Deep Neural Networks: Latent Information Extraction, Domain Adaptation & Uncertainty Estimation

Stefanos Kollias[1,2]

[1]University of Lincoln, UK
[2]National Technical University of Athens, GR

KSEM 2019, Athens, Greece

28 August 2019

Introduction

In this presentation, based on our recent research on Deep Neural Networks (DNNs) at UoL, we focus on:

- extracting latent information from trained DNNs;
- using this information to visualize/explain the DNN decision making;
- adapting the generated knowledge in other domains;
- providing cues about the decision uncertainty.

Examples from real life problems are used to illustrate the achieved performances.

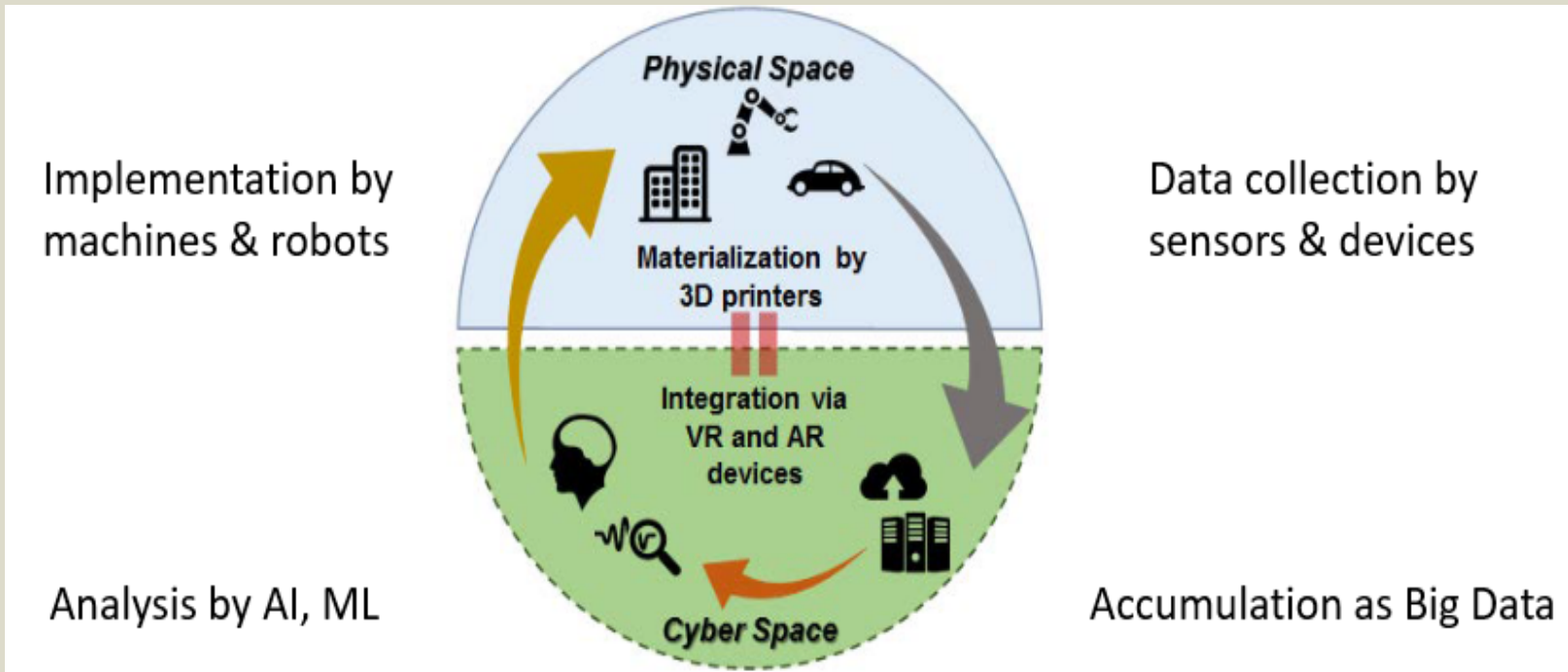# UK Grand Challenge: Artificial Intelligence & Data

Artificial Intelligence and Machine Learning have started to transform the global economy.

They can be seen as new industries in their own right, whilst transforming business models across sectors.
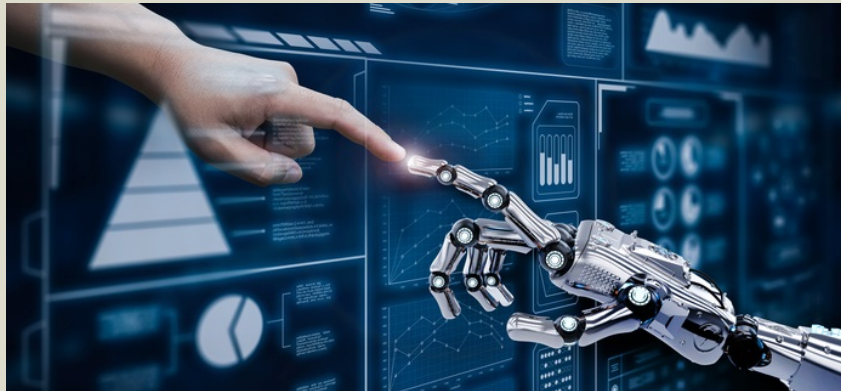
Industry 4.0 to Society 5.0

Trustworthy AI

- European AI Alliance:

  - Transparent AI

  - Explainable AI

  - Responsible AI

  - Trustful AI

  - Human centric AI

  - ...

1. Current Developments at UoL

DNNs are developed, analysed & used in

- Industrial Environments: Anomaly Prediction
- Operational Environments: Monitoring and Improving Productivity
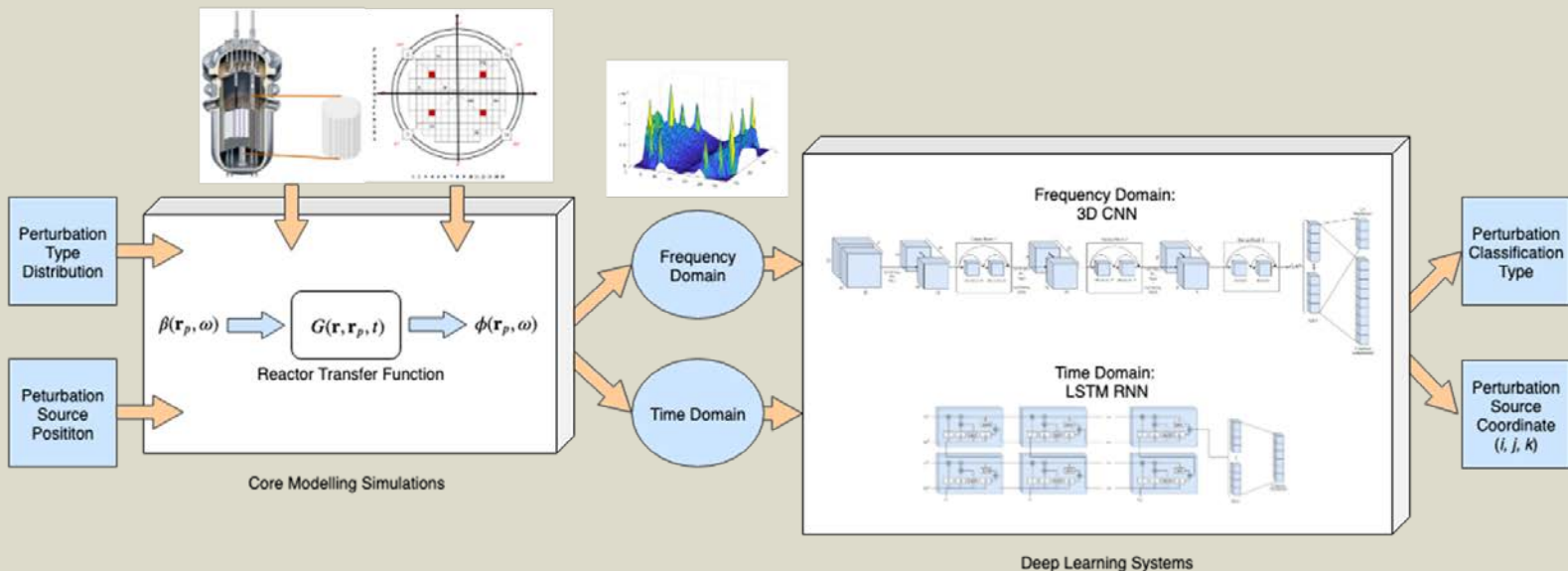- Social Environments: Healthcare, Well being, Culture & Creativity

mlearn

UNIVERSITY OF
LINCOLN

Industrial Environments

1) Siemens Gas Turbine ML/DL Data Analysis, Fault Prediction - Mindsphere Cloud

2) Nuclear Power Plant DL Signal Analysis, Anomaly Prediction – Simulated & Real Data

(CORTEX, EU H2020 project , 2017-21)

Operational Environments

1) Smart DL/AI Agri-Production Prediction - 1st UK Centre for Agri-Robotics

2 Intelligent Refrigeration Systems - Large Energy Supermarket Savings

3) DL based OCR in Food Packaging - Use By Date and/or Ingredient Verification

4) Environmental Data Analysis - Missing Values Prediction - Water/Rainfall Prediction
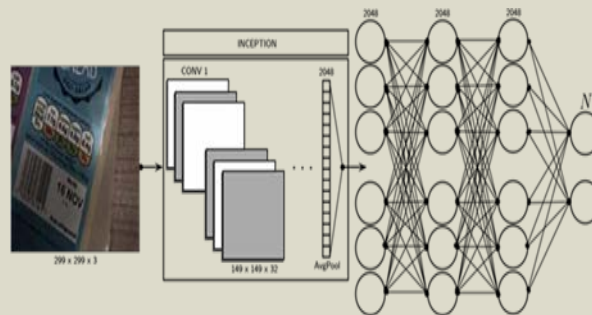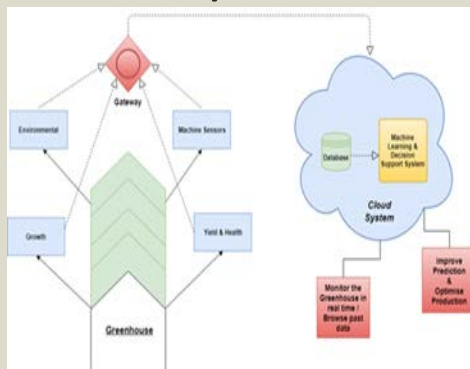
(SmartGreen, Interreg project , 2017-21)

Social Environments: Healthcare, Well being, Culture & Creativity

1) Predictive Modelling of Ambulance calls to Care Homes

2) Predicting Parkinson's Disease from Medical Images

3) Fall Detection of Elderly People Living Alone

4) Video Automated Annotation (Persons, Behaviors)

5) Analysis & Creative Reuse of Cultural Heritage Data
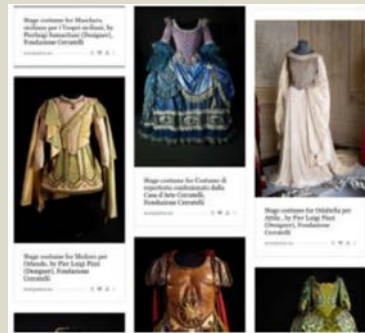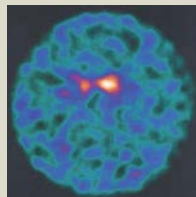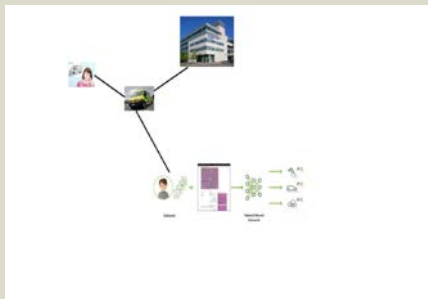
(WeHope, Creative Europe project, 2019-22)

2. DNN Architectures

Various DNN architectures can be used for big data analysis and automatic generation of new features;

to name a few:
- Convolutional CNNs
- Recurrent RNNs, C-RNNs, CNN-RNNs
- Generative Adversarial GANs
- Capsule Nets.

Machine /Deep Learning

2012 –

Deep Learning changed the way to deal with analysis of data.
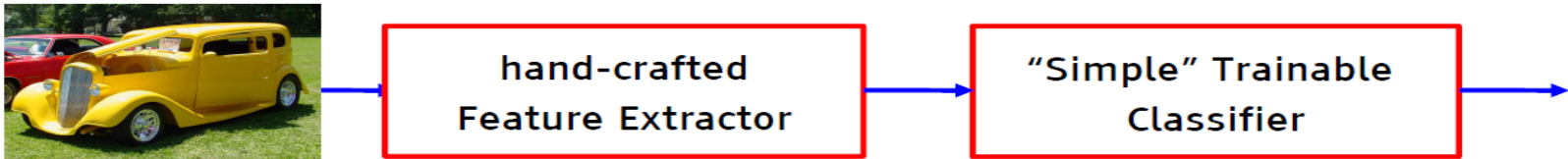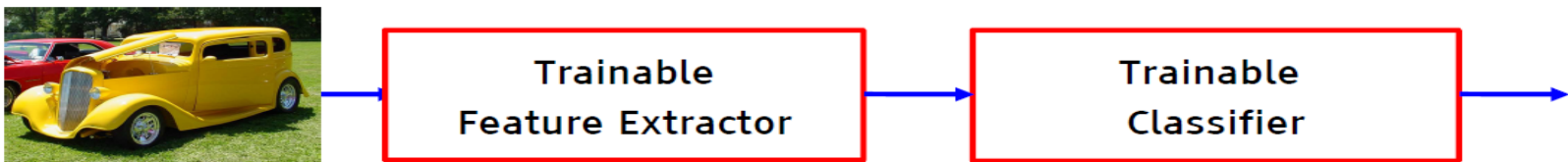
Already Known Architectures (e.g., Neocognitron)

- [Hubel & Wiesel 1962]:
  - simple cells detect local features
  - complex cells "pool" the outputs of simple cells within a retinotopic neighborhood.

U_G U_{S1} U_{C1} U_{S2} U_{C2} U_{S3} U_{C3} U_{S4} U_{C4}

U_0

input layer

contrast extraction

U_M masker layer

recognition layer

"Simple cells"

"Complex cells"

Multiple convolutions

pooling subsampling
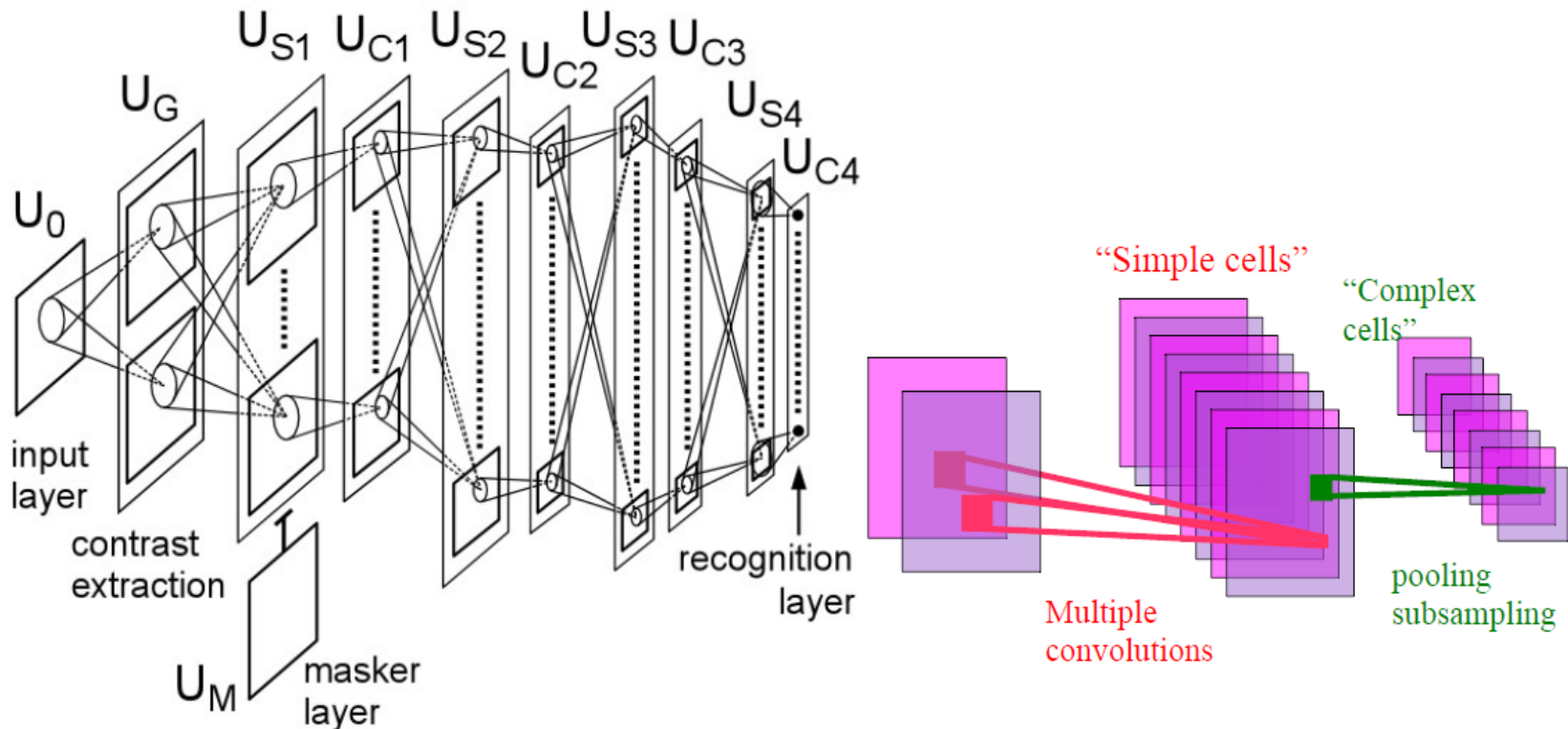
Automatic feature extraction in Deep Convolutional Neural Networks
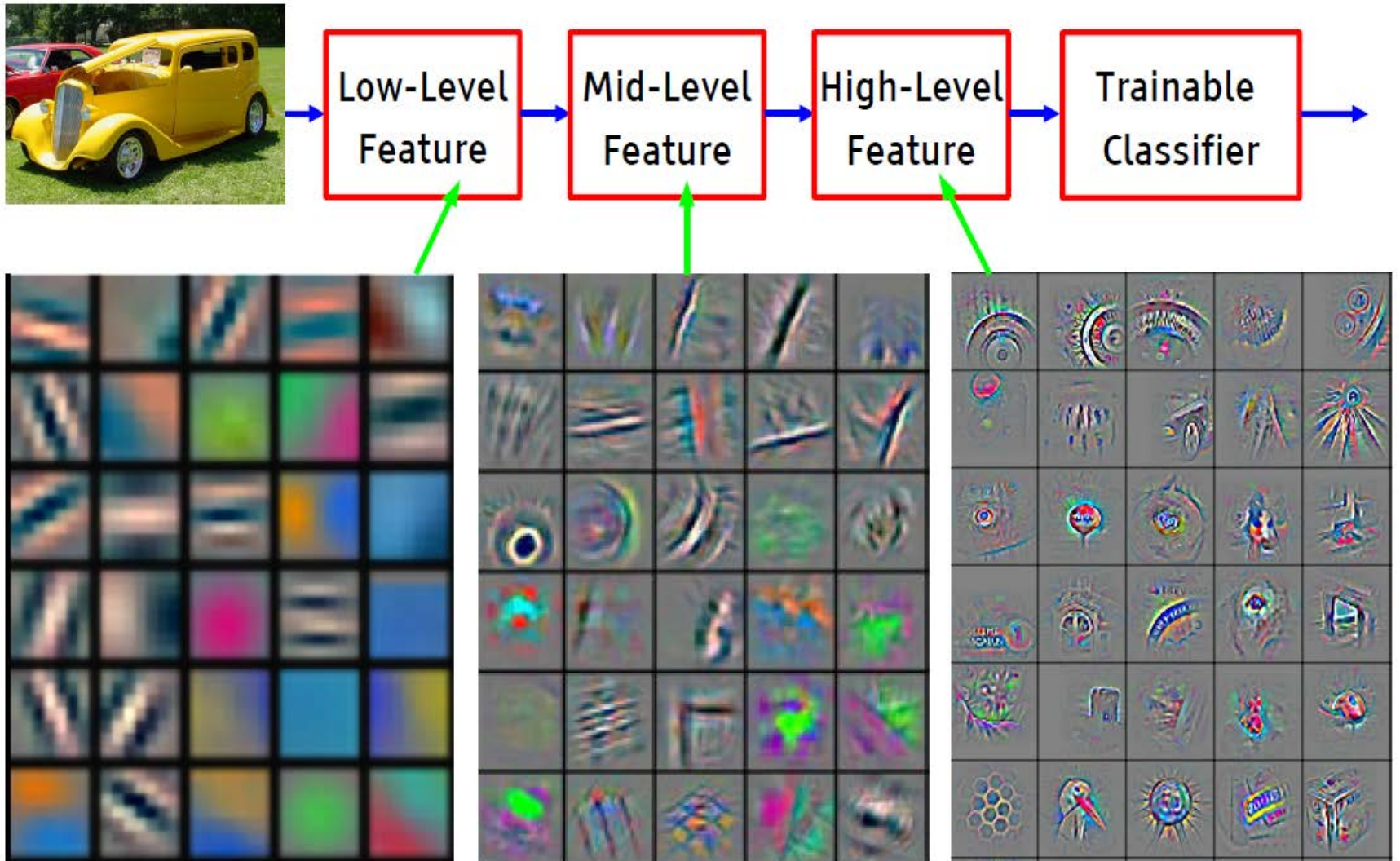
Using CNNs for Facial Expression & Behavior Recognition
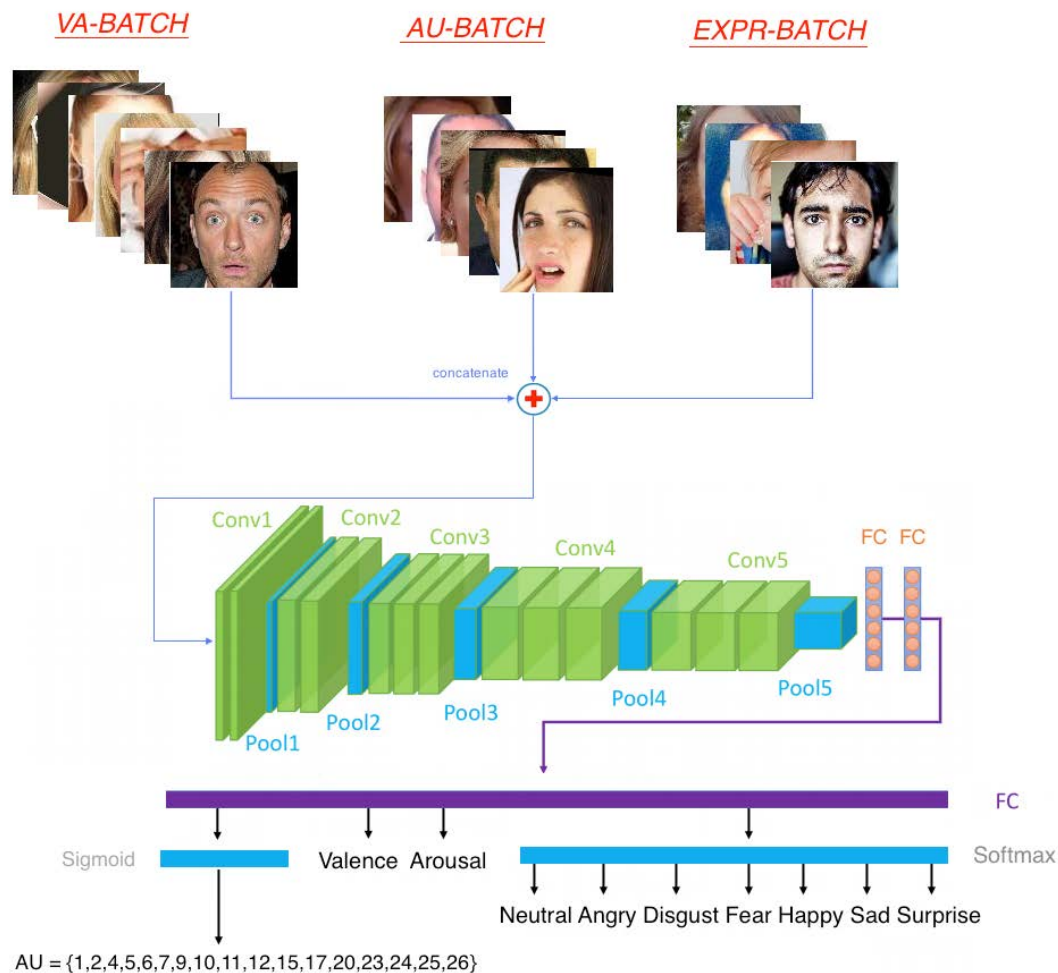
CNN-RNN Architectures (GRU Units)
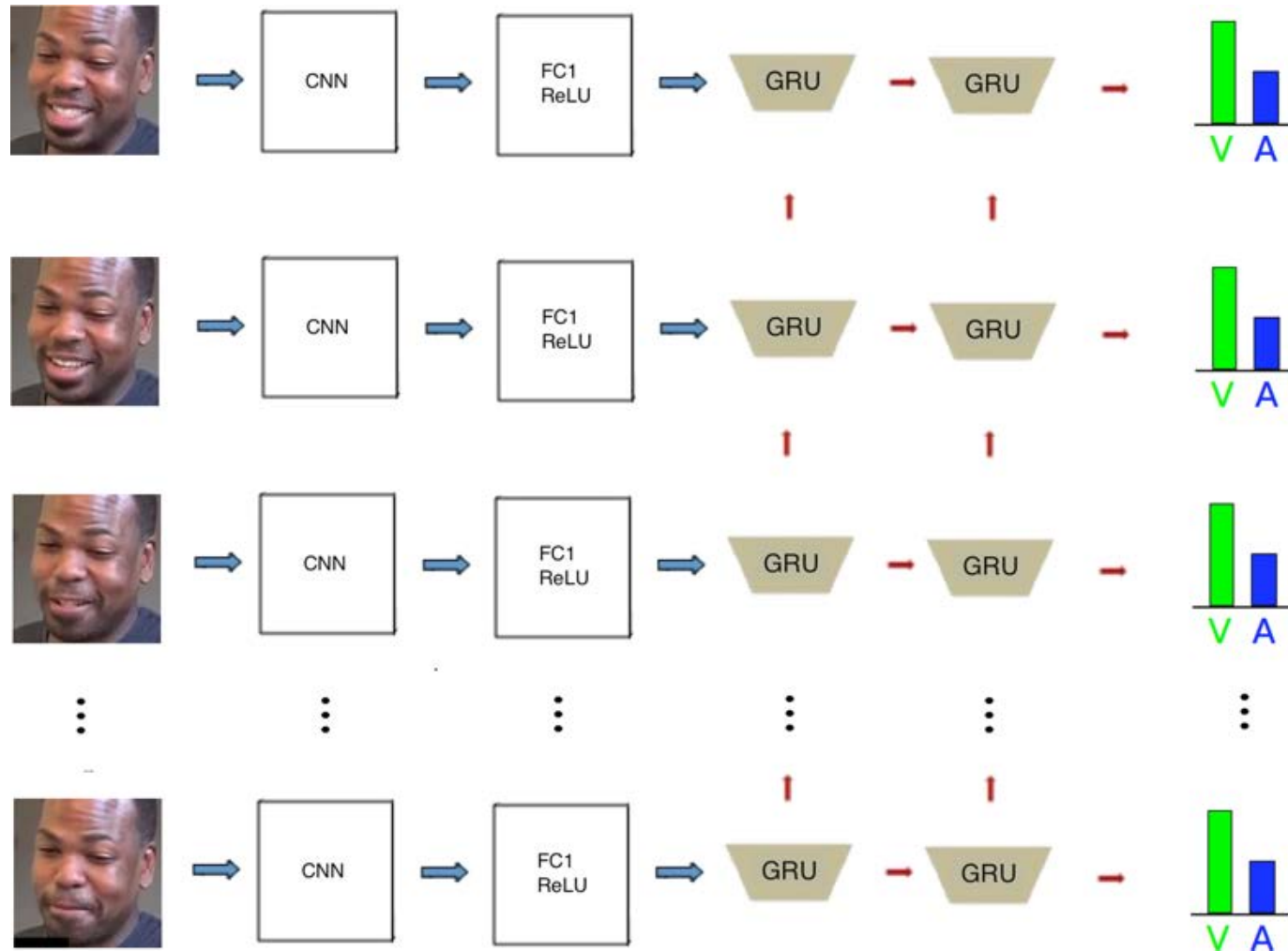
Various CNN-RNN structures

- Aggregate the outputs of neurons from Fully Connected and/or CNN High/Mid/Low Layers

- Feed one or more RNN Hidden Layers

- Provide final classification or regression through CNN-RNN ensembles

Feature Concatenation in CNN-RNN
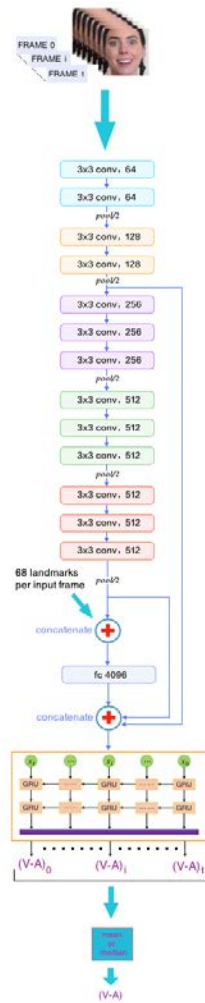
Generative Adversarial Networks
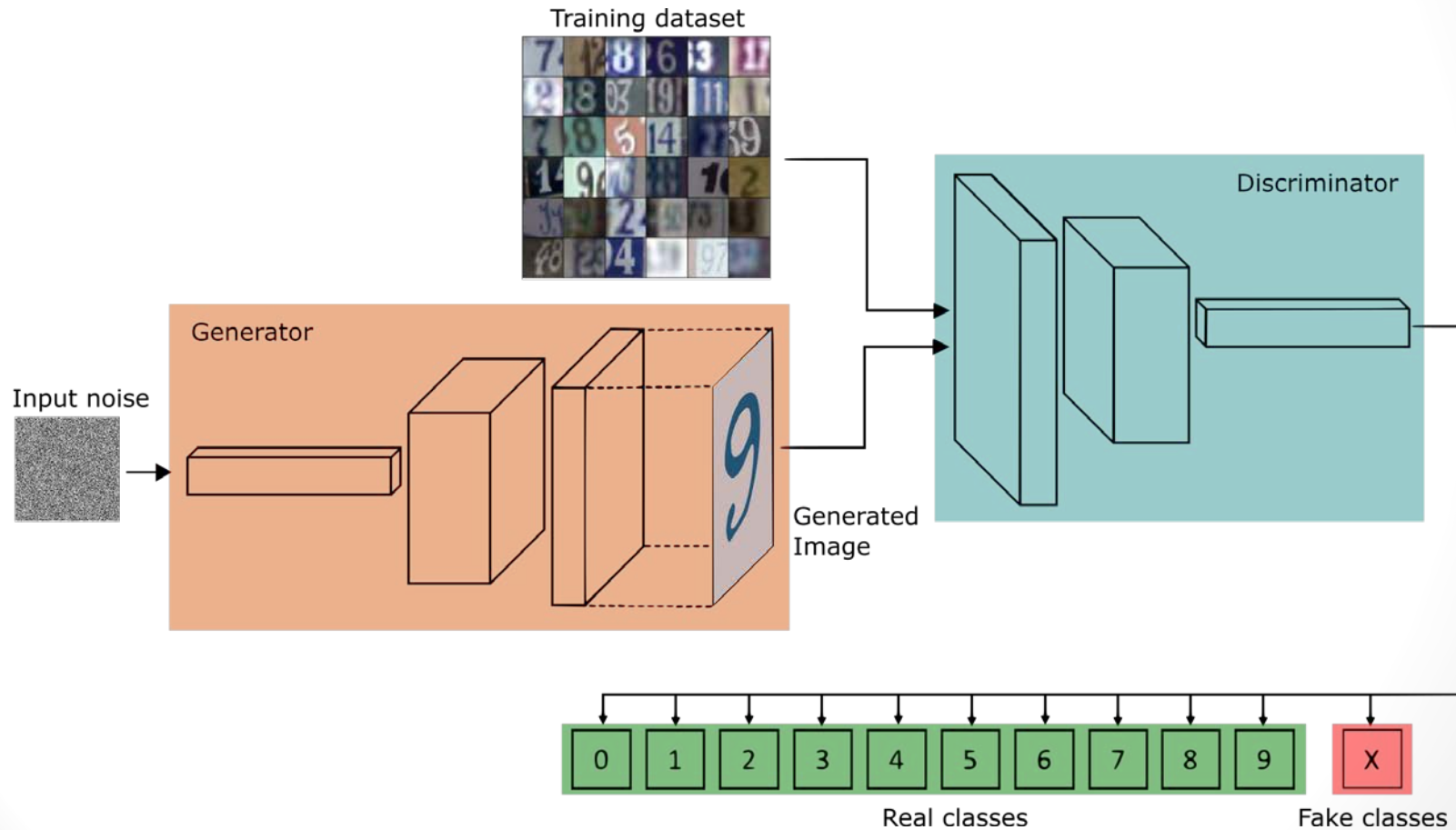
StarGANs for Image-Image Translation
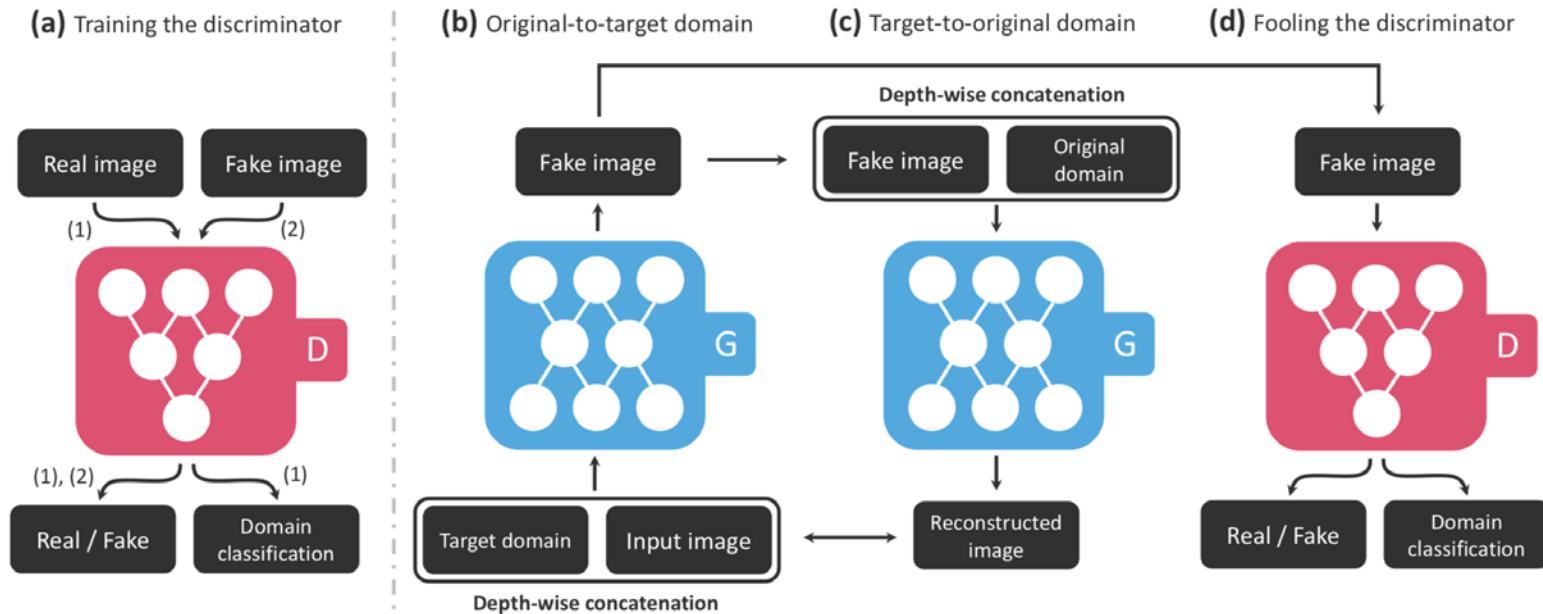
DNN Limitations

However, there are problems in DNN use:

- Need of large training datasets (zero/one shot learning is required)

- Lack of transparent & explainable decision making

- Manual annotation of large amounts of data is required, which is practically intractable.

The following can assist in facing these issues:

- Analysis and use of Learnt Structures (Latent Variables).

- Development of self-annotating predictive models.

3. Latent Variable Extraction

1. Extract DNN Learnt Features over Training Data

$$\mathcal{R}_p = \{(\mathbf{r}(j),\ j = 1,\ldots,n\}$$

p is the Training Set

r denotes the Latent Variables

n is the number of Latent Variables

2. Perform Clustering of these Representations

$$\widehat{T}_{k\text{-means}} = \arg\min_T \sum_{j=1}^{k} \sum_{\mathbf{r} \in R_p} ||\mathbf{r} - \mu_j||^2$$

μ is the mean value of data in each cluster j

3. Compute Cluster Centroids & use them to classify Test Data.

Specific Targets

Using the cluster centroids and their annotations we are able to:

- Predict new subjects' status (through a nearest neighbour criterion); thus explaining why we predict

- Efficiently perform network retraining with new subjects' data (without forgetting the cluster information)

- Transfer the knowledge learnt by the DNN to other domains/environments (where less input information is available).

An Example: Predicting Parkinson's

Prediction of Parkinson's through analysis of medical images (MRI, DaTScans)
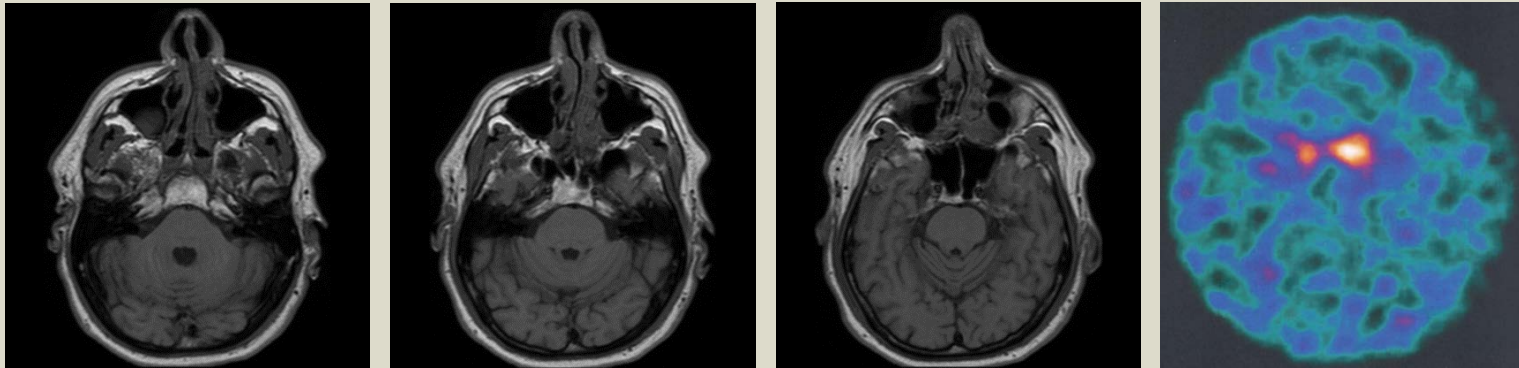
A Parkinson's database comprising 50,000 MRI and 1000 DaTScans from 78 subjects, 55 patients with Parkinson's & 23 controls has been used (available by permission, from UoL mlearn site).

Predicting with CNN/CNN-RNN (1)

- Through data augmentation more than 100,000 data samples were generated, each including a triplet of (consecutive) MRI and a DaTScan.

- These have been used to train CNN/CNN-RNN architectures to predict the status of the subjects (positive/negative).

- The triple MRI slices and the DaTScan are fed to two independent CNNs, one receiving the MRI triplets as inputs and one receiving the (RGB) DaTSCAN images.

Predicting with CNN/CNN-RNN (2)

- The weights of each CNN were initialised using the ResNet-50 CNN. Their outputs were concatenated at the input of the first FC layer of the CNN & any epidemiological data were also concatenated at this point.

- CNN (with 2 ReLU FC layers) and CNN-RNN (with 1 ReLU layer – GRU 2 layers) have provided best results for Parkinson's detection (following a detailed ablation study).

- 128-dimensional vectors were extracted from the last GRU layer as latent variables.

Latent Variable Clustering (1)

- Clustering of the 128-dimensional variables representing the classified images in PD/non-PD classes was performed.

- This resulted in identification of 5 clusters, which were then validated by medical experts as referring to:
    - non-patient (non-PD cases)
    - questionable non-PD (edge cases)
    - early-stage PD (stage 1 primarily)
    - standard-stage PD (stage 2)
    - late-stage PD (stage 4).

- Through this latent variable clustering , we increased the learnt DNN ability to perform 2-class classification (PD/non-PD), so as to manage a 5-stage Parkinson's prediction.

Latent Variable Clustering (2)
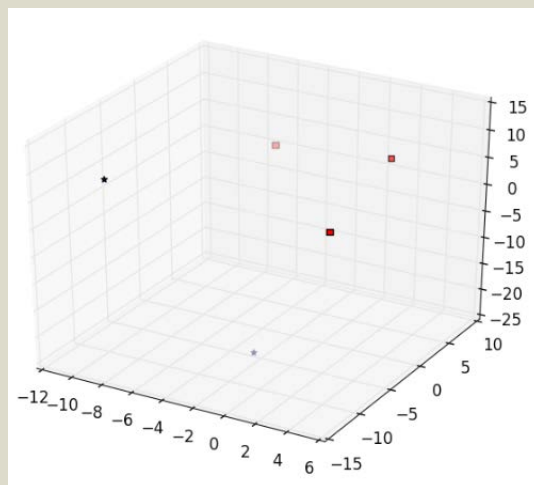
A 3-D projection of the 5 cluster centroids
Squares: patients; Stars: non-patients
The DatScans corresponding to the centroids.
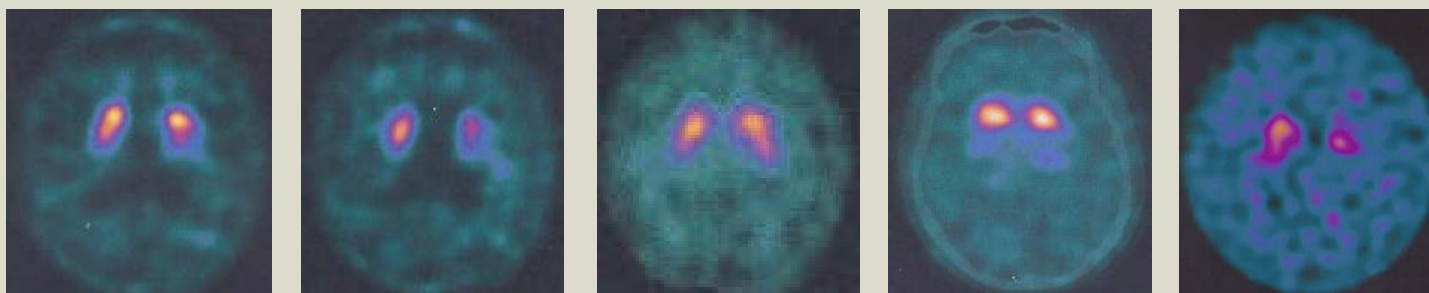
4. Retraining DNNs with Annotated Latent Variables (1)

- Whenever new data are collected (e.g., from one or more new subjects) and annotated, they should be used to adapt the knowledge (i.e., the weights) of the trained CNN/CNN-RNN.

- An approach is developed that computes small weight updates (via Taylor expansion) due to the new data, whilst preserving the annotated latent variables (i.e., data corresponding to cluster centroids).

Retraining DNNs with Annotated Latent Variables (2)

- Modify the Minimised Error Criterion, splitting the input data in two sets, the existing input data (Set P) and the new input data (Set P1, also containing the cluster centroids' inputs):

$$\mathcal{E} = \mathcal{E}_{\mathcal{P}_1} + \lambda \cdot \mathcal{E}_{\mathcal{P}}$$

- Update the CNN/CNN-RNN Weights **W**

$$\mathbf{W}' = \mathbf{W} + \Delta\mathbf{W}$$

Retraining of DNNs with Annotated Latent Variables (3)

When retraining the DNN over the old and new data, we replace the minimised MSE function, over the new data and the latent variable images, with the constraint that:

"DNN outputs and desired outputs are identical"

$$y'(j) = d(j); \ j = 1, \ldots, s$$

Retraining of DNNs with Annotated Latent Variables (4)

- Retraining is then performed through a constrained error minimization algorithm (such as the gradient projection)

$$\mathbf{v} = V \cdot \Delta \mathbf{W}$$

$$\mathbf{v}(j) = d(j) - y(j); \;\; j = 1, \ldots, s'$$

where V includes weights of the original DNN

- The MSE is minimised over old & new data, but with higher attention to the data corresponding to the latent variables and the new data.

5. DNN Domain Adaptation through Annotated Latent Variables(1)

- In many cases, e.g., general purpose medical centers, DaT Scan equipment may not be available, whilst having access to MRI technology.

- A domain adaptation methodology has been developed, for providing an improved PD prediction in such environments.

- This is achieved by using the annotated latent variables from the combined DaTScan/MRI data analysis, to drive DNN learning from only MRI, towards replication of the latent values as well.

DNN Domain Adaptation through Annotated Latent Variables(2)

- A new DNN loss function is introduced and used, including:

  - a normal MSE minimisation function, over the MRI input data

$$\varepsilon_1 = \frac{1}{n'} \sum_{j=1}^{n'} (d'(j) - y'(j))^2$$

  - an additional term which minimises the MSE between the latent variables extracted in the new environment and the original variables extracted in the richer (with DaTScan & MRI medical imaging) environment

DNN Domain Adaptation through Annotated Latent Variables(3)

$$\mathbf{g}(i,j) = \mathbf{u}(i) - \mathbf{r}'(j), \;\; i = 1,\ldots,k; \;\; j = 1,\ldots,n'$$

$$G(i,j) = \mathbf{g}(i,j) * (\mathbf{g}(i,j))^T$$

$$\mathcal{E}_2 = \frac{1}{kn'} \sum_{i=1}^{k} \sum_{j=1}^{n'} (z(i,j) - [1 - f(G(i,j))])^2$$

The training algorithm is adapted to minimise both MSE terms.

$$\mathcal{E}_{new} = \eta \mathcal{E}_1 + (1 - \eta) \mathcal{E}_2$$

Results (1)

TABLE I: DNN best performing structures on DaT Scan and MRI data

Structure	No FC layers	No Hidden Layers	No Units in FC Layer(s)	No Units in Hidden Layers	Accuracy (%)
CNN	2	-	2622-1500	-	94%
CNN-RNN	1	2	1500	128-128	98%

TABLE II: Training data in each generated cluster

Cluster	No of Data (%)
t_1	4,3
t_2	38,4
t_3	27,6
t_4	2,3
t_5	27,4

Results (2)

Performance on 6 test subjects

TABLE III: Classification of 6 subjects' data in clusters t_1-t_5

Test case	t_1	t_2	t_3	t_4	t_5
Non Patient 1	44	398	0	0	0
Non Patient 2	10	90	0	0	0
Patient 1	3	7	94	8	8
Patient 2	1	7	139	17	20
Patient 3	3	0	145	18	38
Patient 4	0	0	0	8	72

By retraining the DNN learnt all former & 3 new subjects; also performing slightly better in the 3 rest

TABLE IV: Classification of 3 subjects' data, after retraining, in clusters t_1-t_5

Test case	t_1	t_2	t_3	t_4	t_5
Non Patient 1	41	401	0	0	0
Patient 1	2	5	99	7	7
Patient 4	0	0	0	7	73

Results (3)
Domain Adaptation raised accuracy 70.6 to 81.1%

TABLE V: MRI-based Classification of 6 subjects' data in clusters t_1-t_5

Test case	t_1	t_2	t_3	t_4	t_5
Non Patient 1	**181**	74	179	8	0
Non Patient 2	14	4	**44**	33	5
Patient 1	16	0	**53**	49	2
Patient 2	6	0	**83**	80	15
Patient 3	26	3	**130**	35	10
Patient 4	12	0	**51**	11	6

TABLE VI: MRI-based Classification of 6 subjects' data, after domain adaptation, in clusters t_1-t_5

Test case	t_1	t_2	t_3	t_4	t_5
Non Patient 1	**176**	147	114	5	0
Non Patient 2	13	**41**	25	18	3
Patient 1	13	0	**70**	35	2
Patient 2	5	0	**116**	54	9
Patient 3	20	2	**140**	34	8
Patient 4	9	0	31	5	**35**

6. Self Annotation through Deep Bayesian Uncertainty Estimation (1)

- We introduce a self-annotating prediction model based on Self-Training of a Bayesian CNN, that leverages variational inference methods of deep models.

- We propose a new inverse uncertainty weighting technique that encourages the Self-training model to learn from more informative data over time, preventing it from becoming lazy by only selecting easy examples to learn from.

Self Annotation through Deep Bayesian Uncertainty Estimation (2)

A real life problem:

Real-time validation of the 'use-by' date on food packaging images; unlimited number of images, different contexts, different backgrounds, occlusions, unfeasible annotation

Per category examples of images in our datasets. (a) Complete Date (day and month visible). (b) Partial Date (no day visible). (c) Partial Date (no month visible). (d) Unreadable. (e) No date (neither day or month visible).

Self Annotation through Deep Bayesian Uncertainty Estimation (3)

- Self-Training is a simple algorithm in which a classifier predicts labels at inference time, and increments the training set with the most confident predictions of the unlabeled data.

- Assuming a Bayesian Neural Network (BNN) formulation, a prior probability distribution $p(W)$ is placed over the set of trainable parameters W, with a Gaussian prior distribution being a sensible choice.

- However, the posterior distribution $p(W/X,Y)$ is intractable.

- Dropout is used at test time to perform a Monte Carlo approximation of the posterior distribution of the parameters.

Self Annotation through Deep Bayesian Uncertainty Estimation (4)

- Aleatoric uncertainty relates to sensory noise in the acquisition process of the data; and is therefore inherently irreducible.

- Epistemic uncertainty relates to our uncertainty about the model parameters; it is in fact reducible as we observe more data.

- We target minimising the epistemic uncertainty w.r.t. the variational interpretation of Dropout.

- By penalising very confident output distributions from our BNN we can improve generalization.

Self Annotation through Deep Bayesian Uncertainty Estimation (5)

Experiments with real data (5000 test data):
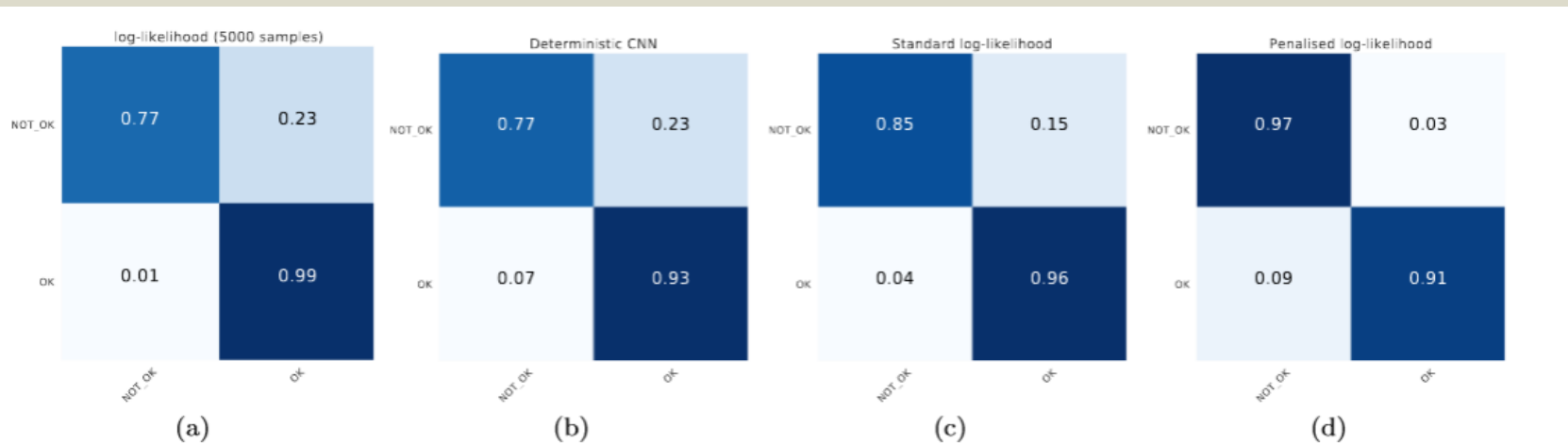
Normalised confusion matrices of the results obtained from our self-annotation procedure. **x** and **y** axes denote the *predicted* and *actual* classes, respectively. **(a)** Refers to the 5000 predicted labels obtained with the lowest prediction uncertainty. **(b)** Deterministic CNN predicted labels, wherein the thresholds were set based on networks sigmoidal output. **(c)** Predicted labels from our Bayesian Self-Training approach, trained with a standard binary log-likelihood loss. **(d)** Similar to (c) but using a Bayesian CNN trained with a penalised binary log-likelihood loss rather than the standard.

Self Annotation through Deep Bayesian Uncertainty Estimation (6)

Experiments with real data (5000 test data):

Bayesian CNN (Penalised log-likelihood)				
Cohen's Kappa: **0.8891**				
Class	Precision	Recall	F1	#Images
NOT-OK	0.9532	0.9694	0.9612	294
OK	0.9427	0.9136	0.9279	162
Avg./Total	0.9494	0.9496	**0.9494**	**456**

Bayesian CNN (Standard log-likelihood)				
Cohen's Kappa: **0.8383**				
Class	Precision	Recall	F1	#Images
NOT-OK	0.9679	0.8538	0.9073	212
OK	0.889	0.9764	0.9306	254
Avg./Total	0.9248	0.9206	**0.9200**	**466**

Deterministic CNN (Standard log-likelihood)				
Cohen's Kappa: **0.6964**				
Class	Precision	Recall	F1	#Images
NOT-OK	0.9158	0.7682	0.8355	453
OK	0.7989	0.9287	0.8589	449
Avg./Total	0.8576	0.8481	**0.8472**	**902**

Conclusions

(1) By analysing the knowledge (latent variables) extracted from a trained DNN with 2-class medical image data of Parkinson's, we were able to reveal a richer facet of this data, through a 5-cluster separation.

(2) By conserving these clusters, we were able to include new knowledge (through retraining with new data) in the DNN, without forgetting its former 5-cluster separation ability.

(3) By reforming the Error minimization function to focus on these clusters, we were able to adapt the learnt knowledge so a to improve DNN performance in less rich input domains.

(4) By Bayesian Self-Annotation we were able to improve prediction of uncertainty in OCV 'use-by date' in food packaging images.

Extending the State-of-the-Art

ML & DNNs

Combine supervised with unsupervised DNN learning, using representations extracted from trained DNNs:

latent variables, structures, graphs, projections, clusters, auto-encoder patterns.

Use these for transparency & interpretability.

DNNs & AI

Combine DNNs with Reasoning Algorithms & semantically update existing Knowledge Bases.

Bibliography

Journals (2018-2019)

1. F.Ribeiro, F. Caliva, M. Swainson, K. Gudmundsson, G. Leontidis, S. Kollias, "Deep Bayesian Self-Training", Neural Computing and Applications, to appear, 2019.

2. A. Chatzitofis, D. Zarpalas, S. Kollias, P. Daras, "Deep Optical Motion Capture Using Multiple Depth Sensors and Retro-Reflectors", Sensors, 19(2), pp. 282-307, 2019.

3. S. Jiang, M. Kaiser, S.Yang, S. Kollias, Stefanos, N. Krasnogor, "A Scalable Test Suite for Continuous Dynamic Multiobjective Optimization", IEEE Transactions on Cybernetics, pp. 1-13, February 2019.

4. B. Chen, J. Huang, S. Kollias, S. Yue, Y. Huang, "Combining Guaranteed and Spot Markets in Display Advertising: Selling Guaranteed Page Views with Stochastic Demand', accepted for publication, European Journal of Operational Research, 2019.

5. N. Papanelopoulos, Y. Avrithis, S. Kollias, "Revisiting the Medial Axis for Planar Shape Decomposition", Computer Vision and Image Understanding, pp. 66-78, vol. 179, February 2019.

6. K. Raftopoulos, S. Kollias, D. Sourlas, M. Ferecatu, "On the Beneficial Effect of Noise in Vertex Localization", International Journal of Computer Vision, vol. 126, no 1, pp. 111-139, 2018.

7. D. Kollias, A. Tagaris, A.-G. Stafylopatis, S. Kollias, G. Tagaris, "Deep Neural Architectures for Prediction in Healthcare", Complex Intelligent Systems, vol. 4, no. 2, pp. 119–131, 2018.

8. A. Tagaris, D. Kollias, A.-G. Stafylopatis, G. Tagaris, S. Kollias, "Machine Learning for Neurodegenerative Disorder Diagnosis: Survey of Practices and Launch of Benchmark Dataset", International Journal on Artificial Intelligence Tools, vol. 27, no. 3, 1850011, 2018.

Bibliography

Conferences (2018-2019)

1. I. Kollia, A. Stafylopatis, S. Kollias, "Predicting Parkinson's Disease using Latent Information extracted from DNNs", IEEE International Joint Conference Neural Networks, Budapest, Hungary, 15-19 July 2019.

2. B. Alhnaity, S. Pierson, G. Leontidis, S. Kollias, "Using Deep Learning to Predict Plant Growth and Yield in Greenhouse Environments", GreenSys 2019, International Symposium on Advanced Technologies and Management for Innovative Greenhouses, Angers, France, 16-20 June 2019.

3. A. Durrant, G. Leontidis, S. Kollias, "3D Convolutional and Recurrent Neural Networks for Reactor Perturbation Unfolding and Anomaly Detection", FISA 2019, Pitesti, Romania, 4-7 June 2019.

4. I. Kollia, A. Stafylopatis, S. Kollias, "Artificial Intelligence and Machine Learning for Cultural Heritage Content Enrichment and Search on the Web", Heritage Dot Conference, 3-5 June 2019.

5. J. Wingate, I. Kollia, L. Bidaut, S. Kollias, "Healthcare Prediction using Latent Information extracted from Deep Neural Networks. Advances in Data Science, Manchester, UK, 20-21 May 2019.

6. F. Ribeiro, F. Caliva, D. Chionis, A. Dokhane, A. Mylonakis, C. Demaziere, G. Leontidis and S. Kollias, "Towards a Deep Unified Framework for Nuclear Reactor Perturbation Analysis", IEEE Symposium Series on Computational intelligence, Bangalore, India, 18-21 November 2018.

7. I. Kollia, S. Kollias, "A Deep Learning Approach for Load Forecasting of Power Systems", IEEE Symposium Series on Computational Intelligence, Bangalore, India, 18-21 November 2018.

8. F. Ribeiro, L. Gong, F. Caliva, M. Swainson, K. Gudmundsson, M. Yu, G. Leontidis, X. Ye and S. Kollias, "An end-to-end deep neural architecture for optical character verification and recognition in retail food packaging", 25th IEEE International Conference on Image Processing, 7-10 October 2018.

9. F. Caliva, F. S. Ribeiro, A. Mylonakis, C. Demaziere, P.Vinai, G. Leontidis, and S. Kollias, "A deep learning approach to anomaly detection in nuclear reactors," IEEE International Joint Conference on Neural Networks, Rio de Janeiro, Brazil, 8-13 July 2018.

10. J. McGinty, G. Leontidis, S. Kollias, "Optimising Remedial Outcomes in Gas Turbines through Data Analysis", Intern. Conf. on Industrial Maintenance & Reliability, Manchester, 13-15 June 2018.

Thank you for your attention