/19q codeletion. The above methods used conventional machine learning methods with hand-crafted features from brain MRIs. Since characterizing glioma features related to molecular (e.g. IDH mutation) by purely using MRIs is very challenging to clinicians, defining hand-crafted features could be difficult.

Deep learning methods can offer solutions for such a glioma characterization issue by automatically learning MRI features. Recently, several deep learning methods for glioma classification have been proposed. Li et al. [38] proposed a six-layer CNN to segment tumors. Fisher vector was then applied to encode deep features from the last convolutional layer using image slices of different sizes, followed by feature selection and classification of IDH mutation using SVMs. Chang et al. [39] proposed to predict IDH mutation status of gliomas by applying residual CNNs on multi-institutional MRI data with four different modalities T1, T1ce, T2 and FLAIR. Dimensional and sequence networks were tested to evaluate the combination of multi-view and multi-modal images. Liang et al. [40] applied 3D DenseNets to predict IDH mutation status with multi-

modal MRIs. The network also showed high generalization to glioma grade classification (e.g. classify LGG and HGG). Although these methods achieved some promising results, research on glioma subtype classification is still in its infant stage. Challenges remain before any clinical usage, such as small clinical datasets and incomplete labels of scans.

1.3 Fall Detection for Assisted Living

Ageing has become a global issue that leads to a rising cost of healthcare every year. According to the WHO [41], the world's population aged 60 years and older is expected to reach 2 billion by 2050, up from 900 million in 2015. There is a growing need for assisted-living in elderly care due to the increased number of elderly people.

A human-centered assisted living system usually includes the following several basic functions. For example, 1) detecting abnormal activities in case of emergencies such as falls and robberies, 2) recognizing daily activities/living patterns to obtain statistics, 3) locating the person if help is needed, 4) giving recommendations to the subject. Figure 1.5 shows some examples of 8 daily activities including eating, drinking, using a laptop, reading, falling, lying down, walking and sitting down.

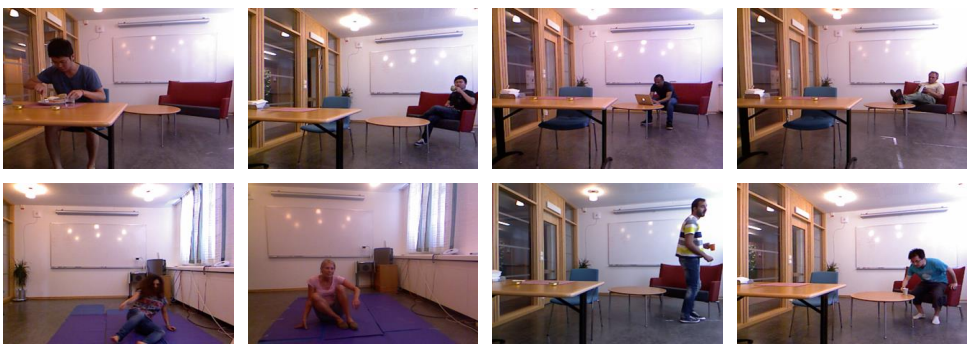

Figure 1.5: Example images of 8 daily activities from a dataset in [42]. The first row from left to right: eating, drinking, using a laptop, reading; the second row from left to right: falling, lying down, walking and sitting down.

Statistics show that falling has been a serious risk to the ageing group, which could lead to bone fracture, coma, and even death. Emergent medical attention is often required after a fall. Considering that many elderly people live alone, it is not easy for them to seek immediate help if severe injuries or unconsciousness occurs after a fall. Thus, there is a great need for automatic fall detection and fall alert. An example of fall detection and emergency alert system [43] is shown in Figure 1.6, where alert cancellation feedback is added for the user to cancel any possible false alarms.

To help fallen people, an airbag system was developed in [44] to protect hips after a

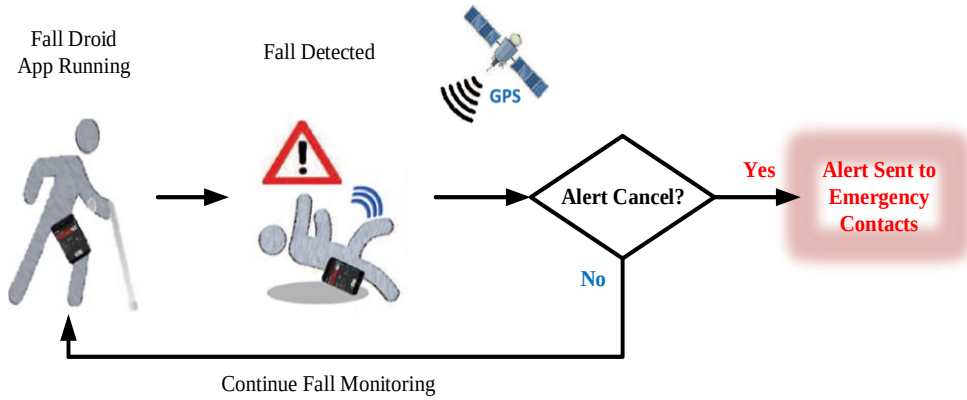

Figure 1.6: Example of a fall detection and emergency alert system from [43].

fall, similar to the function of airbags after a car crash. A triaxial accelerometer and gyroscope was applied as well to make sure that the airbag was filled with air before any collision. [45] proposed a social alarm for the elderly who live alone. It was realized by a wristwatch with a button that can be activated by the subject after a fall. However, this solution of actively seeking help might be unreliable as the person could become unconscious after a fall.

Related work

In recent years, there is a rapid growth of interests on such automatic systems for detecting falls and alarm triggering. Methods of fall detection can be categorized into using sensor-based devices and using video-based analysis. All these methods have a similar pipeline [46] depicted as follows:

1. data acquisition from sensors,
2. signal processing and feature extraction,
3. fall detection,
4. fall alert sent through wired or wireless communications,
5. alert information received by caregivers.

Many current methods exploited wearable devices with motion sensors, such as accelerometers [47], gyroscopes [48] and tilt sensors [49], [50]. Good results were obtained in these methods for fall detection. However, elderly people often feel uncomfortable when wearing such kind of devices for a long time, or forget to wear them at times, let alone the frequent battery charging problems. Zigel et al. [51] presented an innovative

method for fall detection using floor vibration and sound sensing without wearable devices. These sensors can still have a side effect on the health of elderly people. Hence, using cameras to extract visual information may provide a solution to the aforementioned issues when privacy issues are properly handled.

For video-based fall detection, one way is to use a bounding box that compasses the target person in each frame. Qian et al. [52] utilized a two-bounding-box strategy where one was for characterizing the whole body and the other for the lower part of the body. Based on the variances of two boxes, features for fall detection were extracted and then fed into an SVM classifier. Charfi et al. [53] proposed 14 features in the bounding box including height, width, aspect ratio and centroid coordinates, then features were transformed (by Fourier transform, wavelet transform, etc) before classification using an SVM or AdaBoost. Yun et al. [54] represented the dynamic appearance, shape and motion of a target person as points moving on a Riemannian manifold. Based on the dynamics of different features, velocity statistics were computed, followed by feature weighting and a two-stage boosting learning strategy. Another way to address fall detection is to employ multiple cameras or depth cameras. Rougier et al. [55] used shape matching to calculate the cost between consecutive frames as a criterion for shape deformations, and then a majority voting from four camera views was used to decide whether a fall occurred. Ma et al. [56] proposed to learn curvature scale space (CSS) features from human silhouettes based on depth images. Bag of CSS words was utilized to represent actions, which were then classified into falls and other actions by extreme learning machine (ELM). Stone and Skubic [57] developed a two-stage fall detection system, where the vertical state of a person in each depth image frame is modeled. An ensemble of decision trees is then used to compute the likelihood of falls.

Deep learning has been exploited for automatically learning video-based features related to human activities/actions. Simonyan et al. [58] proposed two-stream convolutional networks, where still images and stacks of optical flow fields were used to separately capture spatial and temporal information. Tren et al. [59] exploited 3D CNNs to learn spatio-temporal features from videos without calculating the optical flow. Ng et al. [60] investigated several temporal feature pooling methods and LSTMs to learn CNN features across long periods, and experimental results showed that temporal pooling of CNN features performed better. Fan et al. [61] considered four phases in each fall (standing, falling, fallen and not moving) and trained deep CNNs to distinguish four categories of dynamic images corresponding to the above four phases. Zhang et al. [62] proposed to use trajectory attention maps to enhance the CNN feature of each frame, and rank-pooling-based encoding method was then used to obtain the feature descriptor for each activity. Despite all these promising results achieved for human fall detection, it is still challenging to develop a deep learning strategy that can distinguish activities of different lengths in similar indoor settings.

1.4 Obstacle Avoidance in Prosthetic Vision for Assisted Living

Degenerations of photoreceptor cells such as retinitis pigmentosa and age-related macular degeneration are devastating causes of vision loss. To restore vision to the blind, implantation of prostheses may become a treatment option in the neuroengineering field. Prostheses first transmit image data to information processing units. After stimulation patterns are captured by electrode arrays, surviving neural cells in the visual pathway can be electrically activated, and visual perception is stably restored [63]–[65]. Such electrically induced visual sensations are called "phosphenes", conveying limited but useful visual information to the blind. Discernible phosphenes are usually generated in the following three locations: the visual cortex [66], the optic nerve [67], and the retina [68].

In recent years, retinal prostheses have gained much attention and achieved encouraging performance. Epiretinal prostheses were implanted on the inner surface of the retina, stimulating retinal ganglion cells and axons [69]. A 60-electrode epiretinal prosthetic system improved basic visual tasks such as orientation, mobility and letter reading [70], [71]. Besides, subretinal prostheses are another kind of retinal devices implanted under the transparent retina to replace degenerated photoreceptors [72]. A subretinal prosthesis prototype with 1500 microphotodiodes was implanted in three subjects and demonstrated to be helpful in some visual tasks [64].

Many technical factors such as implant packaging, electrode manufactory and biocompatibility limit the maximum number of implantable electrodes, leading to a low-resolution visual perception and the difficulty of understanding the contents. Hence, researchers find it necessary to improve the image quality of phosphenes in order to assist the prosthesis wearers, so that they can perform better in visual tasks. However, enhancing the high-level functionality of visual prosthesis such as obstacle avoidance is rarely explored, although it is a common but important task in daily lives of visually impaired people.

Related work

Obstacle avoidance was mostly considered as the task of satisfying some control objective subject to non-intersection or non-collision position constraints in robotics. In [73], a vector field histogram method was developed and tested on an experimental mobile robot. A vision-guided local navigation system was proposed in [74] to compute a potential field over the robot heading, steering it towards the target and away from obstacles. By utilizing a vision-based multi-person tracker, a dynamic obstacle map was generated in [75] to enable path planning in complex and highly dynamic scenes. However, all the above methods were not specially designed for the blind to fulfill the obstacle avoidance task.

Current solutions of navigations for the visually impaired individuals can be categorized as:

- white cane,
- dog guide,
- multi-sensor-based travel aid.

According to [76], there were already numerous navigation systems and tools for visually impaired individuals, among which white canes and dog guides were the most popular ones. In order to offer more information such as speed, volume and distances to the visually impaired, a category of devices called electronic travel aids were designed. With the combination of different sensors such as sonars, laser scanners and cameras, multi-sensor information was gathered to guarantee the control of locomotion during navigation.

Researches on visual prostheses were concentrated on how implant recipients interpret visual information from electrical stimulation by simulation of prosthetic vision. In [77], the number of individual Chinese characters required for accurate recognition by blind Chinese subjects was explored. Zhao et al. [78] found out that distortion, dropout percentage, and pixel size had an impact on the recognition of Chinese characters. In addition, many image processing strategies were proposed in simulated prosthetic vision. Parikh et al. [79] first applied a saliency-based method and provided cues for the region of interest detection in simulated vision. By exploring different face detection methods, Wang et al. [80] concluded that such image processing methods can highlight useful information hence improve visual perception of prosthesis wearers. In [81], a background-subtraction-based technique was used to optimize the content of dynamic scenes of daily life in prosthetic vision. Han et al. [82] utilized feature extraction and image enhancement strategies to improve the accuracy and efficiency of object recognition. Aimed at highlighting the main object from a normal image, two different ways of pixelization [83] were proved to be effective in daily object recognition tasks. Despite these promising researches on improving the quality of phosphene images obtained from visual prostheses, challenges remain such as enhancing the high-level functionality of prostheses, where obstacle avoidance is an important task to the blind.

1.5 Outline of this Thesis

To address the above health-related issues, this thesis explores dedicated machine learning techniques. It consists of two parts. Part I gives the general introduction to the research background and a summary of the appended papers. Part II contains the appended papers. The remainder of this introductory part (Part I) is organized as follows: Chapter 2 reviews several fundamental theories and methods on which the proposed methods are built. Chapter 3 summarizes the main work and contributions of each method, followed by Chapter 4 where conclusion and future work are presented.

CHAPTER 2

Background Theories and Methods

2.1 Deep Convolutional Neural Networks

A Convolutional Neural Network (CNN) [84] is a class of deep neural networks. An example is shown in Figure 2.1. In the feature learning module, several convolutional and pooling layers are used. For each convolution, a set of filters is applied to the input feature map/original signal to generate new feature maps followed by a non-linear activation function. For pooling operation, down-sampling is performed to reduce the size of the input feature map. After the feature learning module, the obtained feature maps are flattened and concatenated as the input to the classifier. The classifier consists of several fully-connected (FC) layers that have full connections to all the neurons in the previous layer similar to the regular neural networks. Finally, a prediction is made by a softmax function showing the probability of the input data belonging to each class. To train a CNN, the loss (error) is calculated comparing the predicted result with ground truth, and the parameters of all the layers are updated by backpropagating the loss.

Convolutional layers
The function of a convolutional layer is to generate feature maps from the output of its previous convolutional layer or the original image with filters of a certain size. Small-size filters with odd dimensions in width and height (or depth for 3D cases) are mostly used. When doing convolution on the feature map, a filter with certain parameters is sliding on the input feature map/image with a certain step called "stride". The larger the stride is, the more pixels are skipped during convolution, resulting in a smaller feature map as the output.

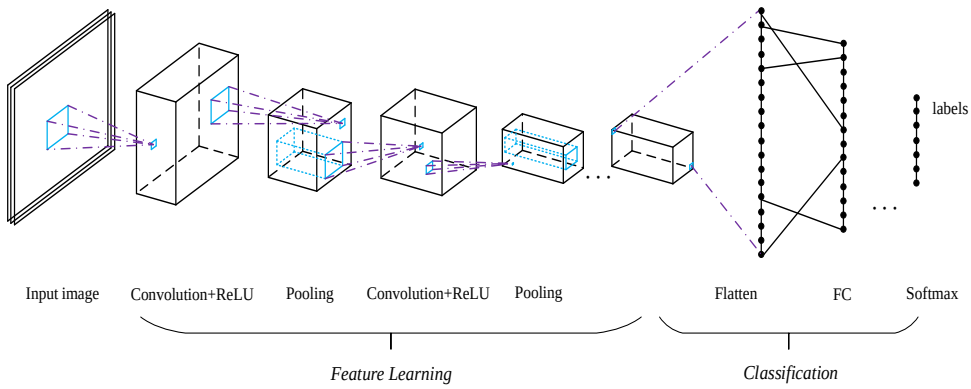

Figure 2.1: Example of a Convolutional Neural Network (CNN).

The resulted feature map will shrink after convolution if the filter size is chosen to have a certain size except one. To preserve the size of the feature map after convolution (otherwise, the information around image borders will be gradually removed), the zero-padding approach is usually used where some zero values are padded around the input feature map/image boundary. Padding size can be chosen based on the filter size so that the output feature map will remain the same size. It turns out that smaller strides work better in practice, and stride one is commonly used in convolutions so that convolutional layers only transform the input volume depth-wise without changing its size. One common strategy used in the convolutional layer is parameter sharing to control the number of parameters. It is based on an assumption that if one feature is useful to compute at one spatial position, then it is also useful to compute at a different position.

An example of convolution operation is shown in the left part of Figure 2.2. Each neuron in the convolutional layer is connected to a small region of the input image/feature volume in the full depth.

Pooing layers

Pooling is usually conducted to perform down-sampling on the feature maps periodically in-between successive convolutional layers. It can reduce the dimension of feature maps and help mitigate potential overfitting. Another function of pooling is to smooth feature maps and reduce noise. It also leads to a decrease in computational cost as the future maps for the following processing will become smaller in size. Similar to the previous convolutional layer, stride can also be used in the pooling operation to skip certain pixels when sliding the pooling filters.

Two most commonly used pooling methods are max pooling and average pooling. Max pooling (as shown in the right part of Figure 2.2) returns the maximum value in the pooling window while average pooling returns the average value. Between these two kinds of

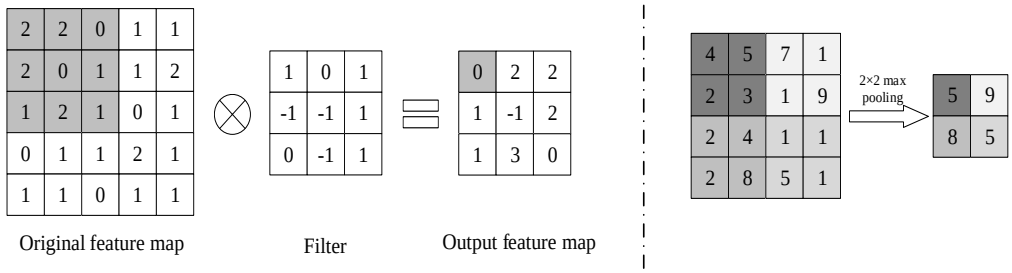

Figure 2.2: Illustrations of convolution operation and pooling operation. Left: example of a 2D convolution with no padding and stride=1. Right: example of a 2D max pooling operation with stride=2. Corresponding input and output are illustrated in the same grey scale.

pooling strategies, max pooling has been shown to work better in practice.

Loss function

In the classification module, a loss function is used to evaluate the performance of the model by measuring the difference between the ground truth label y and the predicted label $\hat{y}$. The performance of a CNN usually improves with the decrease of the loss. The most common loss for a classification task is cross-entropy loss defined as follows in the binary case, where N denotes the number of samples in the training set.

$$L = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)). \tag{2.1}$$

Hinge loss is also widely used for classification tasks, to maximize the margin between the predictions of the true class and the other classes.

$$L = \frac{1}{N}\sum_{i=1}^{N}\sum_{j \neq y_i} \max(0, f_j - f_{y_i} + 1), \tag{2.2}$$

where f_{y_i} denotes the y_i-th element (corresponding to the true class) of $\mathbf{f}$, the vector of activations from the output layer of a CNN.

As for a regression task (e.g., training an autoencoder), mean squared error is commonly adopted as the loss function:

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2. \tag{2.3}$$

CNN training

To train a CNN, backpropagation strategy [85] with the gradient descent method is used,

similar to training a regular neural network. The gradients are used to perform updates for all the parameters in different layers of a CNN. The most common gradient descent method for optimizing a CNN is stochastic gradient descent (SGD), where parameters are moved towards the direction of the steepest descent in the parameter space in each iteration, resulting in the decrease of the loss. The drawback of SGD is that it manipulates the learning rate globally and equally for all parameters. Some variants of SGD are also commonly used to adaptively tune the learning rates such as Adam [86], Adagrad [87], RMSprop [88].

Regularization is widely used in CNN training to mitigate overfitting, as CNNs usually have a large number of parameters while the size of training data is sometimes not sufficiently large. Regularization can be considered as a way to reduce the complexity of a CNN model, and some common methods include L1/L2 regularization and dropout. L1/L2 regularization is realized by adding the L1/L2 norm of all the weights as an additional loss to the original loss function. It limits the capability of CNN by penalizing large weights. Dropout is another straightforward way to limit the capability of CNN by randomly dropping neurons in some layers during the training.

Representative CNN models

- CNN models stem from LeNet [89] that is the first to use weight sharing technique in convolutions for handwriting recognition.
- The first CNN model that significantly drew researchers' attention to deep learning is AlexNet [90]. It won the ImageNet Large-Scale Visual Recognition Challenge 2012 with breakthrough accuracy and greatly outperformed the other competitors. The success of AlexNet was due to the deeper network architecture with a large number of parameters. Activation function ReLU was adopted to accelerate the convergence of training. Techniques such as data augmentation and dropout were employed to alleviate overfitting.
- VGG net [91] improved the performance of AlexNet greatly due to very small (3×3) convolutional filters and the depth extended to 19 layers. The use of small-size filters decreased the number of parameters in convolutional layers and thus made it possible to explore deeper CNN architectures.
- In GoogLeNet [92] an optimal local sparse structure called inception module was proposed to learn the features in a CNN. It consisted of 1×1, 3×3, 5×5 convolutional filters and 3×3 max pooling in parallel followed by concatenation as the output feature map. In addition, a dimension reduction technique was adopted to reduce computation cost by applying 1×1 convolutional filters before the 3×3 and 5×5 convolutional filters.
- ResNet [93] was proposed to ease the training of a very deep CNN by introducing an idea of residual learning. The intuition behind the residual learning is that if

the mapping a CNN is about to learn is closer to an identity mapping than to a zero mapping, it is easier to learn the perturbations based on an identity mapping than to learn a completely new mapping. Residual learning was realized by adding shortcut connections to a common CNN.

Although these CNN architectures show good performance on the ImageNet dataset, different networks need to be explored for specific problems/tasks as the dataset size, image modality and other details can be different from those of the ImageNet dataset. Task-specific configurations and parameter settings should be considered as well.

The success of CNN also inspired the research on a thorough understanding of CNN such as how each neuron is activated to learn the semantic information of an object in an image, instead of treating CNN as a black box. Interpreting and theorizing the mechanisms of CNNs [94], [95] have become a compelling research area, especially in applications such as medical diagnosis where wrong decisions can be costly and dangerous. Hence, much research is still needed to understand the essence of CNN and deep learning.

2.2 Recurrent Neural Networks

A Recurrent Neural Network (RNN) is a kind of network that allows information to persist during different time [96]. It can be considered as multiple copies of the same network, and each passes a message to the successor. An example of RNN is shown in Figure 2.3, where hidden layer vector $\mathbf{h}_t$ takes both original signal $\mathbf{x}_t$ and the output of previous hidden layer $\mathbf{h}_{t-1}$ as the input. By connecting multiple networks of different time steps, information can be passed from one step of the network to the next. With the chain-like structure, RNNs are applied to sequence-related problems such as speech recognition and video analysis. RNNs can be formulated as:

$$\mathbf{h}_t = \sigma(\mathbf{W}_{xt}\mathbf{x}_t + \mathbf{W}_{ht}\mathbf{h}_{t-1} + \mathbf{b}_{ht}), \tag{2.4}$$

$$\mathbf{y}_t = \sigma(\mathbf{W}_{yt}\mathbf{h}_t + \mathbf{b}_{yt}), \tag{2.5}$$

where $\mathbf{W}$ is the weight matrix, $\mathbf{b}$ is the bias vector, and $\mathbf{y}$ is the output vector. RNNs are designed to connect previous information to the present task. However, it suffers from the problem of handing "long-term dependencies" in practice. That is, when the gap between the relevant information and the place where it is needed is large, RNNs are not capable of connecting to the information because of the gradient vanishing problem [97].

As a solution, Long Short Term Memory (LSTM) networks [98] are designed to handle such a long-term dependency problem. LSTMs have a chain-like structure similar to RNNs, but differently, LSTMs have more complex repeating network modules where a

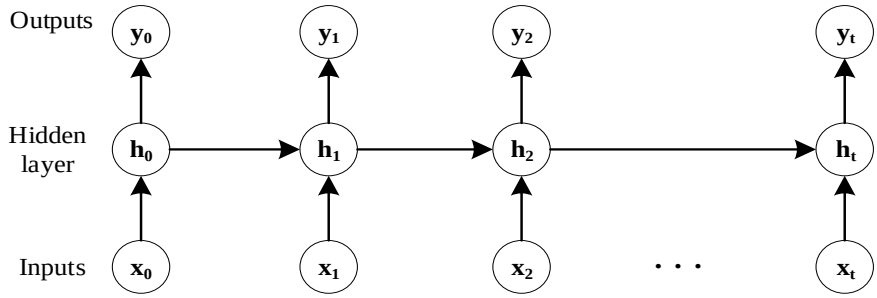

Figure 2.3: Example of a Recurrent Neural Network (RNN). The circles denote network layers and the solid lines denote the weighted connections. For simplicity only one hidden layer is used.

four-layer interactive neural network is used instead of one single-layer neural network in regular RNNs. A basic LSTM architecture is shown in Figure 2.4.

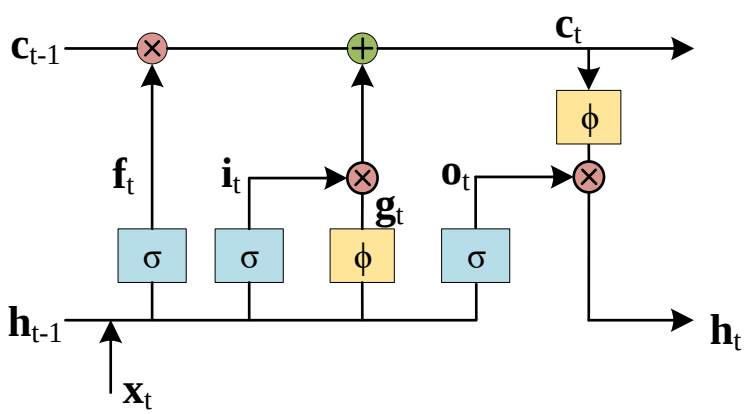

Figure 2.4: A basic LSTM architecture.

The main idea behind LSTMs is the introduction of cell state $\mathbf{c}_i$, as shown in the top horizontal line running through all repeating modules in Figure 2.4, which allows information to flow with minor linear interactions. Another special structure used in LSTMs is called gate consisting of a one-layer neural network with a sigmoid function and a point-wise multiplication operation. It controls how much information can be let through.

A basic LSTM unit contains a single memory cell, an input activation function and four different gates (input gate $\mathbf{i}_t$, forget gate $\mathbf{f}_t$, output gate $\mathbf{o}_t$ and input modulation gate $\mathbf{g}_t$). Input gate controls whether the incoming signal will alter the state of the

memory cell or block it. Forget gate allows the cell to selectively forget something and can somehow prevent the gradient from vanishing or exploding during backpropagation through time. The output gate makes the memory cell have an effect on other neurons or prevent it. The input modulation gate is a function of the current input and previous hidden state. With the help of these gates, LSTM can capture extremely complex, long-term temporal dynamics and overcome the vanishing gradient problems as well. For an input $\mathbf{x}_t$ at time step t, memory cell state $\mathbf{c}_t$ encodes everything the cell has observed until time t, and finally LSTM outputs the hidden/control state $\mathbf{h}_t$:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{b}_i), \\
\mathbf{f}_t &= \sigma(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{b}_f), \\
\mathbf{o}_t &= \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_o), \\
\mathbf{g}_t &= \phi(\mathbf{W}_{xg}\mathbf{x}_t + \mathbf{W}_{hg}\mathbf{h}_{t-1} + \mathbf{b}_g), \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t, \\
\mathbf{h}_t &= \mathbf{o}_t \odot \phi(\mathbf{c}_t),
\end{aligned} \tag{2.6}$$

where $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid function to map the inputs into the interval [0, 1], $\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ is the hyperbolic tangent nonlinear function that maps its inputs into the interval [-1,1], $\odot$ is the element-wise product.

In addition to the classic LSTM described above, it has some variants. LSTM with peephole connections [99] lets the gate layers look at the cell state by adding the cell state vector to the computation of input, forget and output gates. Another variation is to couple forget and input gates by letting $\mathbf{f}_t = 1 - \mathbf{i}_t$ without separately deciding what to forget and what to add. Gated Recurrent Unit (GRU) [100] uses an update gate to combine the forget and input gates, and the cell state and hidden state are merged as well. LSTM with full gate recurrence adds recurrent connections between all the gates similar to the original LSTM [98]. Despite the variants in different architectures, they do not improve the performance of the classic LSTM architecture significantly [101].

Dealing with variable length input sequences

Learning sequences of unequal length is a commonly encountered problem especially in audio processing. Sequence padding is a solution to make all the sequences have the same length by padding the short sequences with zeros. Sequence truncation offers another solution by trimming all sequences to the desired length. When it comes to video classification of variable length, similar strategies can be applied by padding or trimming some frames. However, this way makes it difficult to train RNN-based video classification using CNN features for each frame in an end-to-end way, as the number of CNNs (corresponding to different frames of a video sequence) needs to be fixed to learn features from each frame. A new solution is presented in Paper 6 by first splitting each video activity into a fixed number of segments. A representative frame is then chosen for each segment, followed by a CNN architecture to learn its features. CNN features of each representative frame are finally fed into LSTMs to further learn sequential features

for classification.

2.3 Generative Adversarial Networks

A Generative Adversarial Network (GAN) [102] belongs to the generative model that is capable of generating data. It contains a generator G and a discriminator D for adversarial training, and they can be multilayer perceptrons or convolutional neural networks. A prior on input variable $p_z(\mathbf{z})$ is first defined, then the generator maps it to data space represented as $G(\mathbf{z};\theta_g)$ to learn the distribution p_g of the generator over data $\mathbf{x}$, where θ_g is the parameters of the generator G. The discriminator $D(\mathbf{x};\theta_d)$ is defined to represent the probability of $\mathbf{x}$ coming from the data distribution p_{data}, where θ_d is the parameters of the generator D. It is trained to distinguish between the samples from p_{data} and those from p_g. Simultaneously, the generator G is trained to minimize $\log(1-D(G(\mathbf{z})))$ so that the generated sample $G(\mathbf{z})$ can fool the discriminator D. Alternatively, D and G can be considered to play the two-player minmax game with the following value function $V(G,D)$:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1-D(G(\mathbf{z})))]. \tag{2.7}$$

In practice, optimizing D to completion in the inner loop of training would lead to overfitting given finite datasets. Instead, k steps of optimizing D and one step of optimizing G are conducted in alternation. In this way, D is maintained near optima as long as G changes slowly enough. According to [102], $\log(1-D(G(\mathbf{z})))$ saturates in the early stage of training GAN so that no sufficient gradient is provided for G to learn well. An alternative strategy could be adopted to train G where minimizing $\log(1-D(G(\mathbf{z})))$ is replaced by maximizing $\log D(G(\mathbf{z}))$.

Apart from the original GAN described above, there are some variants of GANs. Some focus on modifying the optimization of GANs:

- Wasserstein GAN (WGAN) [103]: This method is proposed to improve the training of original GAN, since Jensen-Shannon (JS) divergence does not provide sufficient gradient when the generated data distribution does not overlap with the real data distribution. In this method, Earth mover's distance is used as the distance measure for optimization, where $\prod(p_{\text{data}}, p_g)$ is the set of all joint distributions whose marginals are p_{data} and p_g.

$$W(p_{\text{data}}, p_g) = \inf_{\gamma\in\prod(p_{\text{data}},p_g)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma}\|\mathbf{x}-\mathbf{y}\|. \tag{2.8}$$

- Least squares GAN [104]: This method is proposed to remedy the vanishing gradient problem for the generator G. Least squares losses L_D and L_G are used for the

discriminator D instead of the conventional cross-entropy loss, where a is the label for the real samples and b is the label for the generated samples.

$$\min_D L_D = \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[(D(\mathbf{x}) - a)^2] + \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[(D(G(\mathbf{z})) - b)^2], \tag{2.9}$$

$$\min_G L_G = \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[(D(G(\mathbf{z})) - a)^2]. \tag{2.10}$$

Some other variants of GANs focus on network architectures derived from the original GAN targeting at different applications. BiGAN [105] and ALI [106] add an encoder structure for mapping the data back to the feature space. Such an inference mechanism can be useful in discriminative tasks or for a better understanding of what a trained GAN model has learned. Conditional GAN [107] uses extra label information as the condition on both the generator and discriminator. One example is to generate MNIST digits conditioned on class labels. When the condition is an image, conditional GAN can perform image-to-image translation task [108] in a supervised manner. As shown in Figure 2.5, conditional GAN is trained to transform a sketch image to a normal image. The generator takes a sketch shoe as the input and outputs a normal shoe image. Here a noise input to the generator is abandoned as the generator is simply learned to ignore the noise [108]. The discriminator distinguishes a pair of sketch image and normal image, so that the output normal image is constrained on the input sketch image.

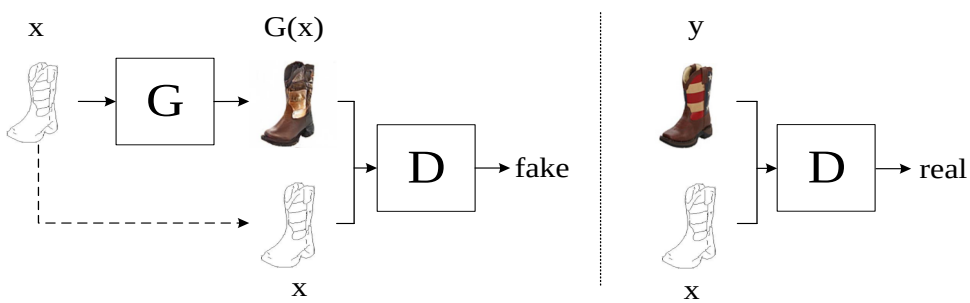

Figure 2.5: Illustration of image-to-image translation using conditional GAN [108]. G represents the generator and D represents the discriminator.

The above supervised image-to-image translation method requires paired images as the input. By using cycle-consistency loss [109] or latent space assumption [110] as the constraint, image-to-image translation can be performed in an unsupervised manner. Such constraints guarantee that input and output images are related.

- Cycle GAN [109]: This method addresses two mappings $G(X) \rightarrow \hat{Y}$ and $F(Y) \rightarrow \hat{X}$ by introducing the cycle-consistency loss, where two generators G, F and two discriminators D_X, D_Y are trained together. The cycle-consistency is based on the intuition that if one image is translated from one domain to the other and back again, the same image will be obtained, as shown in an example in Figure 2.6.

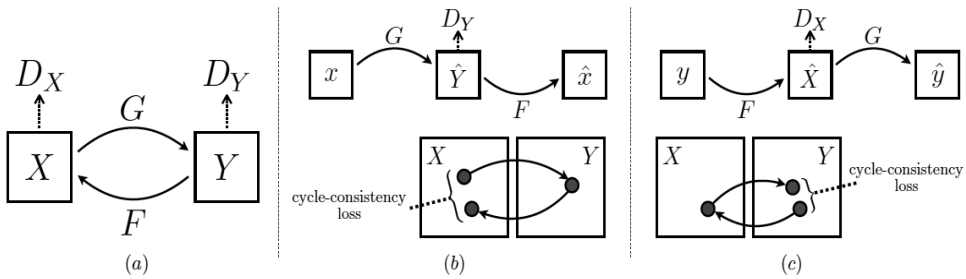

Figure 2.6: Illustration of cycle GAN [109]. (a) Two mapping functions G, F and their associated adversarial discriminators D_Y and D_X. (b) Forward cycle-consistency loss is computed as the difference between $\mathbf{x}$ and $F(G(\mathbf{x}))$. (c) Backward cycle-consistency loss is computed as the difference between $\mathbf{y}$ and $G(F(\mathbf{y}))$. The image is taken from [109].

- Unsupervised Image-to-Image Translation (UNIT) [110]: This method addresses unsupervised image-to-image translation under shared latent space assumption, as shown in Figure 2.7. A pair of images $(\mathbf{x}, \mathbf{y})$ in two different domains X and Y can be mapped to the same latent code $\mathbf{z}$ in a shared-latent space Z. This assumption implies the cycle-consistency assumption (but not vice versa). Shared latent space assumption is implemented by sharing the weights of the last few layers (high-level layers) in E_x and E_y, as well as those of the first few layers (high-level layers) in G_x and G_y, see the dashed lines.

A natural extension to image-to-image translation is video-to-video synthesis [111], with the aim to learn a mapping function from an input source video to an output video that shows the same content as the source video. The source video can be a sequence of semantic segmentation masks or edge maps, similar to the condition defined in an image-to-image translation task. A Markov assumption is made in [111] so that the generation of the t-th frame is only dependent on the previous L frames including the source images and generated images. Optical flow is also added as a constraint to deal with the redundancy in videos. This model is further generalized to the few-shot video-to-video synthesis [112] to synthesize videos of previously unseen subjects or scenes.

The applications of GAN also include super resolution [113], object detection [114], sequential data generation [115], domain adaptation [116], etc. In conclusion, GANs show

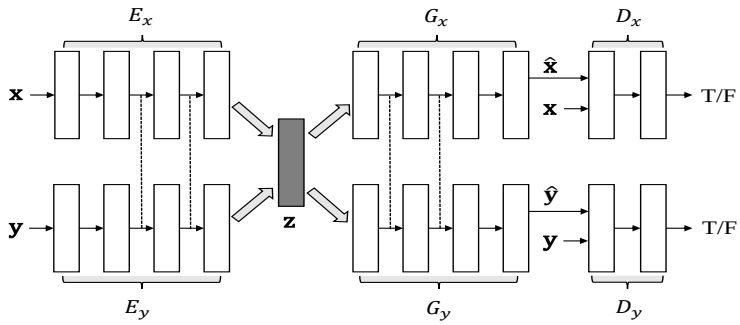

Figure 2.7: Illustration of unsupervised image-to-image translation [110] using shared latent space assumption. A pair of images $(\mathbf{x}, \mathbf{y})$ are mapped to the same latent code $\mathbf{z}$ using two encoding functions E_x and E_y, followed by two generators G_x and G_y to map the latent code to images. D_x and D_y are two corresponding discriminators.

their advantage in generating better and sharper results than other generative models like variational autoencoder [117]. However, it also suffers from well-known problems such as model collapse [118] where GANs fail to generate samples with diversity.

2.4 Spiking Neural Networks

A Spiking Neural Network (SNN) is the third generation of neural network models that employ spiking neurons as the computational units [119]. It is inspired by the experimental evidence that many biological neural systems use spikes to encode information [120]. The most common model for a spiking neuron is "integrate and fire neuron" [121]. For simplicity, one may assume that a neuron fires when its potential P_v reaches a certain threshold θ_v. P_v denotes the electric membrane potential of neuron v at the trigger zone, and it is the sum of excitatory postsynaptic potentials (EPSP) and inhibitory postsynaptic potentials (IPSP). These potentials are obtained from the firing of other neurons u, which are connected to neuron v through a synapse. The firing of the presynaptic neuron u at time s changes the potential P_v with certain amount that is modeled by the product of a weight $w_{u,v}$ and a response function $\varepsilon_{u,v}(t-s)$, as shown in Figure 2.8. The weight term $w_{u,v}$ reflects the strength of the connection (synapse). For mathematical convenience, the potential P_v equals 0 in the absence of postsynaptic potentials, while for a typical biological neuron P_v is around -70 mV in the absence of postsynaptic potentials.

Each spiking neuron has an absolute refractory period that lasts a few milliseconds after firing. This mechanism is modeled by a threshold function $\Theta_v(t - t^{'})$ as shown in Figure 2.9, where $t^{'}$ is the most recent firing time of v.

A spiking neural network [122] consists of a set of spiking neurons V, a set of synapses $E \subseteq V \times V$, weights $w_{u,v}$, response functions $\varepsilon_{u,v}$ for each synapse $\langle u, v \rangle \in E$ and

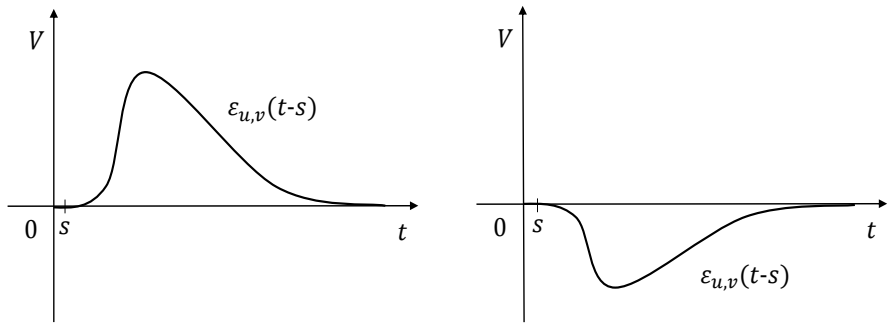

Figure 2.8: The shape of the response functions (Left: EPSP, Right: IPSP) of a biological neuron. The image is reproduced from [119].

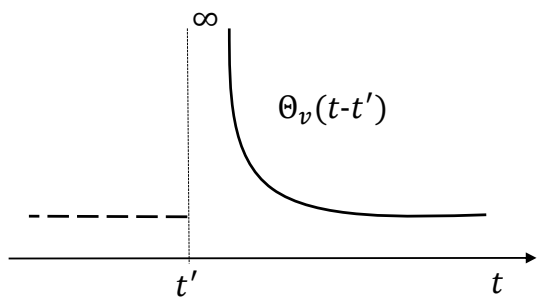

Figure 2.9: The shape of the threshold function of a biological neuron. The image is reproduced from [119].

threshold functions Θ_v for each neuron $v \in V$. Let F_u be the set of firing times for the neuron u, the potential at the trigger zone of neuron v at time t is computed as:

$$P_v(t) := \sum_{u:\langle u,v\rangle \in E} \sum_{s \in F_u : s < t} w_{u,v} \cdot \varepsilon_{u,v}(t-s). \tag{2.11}$$

To train SNNs, input data is first encoded into spike sequences. The spike sequences are then entered into neurons in SNNs, so that the temporal features of the original data are converted to the statuses of neurons in SNNs through spikes [123]. Although SNNs are closer to achieving natural intelligence through brain-like computation compared to other more abstract models, their performance on typical benchmarks has not reached the same level as their machine learning counterparts [124].

2.5 Semi-Supervised Learning

Different from the conventional supervised learning that fully relies on labeled data, semi-supervised learning uses both labeled and unlabeled data for training. It is motivated by the scenario where only a small amount of data has labels and most available data is unlabeled, since a lot of human annotation effort is often required for labeling data. Figure 2.10 shows a comparison between supervised learning and semi-supervised learning. Because a large amount of the unlabeled data is exploited for training the semi-supervised classifier, it can capture the latent data distribution better than the conventional supervised learning with only labeled data.

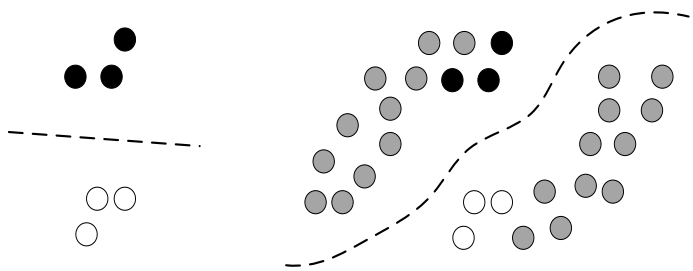

Figure 2.10: A comparison between supervised learning and semi-supervised learning. Left: supervised learning. Right: semi-supervised learning. The black and white dots show samples of two classes, and the grey dots denote the unlabeled samples. Dashed curves in the left and right subfigures indicate decision boundaries.

Graph-based semi-supervised learning

Graph-based semi-supervised learning is a branch of semi-supervised learning methods that infer the labels of unlabeled data using a graph structure derived from both labeled and unlabeled data. Some fundamentals of graph theory are introduced here. A graph model consists of three basic sets: vertex $\mathcal{V}$, edge $\mathcal{E}$ and weight $\mathcal{W}$. A graph can be directed or undirected, depending on whether each edge has a certain direction. In this thesis, only undirected graphs are involved, so all the graphs below refer to undirected graphs. Figure 2.11 shows an example of an undirected graph. It consists of 6 vertices $\{v_1, v_2, v_3, v_4, v_5, v_6\}$, 8 edges $\{e_{12}, e_{13}, e_{23}, e_{25}, e_{35}, e_{45}, e_{46}, e_{56}\}$ and corresponding weights $\{w_{12} = 1, w_{13} = 1, w_{23} = 0.6, w_{25} = 1, w_{35} = 0.8, w_{45} = 0.5, w_{46} = 0.8, w_{56} = 1\}$. The edge matrix $\mathbf{E}$ and weight matrix $\mathbf{W}$ showing the edges and weights can be described as follows:

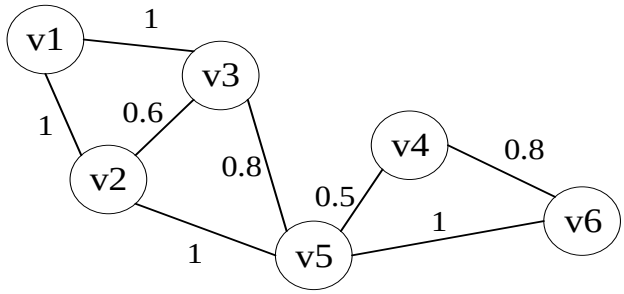

Figure 2.11: Example of an undirected graph.

$$\mathbf{E} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}, \mathbf{W} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0.6 & 0 & 1 & 0 \\ 1 & 0.6 & 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.8 \\ 0 & 1 & 0.8 & 0.5 & 0 & 1 \\ 0 & 0 & 0 & 0.8 & 1 & 0 \end{bmatrix},$$

where $\mathbf{E}$ and $\mathbf{W}$ are both symmetric and with 0 on its diagonal. For each vertex v_i in a graph, its degree $vol(i)$ is defined as:

$$vol(i) = \sum_{j=1}^{n} W_{ij}. \tag{2.12}$$

If the degree of a vertex is zero, this vertex is called an isolated vertex. The degree matrix $\mathbf{D}$ of a graph is defined as a diagonal matrix whose ith diagonal item is $vol(i)$.

The Laplacian matrix of a graph is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. It has some interesting properties. 1) $\mathbf{L}$ is semi-positive definite if all the edge weights are non-negative. 2) One of the eigenvalue of $\mathbf{L}$ is 0, its corresponding eigenvector is constant one vector $(1, 1, ..., 1)^T$.

There are several common ways to construct a graph.

- Full graph: Each vertex is connected to all the other vertices in the graph.
- k-nearest neighbour (kNN) graph: Each vertex is connected to its k-nearest neighbour vertices in the graph. This is the most popular method of constructing a graph in machine learning.
- ϵ-neighbour graph: Each vertex is connected to the vertices whose distance to it is within ϵ.

The followings are three ways to compute the weight of a graph.

- Binary 0-1 weight: It two vertices are connected, the weight between them is 1, otherwise the weight is 0.
- Gaussian similarity: The weight between two connected vertices is set to $\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are feature vectors associated with vertices i and j, σ is the standard deviation of Gaussian function also known as kernel width.
- Cosine similarity: The weight between two connected vertices is set to $1-\mathbf{x}_i^T\mathbf{x}_j/(\|\mathbf{x}_i\| \|\mathbf{x}_j\|)$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are defined similarly as before.

Graph-based semi-supervised learning can be formulated as learning a function $f(\mathbf{x}_i) = y_i$, where $\mathbf{x}_i$ is a sample usually characterized by features and y_i is its corresponding label. Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ be the labeled data with labels and $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$ be the unlabeled data. The vertices of a graph consist of both $\{\mathbf{x}_i\}_{i=1}^{l}$ and $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$. The goal of graph-based semi-supervised learning is to infer the labels of the unlabeled data $\{y_i\}_{i=l+1}^{l+u}$. Some graph-based semi-supervised learning algorithms are briefly reviewed as follows, where only binary labels $y_i \in \{-1, 1\}$ are considered for simplicity.

Harmonic function: f is obtained by solving the energy minimization function:

$$\min_{f(\mathbf{x})|f(\mathbf{x})\in\mathbb{R}} \sum_{i,j=1}^{l+u} w_{i,j}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2, \quad s.t. \quad \{f(\mathbf{x}_i) = y_i\}_{i=1}^{l}, \tag{2.13}$$

where the similar samples ($\mathbf{x}_i$ and $\mathbf{x}_j$) characterized by a large weight w_{ij} are encouraged to take similar labels, with a constraint that labeled samples should stick to their original labels. Since obtained label $f(\mathbf{x})$ will obtain a continuous value after optimization, a certain threshold is necessary for outputting final discrete labels.

Manifold regularization: Different from the harmonic function that fixes $\{f(\mathbf{x}_i) = y_i\}_{i=1}^{l}$, manifold regularization relaxes this constraint by adding an extra loss to penalize the difference between the predicted labels and ground truth of the labeled data.

$$\min_{f(\mathbf{x})|f(\mathbf{x})\in\mathbb{R}} \sum_{i,j=1}^{l+u} w_{i,j}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 + \lambda \sum_{i=1}^{l}(f(\mathbf{x}_i) - y_i)^2. \tag{2.14}$$

Graph-based semi-supervised learning is based on the smoothness assumption that two samples close to each other (in other words, they are connected with a strong edge in a graph) are more likely to share the same label. However, it suffers from the memory cost when the number of samples is large because of the matrix computation. In addition, when new samples (for training or testing) are added to the graph-based semi-supervised learning, the graph construction and label inference usually need to be conducted again since the previously constructed graph does not directly apply to the new data.

CHAPTER 3

Summary of the Work in This Thesis

In this chapter, we summarize the thesis work on machine learning methods for image analysis in medical applications, and the structure is shown in Figure 3.1. In the following subsections, each method and its contributions are summarized.

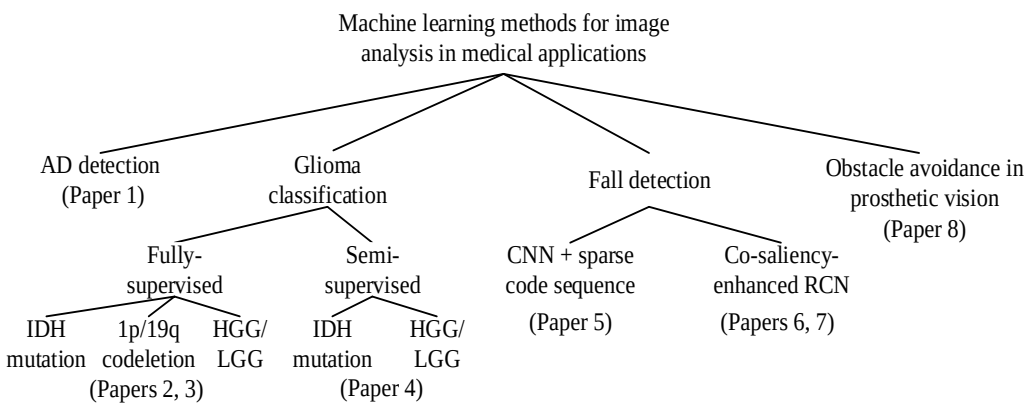

Figure 3.1: A summary of the thesis work on machine learning methods for image analysis in medical applications.

3.1 Deep Learning for Alzheimer's Disease Detection

(Summary of Paper 1)

Problem addressed: This method addresses the problem of Alzheimer's disease detection from MRIs. In this method, AD detection is formulated as a binary classification problem that distinguishes AD patients from NC subjects. Specifically, AD in MRIs is characterized by using multi-stream tissue inputs and fusing low-level detailed image features with high-level semantic features.

Motivations: Existing deep learning methods [16], [19] achieved good performance on AD detection by using a whole brain scan to characterize AD features. They adopted single-scale high-level deep features for classification. A better strategy can be adopted to enhance the performance if one explores both pixel-level and semantic-level information. It is observed that different types of tissues in a brain contain different characteristics with different resolutions for AD, hence they could contribute differently to AD detection. It is desirable to handle them separately by deep learning networks for learning their corresponding features. AD features for different tissue regions show different scales, e.g., changes in CSF and GM are very different for ADs. Furthermore, retaining all resolution levels (i.e., coarse to fine scales) of features enables one to obtain features both from low-level information of volume images to high-level semantic and structural change information in ADs. However, adopting a multi-stream multi-scale network significantly increases the dimension of features, hence it could lead to "the curse of dimensionality" in the classifier given a fixed medium-size dataset.

Basic ideas: The main idea behind this method is to characterize the AD features in a multi-stream multi-scale fashion. Multi-stream inputs (i.e., GM, WM and CSF regions) are used for characterizing AD with multi-scale features in each stream. Two-level feature fusion is then designed to fuse features in different scales in each stream, as well as in the stream level. A feature boosting and dimension reduction method is applied for mitigating overfitting caused by a large number of features.

Main contributions:

- Multi-stream feature extraction from segmented tissue regions (GM, WM and CSF) is proposed to generate complementary features in different tissue levels for AD detection. A 3D multi-scale deep convolutional network (3D MSCNN) architecture is proposed to learn multi-scale features with rich semantics and image details for each type of tissue.

- A two-level feature fusion approach is proposed, which includes fusion in the scale level and the stream level.

- A feature boosting and dimension reduction approach is utilized for post-processing, where a tree boosting method XGBoost is exploited for feature reduction.

- Extensive empirical analysis of the performance is conducted, where the proposed scheme is also compared with several state-of-the-art methods.

Overview: The block diagram of this method is shown in Figure 3.2. It consists of three streams of 3D MSCNNs (each stream is shown in Figure 3.3),

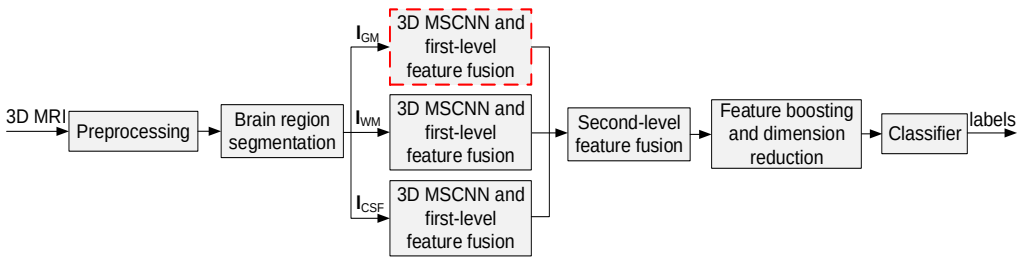

Figure 3.2: Overview of this method for Alzheimer's disease detection from MRIs.

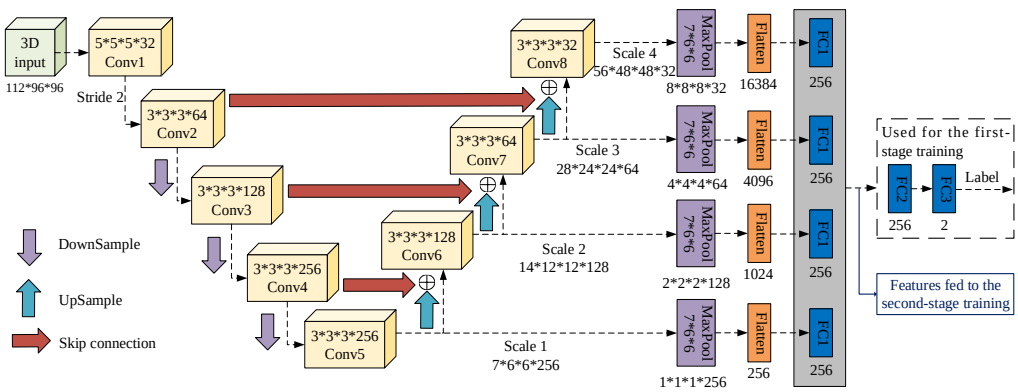

Figure 3.3: The proposed 3D multi-scale deep convolutional neural network architecture and first-level feature fusion, by zooming in the dashed red box in Figure 3.2.

separately applied on the GM, WM and CSF tissue regions. This is then followed by a two-level feature fusion method on the extracted features in different scales and tissue regions, respectively. Let $\mathbf{f}_j^{s_i}$, $i = 1, \cdots, 4$, be the features learned from four different scales for a given j-th tissue region, $j \in \{GM, WM, CSF\}$, then the first-level fusion is described by concatenating the feature vectors as: $\mathbf{f}_j = [\mathbf{f}_j^{s_1} \quad \mathbf{f}_j^{s_2} \quad \mathbf{f}_j^{s_3} \quad \mathbf{f}_j^{s_4}]$. The second-level fusion is performed on features from different tissue regions, by concatenating the feature vectors $\mathbf{f}_j$, $j = GM, WM, CSF$, obtained from different streams of 3D MSCNNs: $\mathbf{f} = [\mathbf{f}_{GM} \quad \mathbf{f}_{WM} \quad \mathbf{f}_{CSF}]$. After these steps, a feature boosting and dimension reduction

method is applied that is designed to retain the important/principal features while reducing the dimension of features. Finally, a classification step is used for AD detection by classifying between AD and NC subjects.

Main results: This method is tested on ADNI dataset from Alzheimer's Disease Neuroimaging Initiative [125] containing 337 subjects and 1198 3D brain scans. In the experiments, T1 MR scans of 2 classes (AD and NC) are used for classification. Comparisons are made with 8 existing methods on two different settings of training and testing datasets, subject-separated dataset partition and random dataset partition.

- For subject-separated dataset partition, brain scans from the same subject are kept together in one kind of set (training, validation or testing). The main results are shown in Table 3.1. The proposed method achieves the second best performance, as the tested dataset is much smaller than that of [126] in terms of subjects.

Table 3.1: Test results on ADNI dataset using subject-separated partition of training and testing sets. MCI: mild cognitive impairment.

Method	# Subjects AD/NC/MCI	#3D scans AD/NC/MCI	Accuracy (%) AD vs. NC
Proposed	**198/139/-**	**600/598**	**94.74**
SAE-CNN [126]	755/755/755	755/755/755	95.39
3DCNN [16]	198/139/-	600/598	90.11

- For random dataset partition, all brain scans are mixed together without considering subject information when partitioning them into training, validation and testing sets. The main results are shown in Table 3.2. The proposed method achieves the best performance.

Table 3.2: Test results on ADNI dataset using random partition of training and testing sets. MCI: mild cognitive impairment.

Method	# Subjects AD/NC/MCI	# 3D scans AD/NC/MCI	Accuracy (%) AD vs. NC
Proposed	**198/139/-**	**600/598**	**99.67**
3D-AE-CNN [19]	70/70/70	-	97.60
AE^+ [127]	65/77/169	-	87.76
SAE [18]	200/232/411	755/1278/2282	94.74
ICA [10]	202/236/410	-	85.70
MIL [11]	198/231/405	-	88.80
3DCNN [16]	198/139/-	600/598	98.74

3.2 Deep Learning for Brain Tumor Characterization and Classification

3.2.1 Fully-Supervised Deep Learning for Glioma Classification

(Summary of Papers 2, 3)

Problem addressed: These two methods address the problem of molecular-based glioma classification, as well as glioma grading by supervised deep learning methods using MRIs with fully-annotated labels.

Motivations: 1) In existing studies [35], [38] one or two types of MR modalities (like T1, or FLAIR) were exploited for glioma classification. Since brain images from different MRI modalities show different sensitivity and provide complemented information on gliomas, using multi-modal inputs followed by a fusion strategy is expected to perform better than using scans of a single modality. 2) Deep learning methods [38]–[40] have been successfully applied in learning glioma-related features for classification. However, effectively enlarging the training dataset for improving the performance of deep learning is hardly explored. Currently most available glioma datasets are relatively moderate in size, and often accompanied with incomplete MRI scans in different modalities. This may lead to overfitting in training and impact the generalization performance of deep learning classifiers on new testing data. Hence, we aim to seek robust methods for enlarging the size of the training dataset in order to cover more tumor statistics in deep learning.

Basic ideas: The main idea behind this method is to learn the characteristics of gliomas from multiple modalities, with enlarged training glioma dataset for improved performance of glioma classification, which can be further split as: 1) Incorporating MRIs from multiple modalities into a novel CNN framework, 2D multistream CNN for glioma classification. In the proposed scheme, modality fusion is conducted in the feature level by an aggregation layer followed by a bilinear layer to model the feature interaction; 2) Using pairwise GAN-based data augmentation for enlarging the size of the training dataset. The pairwise GAN is used to augment synthetic MRIs across different modalities to recover the ones of missing modalities, as well as augmenting synthetic MRIs for fake patients. It also offers more robustness as GAN-augmented MRIs cover more tumor statistics according to their distributions; 3) Using post-processing for 3D scan-level prediction on 3D volume images. Post-processing is used to combine the slice-level glioma classification results for each patient based on 3D volume images.

Overview: The proposed glioma classification scheme is shown in Figure 3.4. It uses four modalities of MRIs as the inputs (T1, T1ce, T2 and FLAIR). 2D image slices are extracted from 3D volume scans in four modalities, and they are partitioned into

training, validation and testing subsets. After that, a pairwise GAN model is employed to generate synthetic MRIs for the training dataset. Real and GAN-augmented MRIs are then utilized to learn the features and the classifier for brain tumor types. Finally, post-processing is conducted for the 3D scan-level diagnosis based on majority voting of slice-level glioma classification results.

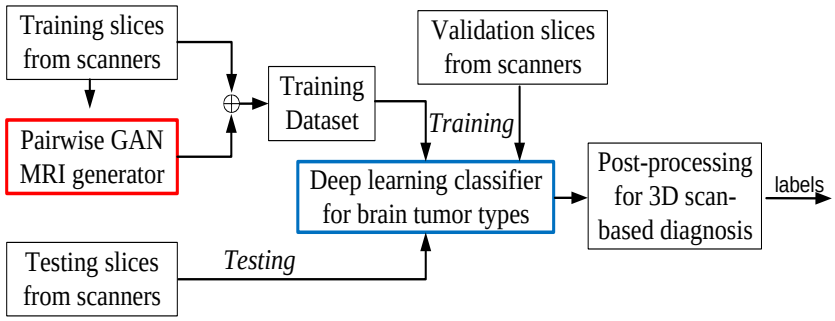

Figure 3.4: Overview of the proposed fully-supervised glioma classification scheme.

Main contributions:

Multi-Stream 2D CNN Scheme for Glioma Classification

The proposed multi-stream 2D CNN scheme for glioma classification is illustrated in Figure 3.5. Input 2D brain image slices from each of the three MRI modalities (e.g., T1ce, T2 and FLAIR) are fed into its corresponding stream of CNN, so that multi-stream CNNs form a set of parallel independent CNN networks. In such a way, modality-specific features can be learned through each CNN stream. For each CNN stream, a 7-layer 2D CNN is used for the extraction of features from one of the modalities, after careful selection through numerous empirical tests. An aggregation layer is then applied to aggregate these three-stream CNN features and obtain a compact feature representation. Let the features from the last convolutional layer be denoted as $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ of size $hw \times c$, where h, w, and c denote the height, width and number of channels of CNN features respectively. Element-wise multiplication is used to fuse the features by $\mathbf{X} = \mathbf{X}_1 \odot \mathbf{X}_2 \odot \mathbf{X}_3$, where $\mathbf{X}$ is the aggregated feature. In this way, features from high-level convolutional layers are aggregated and the spatial relationship between different modalities is retained. A bilinear layer [128], where the resulting feature map is computed as the outer product $\mathbf{y} = \mathbf{X}^T\mathbf{X}$, is adopted for modeling interactions of features from each other at all spatial locations. It also leads to a complement of features from different modalities. FC layers with activation functions are then added to generate the final classification results.

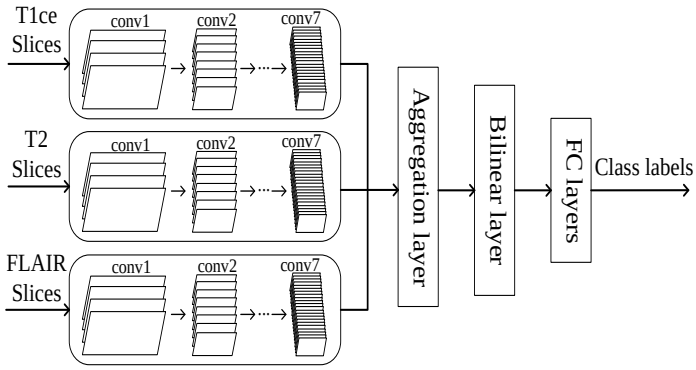

Figure 3.5: Overview of the multi-stream 2D CNN scheme for glioma classification, by zooming in the blue box in Figure 3.4.

Pairwise GANs for Augmenting MR images

To further enlarge the glioma training dataset for improving the classification performance, a pairwise GAN model is employed to generate synthetic MRIs for the training dataset, as shown in Figure 3.6. In pairwise GANs, two streams of GANs (generators G_m, G_n and discriminators D_m, D_n) are interconnected. The loss function consists of adversarial loss L_m, L_n and pixel-level loss $L_1(G_m, G_n)$:

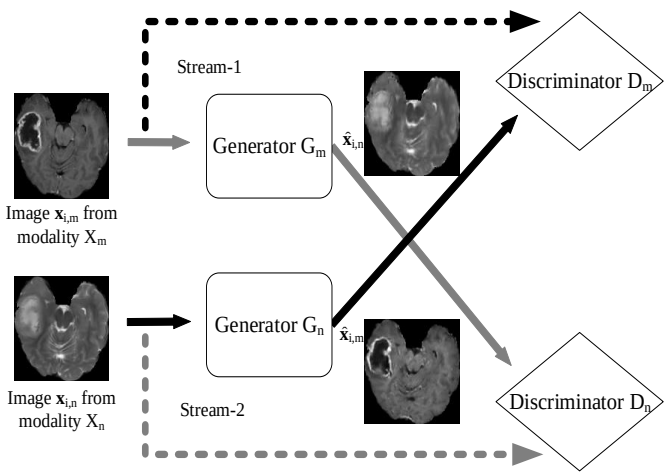

Figure 3.6: Example of the pairwise GAN model, by zooming in the red box in Figure 3.4.

$$L(G_m, G_n, D_m, D_n) = L_n + L_m + \lambda_1 L_1(G_m, G_n), \tag{3.1}$$

$$L_n = \mathbb{E}_{\mathcal{X}_n} ||D_n(\mathbf{x}_{i,n}) - 1||_2^2 + \mathbb{E}_{\mathcal{X}_m} \left(||D_n(G_m(\mathbf{x}_{i,m})||_2^2 \right), \tag{3.2}$$

where D_n distinguishes the fake image $G_m(\mathbf{x}_{i,m})$ from the real one $\mathbf{x}_{i,n}$. L_m is defined in the similar way. $L_1(G_m, G_n)$ describes the loss on the generated images to measure the pixel-level difference between the fake and real images:

$$L_1(G_m, G_n) = \mathbb{E}_{\mathcal{X}_m, \mathcal{X}_n}[\|\mathbf{M}_{i,n} \odot (G_m(\mathbf{x}_{i,m}) - \mathbf{x}_{i,n})\|_1 + \|\mathbf{M}_{i,m} \odot (G_n(\mathbf{x}_{i,n}) - \mathbf{x}_{i,m})\|_1], \tag{3.3}$$

where $\mathbf{M}_{i,m}$ and $\mathbf{M}_{i,n}$ are the tumor masks for the images $\mathbf{x}_{i,m}$ and $\mathbf{x}_{i,n}$ respectively. Real and GAN-augmented MRIs are then utilized to learn the features and the classifier for brain tumor types.

Post-Processing for 3D Scan-Based Diagnosis

As the 2D image-based classifier outputs the glioma type for each image slice, the prediction for each slice can be different even from the same 3D scan, due to variations of slices and image view angles. It is necessary to make a 3D scan-based decision from all slice prediction results. A majority voting-based criterion is proposed for making the final decision of tumor types on each 3D scan. Let $s_i (i = 1, \cdots, N)$ be the i-th slice prediction result of glioma type, N is the total number of extracted tumor image slices of a 3D scan, $s_i = 1$ if the slice belongs to class 1, and $s_i = 0$ if it belongs to class 0. The final prediction result S of glioma type for a 3D scan is determined as follows.

$$S = \begin{cases} 1, & \sum_{i=1}^{N} s_i > N/2, \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

Main results: The multi-stream 2D CNN method is tested on two datasets. 1p19q dataset [129] contains 3D brain volume images with 2 molecular-based subtypes: with-/without 1p/19q codeletion from 159 subjects. MICCAI dataset [130], [131] contains 3D brain volume images including low-grade glioma (LGG) and high-grade glioma (HGG) from 285 subjects. 1p/19q codeletion and LGG are selected as the target class for calculating sensitivity. For all experiments, each dataset is partitioned into 3 subsets: training (60%), validation (20%) and test (20%).

(a) **Molecular-based glioma subtype classification**: The information of 1p19q dataset and the test performance are shown in Table 3.3. The proposed method is shown to be effective and achieved a relatively high accuracy (89.39%) on 1p/19q codeletion classification.

The full scheme with GAN-augmented data and 3D post-processing is tested on a dataset containing 3D brain volume images of 167 subjects from TCGA-GBM [132]

Table 3.3: Information and test results from the proposed scheme on 1p19q dataset.

1p/19q class	# subjects	# 3D scans in T1ce	# 3D scans in T2
With codeletion	102	102	102
Without codeletion	57	57	57

(a) Information of 1p19q dataset.

Dataset	Accuracy	Sensitivity	Specificity
1p19q	89.39%	84.85%	93.94%

(b) Accuracy, sensitivity and specificity on the testing set in one run from 1p19q dataset.

and TCGA-LGG [133]. In the dataset, the MRIs of each patient consist of four modalities (T1, T1ce, T2, FLAIR), the tumor segmentation results, as well as the corresponding molecular-based IDH genotype labels as the tumor subtypes. Figure 3.7 shows the comparison of glioma classification performance on the testing set for the proposed method and the baseline methods in 2 case studies. In Case-A, the training dataset of this method is formed by a subset of original training MRIs from all modalities (S1) and a subset of GAN-generated synthetic MRIs for all modalities (S2), while the training dataset of baseline-1 method is only (S1). In Case-B, the training dataset is formed by the combination of a subset of original training MRIs with missing modalities (S3), a subset of GAN-generated synthetic MRIs that are used to replace the missing modalities (S4), and a subset of GAN-generated synthetic MRIs for all modalities (S5), while the training dataset of baseline-2 method is only (S3), as shown in the illustration in Figure 3.8. Some examples of GAN-augmented images are shown in Figure 3.9.

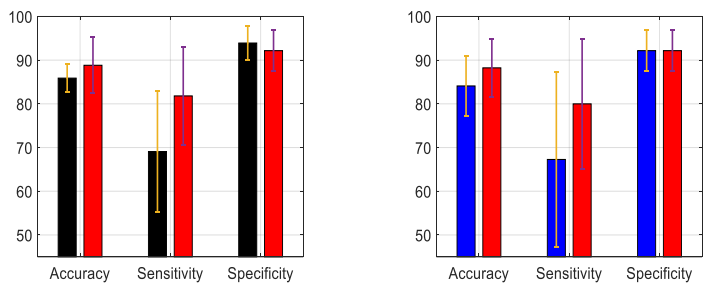

Figure 3.7: Test results on TCGA dataset for IDH mutation classification from two cases, where result is shown in the average of 5 runs including standard deviation ($|\sigma|$). Training datasets in different methods are formed differently, see Figure 3.8. Left: Case-A, proposed (S1+S2) in red, baseline-1 (S1) in black. Right: Case-B, proposed (S3+S4+S5) in red, baseline-2 (S3) in blue.

Observing the results in Case-A, the enlarged dataset with real and augmented

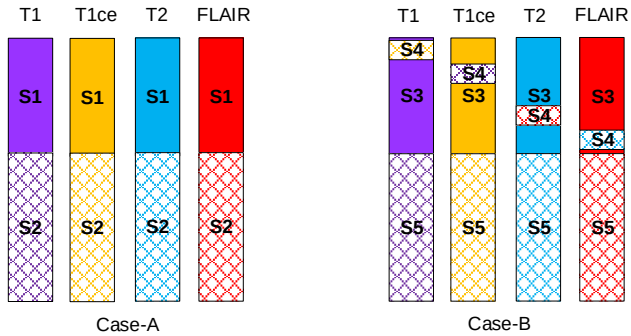

Figure 3.8: Enlarged training datasets formed from the dataset in two cases, where shaded bar areas are GAN-augmented images.

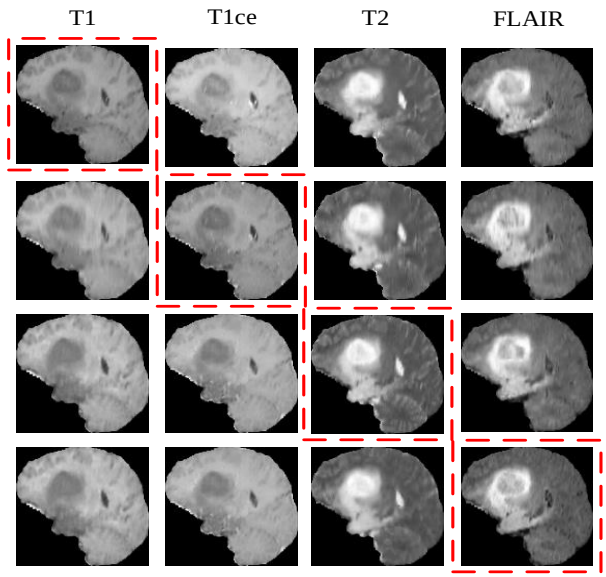

Figure 3.9: Examples of pairwise GAN-augmented synthetic images from TCGA dataset. Four columns correspond to images from modalities T1, T1ce, T2 and FLAIR, while each row contains one real image (in red box) and three synthetic ones generated from this real image.

MRIs across different modalities from fake patients has led to improved classification performance (increased 2.94%) on the testing set for glioma subtypes of IDH mutation/wild-type. In Case-B, the enlarged dataset with real and augmented MRIs across different modalities from fake patients as well as those synthetic ones from missing modalities has led to improved classification performance (increased 4.14%) on the testing set for glioma subtypes of IDH mutation/wild-type. This

indicates that the pairwise GAN can be used for enlarging the training dataset with mixed real and augmented data and recovering MRIs of missing modalities.

(b) Glioma grading: Multi-stream 2D CNN scheme has also been tested on MICCAI dataset. The dataset information and the test performance are shown in Table 3.4. The proposed scheme has generated relatively high accuracy on the testing set (90.87%), and also similar performance on individual LGG and HGG classes.

Table 3.4: Information and test results from the proposed scheme on MICCAI dataset.

Class	# subjects	#3D scans in T1ce	#3D scans in T2	#3D scans in FLAIR
HGG	210	210	210	210
LGG	75	75	75	75

(a) Information of MICCAI dataset.

Dataset	Accuracy	Sensitivity	Specificity
MICCAI	90.87%	90.48%	91.27%

(b) Accuracy, sensitivity and specificity on the testing set in one run from MICCAI dataset.

3.2.2 Deep Semi-Supervised Learning for Glioma Classification

(Summary of Paper 4)

Problem addressed: This method addresses the problem of glioma classification using training datasets containing unlabeled images whose labels are estimated by a deep semi-supervised learning method. A graph-based label propagation method is employed with a 3D-2D consistent constraint to learn the labels of the unlabeled 2D MRIs.

Motivations: Existing glioma classification methods mainly used datasets with all the data labeled. However, in real scenarios, it is quite common that some of the images do not have labels. Motivated by such medical needs, to make the best use of all the images including the unlabeled ones, the labels of the unlabeled data are estimated by semi-supervised learning, so that these images (with the estimated labels) can be used together with the labeled data for training a classifier. For the 2D MRIs from the same 3D scan, they should have the same label. Hence, this method also considers such a 3D-2D consistent constraint to estimate the labels of the unlabeled images.

Basic ideas: 1) Training dataset employs both the labeled dataset as well as the unlabeled dataset with estimated labels obtained from the proposed semi-supervised method. By adding unlabeled data and their corresponding estimated labels to the CNN training, better performance is expected as more training data can mitigate the overfitting of deep learning. 2) Estimating the labels of the unlabeled data by a 3D-2D consistent semi-supervised learning method. The 3D-2D consistent constraint is added to both the way of graph construction and the cost function of label propagation for graph-based semi-supervised learning.

Main contributions:

- We propose to use deep semi-supervised learning for estimating the labels of the unlabeled data, in order to improve the performance of glioma classification by exploring both the labeled and unlabeled data.
- We propose a 3D-2D consistent label propagation method for semi-supervised learning, by adding constraints to both the graph construction and the cost function of label propagation, so that consistent predictions on the 2D slices from the same 3D scan can be made.

Overview: The overview of this method is shown in Figure 3.10. It consists of three modules, semi-supervised learning, data augmentation and deep learning, and 3D volume-based classification. Multi-stream 2D CNN is first trained using only the labeled data in the training dataset. It is then used to extract features from both the labeled and unlabeled data in the training dataset. Graph-based semi-supervised learning is used to learn the estimated labels of the unlabeled data. Its cost function can be described as:

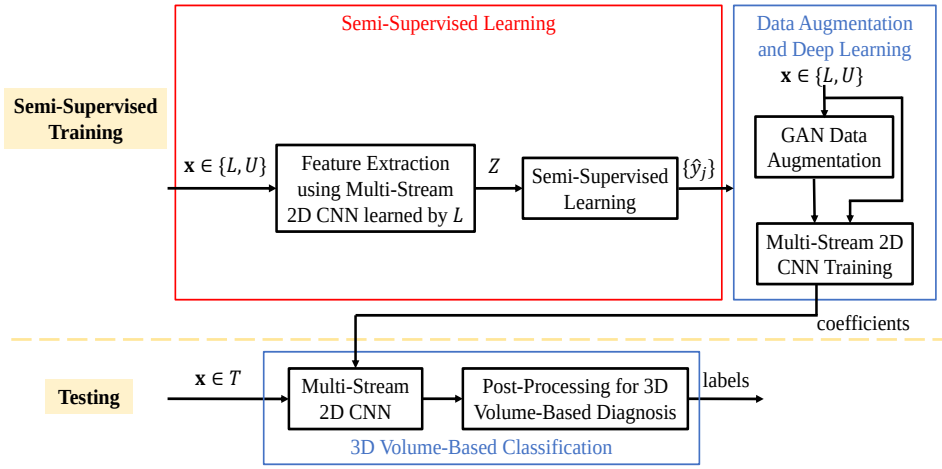

Figure 3.10: Overview of the proposed deep semi-supervised learning scheme for glioma classification, where $\mathcal{L}, \mathcal{U}$ and $\mathcal{T}$ denote the labeled training dataset, unlabeled training dataset and the testing dataset, $\mathcal{Z}$ denotes the feature set, and $\{\hat{y}_j\}$ represents the estimated labels for images in $\mathcal{U}$.

$$E(\mathbf{S}) = \sum_{i,j=1}^{n} W_{i,j} \| \frac{\mathbf{s}_i}{\sqrt{D_{ii}}} - \frac{\mathbf{s}_j}{\sqrt{D_{jj}}} \|^2 + \mu \|\mathbf{S} - \mathbf{Y}\|_F^2 + \lambda \|\mathbf{S} - \mathbf{B}\mathbf{S}\|_F^2, \tag{3.5}$$

where $\mathbf{S}$ is the estimated labels for all the images after graph-based semi-supervised learning, $\mathbf{s}_i$ is the i-th row of $\mathbf{S}$ denoting the label for the i-th image. $W_{i,j}$ is an element of the affinity matrix representing the similarity of images in the feature space. $\mathbf{Y}$ denotes the true labels. $\mu > 0$ and $\lambda > 0$ are the balancing weights. The first two terms are adopted from [134] as the framework for graph-based semi-supervised learning, to let images that are close to each other in the feature space have similar labels, and force the labeled images to remain the initial labels. The third term is the added part using the variance penalty to let the 2D slice images from the same 3D scan share the same label, and $\mathbf{B}$ is defined to compute the average prediction of the 2D slices from the same 3D scan.

After that, training data from both labeled and unlabeled sets form the input to GANs for the augmentation of synthetic MRIs. The labeled training dataset, unlabeled training dataset with estimated labels, as well as the GAN-augmented data are then fed into multi-stream 2D CNN for learning the characteristics of gliomas. In the testing phase, MRIs from the testing dataset are tested using the trained CNN, followed by post-processing to output the glioma type for each 3D brain scan.

Main results: This method is tested on two datasets. TCGA dataset contains 3D brain volume images of 167 subjects from TCGA-GBM [132] and TCGA-LGG [133] with

IDH mutation labels. MICCAI dataset contains 3D brain volume images of LGG and HGG from 285 subjects, downloaded from MICCAI BraTS 2017 competition [130], [131]. The proposed method is compared with two baseline methods. Baseline-1 method uses the training dataset only consisting of the original labeled 2D image slices $\mathcal{L}$. The proposed scheme uses the training dataset consisting of the original labeled 2D image slices $\mathcal{L}$ and unlabeled ones $\mathcal{U}$ whose labels are estimated from graph-based semi-supervised learning. Baseline-2 method uses the training dataset consisting of the original labeled 2D image slices $\mathcal{L}$ and $\mathcal{U}$ in which the ground-truth labels are used. IDH mutation/LGG is selected as the target class for calculating sensitivity. For TCGA dataset the aim is to classify/predict tumor subtypes in the molecular levels, while for MICCAI dataset, the aim is to classify gliomas into low and high grades.

(a) **Molecular-based glioma subtype classification**: Information about TCGA dataset and test results are shown in Figure 3.11. Observing Figure 3.11, the proposed method has achieved high classification performance (86.53%) on the testing set. In addition, the proposed method has achieved increased performance than baseline-1 method and slightly decreased performance than baseline-2 method. It indicates that the proposed deep semi-supervised learning is effective in estimating the labels of the unlabeled data.

Tumor type	#Patients	#3D scans (T1/T1ce/ T2/FLAIR)	#3D scans for training (origial/GAN augmented)	#3D scans for validation (original)	#3D scans for testing (original)
IDH mutation	55	55	33/99	6	16
IDH wild-type	112	112	66/198	13	33

Dataset	Accuracy$\pm\|\sigma\|$ (%)	Sensitivity$\pm\|\sigma\|$ (%)	Specificity$\pm\|\sigma\|$ (%)
TCGA	86.53±4.24	73.75±8.15	92.73±3.45

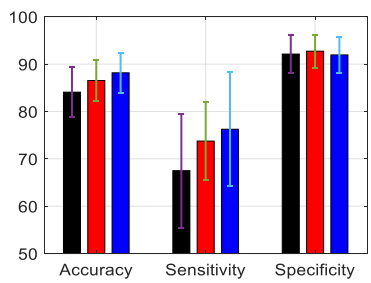

Figure 3.11: Information and test results on TCGA dataset. Top: dataset information, where 9 slices per 3D scan are used. Middle: average accuracy, sensitivity and specificity of 5 runs on the testing sets. Bottom: performance comparison with two baseline methods, Red: the proposed, Black: baseline-1 using $\mathcal{L}$, Blue: baseline-2 using $\mathcal{L}$ and $\mathcal{U}$ with ground truth labels.

(b) Glioma grading: Information about MICCAI dataset and test results are shown in Figure 3.12. Observing Figure 3.12, the proposed method has achieved high classification performance (90.70%) on the testing set. In addition, the proposed method has achieved increased performance than baseline-1 method and similar performance to baseline-2 method, indicating that the proposed deep semi-supervised learning is effective in estimating the labels of the unlabeled data.

Tumor type	#Patients	#3D scans (T1/T1ce/ T2/FLAIR)	#3D scans for training (origial/GAN augmented)	#3D scans for validation (original)	#3D scans for testing (original)
HGG	210	210	126/126	21	63
LGG	75	75	45/45	7	23

Dataset	Accuracy$\pm\|\sigma\|$ (%)	Sensitivity$\pm\|\sigma\|$ (%)	Specificity$\pm\|\sigma\|$ (%)
MICCAI	90.70±1.42	84.35±6.59	93.01±1.42

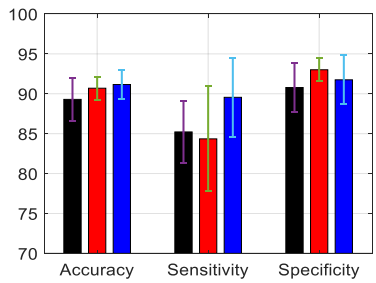

Figure 3.12: Information and test results on MICCAI dataset. Top: dataset information, where 9 slices per 3D scan are used. Middle: Average accuracy, sensitivity and specificity of 5 runs on the testing sets. Bottom: performance comparison with two baseline methods, Red: the proposed, Black: baseline-1 using $\mathcal{L}$, Blue: baseline-2 using $\mathcal{L}$ and $\mathcal{U}$ with ground truth labels.

3.3 Deep Learning for E-Health Care and Assisted Living

3.3.1 Video-Based Human Fall Detection by Deep Learning

3.3.1.1 Segment-Level CNNs + Sparse Code Sequence for Fall Detection

(Summary of Paper 5)

Problem addressed: This method addresses employing segment-level CNN features from both static and optical flow images, to learn sequences of sparse codes for human fall detection.

Motivations: Existing methods for human fall detection [55], [57], [135] mainly focused on hand-crated features. The good performance of deep learning enables automatic feature learning for human fall detection. However, conventional two-stream action recognition framework [58] does not generalize well to fall detection, because it cannot capture the discriminative features for human falls among all the other activities from the similar indoor background.

Basic ideas: The basic idea of this method is to employ high-level CNN features from activity segments for human fall detection. To obtain more discriminative feature representation, a sparse representation framework is used by iteratively learning the codebook and the sparse codes. A residual-based pooling strategy is proposed to obtain more effective code representation by considering residuals reconstructed from the learned codebook. Code for segment t is formed by a linear combination of weighted sparse codes as follows:

$$\mathbf{C}_t = \frac{1}{N}\sum_{i=1}^{N} w_i^t \mathbf{m}_i^t, \tag{3.6}$$

where the weight $w_i^t = 1/\|\mathbf{F}_i^t - \mathbf{H}\mathbf{m}_i^t\|^2$. The larger the reconstruction residual, the smaller the weight w_i^t. $\mathbf{H}$ denotes the codebook and $\mathbf{m}_i^t$ denotes the sparse code corresponding to feature $\mathbf{F}_i^t$ in t-th segment. Given that CNNs only learn appearance features from static images and temporal optical flow features from adjacent frames, we consider long-range temporal representations by concatenating segment-level codes to capture sequential information in an activity.

Main contributions:

- A sparse representation framework with a new residual-based pooling strategy is employed to generate more discriminative feature representations.
- Segment-level code vectors are proposed as long-range dynamic features to capture the sequential information of videos for fall detection.

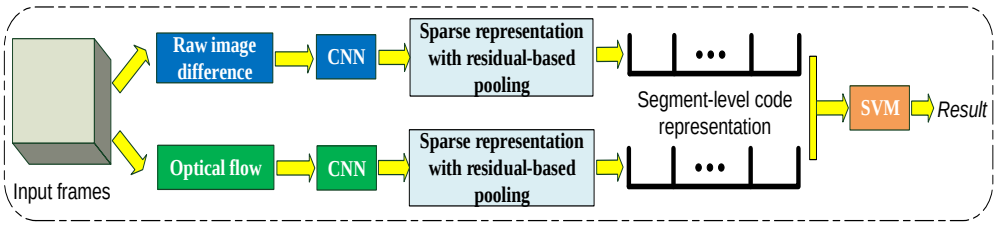

Figure 3.13: Overview of this method for human fall detection.

Overview: The overview of this method is illustrated in Figure 3.13. Two-stream inputs are fed into CNNs to generate high-level appearance and temporal features. For the spatial stream, image difference between two consecutive frames is computed to characterize the appearance change, and irrelevant background is excluded as well. For the temporal stream, optical flow is used to capture the motion information. To normalize each video activity it is divided into M segments, and each being represented by one key frame. CNN features are obtained from the last convolutional layer of fine-tuned VGG-16 network [91]. Different from the conventional two-stream action recognition framework, we generate more discriminative features by sparse representation along with proposed residual-based pooling. In this way, original features are converted to more discriminative sparse codes, and residual-based pooling is used to aggregate the obtained sparse codes within each segment. Segment-level code vectors are then concatenated for video representation, followed by an SVM for final classification.

Main results: This method is tested on 2 datasets. Dataset-A is built on "multiple cameras fall dataset" [136] containing 400 video clips, including 184 video clips of falls and 216 video clips of other activities. Dataset-B is collected from "UR fall detection dataset" [137] that consists of 60 video clips containing falls and 40 video clips containing other activities. Figure 3.14 shows some key frames of videos from Dataset-A and Dataset-B. Test results of this method are evaluated according to the detection rate (true positive rate, TPR) and the false alarm rate (false positive rate, FPR) on the testing set as shown in Table 3.5.

Discussion: It is observed that this method achieves comparable performance to these existing methods. It is worth noting that the multiview methods in [55], [138], [139] need extra information on multiple camera calibration and multiple modalities, [135] requires skeleton information, while the proposed method treats all view video clips equally without using additional information.

Dataset-A

Dataset-B

Figure 3.14: Key frames from the two open datasets on fall detection. For each dataset, upper row: falls; lower row: other activities.

Table 3.5: Comparison of the test results from this method in one run and existing methods on two datasets. TPR: true positive rate, FPR: false positive rate.

Dataset-A

Method	Camera type	TPR(%)	FPR(%)
Auvinet [138]	Multiviews	80.60	0.00
Rougier [55]	Multiviews	95.40	4.20
Hung [139]	Multiviews	95.80	0.00
Yun [135]	Arbitrary Views	91.30	8.33
Proposed	Arbitrary Views	89.40	6.77

Dataset-B

Method	Sensor type	TPR(%)	FPR(%)
Kepski [137]	Depth+Accelerometer	100.00	3.33
Bourke [140]	Accelerometer	100.00	10.00
Yun [135]	Arbitrary Views	96.77	10.26
Proposed	Arbitrary Views	100.00	5.50

3.3.1.2 Co-Saliency-Enhanced RCN for Fall Detection

(Summary of Papers 6, 7)

Problem addressed: This method addresses characterizing the features of human activities by Recurrent Convolutional Networks (RCNs), with activity areas enhanced by co-saliency detection.

Motivations: Existing deep learning methods for human fall detection [62], [141] mainly relied on features from CNNs with temporal-information-embedded pooling strategies to characterize human activities, where complex temporal information cannot be well modeled. In addition, a video object often occupies a small image area, it is desirable to highlight object areas and let the deep learning focus on these specific areas, for more robust classification of human activities.

Basic ideas: The long-term spatio-temporal relationship of human activities is exploited by applying LSTM on a set of CNNs in the segment-level. Applying segment-level CNNs is based on the observation that, by partitioning each video clip into the same number of segments, videos of different lengths can be normalized and images within each segment can be considered as approximately stationary. Hence, a CNN can be used effectively in such segment levels. For a longer time scale, LSTM is then applied to take into account the temporal dependency of segment-level images. In addition, a co-saliency detection method that is designed to detect salient regions from several image frames, is applied to enhance dynamic human activity regions and suppress unwanted background regions in videos. Integrating co-saliency detection with RCN could lead to obtaining more discriminative deep features of human falls, hence the overall performance will be improved.

The basic idea of co-saliency detection method is to treat co-saliency detection as a two-stage saliency propagation problem. The first inter-saliency propagation stage utilizes the similarity between a pair of images to discover the common properties of the images with the help of a single-image saliency map. The propagated saliency from n-th image to m-th image is defined as:

$$S^{n\to m}(i) = g_m(i) * \frac{\sum_{j=1}^{K_n} \exp(-\alpha\|\mathbf{c}_i^m - \mathbf{c}_j^n\|_2) S_o^n(j)}{\sum_{j=1}^{K_n} \exp(-\alpha\|\mathbf{c}_i^m - \mathbf{c}_j^n\|_2)}, \tag{3.7}$$

where $\exp(-\alpha\|\mathbf{c}_i^m - \mathbf{c}_j^n\|_2)$ denotes the color similarity between superpixel i of m-th image and superpixel j of n-th image, whose original saliency value is denoted as $S_o^n(j)$. $g_m(i)$ is a center bias term. The second intra-saliency propagation stage refines the results by a graph-based method combining two cues for better foreground rendering and background suppression. A new fusion strategy combining all these guided saliency maps is then used

to obtain the co-saliency detection maps.

Main contributions:

- A novel integrated fall detection scheme by combining co-saliency-enhancement and a RCN architecture is proposed for fall detection.
- A co-saliency method is employed that is suitable for enhancing dynamic human areas and suppressing static background areas in videos.
- A new co-saliency detection method is proposed, by propagating saliency values between images as well as within images to simultaneously highlight co-salient objects and suppress background information.

Overview: Figure 3.15 illustrates the overview of this method, where co-saliency activity region enhancement is applied to the original video frames, followed by a RCN architecture where a set of CNNs is applied to image frames in the segment-level. Each video clip is first divided into N segments, where one key frame is extracted as the representative for each segment. Hence, video clips of different lengths can be normalized to the same number of segments, independent of the speed of each activity. After that, co-saliency maps that highlight the foreground dynamic human objects are generated based on the segment-level images. The proposed co-saliency detection method consists of four main steps: pre-processing, inter-saliency propagation, intra-saliency propagation and co-saliency integration, as shown in Figure 3.16. Pre-processing is used for superpixel segmentation and initial saliency map generation by a single-image saliency model. After that, saliency is propagated in image pairs to discover the common properties of the images by inter-image saliency propagation. Intra-image saliency propagation is then used for refining previous inter-image propagated results, followed by co-saliency integration to obtain the results. After multiplying the original input images with their corresponding co-saliency maps, human activity areas are greatly enhanced with irrelevant background regions suppressed. Finally, in the proposed RCN there are N CNNs (each is of 5 layers obtained from numerous empirical tests and hyperparameter tunings) followed by a single-layer LSTM. The class labels (fall/non-fall) are obtained from the output of the FC layer after the LSTM layer.

Main results: This method is tested on an open video dataset "ACT4ˆ2 Dataset" [142] containing 768 videos. In the dataset, each activity is performed by 24 subjects, captured in 4 view angles and repeated 2 times. Hence, there are 192 videos for each video activity. Fall detection is formulated as a binary classification problem: We treat the falls (including collapse and stumble) as the positive class and the remaining activities (including pickup and sitdown) as the negative class. Figure 3.17 shows some key video frames from these 2 classes in ACT4ˆ2 Dataset. Experiments are conducted for 5 runs, where partitions of training, validation and test subsets in the dataset are done

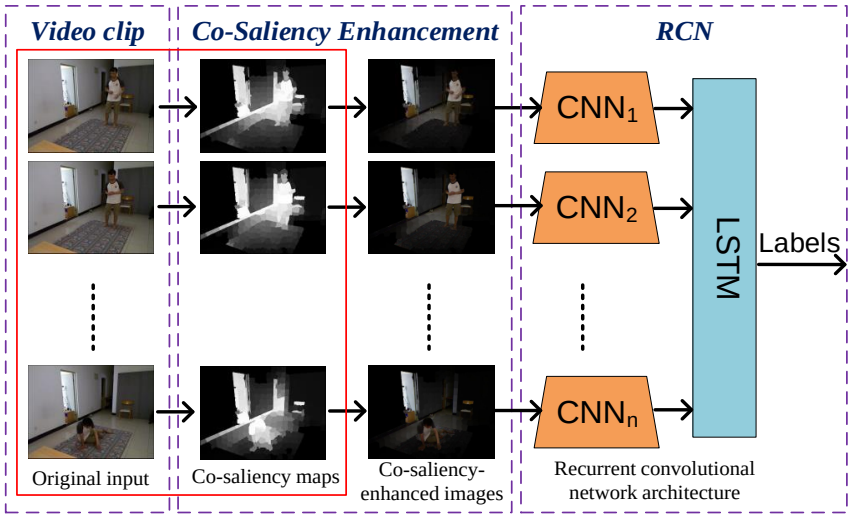

Figure 3.15: Overview of this method for human fall detection from videos.

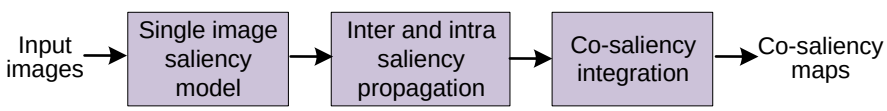

Figure 3.16: Overview of this method for co-saliency detection, by zooming in the red box in Figure 3.15.

randomly in each of the 5 runs. Table 3.6 shows the accuracy of the proposed method on the testing set as well as the sensitivity and specificity, including the average performance of 5 runs and standard deviation $|\sigma|$. Performance of the proposed co-saliency detection method on iCoseg dataset [143] is shown in Figure 3.18.

Table 3.6: Test performance on ACT4ˆ2 Dataset. Result is shown in average performance of 5 runs with standard deviation ($|\sigma|$), and for each run the training/validation/test subsets are randomly re-partitioned.

Accuracy$\pm|\sigma|$ (%)	Sensitivity$\pm|\sigma|$ (%)	Specificity$\pm|\sigma|$ (%)
98.12±0.19	96.93±0.62	99.30±0.70

Discussion: Observing Table 3.6, the proposed method is shown to be effective on the testing set, with a relatively high average classification accuracy 98.12% ($|\sigma|$=0.19%), sensitivity 96.93% ($|\sigma|$=0.62%) and specificity 99.30% ($|\sigma|$=0.70%). Furthermore, the

Figure 3.17: Examples of the key frames from the open ACT4ˆ2 Dataset. Top row: human falls (collapse, stumble). Bottom row: other activities (pickup, sitdown).

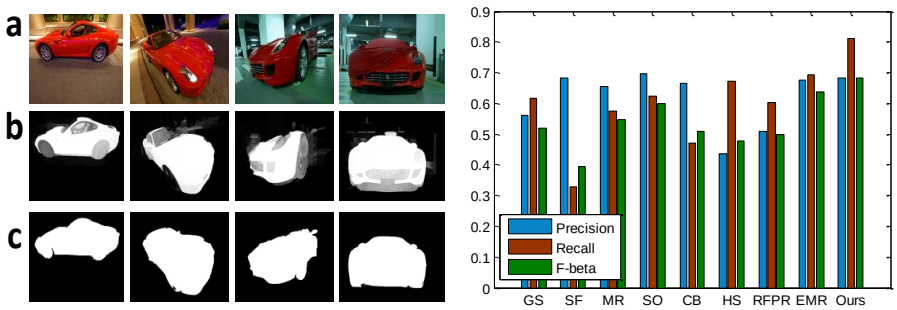

Figure 3.18: Performance of the proposed co-saliency detection method on iCoseg dataset. Left: performance on one set of images, (a) original images; (b) the proposed; and (c) ground truth. Right: comparison with eight saliency detection models on weighed precision, recall and F-beta.

proposed method seems to have relatively balanced performance on two classes since the sensitivity and specificity are relatively close to each other. To conclude, the co-saliency-enhanced RCN architecture is effective for learning the time-dependent features of human activities.

Discussion on Two Human Fall Detection Methods (in Sections 3.3.1.1 and 3.3.1.2)

Two methods have been presented for human fall detection. Method-1 (Paper 5) uses segment-level CNN with sparse dictionary learning to characterize human activity-related features, while Method-2 (Paper 6) uses co-saliency-enhanced RCN to characterize features and conduct classification in an end-to-end fashion. Their differences can be summarized as follows:

- *Characterizing temporal information*: Method-1 uses optical flow to capture the motion information in each segment, and long-term time dependency is modeled by a concatenated sequence of sparse codes obtained from each segment. Method-2 directly uses LSTM to capture the temporal information between different segments.
- *Computational and memory cost*: Method-1 requires more computation, as two streams of CNN features (both appearance and motion) are first calculated. Sparse dictionary learning is then applied to convert features into more discriminative sparse codes, followed by classification using SVM. The optimizations of two-stream CNNs, sparse dictionary learning and SVM classification are done separately. In contrast, Method-2 conducts feature extraction and classification through an end-to-end RCN architecture, thus it is more efficient than Method-1. However, the memory cost of Method-2 is very high especially when the number of segments in a video is large or the CNN structure is deep.

3.3.2 A Spiking Neural Network Model for Obstacle Avoidance in Simulated Prosthetic Vision

(Summary of Paper 8)

Problem addressed: This method addresses the problem of obstacle avoidance in simulated prosthetic vision by using a spiking neural network (SNN) model. The aim is to assist blind people wearing visual prostheses to avoid obstacles during walking, by classifying image sequences in prosthetic vision to classes with/without obstacles.

Basic ideas: Previous studies on visual prosthesis [79], [81], [83] mainly focused on improving the quality of phosphene images obtained from visual prostheses, without enhancing the high-level functionality of prostheses such as obstacle avoidance, which is an important task to the blind. The basic idea of this method is to characterize image sequences with/without obstacles by SNNs. To obtain discriminative features from low-resolution prosthetic images, a new spatio-temporal feature extraction method is presented to capture the pattern changes when approaching an obstacle, followed by a data encoding module for spike representation. To characterize the temporal information of the spikes for classification, we employ a SNN model that is inspired by biologically-realistic model of neurons for computation.

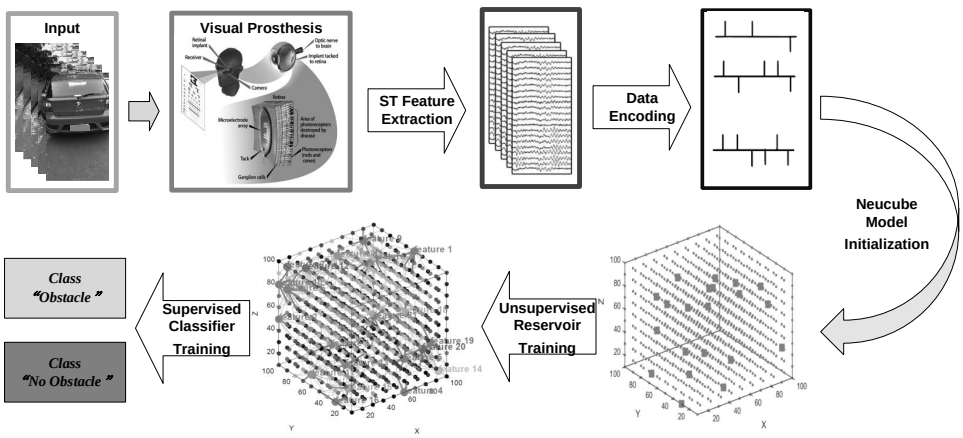

Figure 3.19: Overview of SNN-based obstacle avoidance method.

Main contributions:

- A SNN architecture, NeuCube is first successfully applied to the obstacle avoidance task in simulated prosthetic vision with good performance on two datasets.
- A new spatio-temporal feature extraction method is proposed to generate obstacle-

related features in videos.

- A new data encoding method is presented, which is robust to noise and results in good spike representations as the input to NeuCube.

Overview: The overview of this method is shown in Figure 3.19. First, input videos are captured by a visual prosthesis, then the down-sampled signal is obtained from the output of the prosthesis. After that, a feature extraction algorithm is used to obtain spatio-temporal (ST) features. Finally, such features are fed into a SNN model (NeuCube), which consists of a data encoding module, a NeuCube initialization module, an unsupervised reservoir training module and a supervised classifier training module, to output classification results of obstacle analysis.

Main results: This method is tested on 2 datasets. One contains 20 videos with static obstacles and 20 videos without obstacles, and the other contains 20 videos with moving obstacles and 20 videos without obstacles. Videos are collected by a camera with first-person view during walking, simulating how prosthesis wearers walk. The performance of this method is shown in Table 3.7. It is observed that this method achieves good performance on both datasets.

Table 3.7: Performance of this method and other computational intelligence methods on two datasets.

Static Obstacle Dataset

Classification Methods	Overall Accuracy(%)	TPR(%)	TNR(%)
Adaboost	77.50	75.00	80.00
MLP	32.50	50.00	15.00
SVM(with ST features)	61.25	52.50	70.00
SVM(with CNN descriptor)	71.00	79.00	63.00
Proposed	**90.50**	**88.00**	**93.00**

Moving Obstacle Dataset

Classification Methods	Overall Accuracy(%)	TPR(%)	TNR(%)
Adaboost	67.75	65.00	70.50
MLP	58.00	60.50	55.50
SVM(with ST features)	55.50	42.00	69.00
SVM(with CNN descriptor)	60.75	44.00	**77.50**
Proposed	**81.00**	**86.00**	76.00

CHAPTER 4

Conclusion

In this thesis, several machine learning methods have been tested for healthcare-related applications including Alzheimer's disease detection, brain tumor (glioma) classification, human fall detection and obstacle avoidance in prosthetic vision, from the perspective of computer vision and image processing. Considering the contributions, we have investigated multi-stream and multi-scale CNNs for feature characterization in brain MRIs, graph-based semi-supervised learning for learning the labels of the unlabeled MRIs, pairwise GANs for synthesizing brain MRIs of fake patients and missing modalities, RCNs and SNNs for characterizing spatio-temporal features from videos. Experiments on several datasets have shown that the proposed methods are effective. In addition, the proposed methods have achieved state-of-the-art performance according to the comparisons with some existing methods, although further improvement is required. Considering the real-world applications, these methods are promising and may benefit the areas such as assisted living, computer-aided diagnosis, and E-healthcare.

4.1 Future Work

Despite the research progress we have made, many issues remain in this research field.

- For Alzheimer's disease detection, the proposed method could be extended to predict a new class, mild cognitive impairment (MCI, a stage between AD and NC). It is also an option to use images from other modalities such as PET and CT, or other side information such as symptoms and ages to improve the diagnosis of AD.
- For glioma classification, datasets from different institutions could be merged for

studies. Patient side information (e.g., ages, survival years) could also be incorporated to enhance the performance. Binary subtype classification could be extended to three classes by combining IDH genotype and 1p/19q codeletion status. One may also use bounding boxes to reduce the cost of accurate segmentation.

- For human fall detection, different kinds of daily activities could be added for distinguishing more classes in assisted living. Methods for human activity classification could also be extended to a broader research field: human action recognition.

Bibliography

[1] M. Prince, *World Alzheimer Report 2015: The Global Impact of Dementia: An Analysis of Prevalence, Incidence, Cost and Trends*. Alzheimer's Disease International, 2015.

[2] C. Carrion, F. Folkvord, D. Anastasiadou, and M. Aymerich, "Cognitive therapy for dementia patients: A systematic review.", *Dementia and Geriatric Cognitive Disorders*, vol. 46, no. 1-2, pp. 1–26, 2018.

[3] R. Brookmeyer, M. Corrada, F. Curriero, and C. Kawas, "Survival following a diagnosis of Alzheimer disease", *Archives of Neurology*, vol. 59, no. 11, pp. 1764–1767, 2002.

[4] C. Raji, O. Lopez, L. Kuller, O. Carmichael, and J. Becker, "Age, Alzheimer disease, and brain structure", *Neurology*, vol. 73, no. 22, pp. 1899–1905, 2009.

[5] J. Cummings, T. Morstorf, and K. Zhong, "Alzheimer's disease drug-development pipeline: Few candidates, frequent failures", *Alzheimer's Research & Therapy*, vol. 6, no. 4, p. 37, 2014.

[6] G. Small, "Early diagnosis of Alzheimer's disease: Update on combining genetic and brain-imaging measures", *Dialogues in Clinical Neuroscience*, vol. 2, no. 3, p. 241, 2000.

[7] Garrondo, *Alzheimer's Brain*, `https://commons.wikimedia.org/w/index.php?curid=4387759`, 2017.

[8] MayoClinic, *Alzheimer's Disease*, `https://www.mayoclinic.org/diseases-conditions/alzheimers-disease/diagnosis-treatment/drc-20350453`, 2018.

[9] M. Alipoor, *Computational Diffusion MRI: Optimal Gradient Encoding Schemes*. Doctoral thesis, Chalmers University of Technology, 2016.

[10] W. Yang, R. Lui, J. Gao, T. Chan, S. Yau, R. Sperling, and X. Huang, "Independent component analysis-based classification of Alzheimer's disease MRI data", *Journal of Alzheimer's Disease*, vol. 24, no. 4, pp. 775–783, 2011.

[11] T. Tong, R. Wolz, Q. Gao, R. Guerrero, J. Hajnal, and D. Rueckert, "Multiple instance learning for classification of dementia in brain MRI", *Medical Image Analysis*, vol. 18, no. 5, pp. 808–818, 2014.

[12] E. Arvesen, "Automatic classification of Alzheimer's disease from structural MRI", Master's thesis, 2015.

[13] M. Liu, D. Zhang, E. Adeli, and D. Shen, "Inherent structure-based multiview learning with multitemplate feature representation for Alzheimer's disease diagnosis", *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 7, pp. 1473–1482, 2016.

[14] T. Brosch, R. Tam, A. D. N. Initiative, *et al.*, "Manifold learning of brain MRIs by deep learning", in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2013, pp. 633–640.

[15] S. Sarraf, G. Tofighi, *et al.*, "DeepAD: Alzheimer's disease classification via deep convolutional neural networks using MRI and fMRI", *bioRxiv*, p. 070 441, 2016.

[16] K. Bäckström, M. Nazari, I. Gu, and A. Jakola, "An efficient 3D deep convolutional network for Alzheimer's disease diagnosis using MR images", in *15th International Symposium on Biomedical Imaging*, IEEE, 2018, pp. 149–153.

[17] H. Suk and D. Shen, "Deep learning-based feature representation for AD/MCI classification", in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2013, pp. 583–590.

[18] A. Gupta, M. Ayhan, and A. Maida, "Natural image bases to represent neuroimaging data", in *International Conference on Machine Learning*, 2013, pp. 987–994.

[19] E. Hosseini-Asl, R. Keynton, and A. El-Baz, "Alzheimer's disease diagnostics by adaptation of 3D convolutional network", in *International Conference on Image Processing*, IEEE, 2016, pp. 126–130.

[20] S. Cha, "Update on brain tumor imaging: From anatomy to physiology", *American Journal of Neuroradiology*, vol. 27, no. 3, pp. 475–487, 2006.

[21] M. Goodenberger and R. Jenkins, "Genetics of adult glioma", *Cancer Genetics*, vol. 205, no. 12, pp. 613–621, 2012.

[22] A. Gupta and T. Dwivedi, "A simplified overview of World Health Organization classification update of central nervous system tumors 2016", *Journal of Neurosciences in Rural Practice*, vol. 8, no. 04, pp. 629–641, 2017.

[23] S. Fellah, D. Caudal, *et al.*, "Multimodal MR imaging (diffusion, perfusion, and spectroscopy): Is it possible to distinguish oligodendroglial tumor grade and 1p/19q codeletion in the pretherapeutic diagnosis?", *American Journal of Neuroradiology*, vol. 34, no. 7, pp. 1326–1333, 2013.

[24] N. Jansen, C. Schwartz, *et al.*, "Prediction of oligodendroglial histology and LOH 1p/19q using dynamic [^{18}F] FET-PET imaging in intracranial WHO grade II and III gliomas", *Neuro-Oncology*, vol. 14, no. 12, pp. 1473–1480, 2012.

[25] M. Wijnenga, P. French, *et al.*, "The impact of surgery in molecularly defined low-grade glioma: An integrated clinical, radiological, and molecular analysis", *Neuro-Oncology*, vol. 20, no. 1, pp. 103–112, 2017.

[26] A. Alattar, M. Brandel, *et al.*, "Oligodendroglioma resection: A surveillance, epidemiology, and end results (SEER) analysis", *Journal of Neurosurgery*, vol. 128, no. 4, pp. 1076–1083, 2018.

[27] D. Parsons, S. Jones, X. Zhang, *et al.*, "An integrated genomic analysis of human glioblastoma multiforme", *Science*, vol. 321, no. 5897, pp. 1807–1812, 2008.

[28] J. Eckel-Passow, D. Lachance, *et al.*, "Glioma groups based on 1p/19q, IDH, and TERT promoter mutations in tumors", *New England Journal of Medicine*, vol. 372, no. 26, pp. 2499–2508, 2015.

[29] C. Hartmann, B. Hentschel, W. Wick, *et al.*, "Patients with IDH1 wild type anaplastic astrocytomas exhibit worse prognosis than IDH1-mutated glioblastomas, and IDH1 mutation status accounts for the unfavorable prognostic effect of higher age: implications for classification of gliomas", *Acta Neuropathologica*, vol. 120, no. 6, pp. 707–718, 2010.

[30] C. Houillier, X. Wang, G. Kaloshi, K. Mokhtari, *et al.*, "IDH1 or IDH2 mutations predict longer survival and response to temozolomide in low-grade gliomas", *Neurology*, vol. 75, no. 17, pp. 1560–1566, 2010.

[31] H. Yan, D. Parsons, G. Jin, *et al.*, "IDH1 and IDH2 mutations in gliomas", *New England Journal of Medicine*, vol. 360, no. 8, pp. 765–773, 2009.

[32] Y. Kang, S. Choi, Y. Kim, *et al.*, "Gliomas: Histogram analysis of apparent diffusion coefficient maps with standard-or high-b-value diffusion-weighted MR imaging-correlation with tumor grade", *Radiology*, vol. 261, no. 3, pp. 882–890, 2011.

[33] J. Carrillo, A. Lai, *et al.*, "Relationship between tumor enhancement, edema, IDH1 mutational status, MGMT promoter methylation, and survival in glioblastoma", *American Journal of Neuroradiology*, vol. 33, no. 7, pp. 1349–1355, 2012.

[34] S. Qi, L. Yu, H. Li, Y. Ou, *et al.*, "Isocitrate dehydrogenase mutation is associated with tumor location and magnetic resonance imaging characteristics in astrocytic neoplasms", *Oncology Letters*, vol. 7, no. 6, pp. 1895–1902, 2014.

[35] J. Yu, Z. Shi, Y. Lian, Z. Li, *et al.*, "Noninvasive IDH1 mutation estimation based on a quantitative radiomics approach for grade II glioma", *European Radiology*, vol. 27, no. 8, pp. 3509–3522, 2017.

[36] X. Zhang, Q. Tian, L. Wang, Y. Liu, *et al.*, "Radiomics strategy for molecular subtype stratification of lower-grade glioma: Detecting IDH and TP53 mutations based on multimodal MRI", *Journal of Magnetic Resonance Imaging*, vol. 48, no. 4, pp. 916–926, 2018.

[37] B. Shofty, M. Artzi, D. Bashat, *et al.*, "MRI radiomics analysis of molecular alterations in low-grade gliomas", *International journal of Computer Assisted Radiology and Surgery*, vol. 13, no. 4, pp. 563–571, 2018.

[38] Z. Li, Y. Wang, J. Yu, *et al.*, "Deep learning based radiomics (DLR) and its usage in noninvasive IDH1 prediction for low grade glioma", *Scientific Reports*, vol. 7, no. 1, pp. 1–11, 2017.

[39] K. Chang, H. Bai, H. Zhou, C. Su, *et al.*, "Residual convolutional neural network for the determination of IDH status in low-and high-grade gliomas from MR imaging", *Clinical Cancer Research*, vol. 24, no. 5, pp. 1073–1081, 2018.

[40] S. Liang, R. Zhang, D. Liang, T. Song, T. Ai, C. Xia, L. Xia, and Y. Wang, "Multimodal 3D denseNet for IDH genotype prediction in gliomas", *Genes*, vol. 9, no. 8, p. 382, 2018.

[41] WHO, *Ageing and Health*, `https://www.who.int/news-room/fact-sheets/detail/ageing-and-health`, 2018.

[42] Y. Yun and I. Gu, "Riemannian manifold-valued part-based features and geodesic-induced kernel machine for activity classification dedicated to assisted living", *Computer Vision and Image Understanding*, vol. 161, pp. 65–76, 2017.

[43] A. Shahzad and K. Kim, "Falldroid: An automated smart-phone-based fall detection system using multiple kernel learning", *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 35–44, 2018.

[44] G. Shi, C. Chan, W. Li, K. Leung, Y. Zou, and Y. Jin, "Mobile human airbag system for fall protection using MEMS sensors and embedded SVM classifier", *IEEE Sensors Journal*, vol. 9, no. 5, pp. 495–503, 2009.

[45] J. Porteus and S. Brownsell, "Using telecare: Exploring technologies for independent living for older people", *Anchor Trust, Kidlington, UK*, 2000.

[46] X. Yu, "Approaches and principles of fall detection for elderly and patient", in *International Conference on E-Health Networking, Applications and Services*, IEEE, 2008, pp. 42–47.

[47] Y. Depeursinge, J. Krauss, and M. El-Khoury, "Device for monitoring the activity of a person and/or detecting a fall, in particular with a view to providing help in the event of an incident hazardous to life or limb", 2001, US Patent 6,201,476.

[48] A. Bourke and G. Lyons, "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor", *Medical Engineering & Physics*, vol. 30, no. 1, pp. 84–90, 2008.

[49] G. Williams, K. Doughty, K. Cameron, and D. Bradley, "A smart fall and activity monitor for telecare applications", in *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Vol. 20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No. 98CH36286)*, vol. 3, 1998, pp. 1151–1154.

[50] A. Diaz, M. Prado, L. Roa, J. Reina-Tosina, and G. Sanchez, "Preliminary evaluation of a full-time falling monitor for the elderly", in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 1, 2004, pp. 2180–2183.

[51] Y. Zigel, D. Litvak, and I. Gannot, "A method for automatic fall detection of elderly people using floor vibrations and sound–proof of concept on human mimicking doll falls", *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 12, pp. 2858–2867, 2009.

[52] H. Qian, Y. Mao, W. Xiang, and Z. Wang, "Home environment fall detection system based on a cascaded multi-SVM classifier", in *International Conference on Control, Automation, Robotics and Vision*, IEEE, 2008, pp. 1567–1572.

[53] I. Charfi, J. Miteran, J. Dubois, M. Atri, and R. Tourki, "Optimized spatio-temporal descriptors for real-time fall detection: Comparison of support vector machine and adaboost-based classification", *Journal of Electronic Imaging*, vol. 22, no. 4, pp. 041 106–041 106, 2013.

[54] Y. Yun and I. Gu, "Human fall detection in videos via boosting and fusing statistical features of appearance, shape and motion dynamics on riemannian manifolds with applications to assisted living", *Computer Vision and Image Understanding*, vol. 148, pp. 111–122, 2016.

[55] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Robust video surveillance for fall detection based on human shape deformation", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 5, pp. 611–622, 2011.

[56] X. Ma, H. Wang, B. Xue, M. Zhou, B. Ji, and Y. Li, "Depth-based human fall detection via shape features and improved extreme learning machine", *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 6, pp. 1915–1922, 2014.

[57] E. Stone and M. Skubic, "Fall detection in homes of older adults using the Microsoft Kinect", *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 1, pp. 290–301, 2015.

[58] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos", in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.

[59] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks", in *IEEE International Conference on Computer Vision*, 2015, pp. 4489–4497.

[60] J. Y Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification", in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4694–4702.

[61] Y. Fan, M. Levine, G. Wen, and S. Qiu, "A deep neural network for real-time detection of falling humans in naturally occurring scenes", *Neurocomputing*, vol. 260, pp. 43–58, 2017.

[62] Z. Zhang, X. Ma, H. Wu, and Y. Li, "Fall detection in videos with trajectory-weighted deep-convolutional rank-pooling descriptor", *IEEE Access*, vol. 7, pp. 4135–4144, 2018.

[63] J. Rizzo, J. Wyatt, J. Loewenstein, S. Kelly, and D. Shire, "Perceptual efficacy of electrical stimulation of human retina with a microelectrode array during short-term surgical trials", *Investigative Ophthalmology & Visual Science*, vol. 44, no. 12, pp. 5362–5369, 2003.

[64] E. Zrenner *et al.*, "Subretinal electronic chips allow blind patients to read letters and combine them to words", *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 278, no. 1711, pp. 1489–1497, 2011.

[65] T. Fujikado *et al.*, "Testing of semichronically implanted retinal prosthesis by suprachoroidal-transretinal stimulation in patients with retinitis pigmentosa", *Investigative Ophthalmology & Visual Science*, vol. 52, no. 7, pp. 4726–4733, 2011.

[66] W. Dobelle, "Artificial vision for the blind by connecting a television camera to the visual cortex", *ASAIO journal*, vol. 46, no. 1, pp. 3–9, 2000.

[67] C. Veraart *et al.*, "Visual sensations produced by optic nerve stimulation using an implanted self-sizing spiral cuff electrode", *Brain Research*, vol. 813, no. 1, pp. 181–186, 1998.

[68] M. Humayun *et al.*, "Visual perception in a blind subject with a chronic microelectronic retinal prosthesis", *Vision Research*, vol. 43, no. 24, pp. 2573–2581, 2003.

[69] J. Weiland, A. Cho, and M. Humayun, "Retinal prostheses: Current clinical results and future needs", *Ophthalmology*, vol. 118, no. 11, pp. 2227–2237, 2011.

[70] M. Humayun *et al.*, "Interim results from the international trial of second sight's visual prosthesis", *Ophthalmology*, vol. 119, no. 4, pp. 779–788, 2012.

[71] L. Cruz *et al.*, "The Argus II epiretinal prosthesis system allows letter and word reading and long-term function in patients with profound vision loss", *British Journal of Ophthalmology*, vol. 97, no. 5, pp. 632–636, 2013.

[72] A. Kurtenbach, H. Langrová, A. Messias, E. Zrenner, and H. Jägle, “A comparison of the performance of three visual evoked potential-based methods to estimate visual acuity”, *Documenta Ophthalmologica*, vol. 126, no. 1, pp. 45–56, 2013.

[73] J. Borenstein and Y. Koren, “The vector field histogram-fast obstacle avoidance for mobile robots”, *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991.

[74] W. Huang, B. Fajen, J. Fink, and W. Warren, “Visual navigation and obstacle avoidance using a steering potential function”, *Robotics and Autonomous Systems*, vol. 54, no. 4, pp. 288–299, 2006.

[75] A. Ess, B. Leibe, K. Schindler, and L. Gool, “Moving obstacle detection in highly dynamic scenes”, in *International Conference on Robotics and Automation*, IEEE, 2009, pp. 56–63.

[76] D. Dakopoulos and N. Bourbakis, “Wearable obstacle avoidance electronic travel aids for blind: A survey”, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 1, pp. 25–35, 2010.

[77] X. Chai, W. Yu, J. Wang, Y. Zhao, C. Cai, and Q. Ren, “Recognition of pixelized chinese characters using simulated prosthetic vision”, *Artificial Organs*, vol. 31, no. 3, pp. 175–182, 2007.

[78] Y. Zhao, Y. Lu, C. Zhou, Y. Chen, Q. Ren, and X. Chai, “Chinese character recognition using simulated phosphene maps”, *Investigative Ophthalmology & Visual Science*, vol. 52, no. 6, pp. 3404–3412, 2011.

[79] N. Parikh, L. Itti, M. Humayun, and J. Weiland, “Performance of visually guided tasks using simulated prosthetic vision and saliency-based cues”, *Journal of Neural Engineering*, vol. 10, no. 2, p. 026 017, 2013.

[80] J. Wang, X. Wu, Y. Lu, H. Wu, H. Kan, and X. Chai, “Face recognition in simulated prosthetic vision: Face detection-based image processing strategies”, *Journal of Neural Engineering*, vol. 11, no. 4, p. 046 009, 2014.

[81] J. Wang, Y. Lu, L. Gu, C. Zhou, and X. Chai, “Moving object recognition under simulated prosthetic vision using background-subtraction-based image processing strategies”, *Information Sciences*, vol. 277, pp. 512–524, 2014.

[82] T. Han, H. Li, Q. Lyu, Y. Zeng, and X. Chai, “Object recognition based on a foreground extraction method under simulated prosthetic vision”, in *International Symposium on Bioelectronics and Bioinformatics*, IEEE, 2015, pp. 172–175.

[83] J. Wang, H. Li, W. Fu, Y. Chen, L. Li, Q. Lyu, T. Han, and X. Chai, “Image processing strategies based on a visual saliency model for object recognition under simulated prosthetic vision”, *Artificial Organs*, vol. 40, no. 1, pp. 94–100, 2016.

[84] Stanford, *Convolutional Neural Networks (CNNs/ConvNets)*, `http://cs231n.github.io/convolutional-networks/`, 2019.

[85] D. Rumelhart, G. Hinton, R. Williams, *et al.*, "Learning representations by back-propagating errors", *Cognitive Modeling*, vol. 5, no. 3, p. 1, 1988.

[86] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization", *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[87] D. Kingma and J. Ba, "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980*, 2014.

[88] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning", *Coursera, Video Lectures*, vol. 264, 2012.

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

[90] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks", in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[91] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*, 2014.

[92] C. Szegedy *et al.*, "Going deeper with convolutions", in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[93] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[94] W. Samek, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer Nature, 2019, vol. 11700.

[95] Q. Zhang, Y. Wu, and S. Zhu, "Interpretable convolutional neural networks", in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8827–8836.

[96] L. Medsker and L. Jain, *Recurrent Neural Networks: Design and Applications*. CRC press, 1999.

[97] Y. Bengio, P. Simard, P. Frasconi, *et al.*, "Learning long-term dependencies with gradient descent is difficult", *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

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

[99] F. Gers and J. Schmidhuber, "Recurrent nets that time and count", in *IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, IEEE, vol. 3, 2000, pp. 189–194.

[100] K. Cho, B. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation", *arXiv preprint arXiv:1406.1078*, 2014.

[101] K. Greff, R. Srivastava, J. Koutník, B. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[102] I. Goodfellow *et al.*, "Generative adversarial nets", in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[103] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN", *arXiv preprint arXiv:1701.07875*, 2017.

[104] X. Mao, Q. Li, H. Xie, R. Lau, Z. Wang, and S. Smolley, "Least squares generative adversarial networks", in *IEEE International Conference on Computer Vision*, 2017, pp. 2794–2802.

[105] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning", *arXiv preprint arXiv:1605.09782*, 2016.

[106] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, "Adversarially learned inference", *arXiv preprint arXiv:1606.00704*, 2016.

[107] M. Mirza and S. Osindero, "Conditional generative adversarial nets", *arXiv preprint arXiv:1411.1784*, 2014.

[108] P. Isola, J. Zhu, T. Zhou, and A. Efros, "Image-to-image translation with conditional adversarial networks", in *IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134.

[109] J. Zhu, T. Park, P. Isola, and A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks", in *IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232.

[110] M. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks", in *Advances in Neural Information Processing Systems*, 2017, pp. 700–708.

[111] T. Wang, M. Liu, J. Zhu, G. Liu, *et al.*, "Video-to-video synthesis", in *Advances in Neural Information Processing Systems*, 2018, pp. 1144–1156.

[112] T. Wang, M. Liu, A. Tao, G. Liu, B. Catanzaro, and J. Kautz, "Few-shot video-to-video synthesis", in *Advances in Neural Information Processing Systems*, 2019, pp. 5014–5025.

[113] C. Ledig, L. Theis, F. Huszár, J. Caballero, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network", in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4681–4690.

[114] K. Ehsani, R. Mottaghi, and A. Farhadi, "SeGAN: Segmenting and generating the invisible", in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6144–6153.

[115] L. Yu, W. Zhang, J. Wang, and Y. Yu, "SeqGAN: Sequence generative adversarial nets with policy gradient", in *AAAI Conference on Artificial Intelligence*, 2017, pp. 2852–2858.

[116] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, *et al.*, "Domain-adversarial training of neural networks", *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[117] A. Larsen, S. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric", in *International Conference on Machine Learning*, 2016, pp. 1558–1566.

[118] T. Che, Y. Li, A. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks", *arXiv preprint arXiv:1612.02136*, 2016.

[119] W. Maass, "Networks of spiking neurons: The third generation of neural network models", *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[120] M. Abeles, *Corticonics: Neural Circuits of the Cerebral Cortex*. Cambridge University Press, 1991.

[121] H. Tuckwell, *Introduction to Theoretical Neurobiology: Volume 2, Nonlinear and Stochastic Theories*. Cambridge University Press, 1988, vol. 8.

[122] W. Maass, "On the computational complexity of networks of spiking neurons", in *Advances in Neural Information Processing Systems*, 1995, pp. 183–190.

[123] N. Kasabov, V. Feigin, *et al.*, "Evolving spiking neural networks for personalised modelling, classification and prediction of spatio-temporal patterns with a case study on stroke", *Neurocomputing*, vol. 134, pp. 269–279, 2014.

[124] M. Pfeiffer and T. Pfeil, "Deep learning with spiking neurons: Opportunities and challenges", *Frontiers in Neuroscience*, vol. 12, p. 774, 2018.

[125] ADNI, *Alzheimer's Disease Neuroimaging Initiative*, `https://adni.loni.usc.edu/about/`, 2017.

[126] A. Payan and G. Montana, "Predicting Alzheimer's disease: A neuroimaging study with 3D convolutional neural networks", *arXiv preprint arXiv:1502.02506*, 2015.

[127] S. Liu, S. Liu, W. Cai, S. Pujol, R. Kikinis, and D. Feng, "Early diagnosis of Alzheimer's disease with deep learning", in *International Symposium on Biomedical Imaging*, IEEE, 2014, pp. 1015–1018.

[128] A. Diba, V. Sharma, and L. Gool, "Deep temporal linear encoding networks", in *IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2329–2338.

[129] B. Erickson and Z. Akkus, *Data from LGG-1p19qDeletion*, https://wiki.cancerim-agingarchive.net/display/Public/LGG-1p19qDeletion, 2017.

[130] B. Menze, A. Jakab, *et al.*, "The multimodal brain tumor image segmentation benchmark (BRATS)", *IEEE Transactions on Medical Imaging*, vol. 34, no. 10, pp. 1993–2024, 2015.

[131] S. Bakas, H. Akbari, *et al.*, "Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features", *Scientific Data*, vol. 4, p. 170 117, 2017.

[132] S. Bakas, H. Akbari, H. Sotiras, *et al.*, *Segmentation Labels and Radiomic Features for the Pre-Operative Scans of the TCGA-GBM Collection*, https://doi.org/10.7937/K9/TCIA.2017.KLXWJJ1Q, 2017.

[133] S. Bakas, H. Akbari, *et al.*, *Segmentation Labels and Radiomic Features for the Pre-Operative Scans of the TCGA-LGG Collection*, https://doi.org/10.7937/K9/TCIA.2017.GJQ7R0EF, 2017.

[134] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency", in *Advances in Neural Information Processing Systems*, 2004, pp. 321–328.

[135] Y. Yun and I. Gu, "Human fall detection via shape analysis on riemannian manifolds with applications to elderly care", in *International Conference on Image Processing*, IEEE, 2015, pp. 3280–3284.

[136] E. Auvinet, C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Multiple Cameras Fall Dataset", *DIRO-Université de Montréal, Tech. Rep*, vol. 1350, 2010.

[137] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer", *Computer Methods and Programs in Biomedicine*, vol. 117, no. 3, pp. 489–501, 2014.

[138] E. Auvinet, F. Multon, A. Saint-Arnaud, J. Rousseau, and J. Meunier, "Fall detection with multiple cameras: An occlusion-resistant method based on 3-D silhouette vertical distribution", *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 2, pp. 290–300, 2011.

[139] D. Hung and H. Saito, "Fall detection with two cameras based on occupied area", in *Japan-Korea Joint Workshop on Frontier in Computer Vision*, 2012, pp. 33–39.

[140] A. Bourke, J. Obrien, and G. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm", *Gait & Posture*, vol. 26, no. 2, pp. 194–199, 2007.

[141] C. Ge, I. Gu, and J. Yang, "Human fall detection using segment-level CNN features and sparse dictionary learning", in *International Workshop on Machine Learning for Signal Processing*, IEEE, 2017, pp. 1–6.

[142] Z. Cheng, L. Qin, *et al.*, "Human daily action analysis with multi-view and color-depth data", in *European Conference on Computer Vision*, Springer, 2012, pp. 52–61.

[143] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, "Icoseg: Interactive co-segmentation with intelligent scribble guidance", in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3169–3176.

Part II

Papers