DISCLAIMER: More detailed information on the results and/or performance obtained and their use is available in the Subproject's subsequent Deliverable (D11.4.4) and/or Periodic Report.

Project Number:	604102	Project Title:	Human Brain Project

Document Title:	Applications: First Science Report
Document Filename:	SP11 D11.4.3 FINAL.docx
Deliverable Number:	D11.4.3
Deliverable Type:	First Science Report
Work Package(s):	WP11.1, WP11.2, WP11.3
Dissemination Level:	PU = Public
Planned Delivery Date:	M 18 / 31 March 2015
Actual Delivery Date:	M19 / 30 April 2015

SP Leader(s):	Karlheinz MEIER, UHEI (P45)	
SP Manager(s):	Martina SCHMALHOLZ, UHEI (P45)	Deliverable Template, compiled Document
Contributors:	Gregory FRANCIS	Text for WP 11.1.1, editing for WP 11.1
Contributors:	Eduardo ROS	Text for WP 11.1.1
Contributors:	Florian RÖHRBEIN	Text for WP 11.1.2, editing for WP 11.1
Contributors:	Ferath KHERIF	Text for WP11.2
Contributors:	Frank GOTTFRIED	Text for Task 11.3.1
Contributors:	Chris HUYCK	Text for Task 11.3.2
Contributors:	Bernabe LINARES	Text for Task 11.3.3
Contributors:	Ulrich RÜCKERT	Text for Task 11.3.4
Contributors:	Ryad BENOSMAN	Text for Task 11.3.5
Contributors:	Thomas NOVOTNY	Text for Task 11.3.6
Contributors:	Michael SCHMUCKER	Text for Task 11.3.6
STO Review:	Martina SCHMALHOLZ, UHEI (P45)	
Editorial Review:	EPFL (P1): Richard WALKER, Guy WILLIS, Lauren ORWIN	

Abstract:	This document summarises the achievements and open issues of the Applications Subproject in the Human Brain Project. It is structured according to the three Work Packages in Neuroscience, Medicine and Neuromorphic Computing. The Subproject is progressing according to the original plan.

Keywords:	Neuroscience, medicine, computing, applications, artificial retina, Neurorobotics, disease classification, neuromorphic, spike based computing

Document Status

Version	Date	Status	Comments
0.1	05 March 2015	Draft	Template received from SP Manager
0.2	17 March 2015	Draft	Task 11.1.1: Integrated material from UGR to this template and added material for EPFL.
0.3	19 March 2015	Draft	Included material for Task 11.1.2.
0.3	20 March 2015	Draft	Comments from Martina to WP11.1
0.3	20 March 2015	Draft	Draft of Task 11.3.3 sent to Martina
0.4	23 March 2015	Draft	Task 11.1.1 modified text to address issues raised by Martina
0.4	23 March 2015	Draft	Draft of Task 11.3.2 sent to Martina
0.5	24 March 2015	Draft	Revised text of Task 11.1.1 with information from UGR.
0.5	31. March 2015	Draft	Draft of Task 11.3.4 sent to Martina
0.5	01 April 2015	Draft	Draft of Task 11.3.1 sent to Martina
0.5	02 April 2015	Draft	Draft of Task 11.3.6 sent to Martina
0.5	20 April 2015	Draft	Draft of Task 11.3.5 sent to Martina
0.6	21 April 2015	Draft	Summary Draft compilation and revision
0.7	24 April 2015	Draft	Summary draft sent to Task-Leaders with request for some additional input
0.7	28 April 2015	Draft	Requested additional input incorporated; draft sent to SP-Leader
1.0	15 May 2015	Draft	Copy-edited LO
1.0	22 May 2015	Final	Revised Document sent by MS to Editorial Team for final submissions

Table of Contents

List of Figures and Tables

Executive Summary

This document describes the achievements and work carried out in application areas emerging from the Human Brain Project's (HBP) work in neuroscience, medicine and computing. The Subproject (SP) is structured into Work Packages (WPs) 11.1, 11.2 and 11.3, which relate to the three areas described. The SP is scientifically rather disjunct, so this summary is structured according to the three WPs.

Work Package 11.1 developed a retina model, to be used as a sensor as input for cortical models. The retina model has been successfully validated against neurophysiological and psychophysical data. The WP also developed a cortical model of visual processing, using HBP software tools. This cortical model has been validated against psychophysical data, and has been shown to match human behaviour in several situations. Finally, the HBP Neurorobotics tool chain was used to calibrate the simple neural network of the Braitenberg vehicle.

Work Package 11.2 successfully delivered a standardised description format for biological signatures of brain diseases. The on-going study of the Alzheimer's disease (AD) phenotype, using machine learning methods and brain pathology, identified several subgroups or subtypes of AD. Therefore, the first standard description of the biological signatures of brain diseases was defined. The work then progressed to function 11.2.1.2: informatics-based model for generating biological disease signatures. In this Task, other types of algorithms for building disease models were tested and benchmarked. An important aspect of this work is to design a method that can select the best features.

Work Package 11.3 prepared for application cases from the Neuromorphic Systems. The industry Partner SAP AG (P47) is investigating the potential of spiking neural networks in business applications. For that purpose, we are preparing a PyNN application for use on Neuromorphic Systems. In addition, the CABot3 agent now runs on SpiNNaker. This is an agent that takes natural language commands, views the environment, executes plans, and learns a simple cognitive map. Two spiking associative memory (SAM) models were implemented and simulated with different parameter sets. Automatic performance evaluation tools were developed to analyse simulations, and the first benchmark data sets were generated. We have also performed a successful mapping of SAM models on the Heidelberg hardware emulator (ESS). With an interface to connect two ATIS cameras to SpiNNaker, it is now possible to transmit and process events provided from ATIS cameras. The maximum input flow rate has been estimated to 1.4 million events per second for each camera. Finally, we scaled-up the size of a spiking network for neuromorphic pattern recognition to several thousand neurons on a classical GPU-simulator. The network has been applied to odorant recognition from electronic nose recordings, and is now ready for neuromorphic use.

1. Introduction

1.1 Overview of Subproject 11 Achievements

1.1.1 WP11.1

- One of the WP's major achievements was the development of a retina platform that acts as a sensor for the cortical models. This platform has been validated against neurophysiological and psychophysical data, so that it can reproduce different retina models as needed. Another major achievement was the development of a cortical model of visual processing using HBP tools (e.g. NEST). This cortical model has been validated against psychophysical data, and has been shown to match human behaviour in several situations. A third major achievement was the successful use of the SP10 Neurorobotics tool chain to calibrate the simple neural network of the Braitenberg vehicle.
- Task 11.1.1: Using the retina platform and the cortical visual processing model, we explored properties of Weber's Law for brightness perception. The retina model exhibits Weber-like processing of brightness information, and the cortical model has been shown preserve such representations.
- Task 11.1.2: As a first test of the Neurorobotics tool chain, a Braitenberg vehicle experiment was developed in close collaboration with SP10. This demonstrates the interplay of virtual robots with a virtual environment under different light conditions. More detailed results will be reported in the resubmitted D11.4.2.

1.1.2 WP11.2

- The computer-aided diagnosis of AD has proven to be a promising method of early detection, an important condition for treating the disease more effectively. Most of the diagnostic tools that have been developed are based on the evaluation of magnetic resonance imaging (MRI) scans with multiple modalities. These are sometimes supplemented with additional information, such as positron emission tomography (PET) scans, or genetic or cerebrospinal fluid values to improve the classification accuracy.
- Milestone 204, "standardised description format for biological signatures of brain disease", was achieved. The on-going study of the AD phenotype, using machine learning methods and brain pathology, identified several subgroups or subtypes of AD. It therefore defined the first standard description of the biological signatures of brain diseases. The work then progressed to function 11.2.1.2: informatics-based model for generating biological signature of a disease. In this task, we tested and benchmarked other types of algorithms for building the disease model. An important aspect of this is to design a method that can select the best features.

1.1.3 WP11.3

- Task 11.3.1: SAP AG is working on two specific use cases to investigate the possibilities and potential advantages of algorithms based on spiking neural networks in business applications. For that purpose, a PyNN implementation is underway, preparing for upcoming use on the Neuromorphic Platform.
- Task 11.3.2: The CABot3 agent now runs on SpiNNaker. This is an agent that takes natural language commands, views the environment, executes plans, and learns a simple cognitive map. A simple virtual environment has also been developed. The agent exists in the environment, but all of its processing is done by neurons on

SpiNNaker. The virtual environment sends bitmaps and natural language user commands to the SpiNNaker board in the form of particular neurons spiking, and collects spikes from four neurons that correspond to the agent moving forward or backward, or turning left or right.

- Task 11.3.3: We have progressed as follows: two people attended the January 2015 SpiNNaker workshop in Manchester. Regarding interfaces, we are working on a modified version of the interfacing between SpiNNaker chips and FPGAs on the 48-chip SpiNNaker board. This will eventually allow us to maximise the event throughput to its physical maximum: 6 Meps to 64 bit events. We are still performing tests and characterisations in this respect, to make sure performance and reliability are satisfactory. Regarding PyNN descriptions of neurons models, we have three on-going descriptions. These are: (a) a slightly modified neuron model taken from the standard SpiNNaker library, that allows us to combine pairs of neurons to emulate positive and negative weight events from a single neuron; (b) a heavily modified neuron model which performs instantaneous events updates (no need to wait for the 1 ms global update step); and (c) a full population model of a convolution population directly emulating a large amount of neurons. Of these, (a) is fully compatible with the official PACMAN release, while (b) and (c) require a modified custom PACMAN version. We have performed some initial work to allow for fast parameter updates/perturbations. This will be required for Tasks on optimisations and mismatch characterisations, and hence is part of these Tasks. We have also performed some initial work on the feedback analyses Task, and tested some preliminary PyNN descriptions.
- Task 11.3.4: Two SAM models in PyNN were implemented and simulated with different parameter sets. To analyse the simulations, we implemented automatic performance evaluation tools, and generated the first benchmark data sets. Following successful mapping of SAM models on the Heidelberg hardware emulator (ESS), we are now prepared for the use of the Neuromorphic Platform NM-PM in Heidelberg, and will fulfil our first Milestone on schedule.
- Task 11.3.5: All hardware implementations and the database were completed. With the interface that connects one ATIS camera to SpiNNaker (function 11.3.5.1), we are able to transmit and process ATIS camera events. This has been accepted as a Live Demonstration that will be presented at the IEEE International Symposium on Circuits and Systems (ISCAS) 2015 (Garrick ORCHARD, Xavier LAGORCE, Christoph POSCH, Steve FURBER, Ryad BENOSMAN. Live Demonstration: Real-Time Event-Driven Object Recognition on SpiNNaker). We are also able to transmit and process events provided by two ATIS cameras, using the interface to connect two ATIS cameras into SpiNNaker (function 11.3.5.2). The maximum input flow rate has been estimated to 1.4 million events per second, per camera. The stimulation platform (function 11.3.5.3) -a table XY- allows us to automatically acquire datasets of different patterns from one (or several) ATIS cameras, with different motions and speeds. The platform is able to reach more than 1 m/s. Above this speed, the data flow is irrelevant, because the sensor tends to saturate. This platform will be described in an article we expect to submit to a Special Issue in *Frontiers of Neurosciences*, on "Benchmarks and Challenges for Neuromorphic Engineering". The database (function 11.3.5.4) has been built using the aforementioned platform. Several data flows have been acquired for benchmarking. In addition, data flows have been acquired in natural environment conditions. A software interface has been developed with Matlab to allow users to visualise these data, and to analyse them in terms of activity, which is the principal criteria we will use to characterise the software implementations. Again, we are currently working on an article using a part of this dataset for a Special Issue in *Frontiers of Neurosciences* on "Benchmarks and Challenges for Neuromorphic Engineering". Technical reports on the hardware platforms have been provided as

scheduled. More details can be found at https://collaboration.humanbrainproject.eu/web/collab/wiki/-/wiki/Main/WP11.3.

- Task 11.3.6: Thomas NOWOTNY's group (University of Sussex – P112) achieved their intermediate goal of scaling up the size of a spiking network for neuromorphic pattern recognition to several thousand neurons on a GPU-simulator. They applied the network to odorant recognition from electronic nose recordings. An article that documents their findings is currently under review at a peer-reviewed journal.

1.2 Overview of Subproject 11 Problems and Corrective Actions

1.2.1 WP11.1

- Milestone 202 (due Month 18) was to integrate the visual processing models with SP10's Neurorobotics Platform. This Milestone was not reached on time, because the Neurorobotics Platform was not due to be released to the HBP until Month 18. Our corrective action was to engage with members of SP10 to plan our integration for when the platform is released. We hope to reach the Milestone four months after the release of the Neurorobotics Platform. This delay should not prove too troublesome for the WP, because other topics can be explored in parallel.
- Task 11.1.1: The plan was to combine the retina and cortical models into a unified system. The team at *Universidad de Granada* (UGR – P58) has found a solution, but technical issues (perhaps missing libraries) stopped the solution from working at *École Polytechnique Fédérale de Lausanne* (EPFL – P1). We could not fully evaluate the systems until this technical problem was addressed, which finally happened in mid-March. We should now be able to catch up on some KPIs, which we had fallen behind on.
- Task 11.1.2: It proved difficult to create a full set of benchmarks for the Neurorobotics tool chain while its properties were still under development, but we should catch up once the Neurorobotics Platform is released in Month 18.

1.2.2 WP11.2:

- Task 11.2.1: The project is on schedule and no problems have occurred.

1.2.3 WP11.3

- Task 11.3.1: We expect a short delay in finishing the PyNN implementation. This will lead to an approximate two-month delay in completing Milestone 210. The revised timeline will fit with the expected availability of the hardware systems, and should not cause any further delay to the planned proof-of-concept implementation.
- Task 11.3.2: Some of our internal waypoints are slightly behind schedule, but we hope to catch up on all aspects. One remaining problem is that we do not have a HICANN chip that we can use for our agent. Hopefully, progress will be straightforward once it arrives, but we will proceed with the simulator in the meantime.
- Task 11.3.3: We are heavily dependent on the evolution of software support developments for SpiNNaker. Until this point, we have not encountered a major problem due to this, and we have been able to progress more or less as planned. At present, there is an issue with the new release of PACMAN, designed to support interfacing in real time with an external retina. However, we expect this problem to be solved in the near future. We still rely on previous PACMAN releases for retina interfacing.

- Task 11.3.4: The project is on schedule and no problems have occurred.
- Task 11.3.5: The project is now on schedule and no further problems have occurred.
- Task 11.3.6: Our initial plan to implement the pattern recognition network on the HICANN chip had to be postponed due to the limited availability of that chip. Instead, we are now using the SpiNNaker system as a basis for network hardware implementations.

2. Future Neuroscience (WP11.1)

2.1 Application/Experimental Set-up and Results

The retina platform has been evaluated against published neurophysiological data [1, 2, 3, 4, 5, 6, 7], that describe three significant features of retina processing: contrast, mean luminance adaptation, and chromatic opponency pathways (KPIs 11.1.1.1, 11.1.1.2, 11.1.1.3, 11.1.1.4). The experiment setups implement the same set of synthetic visual stimuli (e.g. drifting sinusoidal gratings and flashes) used by neuroscientists to perform intracellular recordings of retina cells (KPI 11.1.1.5). Conference papers that have already been submitted [1, 2, 3] give the preliminary results of neural simulations. They also show the potential of this tool to feasibly reproduce different retina models, and to be easily connected with other HBP platforms, such as the NEST simulation platform being developed in SP6 (KPIs 11.1.1.6, 11.1.1.7).

The LAMINART cortical model of visual processing [8, 9] has been implemented in NEST (KPIs 11.1.1.8, 11.1.1.9, 11.1.1.10), which should enable it to integrate with the Neurorobotics Platform (KPI, 11.1.1.7). Cortical processing has been shown to preserve brightness information representations that come from the retina model, which is important for explaining some brightness percepts (KPIs 11.1.1.13, 11.1.1.15, 11.1.1.17, 11.1.1.20). The model has also been shown to produce illusory contours in a way that mimics human perception (KPI 11.1.1.16).

Task 11.1.2 ensures that the models being developed in Task 11.1.1 integrate smoothly with the Neurorobotics Platform. To do so, many SP10 meetings, and meetings between SP10 and WP11.1, were held. These established what features would be needed in the Neurorobotics Platform, and their properties. A first series of experiments using the complete Neurorobotics Platform tool chain has already been successfully completed.

2.2 Interaction with Neurorobotics Platform

The visual processing models have been designed with to use the Neurorobotics Platform. We have used HBP technologies that should integrate smoothly with the Platform and related technologies. To ensure integration, we have provided SP10 with guidance on necessary features for Future Neuroscience, and contributed to meetings on the Platform's development. This guidance included informal discussions about the kinds of experiments that are likely to be performed by vision scientists, and the necessary properties of the simulation environment, such as how to measure simulated luminance. We anticipate even closer interaction when the Neurorobotics Platform is released in Month 18.

2.3 Interaction with other HBP Subprojects

Our work has focused on making a system that will integrate with the Neurorobotics Platform. Once the visual processing model is implemented in the Neurorobotics Platform, we anticipate that there will be many opportunities to work with other Subprojects, especially SP3, SP6, SP7 and SP9, which involve visual processing.

2.4 Outreach

- Clarke, A. M., Herzog, M. H. & Francis, G. (2014). Visual crowding illustrates the inadequacy of local versus global and feedforward versus feedback distinctions in modelling visual perception. *Frontiers in Psychology: Perception Science*, **5**, 1193. doi: 10.3389/fpsyg.2014.01193.
- Francis, G. (under review). Contour erasure and filling-in: Old simulations account for most new observations. *I-Perception*.
- Francis, G. (03 March 2015). Thoughts from a cognitive psychologist. *Are we building the right thing? - Requirements from theory for simulation environments and neuromorphic computing*. European Institute for Theoretical Neuroscience, Paris.
- Manassi, M., Hermens, F., Francis, G. & Herzog, M. H. (under review). Release of crowding by pattern completion. *Journal of Vision*.
- Martínez Cañada, P., Morillas, M., Nieves, J.L., Pino, B., Pelayo, F. (2015). *First Stage of a Human Visual System Simulator: the Retina*. In Computational Color Imaging Workshop (CCIW'15). Accepted, Paper in Press.
- Martínez Cañada, P., Morillas, M., Pino, B., Pelayo, F. (2015). *Towards a Generic Simulation Tool of Retina Models*. In International Work-Conference on the Interplay between Natural and Artificial Computation (IWINAC'15). Paper under review.
- Martínez Cañada, P., Morillas, M., Romero, S., Pelayo, F. (2015). *Modeling Retina Adaptation with Multiobjective Parameter Fitting*. In International Work-Conference on Artificial Neural Networks (IWANN'15). Paper under review.

3. Future Medicine (WP11.2)

Data Set	Classification and clustering	Informatics-based model	Algorithms and Benchmarks [1]	Comments
Research data from ADNI and 3C: imaging, genetic and clinical variable	Semi-supervised clustering algorithm	Rules-based classification Six rules were derived for explaining AD	Density-based algorithm compared to the use of state of the art "black box" methods	The results show that low dimension factors can explain whole datasets
Research data from ADNI and 3C: imaging, genetic and clinical variable	Support vector machine classifier trained on the pathology proven data and tested on the previous research data (ADNI)	Automated diagnostic	The results were compared to clinical diagnostic performed by neurologists (expert knowledge)	The results show that automated biological based classification can identify clinically healthy controls at risk of dementia
Research data from ADNI and 3C: imaging, genetic and clinical variable	Deep learning algorithm	Automated feature learning	Neural net/stacked auto-encoder Compared atlas based features vs. random based features Compared to clinical label	The results show that the algorithm was able to learn the best features for optimum accuracy

[1]: Please link to any kind of reference or description in text

3.1 Model Set-up and Results

Computer-aided diagnosis of AD has proven to be a promising method of early detection, which is an important factor in the effective treatment of the disease. Most of the diagnostic tools developed are based on the evaluation of MRI scans with multiple modalities, sometimes supplemented with additional information, such as PET scans, and genetic or cerebrospinal fluid values, to improve the accuracy of classification.

Milestone 204 "standardised description format for biological signatures of brain disease" has been achieved. The on-going study of the AD phenotype, using machine learning methods and brain pathology, has identified several subgroups or subtypes of AD. This therefore defined the first standard description of biological brain disease signatures. The work progressed onto function 11.2.1.2: informatics-based model for generating biological disease signatures. In this task, we tested and benchmarked other types of algorithms for

building the disease model. An important aspect of this was to design a method that could select the best features.

Function	Function Name	Possible KPI statuses	Current KPI status	Target
11.2.1.1	Description format for the biological disease signature	Identify multimodal clinical data Data pre-processing Data aligned Feature selection	All achieved	M3 M6 M9 M12
11.2.1.2	Informatics based model for generating biological disease signature	Implement test different algorithms Model configuration Benchmark algorithms Select algorithms	All achieved	M12 M18 M18 M18

3.1.1 Informatics-based Model One: Unsupervised Rule-based Clustering

Objectives: There is a great amount of uncertainty regarding the accuracy of diagnostic classification in the early stages of AD. This is due to the underlying heterogeneity in etiologies leading to similar phenotypes. To explain the observed heterogeneity, we use a rule-based clustering algorithm, and identify homogeneous subgroups of patients. The hypothesis is that such subgroups have the same underlying causes.

Methods: We used high resolution T1-weighted 3D data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) datasets, which included 66 healthy controls and 48 AD patients. Participants were matched for age and gender. Firstly, the data were normalised to a common template, using new segmentation and Diffeomorphic Anatomical Registration using Exponentiated Lie algebra (DARTEL) processing in SPM8. This allowed us to extract measures of Grey Matter volume (GMv) from each voxel. Next, we summarised the data into regions of interest, based on the AAL atlas (Figure 1).

The rule-based algorithm aims to explain the variability between individuals, and describes a population by a group of “local over-densities”. These are defined as subspaces over combinations of variables. The algorithm performs an exhaustive search of the data space to predict the outcome variables; in this case, the health status of each subject in terms of the presence or absence of AD. In our experiment, the predictive variables are the 90 brain region volumes, age, gender, and individual subject global volumes.

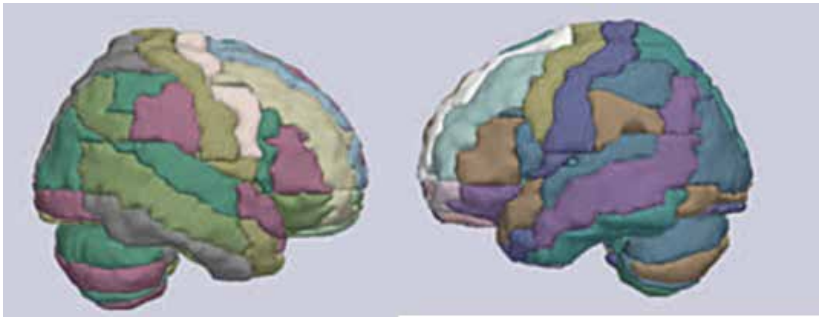

Figure 1: 3D view of the AAL Atlas

The AAL atlas is composed of 45 anatomical regions for each brain hemisphere. In our sample, we extracted mean volumes of each anatomically defined region as features for analysis by the data-mining algorithm.

Results: After convergence and cross-validation tests, Hypercube showed that the data could be explained by six different rules for AD patients, and five rules for healthy controls. Bringing these rules together maximises the difference between healthy controls and AD patients. At population level, this result shows that there are six ways of presenting with an Alzheimer phenotype. These six ways correspond to six different sets of regions (see figures 2 and 3). At individual level, nonlinear effects are captured by the fact that each participant can be explained by more than one rule (see Table 1 for the proportion of overlap between rules). Critically, in prediction mode, AD and control rules explain 98% of AD patients and 100% of controls.

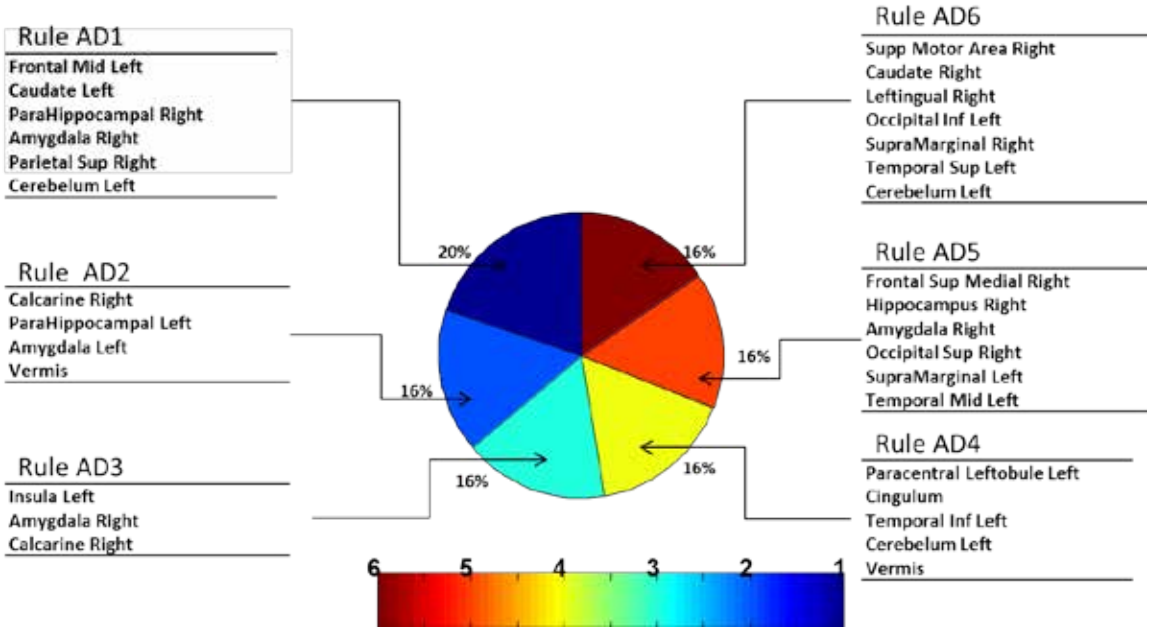

Figure 2: Anatomical Label of the Six Rules, and Proportion of Data Space Explained by Each Rule

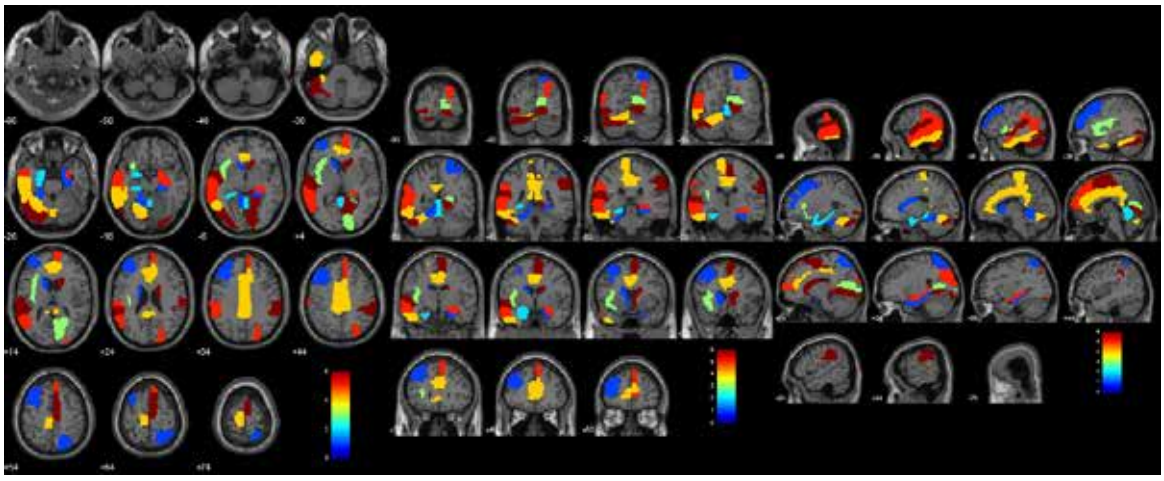

Figure 3: Brain Regions Contributing to Each Set of Rules

AD patients	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6
Rule 1						
Rule 2	44%					
Rule 3	58%	33%				
Rule 4	58%	28%	74%			
Rule 5	50%	39%	44%	44%		
Rule 6	56%	39%	44%	56%	33%	

Table 1: Proportion of Subjects described by Two Rules

Conclusion: The results show that there is more than one pattern of regional brain pathology characterised by an Alzheimer phenotype. The regions contributing to each set of rules or subgroups are very specific. Patients differ from controls according to a very systematic pattern, which involves regions known to show atrophy from pathological examination. Specifically, the results showed that firstly, the pattern of differences between AD patients and controls involved regions beyond the medial temporal lobes, secondly, that there is evidence for the existence of several subgroups of AD patients, and thirdly, that these subgroups can be predicted with high accuracy from a low number of deterministic rules.

Given the small number of patients and variables included in the pilot study, we have concluded that it would be very worthwhile to carry out a further study. This would include a much greater number of subjects, and a considerably greater range, diversity, and amount of data describing their states. This follow-up would aim to confirm the preliminary results, and specify with greater precision how many sets of rules are needed to identify patients with the AD phenotype. Adding more data may result in fewer rules, if some of the patterns identified in our small pilot sample are idiosyncratic and replaced by a more consistent set of factors. The inclusion of new data and auxiliary variables (genetic, cognitive, etc.) would provide a smaller-grained, direct descriptive explanation of the underlying causative pattern identified in this preliminary analysis.

3.1.2 Informatics-based Model Two: Enriched Automated Diagnostic Tools

Objectives: It has been predicted that delaying AD onset by just one to two years would result in 9.5 to 23 million fewer symptomatic and dependent cases by 2050 [10]. Alzheimer's Disease pathology, like that of Parkinson's disease, precedes symptoms. This is demonstrated in the significant redundancy of brain organisation, with a resulting capacity for reorganisation in the face of pathology [11]. In light of this, it is legitimate to propose a strategy of preventive therapy for dementia patients. This would require accurate diagnosis prior to the onset of symptoms, or demonstrable signs and syndromes. Identifying accurate biomarkers, independent of symptoms, is critical to such a strategy.

Methods: We built an automated classifier from a set of MRI scans that came from deceased, pathologically diagnosed individuals. This classifier was evaluated for its prognostic value on clinically categorised living people. Subjects were clinically diagnosed as either healthy control (HC) or AD, and then grouped by the presence or absence of AD related atrophy into probable AD or NC. Recent evidence suggests that a clinical diagnosis of AD has 70% sensitivity and 44% specificity when patients are followed to autopsy [12]. We compared the clinical diagnosis with one based on an AD-typical pattern of brain atrophy and biomarker profiles (genetic and proteomic), and aimed to correlate the results with clinical evidence of subsequent cognitive decline. We tested the idea that cognitively

normal people with an AD related brain atrophy pattern were at high risk for conversion to memory impairment and dementia. We evaluated the relative risk of such conversion, and compared it with other conventional AD risk factors, such as the Apolipoprotein E4 (APOE-ε4) genotype, and AD-associated single nucleotide polymorphisms (SNP).

Results: We used all 33 pathologically verified subjects to train an SVM classifier, and evaluated the performance using a leave-one-out cross-validation. We achieved 88% accuracy in diagnostic discrimination (sensitivity=88.8%, specificity=86.6%). The classifier was then used on ADNI subjects to predict pathological diagnoses based on anatomical patterns of atrophy. All ADNI subjects had been clinically diagnosed. By adding predicted pathological diagnoses, all subjects received a binary label that referred to clinical diagnosis and SVM prediction, e.g. “clinically healthy/predicted AD” or HC_AD. ADNI subjects fell into four groups: 275 clinical healthy controls with normal anatomical patterns: HC_HC, 192 clinical AD with an AD atrophy pattern: AD_AD, 91 clinically diagnosed AD subjects were classified as HC by the SVM, and 83 clinically diagnosed HC subjects had an AD-specific atrophy pattern.

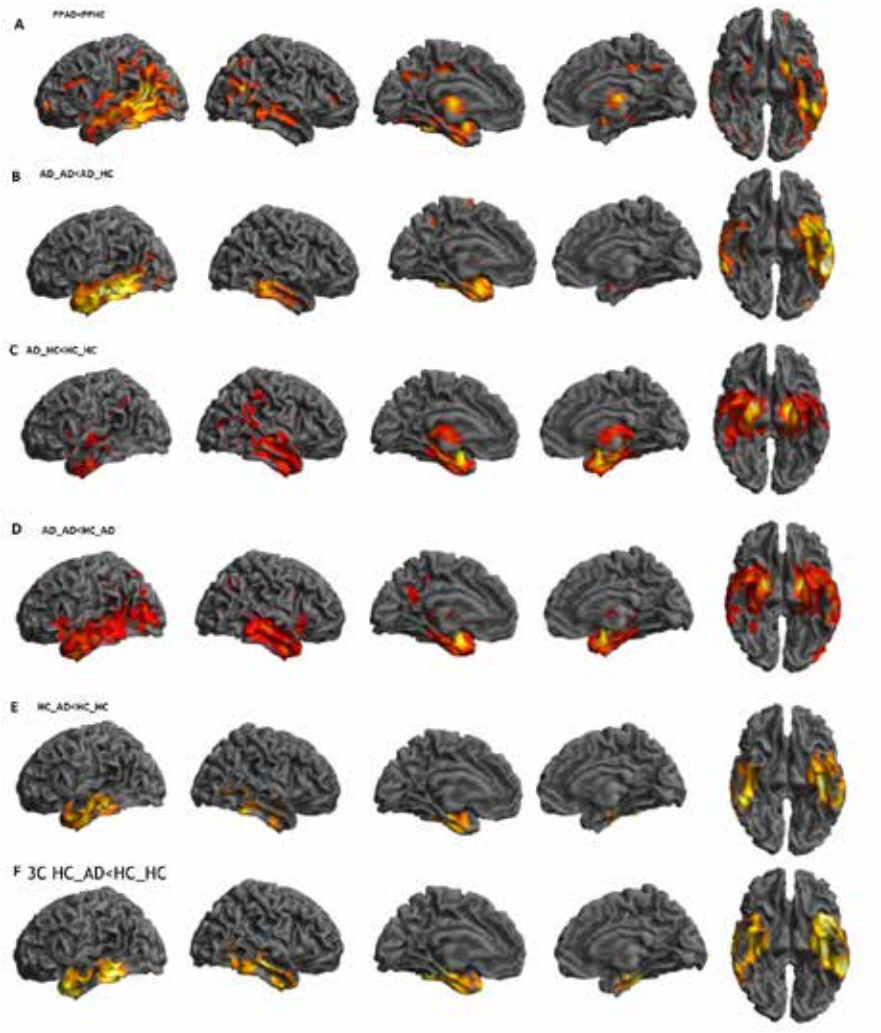

Figure 4:

A shows a comparison of pathologically validated AD and HC revealed atrophy. B—E show group comparisons among ADNI subgroups. F compares the atrophy pattern of cognitively normal participants classified as AD (HC_AD) in the 3C study, to those classified as HC_HC.

Survival analysis: Clinical HC participants were followed for a median of 35.8 months. We examined whether people diagnosed as healthy (based on clinical scores) at their first visit developed memory impairment during follow-up, as a function of SVM predicted outcome (HC_AD vs. HC_HC). Five years after prediction at the first screening visit, clinical HC classified as AD patients had a survival rate of 66.6% (95% confidence interval [CI], 52% to 83%) in terms of conversion to memory impairment or dementia. On the other hand, HC_HC had an 83% survival rate (95% CI, 76% to 89%). Log-rank testing showed a significant difference in conversion time between the two groups (p= 4.3e-03).

All AD related factors were tested for associations with conversion to memory impairment or dementia using Cox's proportional hazard regression. After adjustment for gender, age, APOE-ε4 genotype and education, we found that, cognitively, HC subjects with atrophy had a 2.5 times higher risk of developing memory impairment than those without.

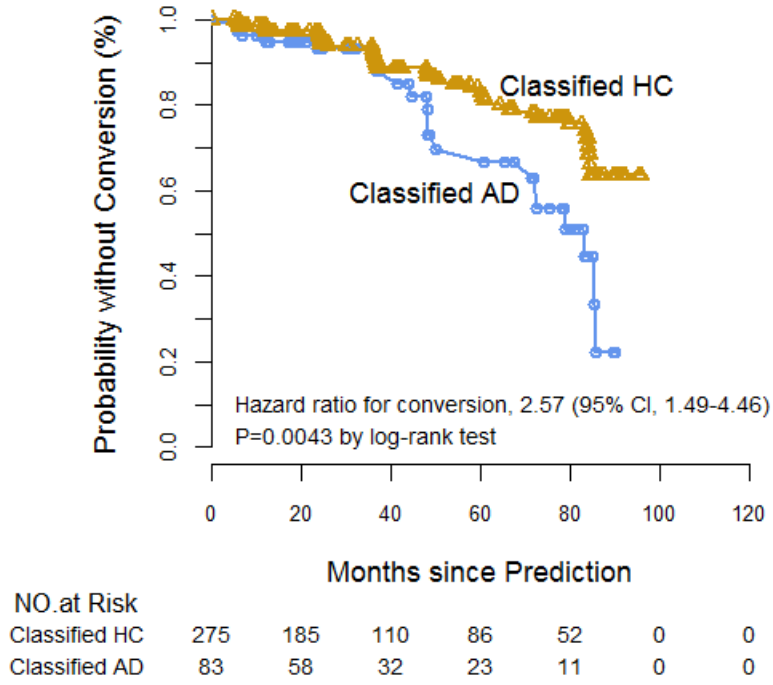

Figure 5: Survival Analyses

Our methods identified that cognitively healthy individuals with atrophy have a 2.5 times higher risk of developing memory impairment than those without atrophy.

Conclusion: Individuals with mismatched labels showed intermediate characteristics in both anatomy patterns and memory performance. This characterises different mechanisms related to AD.

3.1.3 Informatics-based Model Three: Deep Learning for Automated Features Extractions

Objectives: The increasing calculation power of computers has led to a rising interest in complex machine learning methods. In particular, the investigation of artificial neural networks with many hidden layers continuously results in promising new applications. These include image and face recognition [13, 14, 15], speech recognition [16, 17] and signal processing [18]. Very recently, these deep learning networks have also been used in the

classification of AD patients versus healthy control subjects, resulting in accuracies of up to 95% ([20]).

Methods: We used MRI scans from the publicly available ADNI database. T1-weighted scans were used from 359 HCs and 284 mild AD patients. The groups were matched for age and gender. We tested two methods of feature extraction for training deep learning networks, both of which ensured the anonymity of the individual subjects, and resulted in manageable input vectors. The first of these was the classical ROI approach, using volumes of grey matter regions as input values. The second feature set consisted of a random set of two-dimensional sub-images extracted from each of the MRI scans. The pixel values of these patches were fed directly into the neural network for training and classification. The main difference between our study and previous attempts to use small sub regions for AD diagnosis, is that in our study, no preliminary group comparison was performed to select affected brain regions. On the contrary, proper analysis of the classification results, based on randomly located training patches, can be used to locate affected grey matter regions in individual AD patients.

One of the most widely used deep neural networks for classification is the stacked auto-encoder. An auto-encoder contains three layers, the first (input) and last (output) layer of which are identical and known. These are connected to the neurons in the middle layer (the hidden layer) by means of weight matrices W_{ji}, where the indices refer to connections from layer i to layer j. There are no connections between neurons in a single layer. For a certain input vector, each neuron in the hidden layer produces an output value defined by the relation $y = f(W_{21}.x + b_2)$. The value y is called the activation of a neuron. The function f can take any form, but because of its saturation properties, and advantageous mathematical properties, mostly sigmoid functions (varying from 0 to 1) or hyperbolic tangents (varying from -1 to 1) are used. The value b_2 is a bias linked to each neuron in a hidden layer. Its value has to be optimised, together with the weights in W_{ij}. In a second step, the output vector y is used as input to reproduce the original input layer: $\hat{x} = f(W_{32}.y + b_3)$. The aim of an auto-encoder is to minimise the difference between $\hat{x}$ and x, by finding an optimal value for W_{ji} and b_j. This way, an initial input vector can be encoded and decoded using the optimal weight matrices and bias vectors. If the number of hidden neurons is lower than the number of elements in the input vector, an auto-encoder can be used as a data compression mechanism. For this type of training, only input values are required; it is therefore called unsupervised learning.

Several of the hidden layers can be stacked in order to capture more complex properties of the input, and these deep structures have been used successfully in many classification tasks. The top layer represents the classification label of the input vector, and once again the weights in the network have to be optimised to obtain good agreement with the given labels. Since output values are now compared to known labels, this technique is referred to as supervised learning.

Due to the very high amount of parameters to fit, training the entire neural network based on an input vector and a classification label is very slow, and often results in low quality local optima. However, it has been shown empirically that unsupervised pre-training of each individual layer, followed by a supervised optimisation of the entire network, yields good results.

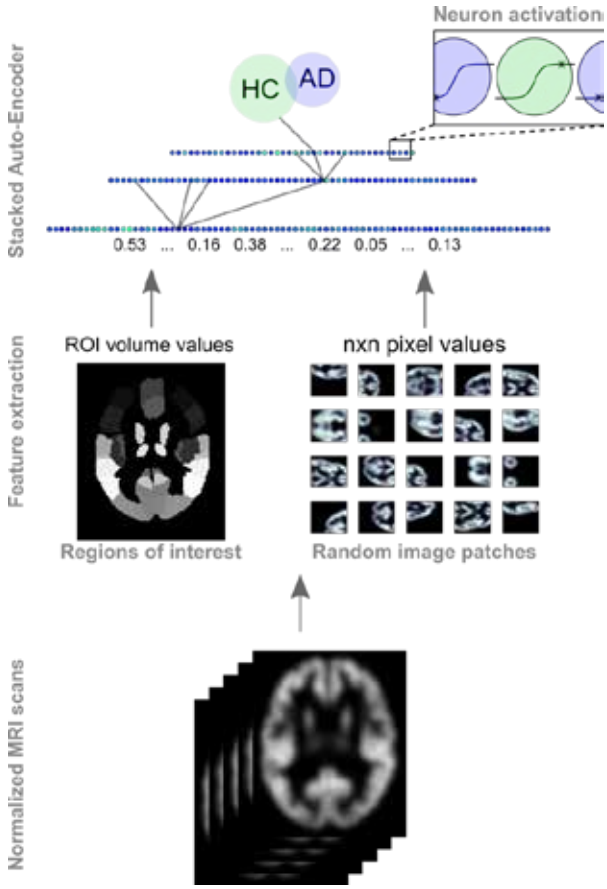

Figure 6: Feature Extraction for a Stacked Auto-encoder Containing Two Hidden Layers

Results: Using volumes of grey matter ROI defined by a brain atlas, HC subjects were correctly classified in 75% of cases, and only 7% were misclassified. The performance accuracy is lower for AD classification (57% correct vs. 19% misclassified), but the misclassified subjects actually show structural properties of the HC subjects (almost no atrophy in temporal and hippocampal areas). Additionally, the MMSE scores of misclassified AD subjects were significantly higher than those of the true AD subjects. This could mean that we have identified a subgroup, which needs a different label referring to a mild or early stage of the disease.

The alternative approach for feature extraction, using random two-dimensional patches of normalised grey matter scans, resulted in distinctive results for HC and AD subjects. The most striking of these was the fact that patches from temporal and hippocampal regions performed much better for AD classification, whereas we noticed no significant variability in classification accuracy of individual patches for HC patients. It is, however, not straightforward to put a single label on a subject, due to the classification distribution of the patches. A possible approach might be to define a threshold of the number of correctly classified patches, above which the subject is supposed to be healthy. Based on our results, a possible threshold value might be 70%, leading to an accuracy of 84% for HC subjects, and 93% for AD subjects.

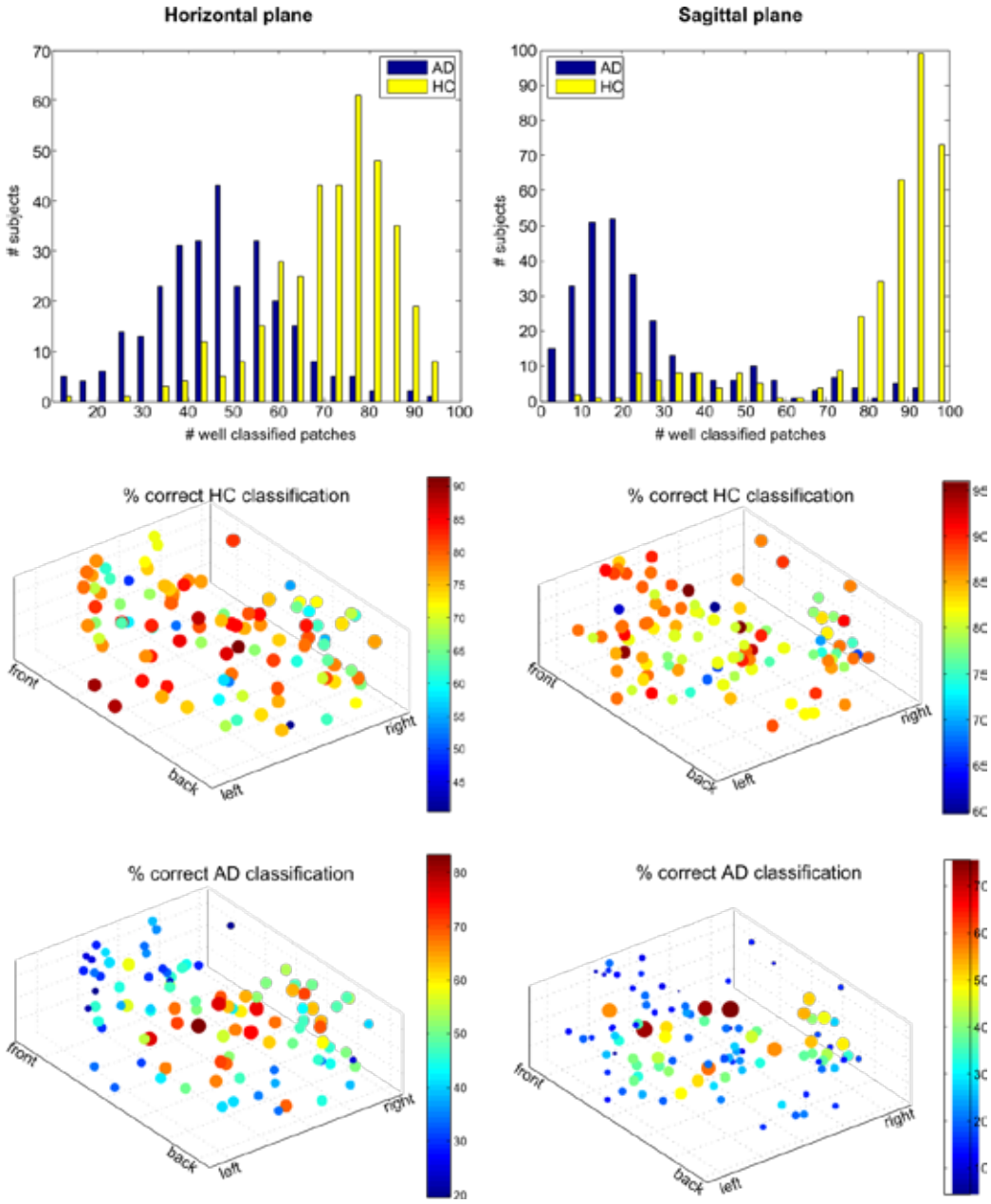

Figure 7: Results for Classification Based on Random Patches Used as Input Features

Histograms of well classified patches (top) per group. Location of patches showing the percentage of correct classifications for HC (middle) and AD (bottom). The size and colour of the dots refer to the percentage of correct classifications.

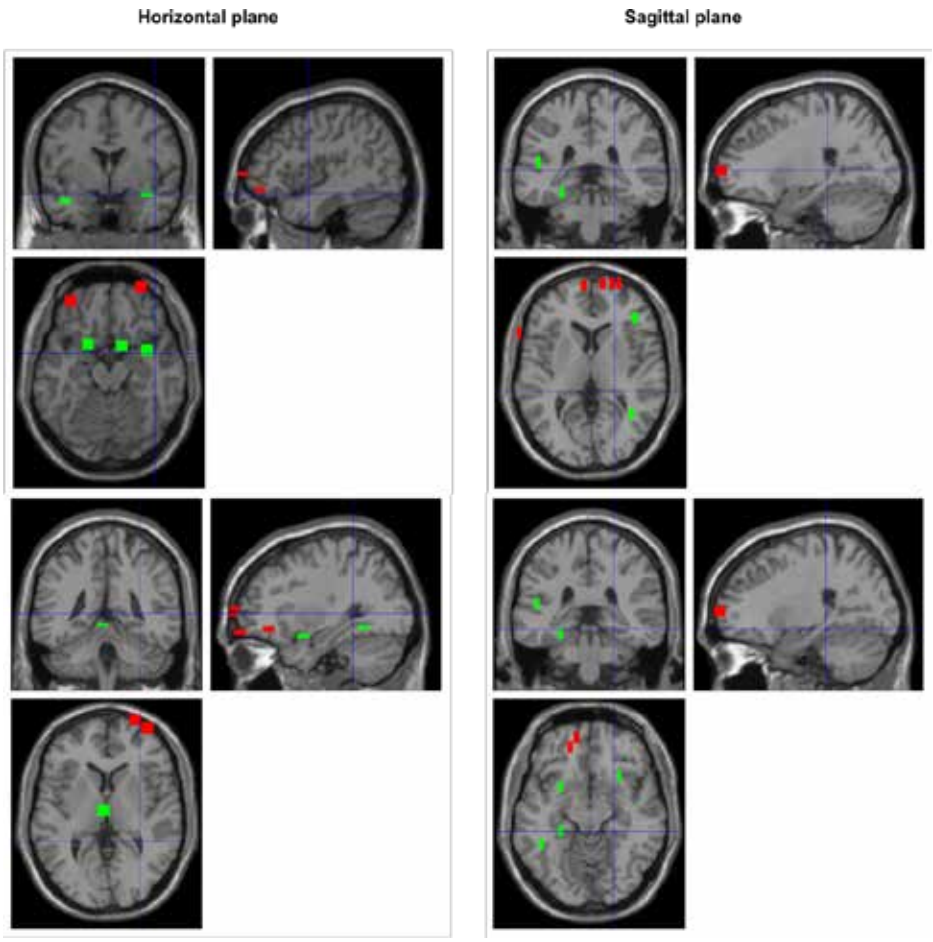

Figure 8: Location of the Patches that Perform Best (green) and Worst (red) for AD Classification

For clarity, only the central 7×7 voxels of the full 27×27 patches are displayed.

Conclusion: The results show that low-level features extracted from the scans without any preliminary knowledge, combined with a fairly easy deep learning scheme, has promising classification potential. Apart from being able to classify single subjects (which is not possible using statistical methods such as VBM), additional conclusions such as the definition of subgroups or finding brain regions affected by a disease.

3.2 Interaction with Medical Informatics Platform

- This Task relies on data accessible through the Medical Informatics Platform (MIP) WP8.2, including data on the longitudinal study of the large cohort of "control" Alzheimer's patients.
- This task will identify biological signatures of brain disease, and ultimately, MIP end-users will be able to query the construct. End-users will also be able to compare the derived biological signatures of disease to standard classification (e.g. ICD-10).
- The next step is to create comprehensive, simple and causal models of brain diseases that can be run with live data collected by the MIP. The model will be executed locally within each hospital (preserving privacy). This step will allow us to collect additional clinical data for the definition of the subgroups.

3.3 Interaction with other HBP Subprojects

This Task interacts with SP8 and SP5 to make available new disease ontologies, and to use the extent ontology for brain atlases.

3.4 Outreach

Abstracts submitted:

- Jing Cui, Sandrine Muller, Valérie Zufferey, Juergen Dukart, Ahmed Abdulkadir, Stefan Klöppel, Bogdan Draganski, Kutalik Zoltán, Richard Frackowiak, Ferath Kherif. Computation based diagnosis reveals intermediate Alzheimer's disease phenotypes. Joint Congress of European Neurology, Istanbul, Turkey, 31 May—3 June 2014. Accepted: oral presentation.
- Jing Cui, Sandrine Muller, Valérie Zufferey, Juergen Dukart, Ahmed Abdulkadir, Stefan Klöppel, Bogdan Draganski, Kutalik Zoltán, Richard Frackowiak, Ferath Kherif. Computation based diagnosis reveals intermediate Alzheimer's disease phenotypes: the follow up study. Organisation for Human Brain Mapping, Hamburg, Germany, June, 2014. Accepted: poster presentation.

Papers submitted/in preparation:

- Jing Cui, Sandrine Muller, Valérie Zufferey, Juergen Dukart, Ahmed Abdulkadir, Stefan Klöppel, Bogdan Draganski, Kutalik Zoltán, Richard Frackowiak, Ferath Kherif. In preparation: "Computation Based Diagnosis Reveals Intermediate Alzheimer's Disease Phenotypes."
- Jing Cui, Valérie Zufferey, Ferath Kherif. In press: "The Never-Ending Hunt for Neuroimaging Biomarkers".
- Bart Van Damme, Jing Cui, Bogdan Draganski, Ferath Kherif. In preparation: "MRI Feature Selection for Deep Learning Applied to the Classification of Alzheimer's Disease".

4. Future Computing (WP11.3)

4.1 T11.3.1: Neuromorphic Data Mining Systems

4.1.1 Application/Experimental Set-up and Results

Use Case	Data Set	Classification and clustering	Informatics-based model[1]	Algorithms and Benchmarks[2]	Comments
Predictive Maintenance	Model data	Classification and prediction	Network implemented in Python; PyNN implementation underway	Modified CMAC	See details in text
Hardware assisted NUMA job placement	Real SAP HANA workload traces - learning supervised by simulation tool	Classification	Classifier implemented in Python, classifier in PyNN, running on Neuromorphic HW available	Standard machine learning classifier like Naïve Bayes and Neuromorphic classifier implemented (M. SCHMUKER T11.3.6)	See details in text

[1]: Please refer also to the implementation status of network architectures in hardware description language

[2]: Please link to any kind of reference or description in text

Use case one: Predictive Maintenance

Introduction: Predicting when a piece of technical equipment, which is deviating or degrading from its normal operating conditions, will stop performing its intended function, is an important input into contingency planning. This time is usually known as the equipment's remaining useful life (RUL). In recent years, predicting the RUL of machines has been an active field. Technical approaches to building models for the degradation of equipment can be put into different categories, one of which is the data-driven approach. This relies purely on the analysis of (sensory) data, usually by applying pattern recognition and machine learning techniques to detect changes in system states.

Experiment Setup: The exponential behaviour of the fault evolution is common to almost all "Damage Propagation Models". Assuming that an upper threshold describes the operational limit, after which the system can no longer be used, a Health Index Function was defined.

The trained network was based on a modified CMAC algorithm. This algorithm quantised the input data, and then transformed them into an internal representation using a non-linear transformation. In practical implementations, the CMAC is usually based on an equivalent representation, and presented as a look-up table. However, our implementation actually uses a neural network structure. The modified mapping to the internal representation leads to a sparse distributed representation, and the connections (weights) to the output layer are trained to match the desired target signal.

Current State: The simulation environment is coded in Python, and initial tests to verify the validity of the approach have been performed. A PyNN implementation suitable for Neuromorphic use is currently being developed. It has not yet been finished, and therefore Milestone 210 needs to be delayed by one to two months.

Use case two: HW assisted NUMA job placement in a combined OLTP/OLAP in-memory database

Introduction: The SAP HANA in-memory database system is a very high performance system underlying most of SAP's newer solutions. Conventional database systems rely on a particular efficient structuring of the data for a given task (i.e. indexes, prebuilt aggregates and features). In contrast, SAP HANA allows multiple simultaneous usage scenarios on the same consistent data sets. This is achieved through the effective use of brute force algorithms (i.e. to locate an item, search all items; to provide an aggregate value, sum all values on each request). To make such a flexible brute force approach efficient, clever low-level implementation techniques are employed. One central challenge of modern computer hardware is the placement of jobs within a machine, i.e. a classical computer separates storage (RAM) and processing (CPU), with different access paths and bandwidths across interconnects between them. Dynamically scheduling jobs onto a modern machine with hundreds of cores and thousands of concurrent jobs is challenging. This is because optimal scheduling is NP-hard, and actual hardware exhibits dynamic behaviour that introduces uncertainty and noise to the optimisation parameters. It is therefore infeasible for software to perform at the required speeds.

Systems such as the "Omega" scheduling system at Google tackle these kinds of problems, but only for pure batch workloads, treating interactive jobs as isolatable error signals. With SAP HANA, we need to treat various kinds of jobs in a single integrated system at interactive (not batch) speeds. This is because HANA systems concurrently run anything from sub-millisecond interactive OLTP jobs, to multi-hour planning and optimisation runs. These are done simultaneously, in the same machine, and on the same datasets.

Experiment Setup: We placed workload mixes onto a machine and trained a naïve Bayesian classifier to make "optimal" placement decisions for incoming jobs. These were based on the job's properties, the placement of the data, and the current and predicted load of the machine's CPUs, interconnects and memory modules. To generate the supervision signal for training, we used two different approaches.

In the first approach, we used existing heuristic schedulers to place and orchestrate jobs, and had the classifier learn to mimic their behaviour. This, together with baseline approaches, such as round-robin or random schedules, provided the baseline against which we could measure improvement potential.

The second approach used an elaborate whole system model to calculate step-by-step optimal schedules, and provide these as supervision feedback to the Bayesian classifier. This cannot be done in a real machine, and therefore required the implementation of a simulation environment to enable closed-loop interaction between the supervisor and the Bayesian classifier.

Current State: The simulation environment is finished, and we are now in the process of running actual experiments.

Milestones at M18:

- MS210: implementation of network architectures in hardware description language.

The implementation has started but not yet finished. Therefore, the completion of this milestone is delayed by one to two months. As with the slight delay in the previous Task "network architectures evaluated", this delay should not lead to major problems, since

parts of the implementation can be tested when the Neuromorphic hardware platforms are released. For the second use case, we will use an available Neuromorphic classifier.

4.1.2 Interaction with the Neuromorphic Systems

The requirements imposed by the Predictive Maintenance scenario on Neuromorphic hardware are a combination of accuracy, processing speed and power efficiency, assuming that the processing will be de-centralised, i.e. close to the point of data creation, and not in a central backend. They requirements are:

- Data streams need to be processed in real-time (limited local buffering).
- Hardware variability and noise need to be controlled for the algorithm to work efficiently.

The NUMA scheduling and placement problem was selected for a potential hardware implementation with the Neuromorphic systems. This is because:

- The classifier needed is known to be efficiently implementable on the spiking neural chips.
- The complexity of the classifier very likely fits into the rather limited neuron budget of the early chips.
- The problem stresses the need for fast hardware response over accuracy. Scheduling the database jobs is a real-time task; otherwise query response times can be negatively impacted.

We can estimate the required throughput of the Neuromorphic scheduling and placement solution from our workload traces. We can also derive the required latency and throughput parameters for potential interfaces between the Neuromorphic hardware and the HANA system. These requirements far exceed those of biological systems, and hence benefit from the much-faster-than-biology timing of the Neuromorphic hardware. As a typical query can generate execution plans containing hundreds to a few thousand individual sub-jobs, a medium scale workload with 100 user queries per second needs to place 50000 jobs per second. Extreme use cases are heavy OLTP workloads with 3000—6000 user transactions per second, but typically there are much lower numbers of sub-jobs.

4.1.3 Interaction with other HBP Subprojects

- Discussions with Michael SCHMUKER (T11.3.6) about the classification algorithm.
- Discussions with Eduardo ROS (T11.1.1) on Cerebella algorithms.

4.2 Outreach

None.

4.3 T11.3.2: Port CABot3 to Neuromorphic Chips and Extend

4.3.1 Application/Experimental Set-up and Results

Task No.	Model [1]	PyNN Implementation [2]	Comments
T11.3.2	Spiking neurons	Runs on SpiNNaker Board	Late, but Milestone reached

[1]: e.g. model uses spiking neurons

[2]: PyNN running in simulation, on NM-MC, NM-PM, or ESS

This Task has managed to complete the agent and environment, and run the agent on SpiNNaker, with communication to the environment running on the attached PC. The 3D environment set-up is written in Python, and is also running on SpiNNaker. There are four rooms, connected by four corridors. Each room has a unique shape and colour (pyramid or stalactite, blue or red). The user types natural language commands into the environment, and those commands are sent to the SpiNNaker board.

The last component completed was cognitive mapping. The user types in “explore”, and the agent explores the environment, associating each room with the shape inside it. The user then issues a command, such as "go to the room before the red pyramid", and the agent moves around the environment and stops at the appropriate room. This shows that it has learned the map. Testing has not been thorough, but the experiment typically works. The agent views the environment from the picture on the camera. This picture is then sent to the SpiNNaker chips. The agent follows plans, the goals of which are set by the NL commands. There are four actions that can be used to move the agent (left, right, forward and back). These are sent from the board to the environment.

In summary:

- Milestone 308 has been achieved. There are 20 subnets working. Binding is still not working for the parser, and this will require the development of new synaptic models for SpiNNaker and PyNN.
- Progress has been made on learning visual objects in an environment, persistent Cell Assemblies (CAs), and caching plans.
- We still need to explore using HICANN in Stockholm.

4.3.2 Interaction With the Neuromorphic Platform

Our system uses the SpiNNaker platform (SP9, NM-MC1) in a closed loop experiment. As such, it is a test-bed for the system for robotics applications (SP10). It also is extensible, so future systems can use this as a template to start developing neuromorphic agents.

4.3.3 Interaction with other HBP Subprojects

We have spoken with the Neurorobotics group, and asked to use their virtual environment. Unfortunately, it is not ready yet, so we have implemented our own virtual environment.

4.3.4 Outreach

None.

4.4 T11.3.3: Exploitation of Feedback in Ultra-fast Spiking Visual Architectures

4.4.1 Application/Experimental Set-up and Results

Task No.	Model [1]	PyNN Implementation [2]
11.3.3.2	Spiking neurons n1, n2, n3 models	Running on NM-MC
11.3.3.3	Spiking neurons of n3 models	Running on NM-MC
11.3.3.5	Spiking neurons of n2 models	Running on NM-MC
11.3.3.7	Spiking neurons of n3 models	Running on NM-MC

[1]: e.g. model uses spiking neurons

[2]: PyNN running in simulation, on NM-MC, NM-PM, or ESS

[3]: If yes, please link to any kind of reference or description in text

During this period, the group focused on two main aspects:

- (a) More exhaustive PyNN descriptions for functions 11.3.3.2 (simulations for exhaustive optimisations), 11.3.3.3 (PyNN of feedback architectures), and 11.3.3.7 (PyNN for mismatch), while porting them onto the Neuromorphic Computing Platform for function 11.3.3.5.
- (b) Improving sensor interfaces, also for function 11.3.3.5.

Regarding (a), we have so far three separate PyNN basic descriptions of a ConvNet recognition system. Each of these uses a different neuron model, and is also developed for a different PACMAN (SpiNNaker) version. Each of the three PyNN basic descriptions can be used for functions 11.3.3.2, 11.3.3.3, 11.3.3.5, and 11.3.3.7. The three different neuron models are as follows:

- (n1) uses a single population model for each Feature Map in a ConvNet with instantaneous integrate-and-fire dynamics.
- (n2) neurons are modelled individually with instantaneous integrate-and-fire dynamics.
- (n3) neurons use a PyNN package integrate-and-fire model, slightly re-touched to allow for signed output events.

Regarding (b), we have preliminary results on a higher speed SpiNNaker-FPGA interface that would allow for a speed of up to six mega events per second for 32-bit events.

4.4.2 Interaction with the Neuromorphic Platform

We are already running our architectures on the NM-MC platforms. We have two four-chip SpiNNaker boards and one 48-chip SpiNNaker board. Most of our tests are being conducted on the 48-chip board. This allows us to run simulations and tests very quickly.

- For the feedback structure, we have started to use weak feedback connections from a later layer to an earlier layer. To assess the real benefits of feedback, we still need to make progress on the exhaustive optimisations of the fully feed-forward versions. So far, we are preparing the infrastructure to allow for feedback to be incorporated quickly.

4.4.3 Interaction with Other HBP Subprojects

Our group maintains a strong interaction with Univeristy of Manchester (UMAN – P73) SpiNNaker group. Two of our members recently attended the SpiNNaker Workshop in Manchester, and we have three SpiNNaker boards in our lab (two four-chip ones and one 48-chip one). We also collaborate with the group at *Université Pierre et Marie Curie - Paris 6* (UPMC – P107) on computational aspects of event-driven systems for vision.

4.4.4 Outreach

LA Camuñas-Mesa, T Serrano-Gotarredona, B Linares-Barranco, "Event-driven sensing and processing for high-speed robotic vision, *Biomedical Circuits and Systems Conference* (BioCAS), 2014, IEEE, 516—519.

T Iakymchuk, A Rosado, T Serrano-Gotarredona, B Linares-Barranco, et al., "An AER handshake-less modular infrastructure PCB with x8 2.5 Gbps LVDS serial links" International Symposium on Circuits and Systems (ISCAS), 2014, IEEE, 1556—1559.

4.5 T11.3.4: Spiking Associative Networks for Neuromorphic Computing Systems

4.5.1 Application/Experimental Set-up and Results

Task No.	Model [1]	PyNN Implementation [2]	Benchmarks evaluated [3]	Comments
11.3.4.1	Binary SAM	PyNN/Nest simulations	12	None
11.3.4.1	Analog SAM	PyNN/Nest simulations	4	None
11.3.4.4	Binary SAM	PyNN/ESS	3	None

[1]: e.g. model uses spiking neurons

[2]: PyNN running in simulation, on NM-MC, NM-PM, or ESS

[3]: If yes, please link to any kind of reference or description in text

In the first year of our project, the group implemented two spiking associated memory (SAM) models in PyNN (Function 11.3.4.1, KPI=2), and conducted many simulations with different parameter sets. To analyse the simulations, we implemented automatic performance evaluation tools (11.3.4.2), and generated the first benchmark data set (11.3.4.3, KPI=1). To prepare for the use of the Neuromorphic Platforms, we implemented the first SAMs on the Heidelberg virtual hardware emulator ESS (11.3.4.4). With the successful mapping of SAM models on the virtual hardware ESS, we fulfilled our first Milestone on time. The same holds for the planned KPIs. In summary, the project is on schedule.

4.5.2 Interaction with the Neuromorphic Platform

Following the successful mapping of SAM models on the Heidelberg hardware emulator, we are now prepared for the use of the Neuromorphic Platform NM-PM in Heidelberg. Because of the relatively long simulation times for NEST and ESS, use of the Neuromorphic Platforms is mandatory in respect to the planned parameter space explorations for this project. Therefore, we have a close cooperation with the Heidelberg group (neuromorphic system NM-PM1) at present, and will intensify our cooperation with the Manchester group (neuromorphic system NM-MC1) in the second half of the project.

4.5.3 Interaction with other HBP Subprojects

We interact with SP3, SP4 and SP6, in order to have a theoretical and biological basis for our spiking associative memory simulations.

4.5.4 Outreach

As our project is only in its first year, we have just started to publish the first results.

4.6 T11.3.5 (UMPC): Asynchronous Computational Retina

4.6.1 Application/Experimental Set-up and Results

Four "hardware" platforms are scheduled for the end of Month 19. These hardware implementations have been achieved:

- Interface to connect one ATIS camera into SpiNNaker (function 11.3.5.1): we are able to transmit and process events provided by an ATIS camera. This result has been accepted as a Live Demonstration that will be presented at the IEEE International Symposium on Circuits and Systems (ISCAS) 2015 (Garrick ORCHARD, Xavier LAGORCE, Christoph POSCH, Steve FURBER, Ryad BENOSMAN).
- Interface to connect two ATIS cameras into SpiNNaker (function 11.3.5.2): we are able to transmit and process events provided by two ATIS cameras. The maximum input flow rate has been estimated to 1.4 million events per second, per camera.
- Stimulation platform (function 11.3.5.3): a table XY allows us to automatically acquire datasets of different features with one or more ATIS cameras, with different motions and speeds. The platform is able to reach more than 1 m/s. Above this speed, the data flow is not useful, because the sensor tends to saturate. This platform will be described in an article we expect to submit into a Special Issue of *Frontiers of Neurosciences* about "Benchmarks and Challenges for Neuromorphic Engineering".
- Database platform (function 11.3.5.4): several data flows have been acquired for benchmarking using the previously described platform. In addition, data flows have been acquired in natural environment conditions. A software interface has been developed under Matlab to allow users to visualise these data, and analyse them in terms of activity. This is the main criteria we will use. Again, we are currently working on an article using a part of this dataset for a Special Issue of *Frontiers of Neurosciences* about "Benchmarks and Challenges for Neuromorphic Engineering".

Reports on these platforms will be provided as scheduled at the end of Month 19.

4.6.2 Interaction with the Neuromorphic Platform

Several tasks have been carried out to prepare for the implementation of our algorithm models into the SpiNNaker electronic board. Several articles present these preliminary results:

Galluppi, F., Lagorce, X., Stromatias, E., Pfeiffer, M., Plana, L.A., Furber, S.B., AND Benosman, R. (2015). A framework for plasticity implementation on the SpiNNaker neural architecture. Frontiers in Neuroscience, 8, 429.

Lagorce, X., Stromatias, E., Galluppi, F., Plana, L.A., Liu, S.-C., Furber, S.B., & Benosman, R.(2015). Breaking The Millisecond Barrier On SpiNNaker : Asynchronous Event-Based Models With Microsecond Resolution And Plasticity. Frontiers in Neuroscience (under review).

Himanshu Akolkar, Cedric Meyer, Xavier Clady, Olivier Marre, Chiara Bartolozzi, Stefano Panzeri, Ryad Benosman (2015), What Can Neuromorphic Event-Driven Precise Timing Add to Spike-Based Pattern Recognition?, Neural computation, March 2015, Vol. 27, No. 3, Pages 561—593.

The two first articles describe how to implement neuron models and precise timing based computations into the SpiNNaker's board. The third article explains how precise timing-based computation provided by a neuromorphic platform can increase performance results in high-level tasks, such as pattern recognition. Therefore, it demonstrates the usefulness

of an event-based and precise timing-based extraction of low-level information, such as we propose to do in this Task.

4.6.3 Interaction with Other HBP Subprojects

Our team is constantly interacting with Steve FURBER's team (SP9, SP13) about the SpiNNaker implementation, as reflected in the two first publications cited above, and in the ISCAS 2015's Live Demonstration. In addition, the work on the usefulness of precise timing based computations [19] was carried out in collaboration with Olivier MARRE (HBP CLAP project, SP4).

4.6.4 Outreach

Publications:

Galluppi, F., Lagorce, X., Stromatias, E., Pfeiffer, M., Plana, L.A., Furber, S.B., And Benosman, R. (2015). "A framework for plasticity implementation on the SpiNNaker neural architecture". *Frontiers in Neuroscience*, 8, 429.

Himanshu Akolkar, Cedric Meyer, Xavier Clady, Olivier Marre, Chiara Bartolozzi, Stefano Panzeri, Ryad Benosman (2015), "What Can Neuromorphic Event-Driven Precise Timing Add to Spike-Based Pattern Recognition?", *Neural computation*, March 2015, Vol. 27, No. 3, Pages 561—593.

Publications under review:

Lagorce, X., Stromatias, E., Galluppi, F., Plana, L.A., Liu, S.-C., Furber, S.B., & Benosman, R.(2015). "Breaking The Millisecond Barrier On SpiNNaker: Asynchronous Event-Based Models With Microsecond Resolution And Plasticity". *Frontiers in Neuroscience*

Live Demonstration:

Garrick Orchard, Xavier Lagorce, Christoph Posch, Steve Furber, Ryad Benosman. Live Demonstration: Real-Time Event-Driven Object Recognition on SpiNNaker, ISCAS 2015.

4.7 T11.3.6: Implementing a Spiking Classifier Network on HiCANN

4.7.1 Application/Experimental Set-up and Results

Task No.	Model [1]	PyNN Implementation [2]	Benchmarks evaluated [3]	Comments
11.3.6	eNose Classifier on GeNN	No	N/A	
11.3.6	MNIST classifier on GeNN	No	N/A	
11.3.6	MNIST on SpiNNaker	Yes		Testing phase

[1]: e.g. model uses spiking neurons

[2]: PyNN running in simulation, on NM-MC, NM-PM, or ESS

[3]: If yes, please link to any kind of reference or description in text

The team completed work with the GPU-based neuromorphic classifier to classify continuous olfactory data (eNose sensor recording), and prepared and submitted a full journal paper to IEEE Transactions on Neural Networks and Learning Systems (TNNLS) covering this work. The paper is titled: "A GPU-based neuromorphic classifier for chemosensing applications.

We then adapted the classifier design, model parameters and code base to classify the full (10-digit) MNIST high dimensional dataset, comprising 60 000 training examples and 10 000 testing samples. The adapted classifier was again implemented using GPU enhanced neuronal networks (GeNN). We are planning to submit the results as part of an abstract to the September 2015 Brain Informatics and Health conference. We plan to describe and demonstrate the application of the Sussex GPU neuronal simulation software (GeNN) to a large-scale data problem.

Our next objective was originally to port the classifier for large, high-dimensional problems to the HICANN chip. However, due to the fact that the HICANN chip and the software infrastructure is still heavily evolving, we decided to first target the SpinNNaker architecture, which is readily useable in its current state. We have borrowed two small SpiNNaker boards from UMAN, with the view to borrow a large board when our application software is ready.

We subsequently implemented and tested a prototype PyNN model of the classifier design for use on SpiNNaker boards. Most recently, we have begun implementing a 10 digit MNIST classifier test on SpiNNaker, with promising initial results.

KPIs:

- 11.3.6_sKPI_001: Approximately 600 checked in code revisions on software management system.
- 11.3.6_sKPI_002: Number of neurons used in the software model. The largest model used for the MNIST classifier on GeNN uses around 25,000 neurons
- 11.3.6_sKPI_003: Number of neurons used in the Spikey/HICANN hardware model. No HICANN access as of yet; for the SpiNNaker we have a model functional at around 2000 neurons and anticipate this reaching at least 6000 before needing to upgrade to a larger board.

- 11.3.6_sKPI_004: Number of neurons used on the wafer-scale hardware model. Not implemented yet.
- 11.3.6_sKPI_005: Number of classification problems to which the classifier network is applied. The developed classifier has been applied to four problems (eNose static, eNose continous, IRIS and MNIST 10 digit).

4.7.2 Interaction with the Neuromorphic Platform

We have two small (four-chip) SpiNNaker boards on loan, and are using them for our classifier system. We are planning to benchmark on a larger SpiNNaker board soon. We have access to the Spikey hardware in Heidelberg via remote login. We plan to obtain a Spikey-USB-board. We are still awaiting access to the HICANN. We are in constant email contact with the SpiNNaker and Spikey developers for feedback and feature requests.

4.7.3 Interaction with other HBP Subprojects

Dr. Diamond attended a five-day workshop at the University of Manchester on developing neuronal simulations using SpiNNaker. Michael SCHMUKER is providing a Benchmark (multivariate classification on Spikey) within Task 9.3.4 (SP9).

4.7.4 Outreach

Dr. Diamond attended a workshop for electronics-related researchers in the UK (eFutures). There he discussed current work and possible synergies with researchers form other UK institutions. Michael Schmucker presented his work on Spikey at the HBP Summit in October 2014. The results were also presented at a workshop at the Bernstein Conference 2014, as part of a workshop presentation by T. Pfeil, UHEI, and in poster form.

5. References

[1] Baccus, S. A., & Meister, M. (2002). Fast and slow contrast adaptation in retinal circuitry. Neuron, 36(5), 909-919.

[2] Ozuysal, Y., & Baccus, S. A. (2012). Linking the computational structure of variance adaptation to biophysical mechanisms. Neuron, 73(5), 1002-1015.

[3] Smith, V. C., Pokorny, J., Lee, B. B., & Dacey, D. M. (2001). Primate horizontal cell dynamics: an analysis of sensitivity regulation in the outer retina. Journal of Neurophysiology, 85(2), 545-558.

[4] Lee, B. B., Dacey, D. M., Smith, V. C., & Pokorny, J. (2003). Dynamics of sensitivity regulation in primate outer retina: The horizontal cell network. Journal of Vision, 3(7), 5.

[5] Lee, B. B., Shapley, R. M., Hawken, M. J., & Sun, H. (2012). Spatial distributions of cone inputs to cells of the parvocellular pathway investigated with cone-isolating gratings. JOSA A, 29(2), A223-A232.

[6] Crook, J. D., Manookin, M. B., Packer, O. S., & Dacey, D. M. (2011). Horizontal cell feedback without cone type-selective inhibition mediates "red-green" color opponency in midget ganglion cells of the primate retina. The Journal of Neuroscience, 31(5), 1762-1772.

[7] Crook, J. D., Davenport, C. M., Peterson, B. B., Packer, O. S., Detwiler, P. B., & Dacey, D. M. (2009). Parallel ON and OFF cone bipolar inputs establish spatially coextensive receptive field structure of blue-yellow ganglion cells in primate retina. The Journal of neuroscience, 29(26), 8372-8387.

[8] Cao, Y. & Grossberg, S. (2005). A laminar cortical model of stereopsis and 3D surface perception: Closure and da Vinci stereopsis. Spatial Vision, 18, 515-578.

[9] Cao, Y. & Grossberg, S. (2012). Stereopsis and 3D surface perception by spiking neurons in laminar cortical circuits: A method of converting neural rate models into spiking models. Neural Networks, 26, 75-98.

[10] Brookmeyer, R., Johnson, E., Ziegler-Graham, K. & Arrighi, H.M. (2007). Forecasting the global burden of Alzheimer's disease. Alzheimers Dement. 3(3), 186-91. doi: 10.1016/j.jalz.2007.04.381.

[11] Jack, C.R. Jr., Knopman D.S., Jagust W.J., Petersen R.C., Weiner M.W., Aisen P.S., Shaw L.M., Vemuri P., Wiste H.J., Weigand S.D., Lesnick T.G., Pankratz V.S., Donohue M.C. & Trojanowski J.Q. (2013). Tracking pathophysiological processes in Alzheimer's disease: an updated hypothetical model of dynamic biomarkers. Lancet Neurol., 12(2), 207-16. doi: 10.1016/S1474-4422(12)70291-0.

[12] Beach, T.G., Monsell, S.E. Phillips, L.E. & Kukull , W. (2012). Accuracy of the clinical diagnosis of Alzheimer disease at National Institute on Aging Alzheimer Disease Centers, 2005-2010. J Neuropathol Exp Neurol. 71(4), 266-73. doi: 10.1097/NEN.0b013e31824b211b.

[13] Larochelle, H., Bengio, Y., Louradour, J. & Lamblin P. (2009). Exploring strategies for training deep neural networks. The Journal of Machine Learning Research 10, 1-40.

[14] Le, Q.V., Ranzato, M.A., Monga, R., Devin, M., Chen, K., Corrado, G.S., Dean, J. & Ng, A.Y. (2012). Building High-level Features Using Large Scale Unsupervised Learning. International Conference in Machine Learning (2012).

[15] Salakhutdinov & Hinton, 2009

[16] Deng et al., 2013

[17] Hinton et al., 2012

[18] Yu & Deng, 2011

[19] Himanshu Akolkar, Cedric Meyer, Xavier Clady, Olivier Marre, Chiara Bartolozzi, Stefano Panzeri, Ryad Benosman (2015), "What Can Neuromorphic Event-Driven Precise Timing Add to Spike-Based Pattern Recognition?", *Neural computation*, March 2015, Vol. 27, No. 3, Pages 561—593.

[20] Suk, Heung-Il; Shen, Dinggang; "Deep learning-based feature representation for AD/MCI classification", *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2013*, 2013, 583—590.

Annex A: Milestones

No.	Milestone Name	WP	Month Due	Month Achieved
MS200	Preparation of simplified virtual sensor and motor models for virtual robots	11.1	6	6
MS208	Specification of benchmark tasks	11.3	6	6
MS201	Experimental design for the first experimental task	11.1	12	12
MS204	Standardised description format for biological signatures of brain disease	11.2	12	12
MS209	Evaluation of suitable network architectures for benchmark tasks	11.3	12	12
MS310	Initial sensor interfaces operative	11.3	12	12
MS315	Acquisition and interface platforms and database	11.3	12	20 est.
MS308	CABot3 on SpiNNaker	11.3	15	18
MS202	Implementation of virtual robot, environment and experiment complete	11.1	18	22 est.
MS205	First draft informatics-based model generating a biological signature of a disease	11.2	18	18
MS210	Implementation of network architectures in hardware description language.	11.3	18	20 est.

Annex B: Scientific Key Performance Indicators

View SP11's Key Performance Indicators (KPIs) on the KPI webpages hosted by the Science and Technology Office.