

Adaptation and Contextualization of Deep Neural Network Models

Dimitrios Kollias
*School of Electrical & Computer
Engineering
National Technical University of
Athens*
Athens, Greece
dkollias1@gmail.com

Miao Yu
*School of Computer Science
University of Lincoln
Lincoln, United Kingdom*
myu@lincoln.ac.uk

Athanasios Tagaris
*School of Electrical & Computer
Engineering
National Technical University of
Athens*
Athens, Greece
thanostagaris@gmail.com

Georgios Leontidis
*School of Computer Science
University of Lincoln
Lincoln, United Kingdom*
gleontidis@lincoln.ac.uk

Stefanos Kollias
*School of Computer Science
University of Lincoln
Lincoln, United Kingdom*
skollias@lincoln.ac.uk

Andreas Stafylopatis
*School of Electrical & Computer
Engineering
National Technical University of
Athens*
Athens, Greece
andreas@cs.ntua.gr

Abstract—The ability of Deep Neural Networks (DNNs) to provide very high accuracy in classification and recognition problems makes them the major tool for developments in such problems. It is, however, known that DNNs are currently used in a ‘black box’ manner, lacking transparency and interpretability of their decision-making process. Moreover, DNNs should use prior information on data classes, or object categories, so as to provide efficient classification of new data, or objects, without forgetting their previous knowledge. In this paper, we propose a novel class of systems that are able to adapt and contextualize the structure of trained DNNs, providing ways for handling the above-mentioned problems. A hierarchical and distributed system memory is generated and used for this purpose. The main memory is composed of the trained DNN architecture for classification/prediction, i.e., its structure and weights, as well as of an extracted - equivalent - Clustered Representation Set (CRS) generated by the DNN during training at its final - before the output - hidden layer. The latter includes centroids - ‘points of attraction’ - which link the extracted representation to a specific area in the existing system memory. Drift detection, occurring, for example, in personalized data analysis, can be accomplished by comparing the distances of new data from the centroids, taking into account the intra-cluster distances. Moreover, using the generated CRS, the system is able to contextualize its decision-making process, when new data become available. A new public medical database on Parkinson’s disease is used as testbed to illustrate the capabilities of the proposed architecture.

Keywords—Deep neural networks, clustered representation sets, classification, prediction, adaptation, contextualization, Parkinson’s disease

I. INTRODUCTION

Current signal processing and analysis of multimodal data, including visual data and medical images, is based on feature extraction, segmentation, followed by quantitative and qualitative analysis of the features and segments. Recent advances in machine learning and deep neural networks provided state-of-the-art performance in all significant signal processing tasks, being used in a large number of applications, ranging from healthcare and question answering systems, to human computer interaction, surveillance and defense [4-6]. Deep neural networks are also applied as end-to-end-architectures which include different network types in their structure and are trained to analyse signals, images, text and other inputs [3, 4]. A recent survey on the use of DNNs for medical image analysis can be found in [2]. However, DNNs lack efficient on-line adaptation capability, as well as transparency and interpretability of their decision making. This makes their use difficult in fields such as healthcare, industrial monitoring and defence, where safety & privacy are key issues. Extracting and using latent variables, for example, from stacked, or variational autoencoders [4, 9] can be a way to tackle these problems, which constitute open areas of research in deep learning [10].

This paper proposes a class of novel deep neural architectures, which are able to perform trusted analysis of complex data. Prediction of Parkinson’s disease from medical images is used as the testbed application, based on a new public dataset of medical images we have been generating. The architectures are trained to predict the status of subjects, being able to efficiently adapt to new patient-specific cases. On-line

deep neural network adaptation for personalized prediction and disease diagnosis is a difficult and computationally inefficient problem, since it requires retraining of the deep neural network, or fine-tuning of it, the latter causing the well known catastrophic forgetting problem. In our approach, we combine the deep neural network with clustering of representations extracted from the trained network, showing that this provides the possibility to keep former knowledge, while including new clustered information in it.

Our targeted objectives are the following:

- (i) designing novel end-to-end system architectures, incorporating both deep neural networks, either convolutional (CNN), or convolutional - recurrent neural networks (CNN-RNN) and clustered representation sets (CRS) extracted from the trained deep networks;
- (ii) using these architectures for efficient visual data analysis, prediction and/or classification;
- (iii) designing a novel adaptation methodology, which permits learning of new data cases, without forgetting the previous system knowledge and expertise;
- (iv) obtaining contextualization and transparency, by analyzing the CRS system provided information; thus, enriching the representations and decisions with domain knowledge;
- (v) demonstrating applicability of the above concepts and developments on a significant medical application field, i.e., predicting Parkinson's disease based on subjects' MRI and DaT Scans and respective clinical data.

The rest of the paper is organized as follows: Section II defines the problem we are facing and presents the design of the new system architecture. Section III describes the operation of the proposed systems. Treatment of drift detection and system adaptation, as well as contextualization are described in Section IV. Section V presents the experimental study on Parkinson's dataset. Conclusions and future work are presented in Section VI of the paper.

II. DESIGN OF SYSTEM ARCHITECTURE

The proposed methodology is based on the following principles:

Let us assume, that we have a large set of data collected from subjects, including images, video frames, textual data I , $I = \{I_1, \dots, I_n\}$, with corresponding annotations, $\{d_1, \dots, d_n\}$ and wish to solve a related classification/prediction problem. Let us denote by: D , the (end-to-end) Deep Neural Architecture, say a convolutional (CNN), or convolutional - recurrent (CNN-RNN) neural system, which is trained to provide the best performance on the above dataset; w , a vector, containing all the weights of the trained DNN system; $y = [y_1, \dots, y_n]$, a vector with all DNN outputs; $u = [u_1, \dots, u_n]$ a vector with the DNN last (before the output) hidden layer neurons' outputs.

In CNN-RNN architectures, each y_i is generally computed through a *tanh* activation function operation on u_i (a soft max one is usually adopted in CNN cases). If the network achieves a very high accuracy, this means that the CNN-RNN model has managed to almost achieve linear separation of the computed u_i

representations, over all input data. Thus, we may use the values of either, y , or u vectors, as outputs, or DNN derived representations, in the classification/prediction process.

In particular, the y outputs provide the predictions, or classification decisions, made by the DNN system. In addition to this, by analyzing the u vectors in the respective multidimensional space, we can provide hierarchical clustered representation sets (CRS), which are better suited for visualizing, adapting, contextualizing and semantically interpreting the generated by the DNN knowledge.

This is the procedure which we follow next to design the proposed system architecture. We target to minimize a global objective function, for learning the CRS which possesses the highest expressivity, as far as semantic information is concerned:

$$\underset{c,s}{\operatorname{argmin}} L(c, s/w, I) \quad (1)$$

where L is a loss function, such as the Mean-Squared-Error of the distances of each data sample from the targeted categories and the respective cluster centers; c denotes the cluster ids for all input data; s denotes the semantic information to be extracted from the CRS, given the input data and the DNN learnt weight values.

We propose a hierarchical (agglomerative) clustering approach, in which we first assume that semantics are fixed, based on existing domain knowledge about the problem and tackle CRS as a conventional clustering problem [11]; then we extract higher level semantic information from the CRS.

It should be added that the derived CRS can be extended in a one-shot manner, whenever adaptation to new data cases is required. This is done by adding, a new data element, either in a specific cluster, or as a new cluster centroid. The decision on this will be taken, by comparing the distance, of the corresponding DNN representation from each existing cluster center, to the maximum intra-class distance of the cluster.

A drift detection approach is then derived, by examining whether the above defined distances are larger than all cluster intra-class distances. In drift cases, appropriate adaptation of the original DNN architecture can also be performed, as described in [1,3], through re-training, using the extracted CRS and minimizing the following criterion:

$$\underset{w}{\operatorname{argmin}} L(w/I, c, s) \quad (2)$$

The proposed system architecture is shown in Fig. 1. It contains the Primary System Memory (PSM) which is mainly used during normal system operation, providing the current focus of attention based on the existing system knowledge.

The DNN – its structure and the w weight vector - forms the classification component of the PSM. The CRS centroids and intra/inter cluster distances, with the respective input data, are also included in the PSM, forming the basis of efficient system adaptation and contextualization. The members of each clustered set, with the respective input data, are stored and then

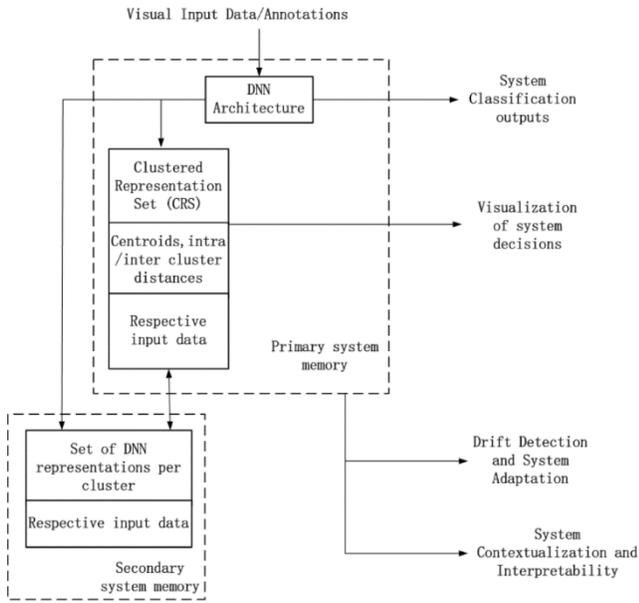


Fig. 1 The Proposed System Architecture

retrieved within the secondary, possibly distributed, system memory (SSM).

III. SYSTEM IMPLEMENTATION AND OPERATIONAL PHASE

In the following we describe the implemented system operation and decision making process. The technologies we use to derive the DNN architecture include:

- Deep Convolutional Neural Networks: Deep CNNs are architectures that try to exploit the spatial structure of the input information [4]. They have been used with great success in various applications, including image analysis, vision, object and emotion recognition. The most successful CNN was used for classifying millions of images in a 1000 classes [5].
- Transfer Learning: Transfer learning [6] is the main approach to avoid learning failure due to overfitting, when training complex CNNs with small amounts of (image) data. In transfer learning, we use networks previously trained with large image datasets (even of generic objects) and fine-tune the whole, or parts of them, using the small training datasets.
- Recurrent Neural Networks: RNNs are very powerful for processing sequential data [7]. A very successful model, the Long Short-Term Memory (LSTM) [8], uses hidden units with gates that explicitly control data flow in terms of both hidden states and inputs. Bidirectional (B-LSTM) models are obtained by combining forward and backward processing of input data. Gated Recurrent Units (GRUs) [12] have also been used for providing best RNN system performances.

The DNN systems that we use are end-to-end architectures including both CNNs and RNNs. CNNs derive rich internal representations from input data; B-LSTM/GRU RNNs correlate/analyze time evolution of the inputs, providing the final classification. CNNs can have the basic structure of the so called VGG-16 network [19], consisting of 13 convolutional, 5 pooling and 3 FC (fully connected) layers, or of the Deep Residual Net with 50 layers [20]; neurons have the non-linear activation (Relu) function.

Figure 2 shows the targeted system operation. Input data are fed into the Primary System Memory, where DNN-based classification and CRS-based analysis, including drift detection are implemented. As a consequence, both vectors y and u are calculated for these data. The y values provide the system classification outputs to be subsequently checked and validated. The u values are used to compute distances from the existing CRS centroids, taking into account the stored intra-cluster distances, so as to decide whether to link this data to a specific cluster - through a nearest neighbor criterion -, or else to detect a drift in the input data.

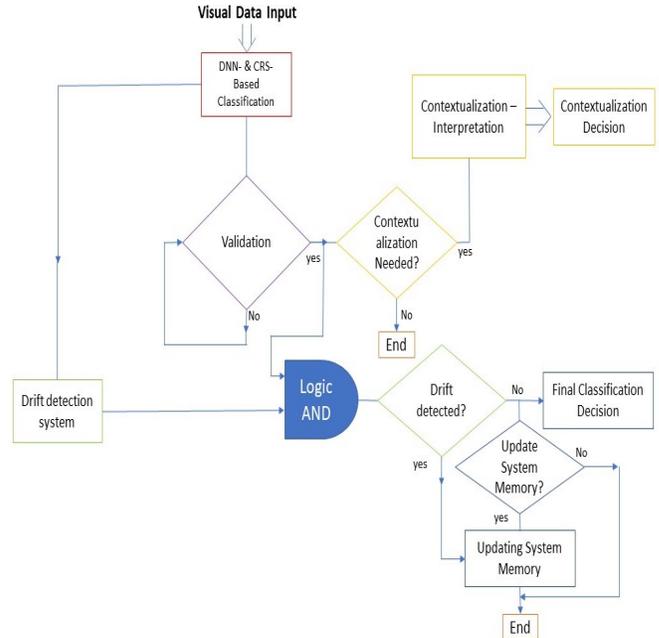


Fig. 2 The Operation Phase of the Generated System

Indexing the respective secondary memory, i.e., matching specific members of the CRS, is possible, if deeper analysis is deemed necessary.

After validation of the derived results, either automatically, based on evaluation of DNN and CRS outputs, or by the system operator/expert:

- if needed, contextualization of the made decisions is implemented;
- if drift is detected, adaptation of clusters' characteristics is made to the new context;

- adaptation of the DNN weights is also possible, whenever this is decided.

In case of no drift detection, the outcomes of the DNN and CRS classification subsystems are fused to provide the final classification decision.

Fig. 3(a) shows the CNN part of the system, indicatively showing a linear FC3 layer, if continuous output estimation is targeted. The CNN feeds the RNN part with the neuron outputs of its second FC layer (F). The RNN accepts $F_1, F_2, F_3, \dots, F_N$ and provides predicted/classified values $O(1), \dots, O(N)$ through time, at its output, as shown in Fig. 3(b). We perform transfer learning of the weights of the convolutional and pooling part of VGG or ResNet network to our neural architecture. This part is then fixed and training focuses on remaining FC layers and the RNN part.

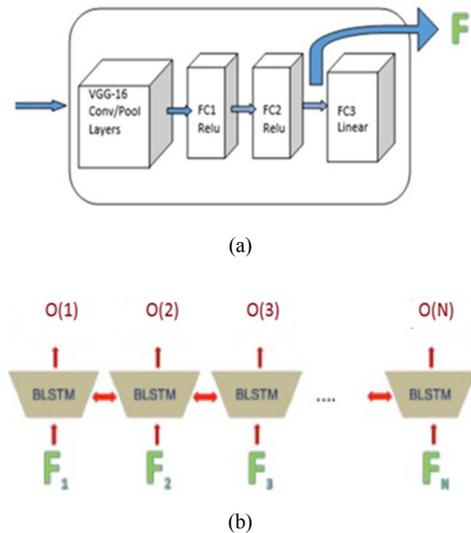


Fig. 3 The end-to-end CNN-RNN architecture: (a) CNN FC layer outputs feed the (BLSTM) RNN part (b) the RNN provides the final system predictions

We have used the Tensor Flow Platform as the main tool for generating software implementation of the presented architecture.

IV. SYSTEM ADAPTATION AND CONTEXTUALIZATION

In the following, let us first focus on drift detection and system adaptation, when new input data are processed and cannot be correctly classified based on system's prior knowledge.

This scenario is illustrated in Figure 4. Whenever all distances of the new data from the existing CRS centroids are larger than a selected threshold, then a drift is detected. In this case the new data are directly inserted, usually in the form of a new cluster, in the CRS (in the primary and respective secondary memory). This slightly extends CRS and updates some of its characteristics, without forgetting its former structure and characteristics.

Adaptation of the DNN classifier is also possible, if considered necessary by the expert/user of the considered application. If this is not the case, the specific data are stored in a temporary memory for future retraining of the CNN, or CNN-RNN classifier. Whenever such a DNN retraining is performed, this is done through an approach which extends the work described in [1-3]. In particular, DNN retraining is performed, using, on the one hand, the input data corresponding to the clusters' centroids (Existing Knowledge) and, on the other hand, the new data. Following this retraining procedure, we avoid the DNN forgetting problem, which occurs when we repeatedly fine-tune them to new data cases.

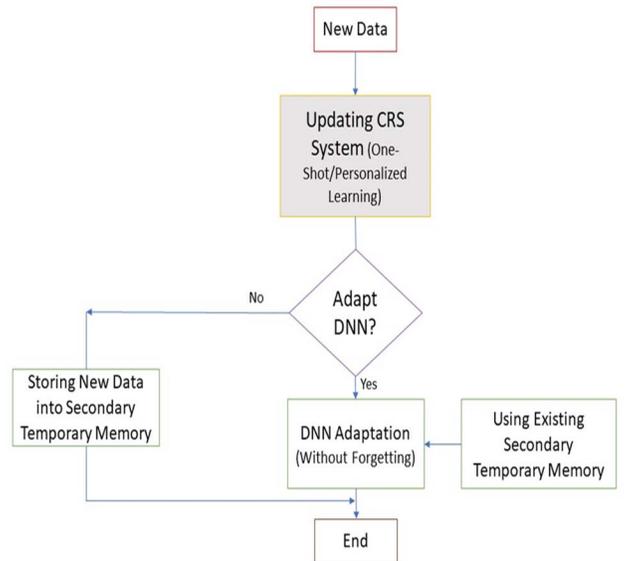


Fig. 4 System Adaptation

Let us now move to system contextualization. In the current paper we use the input data corresponding to the centroid of the cluster in which the new data is classified - based on the nearest neighbor criterion - for contextualization and interpretation purposes. In particular, we assume that the centroid input data represent the semantics related to all data belonging to the specific cluster. Consequently, we justify the classification our system provides on the new data, by showing the input data of the respective cluster centroid.

Our target is to match the specific representation data vector produced by the DNN and clustered according to the CRS, with semantic information provided by experts, or being a-priori known in the specific application. This is illustrated in the provided experimental study on Parkinson's. Furthermore, existing domain knowledge, in the form of knowledge representation or rules, can be combined with the Clustered Representation Sets for this purpose [13-15]. In our future research, the knowledge base of characteristics per derived cluster will be the basis for contextualizing the provided decisions, thus, rendering its use further transparent and trustful.

V. EXPERIMENTAL STUDY

Parkinson's, is a neurodegenerative disease which develops progressively due to the lack of dopamine in patients' brains. It is one of the most common neurodegenerative disorders which usually starts between the ages of 50 to 70 years. In countries with ageing population, as in most EU countries and in the US, the number of patients is expected to triple during the next 50 years. Although there is no definite treatment for Parkinson's disease, the early detection and appropriate management may highly improve the quality of patients' lives.

The presented approach is based on a large database related to Parkinson's disease, which is currently populated and includes the following: Magnetic Resonance Images (MRI) of the brain; images obtained through scintigraphy with 123-iodoflupane; different rating scales [16-17], including a binary classification rating between subjects with Parkinson's and non-patients.

All data are anonymous and, when completed, the database will refer to about 100 patients with Parkinson's disease and 40 people with other neurological diseases

This database is being constructed, based on collaboration of the Intelligent Systems Lab of the National Technical University of Athens, with the Department of Neurology, Georgios Gennimatas General Hospital in Athens, Greece.

Most than 50% of the database is currently available at [18] and constitutes the testbed of the developments presented in this Section, including subjects' MRI, DaT Scans and clinical data. In particular, it contains:

- MRI data: These are constructed as shown in Fig. 5 and analyzed to identify variations in corpus striatum, caudate nucleus, putamen, substantia nigra, thalamus, cerebellum, frontal cortex and hippocampus.

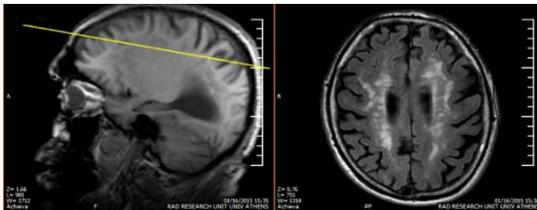


Fig. 5 A frame of an axial T1 sequence from a brain MRI (right)

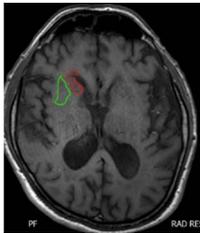
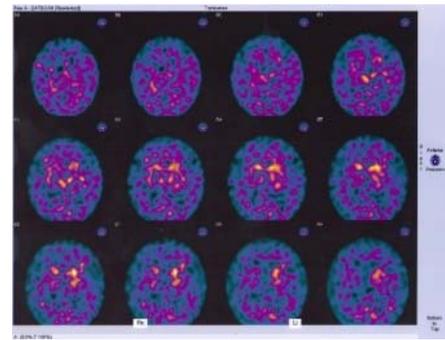


Fig. 6 An image from an axial T1 sequence

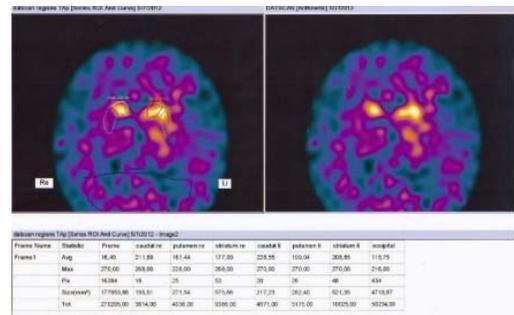
Of special interest is the volume of the surface of substantia nigra, the lenticular nucleus (green line) & head of caudate nucleus (red line in Fig. 6), normalised by total brain volume. Since the focus is on volume estimation, we process the MRI image sequences in batches of 3-4 consecutive frames.

- DaT (Dopamine Transporters) Scans: These images are created through Single-Photon Emission Computer Tomography (SPECT), with the patient taking 123-I-iodoflupane. Through them we can detect whether striatum gets dopaminergic links from substantia nigra and differentiate Parkinson's from other syndromes.

Doctors select the most representative image (the 8th in Fig. 7a), mark the areas around the caudate nucleus head, compare this with a neutral one (e.g. cerebellum), compute the ratios shown in Fig. 7b and use them next for diagnosis.



(a)



(b)

Fig. 7 (a) A sequence of frames from a DaT scan (b) DaT scan with expert selection, with and without markings and diagnostic ratios

Each input to the CNN-RNN network consists of three consecutive MRI gray-scale images (as was above-described) and one color (RGB) DaT scan image. By combining both types of images across subjects, we generated a training set composed of about 230000 input vectors from 50 subjects. Then we created validation and test datasets from 6 different subjects each, by combining the two types of images per single subject; each set was composed of about 1200 input vectors. The ratio of patients to non-patients was about 2:1 in all cases.

It should be mentioned that our target has been to test the ability of the network to learn to solve the problem in a person independent way, i.e., be trained with a number of patients/non-patients and generalize its performance to other subjects who

have not been used in the training set. The network has two linear outputs, (1,0) and (0,1), respectively, for the two categories.

Our approach to CNN-RNN network design has been through transfer learning, i.e. transferring the weights of the convolutional and pooling layers of a pretrained CNN, to the generated network and training the ‘upper’ FC and RNN hidden layers of the CNN-RNN with the above-described database.

We used the ResNet-50 CNN, which has been trained with millions of general-type RGB images for this purpose. In particular, we used a separate ResNet pre-trained convolutional part for each of the input image types, i.e., one focusing on the MRI triplets and another focusing on the DaT scan. We concatenated the outputs of these two ResNet substructures at the input of the first FC layer of the CNN network.

Table 1 summarizes the accuracy obtained by the CNN-RNN (with GRU neuron model) architecture, with different respective structures. Two hidden layers were used in the RNN part, each of which was composed of 128 units based on the GRU neuron model. The best performance of the CNN-RNN architecture was 99,97% on the training data and about 98% on the validation data. Fig. 8 shows the accuracy obtained by the end-to-end deep CNN-RNN architecture on the validation data set during training, with reference to the number of training epochs.

The performance on test data was very good, equal to 98%; it was 96% on patient data and perfect, i.e., 100% on non-patient data. For comparison purposes, we repeated the training with a CNN only network. The respective best performance of the CNN network was 94%. It is evident that the RNN part provided the deep neural architecture with the ability to follow better the time varying (volume) correlations provided in the MRI sequence of triplets of frames.

TABLE I. PERFORMANCE OF CNN-RNN ARCHITECTURE

<i>Configuration</i>	<i>Number of Units in FC Layer</i>	<i>Accuracy</i>
1	500	91.00%
2	1000	96.00%
3	1500	98.00%
4	2000	97.00%

Moreover, in the best reported case (line 3 of Table 1) the MSE value was very low, equal to 0,02. Precision reached 1,00 and Recall 0,96. The F1 value was 0,98.

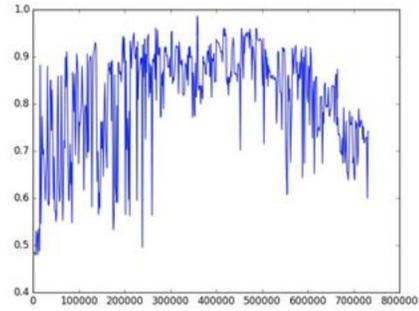


Fig. 8 CNN-RNN performance on test data, during training epochs

We have used the Adam optimizer algorithm, in mini batches, for the CNN-RNN training. The cost function was the Mean Squared Error (MSE). The hyper-parameters we selected to use were: a batch size of 40 (20 examples from each category), a constant learning rate of 0.001; fully connected layer included biases and a dropout with a value of 0.5. The weights of the fully connected and hidden layers were initialized from a Truncated Normal distribution with a zero mean and a variance equal to 0.1 and the biases were initialized to 1. Training was performed on a single GeForce GTX TITAN X GPU and the training time was about two days.

The weights of this CNN-RNN architecture have been stored in the Primary Memory of the developed system shown in Fig.1 of this paper. Then we applied the proposed procedure to derive the Clustered Representation Set.

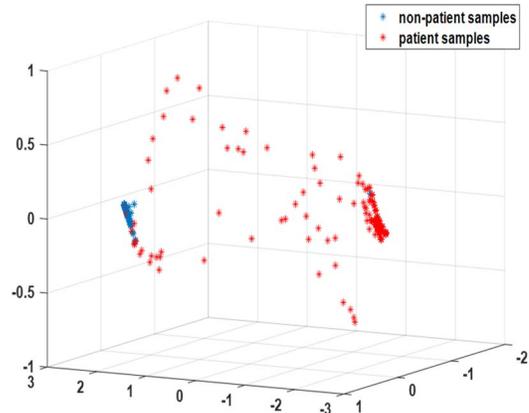


Fig. 9 Visualization (in 3-D) of CNN-RNN generated CRS in test data

Fig. 9 shows the 128-dimensional input vectors generated by the CNN-RNN network at its last hidden layer in the case of test data. They are visualized in a 3-D space, after using principal component analysis. The respective annotation, i.e., if it refers to a patient, or non-patient case, is also shown.

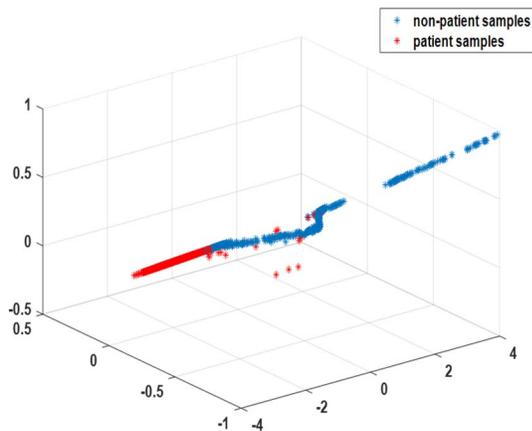


Fig. 10 Visualization (in 3-D) of CNN generated vectors in test data

For comparison purposes, in Fig. 10, we show the respective visualization of input vectors when a single CNN network has been trained with the input data for Parkinson’s prediction. It can be seen that the CNN-RNN architecture managed to provide a much more compact representation, especially of the non-patient cases.

We then performed clustering of these input vectors to generate the CRS and evaluate the implementation and operation of the proposed system, as shown in Fig.1 and Fig.2 respectively. We applied the k-means algorithm on the training dataset, using the validation data set to define the best number of clusters for the Parkinson’s prediction problem.

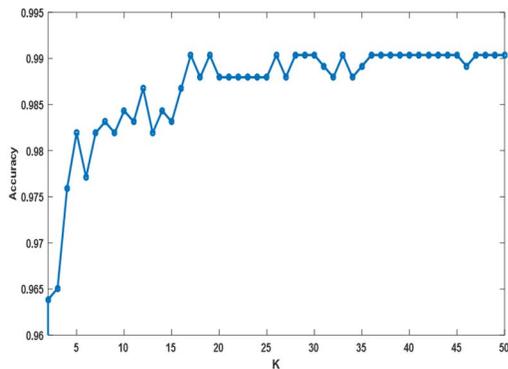


Fig. 11 Accuracy in Classification with reference to number of Clusters

Fig.11 presents a plot of the classification accuracy achieved in the validation data set when using different number, K , of clusters in the CRS. The best accuracy was achieved for a number $K = 17$.

In this configuration, 3 clusters correspond to the Non-Patient category and 14 clusters corresponded to the Patient one. The obtained accuracies are shown in Table II. We computed the Precision metric for all clusters. The computed values were 99% - 100% in all cases, which shows the achieved excellent purity performance of clustering.

TABLE II. CRS-BASED CLASSIFICATION ON VALIDATION SET

<i>True Positive Rate</i>	<i>False Positive Rate</i>	<i>Accuracy</i>
99.55 %	1.55 %	99.04 %

Fig. 12 shows a visual illustration of the centroids and main components of the 3 clusters corresponding to the Non-Patient category, while Figure 13 presents the centroids and main components of the 4 most significant clusters of the Patient category.

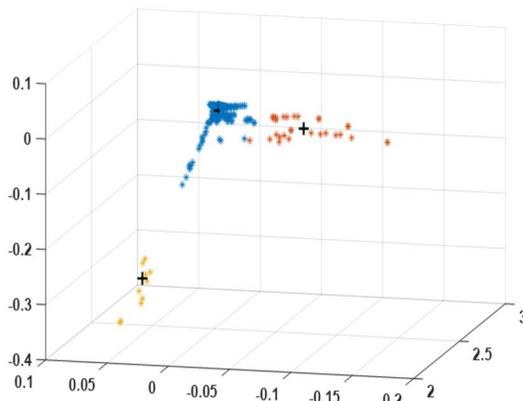


Fig. 12 Visualization (in 3-D) of CRS (centroids & representative components) for the Non-Patient category

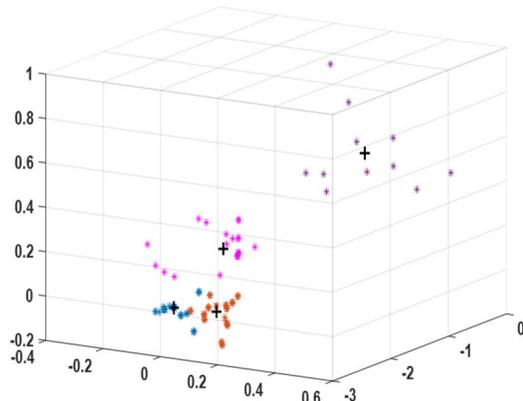


Fig. 13 Visualization (in 3-D) of the major part of the CRS (centroids & representative components) for the Patient category

Through postprocessing, we merged the obtained clusters to five - two for the non-patient and three for the patient category - without significantly changing the obtained CRS-based accuracies. We then provided the five respective inputs (MRI triplets and DaT Scan image) to medical experts, who came out that the five centroids differ in the level of degeneration of the basal ganglia (Lentiform Nucleus, Caudate Nucleus) and that they roughly represent the 3 stages of DaT loss in Parkinson’s. In particular: the first cluster center corresponded to a typical frame from a DaT scan of an individual not suffering from PD; the second cluster center represented a case that seems to be pathological but belongs to a healthy individual, since, though

the Lentiform Nucleus appears to be completely gone, there is no diffusion of the contrast agent in the brain; the third cluster center represented the early stages (1-2) of the degeneration associated with Parkinson's; the fourth cluster center showed a typical stage 2 DaT loss; the fifth cluster center represented the most advanced, stage 3, of DaT loss.

We then examined the drift detection and adaptation phase of the proposed system, shown in Fig.4, when new subjects' data are presented at its input. We used the test dataset we have created for this purpose. We presented the test data to the Primary Memory of the generated system composed of the DNN and CRS components and performed drift detection and updating of the CRS system. The obtained classification of the new data is summarized in Table III, showing that the performance of the generated system was very good. There were only 21 input combinations (out of 508) from 3 patients that have been erroneously classified, as NPD cases, both by the CNN-RNN and the CRS architecture. It should, however, be stressed that in all these cases the distances of the computed representations from the 5 cluster centers have been larger than the respective maximum intra-cluster distances. This has been the criterion for drift detection. Two additional cluster centers were added in the CRS to model these personalized cases. The CNN-RNN was also retrained with these data, being then able to correctly classify them as well.

TABLE III. CRS-BASED CLASSIFICATION ON NEW DATA

<i>True Positive Rate</i>	<i>False Positive Rate</i>	<i>Accuracy</i>
98.00 %	1.00 %	98.67 %

VI. CONCLUSIONS AND FURTHER WORK

In this paper, we used recent developments in designing end-to-end deep neural networks, composed of CNN and RNN components, refined them and appropriately trained them with medical imaging data, composed of MRI and DaT scans. We managed to obtain very good performance in prediction of Parkinson's disease, treating it as a binary classification problem (to patient and non-patient categories). Moreover, we proposed a novel unsupervised approach, based on clustering of trained DNN internal representations, which provides the deep neural architecture with the ability to adapt to new data cases in an efficient and transparent way.

In our future research, with the aid of medical experts, we will extend our results for Parkinson's, with more data and modelling the evolution of the disease as well. We will also extend the semantic descriptions of the generated CRS, relating the DNN decisions with the developed cluster characteristics, thus, rendering them really transparent and trusted.

ACKNOWLEDGMENT

The work of the NTUA team was financed by Greek State Scholarships Foundation (IKY) through "Research Projects for Excellence IKY/Siemens" Programme in the framework of the Hellenic Republic – Siemens Settlement Agreement.

REFERENCES

- [1] A. Doulamis, N. Doulamis, S. Kollias, "On-Line retrainable neural networks: Improving the performance of neural networks in image analysis problems," IEEE Transactions on Neural Networks, vol. 11, no 1, pp. 137-156, 2000.
- [2] G. Litjens, T. Kooi, B.E. Bejnordi, et al., "A survey on deep learning on medical image analysis," Computer Vision Pattern Recognition, <https://arxiv.org/abs/1702.05747v2>, 2017.
- [3] D. Kollias, T. Tagaris, A. Stafylopatis, "On line emotion detection using retrainable deep neural networks," IEEE Symposium Series on Computational Intelligence 2016, Athens, Greece, 6-9 December 2016.
- [4] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, The MIT Press, Cambridge, MA, 2016.
- [5] A. Krizhevsky, I. Sutskever, G. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems, vol. 25, pp. 1106-1114, 2012.
- [6] H.W. Ng, et al., "Deep learning for emotion recognition on small datasets using transfer learning," Proceedings ACM International Conference Multimodal Interaction, Seattle, USA, 9-13 November 2015.
- [7] S. E. Kahou, et al., "Recurrent neural networks for emotion recognition in video," Proceedings ACM International Conference on Multimodal Interaction, Seattle, USA, 9-13 November 2015.
- [8] M. Wollmer, M. Kaiser, et al., "Lstm-modeling of continuous emotions in an audiovisual affect recognition framework," Image and Vision Computing, vol. 31, no 2, pp. 153-163, 2013.
- [9] P. Vincent, H. Larochelle, et al., "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," The Journal of Machine Learning Research, vol. 11, pp. 3371-3408, 2010.
- [10] A. Wilson, J. Yosinski, et al., "Interpretable Machine Learning," NIPS 2017 Symposium, LA, USA, 7 December 2017.
- [11] M. Yu et al., "An on-line one class SVM-based person-specific fall detection system," IEEE Journal of Biomedical and Health Informatics, vol. 17, no. 6, pp. 1002-1014, 2013.
- [12] K. Cho, B. Merriënboer, et al., "On properties of neural machine translation: Encoder-decoder approaches," arXiv, 2014.
- [13] G. Stamou, S. Kollias, Multimedia Content and Semantic Web, John Wiley and Sons, NY., 2005.
- [14] A. d' Acila Garcez et al., "Neural-symbolic learning and reasoning: Contributions and challenges", Proceedings of AAAI Spring Symposium, Stanford University, CA, March 2015.
- [15] Z. Hu et al., "Harnessing deep neural networks with logic rules", arXiv:1603.06318v4, 2016
- [16] C. Goetz et al., "Movement Disorder Society - sponsored revision of the Unified Parkinson's Rating Scale and clinimetric tests," Movement disorders, vol. 23, no 15, pp. 2129-2170, 2008.
- [17] C. Goetz et al., "U.D.R. Scale," Movement Disorders, vol. 23, no 16, pp. 2398-2403, 2008.
- [18] <http://www.islab.ntua.gr/datasets/parkinson/>
- [19] K. Simonyan et al., "Convolutional neural networks and large-scale image recognition (IR)," arXiv:1409.1556, 2014.
- [20] K. He et al., "Deep residual learning for image recognition," <https://arxiv.org/abs/1512.03385v1>