PremiUm-CNN: Propagating Uncertainty Towards Robust Convolutional Neural Networks

2022 IEEE International Conference on Acoustics, Speech and Signal Processing

Dimah Dera[*], Nidhal Buaynaya[†], Ghulam Rasool[†], Roman Shterenberg[‡] and Hassan Fathallah-Shaykh[§]

[*†]Department of Electrical and Computer Engineering, [‡]Department of Mathematics and [§]Department of Neurology

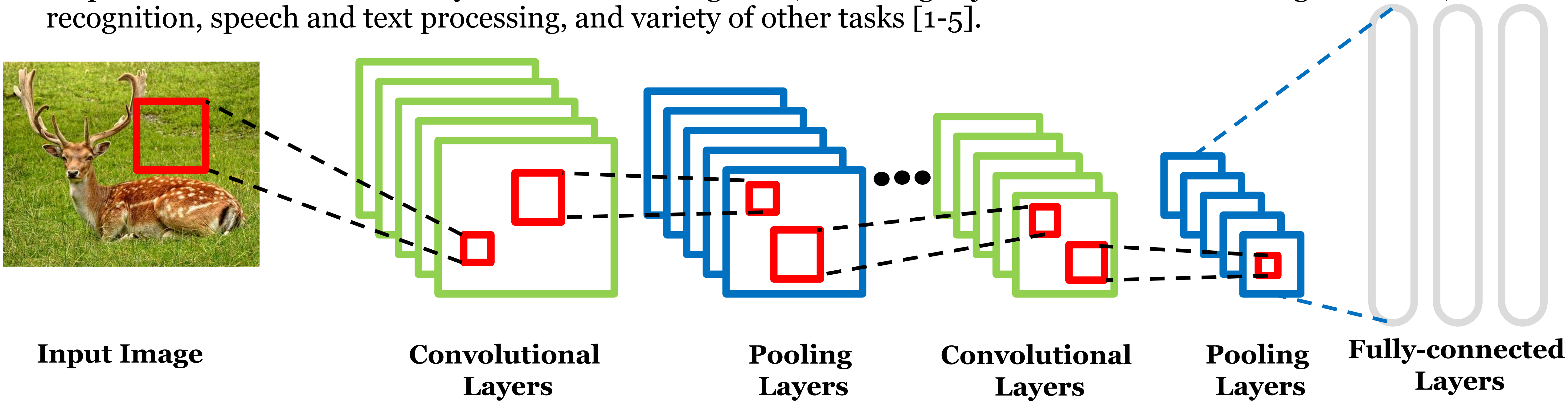
[*]The University of Texas Rio Grande Valley, [†]Rowan University,
[‡§]University of Alabama at Birmingham, USA

May 9, 2022

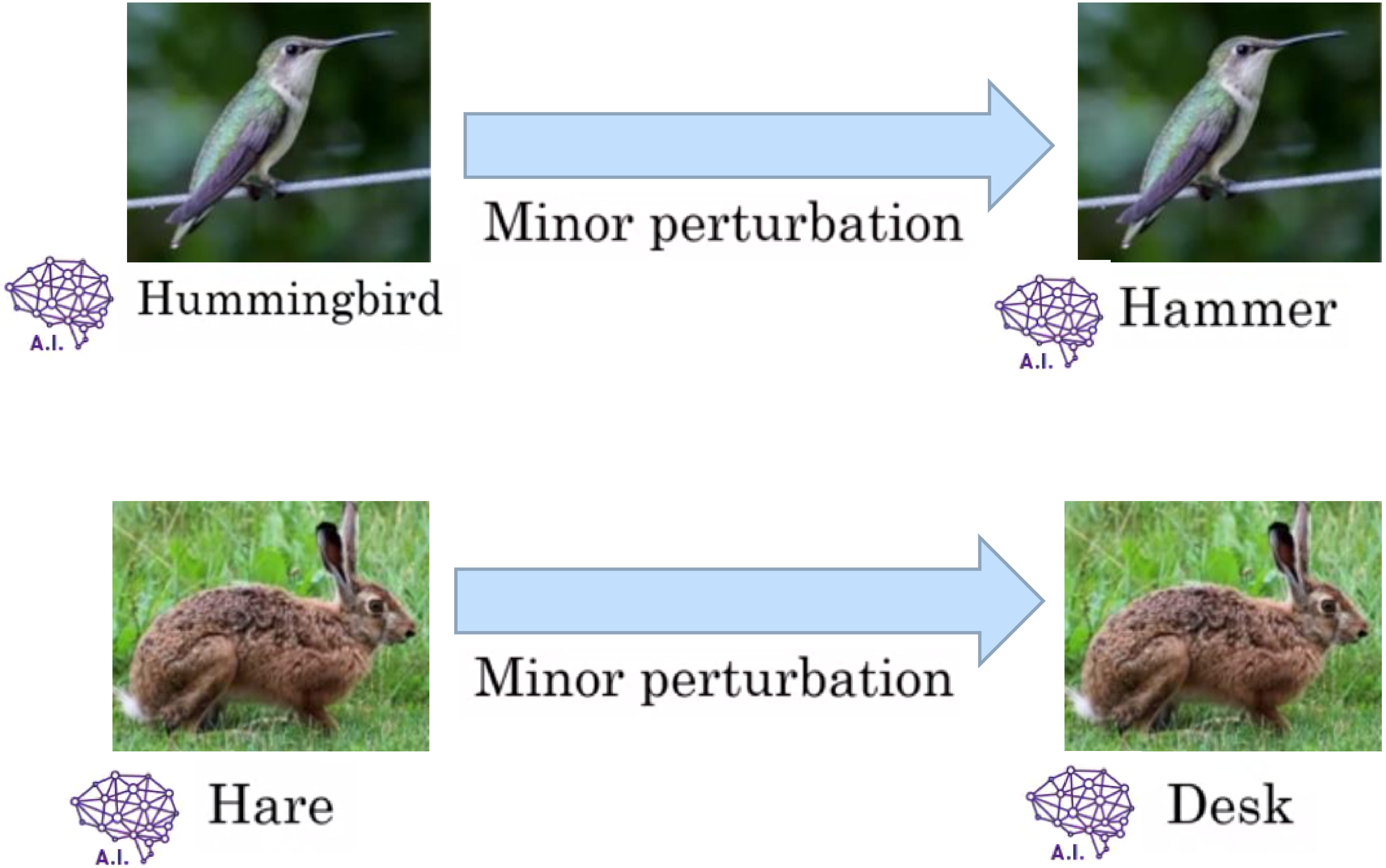
Neural Networks

***Deep Neural Networks** also known knows as convolutional neural networks are composed of multiple levels of nonlinear operations that aim at learning features hierarchies.*

Machine learning models based on deep neural networks (DNNs) have achieved significant improvements and surpassed human-level accuracy in various learning tasks, including object identification and segmentation, face recognition, speech and text processing, and variety of other tasks [1-5].



Robustness and Trustworthiness



Limitations of Deep Neural Networks

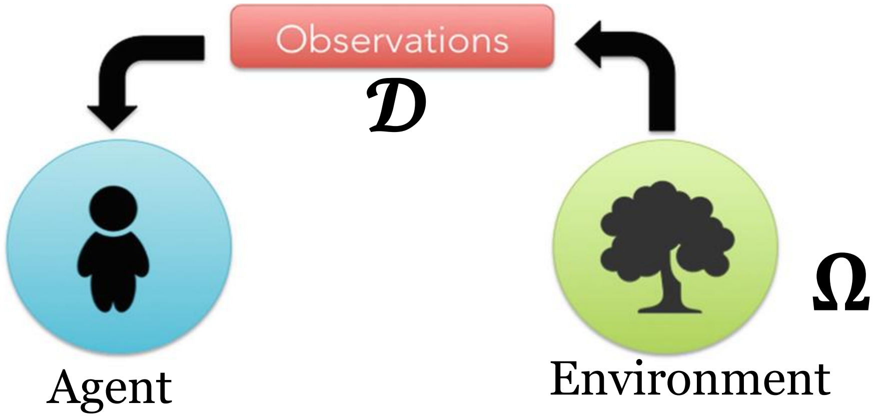
Detected as a speed sign

1. **<u>Lack of Uncertainty Estimation</u>**
   DNNs are unable to provide calibrated confidence or a measure of uncertainty in their predictions [6].

2. **<u>Vulnerability to Noise and Adversarial Attacks</u>**
   DNNs are vulnerable to noisy or perturbed inputs which might easily drive the model towards an incorrect prediction [7].

<u>Quantifying the confidence of a model's prediction is crucial in applications, where decision-making and control is handed over to autonomous systems, such as autonomous control of drones and self-driving cars and healthcare diagnosis systems.</u>

Bayesian Inference in Deep Neural Networks



Prior: $p(\boldsymbol{\Omega})$ prior knowledge about the parameters, $\boldsymbol{\Omega}$, before observing any data.

Likelihood: $p(\boldsymbol{\mathcal{D}}|\boldsymbol{\Omega})$ the process by which the data is generated given a particular $\boldsymbol{\Omega}$.

Posterior: $p(\boldsymbol{\Omega}|\boldsymbol{\mathcal{D}}) = \frac{p(\boldsymbol{\Omega})p(\boldsymbol{\mathcal{D}}|\boldsymbol{\Omega})}{\sum_{\Theta} p(\boldsymbol{\Omega})p(\boldsymbol{\mathcal{D}}|\boldsymbol{\Omega})}$ captures the total knowledge about $\boldsymbol{\Omega}$ after observing $\boldsymbol{\mathcal{D}}$.

- ❑ The posterior distribution of the parameters is used to find the predictive distribution of any new data point $\boldsymbol{\mathcal{X}}^*$ by marginalizing out the model's parameters,

$$p(\boldsymbol{y}^*|\boldsymbol{\mathcal{X}}^*, \boldsymbol{\mathcal{D}}) = \int p(\boldsymbol{y}^*|\boldsymbol{\mathcal{X}}^*, \boldsymbol{\Omega})\, p(\boldsymbol{\Omega}|\, \boldsymbol{\mathcal{D}}) d\boldsymbol{\Omega}$$

Variational Inference (VI) Framework

- Exact Bayesian inference on the parameters of a DNN is intractable due to the functional form of a DNN that consists of multiple layers of non-linearities and the high dimensionality of the parameter space [13].
- Various approaches have been proposed to approximate the posterior distribution of the weights given the data including the well-known Variational Inference (VI) [8 – 14].
- VI methods approximate the true posterior $p(\boldsymbol{\Omega}|\,\boldsymbol{\mathcal{D}})$ with a simpler parametrized variational distribution $q_\phi(\boldsymbol{\Omega})$.
- The optimal parameters of the variational posterior ϕ^* are found by minimizing the Kullback-Leibler (KL) divergence between the approximate and the true posterior,

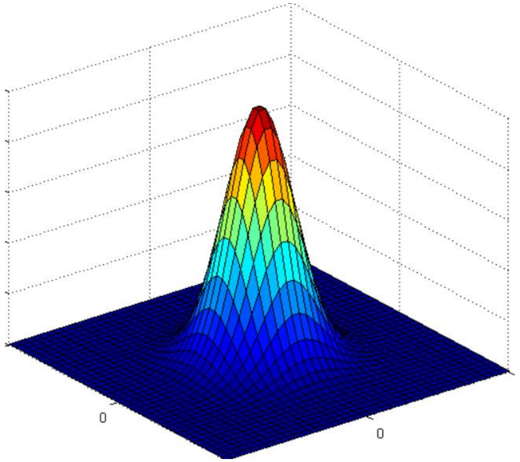
$$\begin{aligned}\phi^* &= \operatorname{argmin} \mathrm{KL}\left[q_\phi(\boldsymbol{\Omega})\,||p(\boldsymbol{\Omega}|\,\boldsymbol{\mathcal{D}})\right] \\ &= \operatorname{argmin} \mathrm{KL}\left[q_\phi(\boldsymbol{\Omega})\,||p(\boldsymbol{\Omega})\right] - E_{q_\phi(\boldsymbol{\Omega})}\{\log p(\boldsymbol{\mathcal{D}}\,|\boldsymbol{\Omega})\}\end{aligned}$$

- The optimization objective is given by the evidence lower bound (ELBO) $\mathcal{L}(\phi;\boldsymbol{\mathcal{D}})$

$$\mathcal{L}(\phi;\boldsymbol{\mathcal{D}}) = \underbrace{E_{q_\phi(\boldsymbol{\Omega})}\{\log p(\boldsymbol{\mathcal{D}}\,|\boldsymbol{\Omega})\}}_{\textbf{Log-Likelihood}} - \underbrace{\mathrm{KL}\left[q_\phi(\boldsymbol{\Omega})\,||p(\boldsymbol{\Omega})\right]}_{\textbf{Regularization}}$$

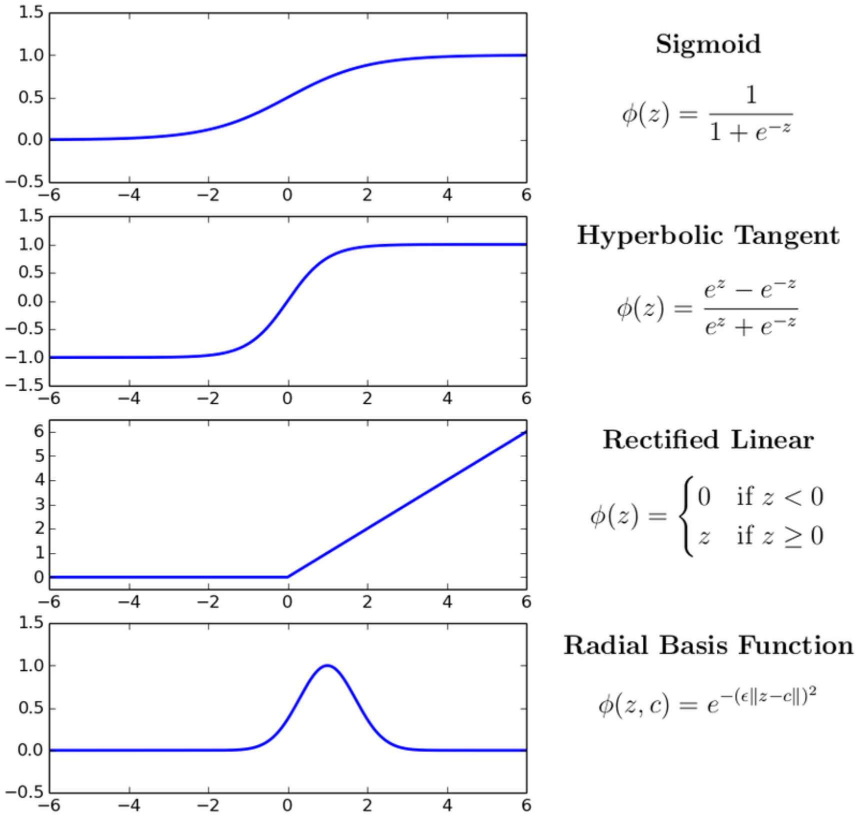
The Challenge in Density Propagation

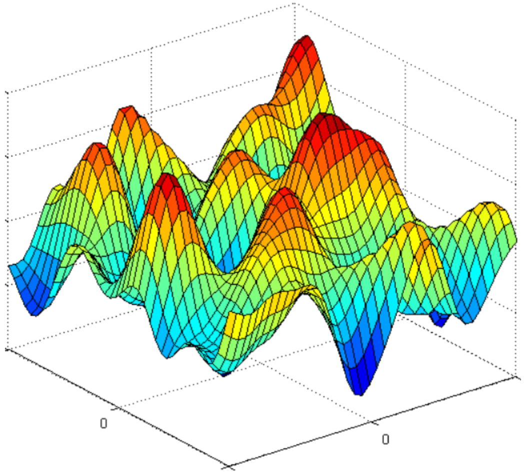
- The challenge remains in propagating the variational distribution $q_\phi(\boldsymbol{\Omega})$ over the parameters of a DNN through stacked layers of non-linearities.



Multivariate Gaussian Distribution

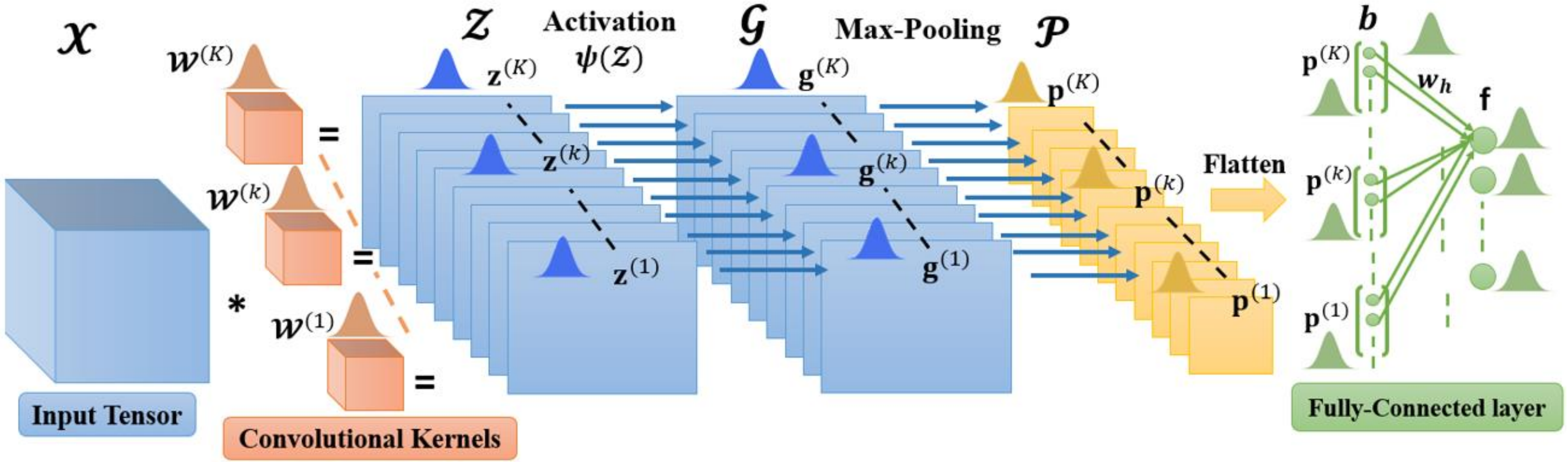
Nonlinear Activation Functions





Unknown Distribution

Variational Density Propagation – Convolutional Neural Network



We consider a convolutional neural network with:

- ❑ One convolutional layer
- ❑ Nonlinearity (e.g. ReLU activation),
- ❑ Max-pooling layer,
- ❑ One fully connected.

Extended Variational Density Propagation (*exVDP*)
Propagation of Mean and Covariance

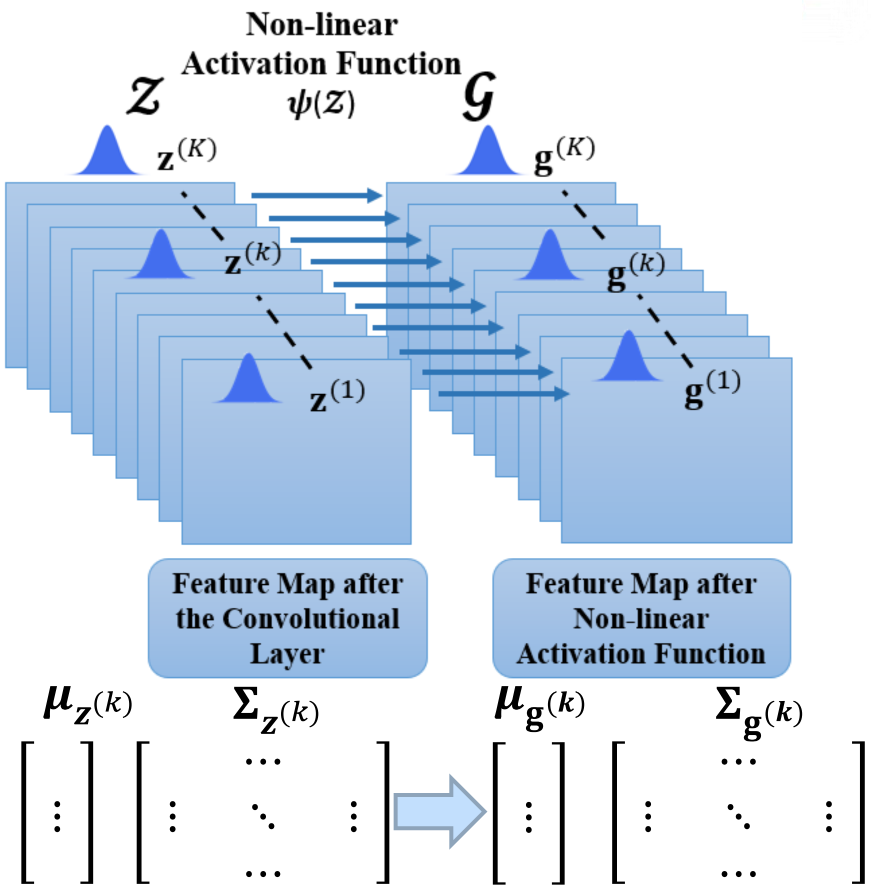
Non-linear activation layer – first-order Taylor approximation:

$$\boldsymbol{\psi}(z_i) = \boldsymbol{\psi}(\mu_{z_i}) + (z_i - \mu_{z_i})\frac{d\boldsymbol{\psi}(\mu_{z_i})}{dz_i} + \frac{1}{2!}(z_i - \mu_{z_i})^2 \frac{d^2\boldsymbol{\psi}(\mu_{z_i})}{dz_i^2} + \cdots$$

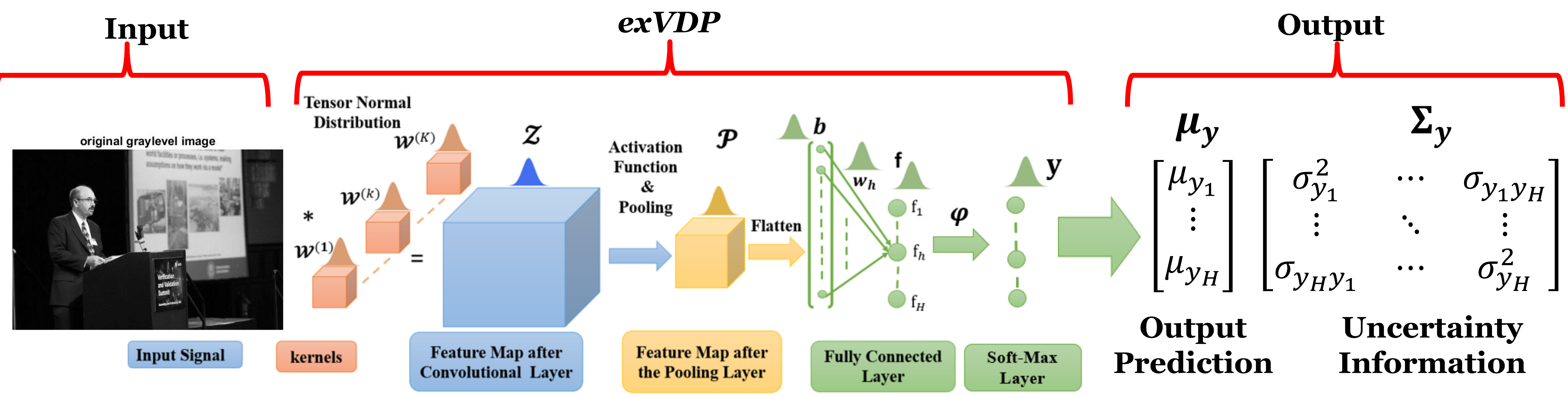
$$\mu_{g_i} = E(g_i) \approx \boldsymbol{\psi}(\mu_{z_i})$$

$$\boldsymbol{\Sigma}_{\mathbf{g}^{(k)}} = \begin{cases} \sigma_{g_i}^2 = Var(g_i) \approx \sigma_{z_i}^2 \left(\frac{d\,\boldsymbol{\psi}(\mu_{z_i})}{dz_i}\right)^2, & \text{if } i = j \\ \sigma_{g_i g_j} = Cov(g_i, g_j) \approx \sigma_{z_i z_j} \left(\frac{d\,\boldsymbol{\psi}(\mu_{z_i})}{dz_i}\right)\left(\frac{d\,\boldsymbol{\psi}(\mu_{z_j})}{dz_j}\right), & \text{if } i \neq j \end{cases}$$

g_i is the i^{th} element of the feature map, $\boldsymbol{\psi}$ is element-wise activation function.
We remove the superscript k for simplicity.



Extended Variational Density Propagation (*exVDP*)
Propagation of Mean and Covariance



Soft-Max Layer:

$$\boldsymbol{\mu}_{\mathbf{y}} \approx \varphi(\boldsymbol{\mu}_{\mathbf{f}}), \qquad \boldsymbol{\Sigma}_{\mathbf{y}} \approx \mathbf{J}_{\varphi}\, \boldsymbol{\Sigma}_{\mathbf{f}}\, \mathbf{J}_{\varphi}^{T}$$

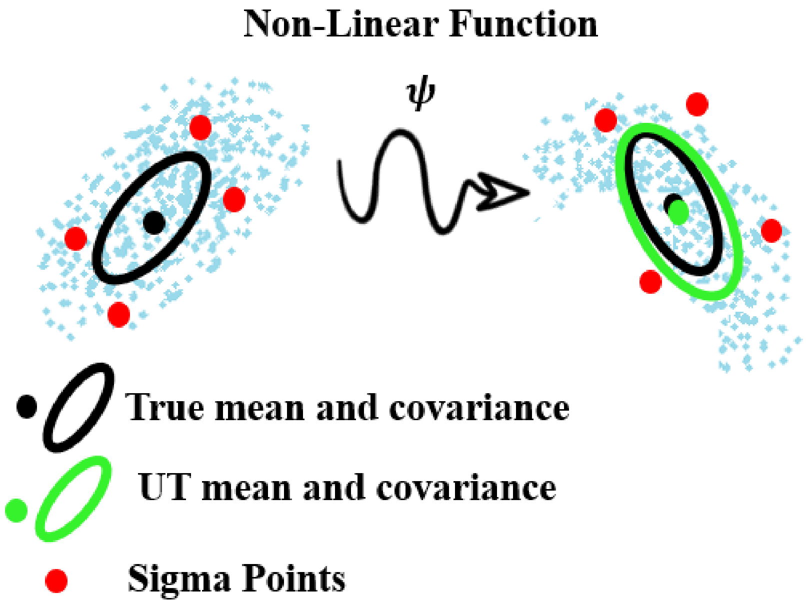
$$\mathbf{J}_{\varphi}(\boldsymbol{\mu}_{\mathbf{f}}) = \mathrm{diag}(\boldsymbol{\mu}_{\mathbf{y}}) - \boldsymbol{\mu}_{\mathbf{y}}\, \boldsymbol{\mu}_{\mathbf{y}}^{T}$$

where φ is the Soft-max function and $\mathbf{J}_{\varphi}$ is the Jacobian matrix of φ with respect to $\mathbf{f}$ evaluated at $\boldsymbol{\mu}_{\mathbf{f}}$.

Unscented Variational Density Propagation (*unVDP*)- Propagation of Sigma Points

Unscented Transformation:

- The linearization, performed in the exVDP propagation may result in accumulation of errors especially in deep neural networks with a large number of stacked non-linear activations.
- The unscented transformation (UT) can provide estimates of the mean and covariance after non-linear transformation which are correct at least up to the third order [15].
- In the UT framework, the probability density function (pdf) is specified using a set of carefully chosen samples, called sigma points.



Evidence Lower Bound (ELBO) Objective Function

$$\mathcal{L}(\phi; \boldsymbol{y} \,|\boldsymbol{\mathcal{X}}) = E_{q_\phi(\boldsymbol{\Omega})}\{\log p(\boldsymbol{y} \,|\boldsymbol{\mathcal{X}}, \boldsymbol{\Omega})\} - \mathrm{KL}\left[q_\phi(\boldsymbol{\Omega}) \,||p(\boldsymbol{\Omega})\right]$$

Backpropagation

- In the forward pass, we propagated the mean and covariance matrix of the variational distribution $q_\phi(\boldsymbol{\Omega})$ across the network layers and calculated the objective function $\mathcal{L}(\phi; \boldsymbol{y} \,|\boldsymbol{\mathcal{X}})$.

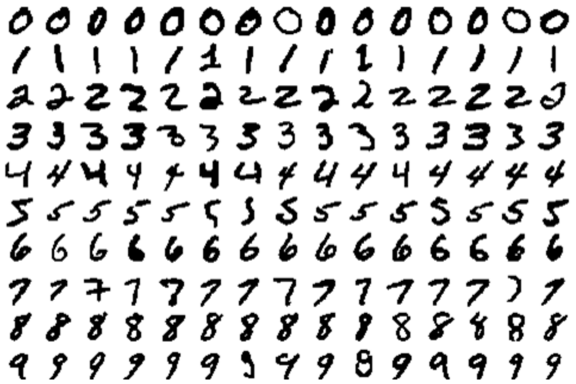
- In the back-propagation pass, we compute the gradient of the objective function $\nabla\mathcal{L}(\phi; \boldsymbol{y} \,|\boldsymbol{\mathcal{X}})$ w.r.t the variational parameters ϕ and update ϕ using the gradient descent update rule.

Simulation Results and Discussion

Image Classification on MNIST and CIFAR-10

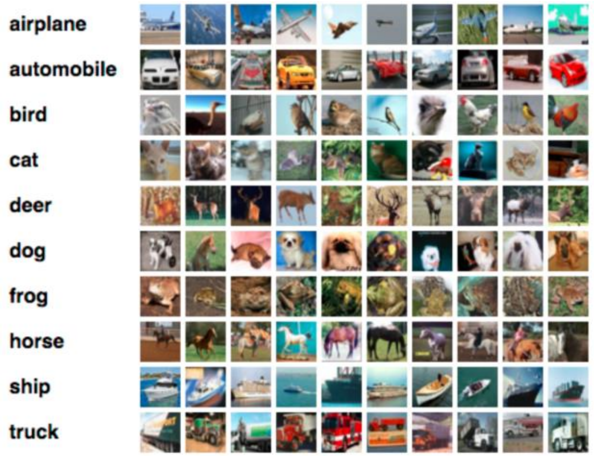
- We present test accuracy of *unVDP, exVDP* compared with Bayes-by-Backprop (BBB), and a deterministic CNN for the MNIST and with Bayes-CNN, and Dropout CNN for CIFAR-10 with varying levels of adversarial and Gaussian noise added to the test set.

MNIST Dataset



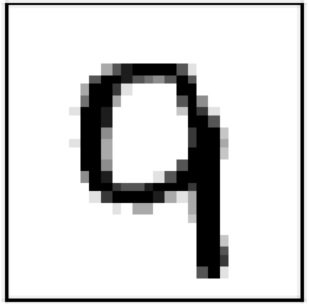
Gaussian noise level	unVDP	exVDP	BBB	CNN
Zero (No noise)	97.9%	97.8%	97.8%	97.7%
Low	95.1%	94.1%	86.4%	79.6%
Medium	86.7%	84.6%	76.7%	70.5%
High	74.8%	73.4%	63.8%	55.9%
Adversarial noise level				
Low	97.5%	96.6%	91.5%	58.7%
Medium	84.9%	84.4%	45.9%	14.7%
High	66.1%	51.6%	16.5%	14.5%

CIFAR-10 Dataset



Gaussian noise level	*unVDP*	*exVDP*	Bayes-CNN	Dropout CNN
Zero (No noise)	92.5%	91.8%	92.1%	91.0%
Low	92.3%	91.4%	87.0%	89.0%
Medium	91.9%	90.9%	86.8%	87.2%
High	90.1%	89.1%	85.2%	86.0%
Adversarial noise level				
Low	88.2%	88.1%	76.2%	77.0%
Medium	85.4%	82.3%	69.1%	53.0%
High	76.5%	67.7%	42.2%	33.0%

Self-Awareness and Robustness Analysis of the Output Covariance Matrix



True: 9, Pred: 9

Output Mean and Covariance Matrix of exVDP for Noise-free Input (Correctly Classified Input)

Output Covariance Matrix / Output Prediction

	0	1	2	3	4	5	6	7	8	9	0	
0	1.3E-12	-1E-20	6.7E-16	6.9E-15	3.4E-13	3.2E-14	2.9E-18	-7E-15	1.7E-16	8.3E-13	1.2E-08	**0**
1	-1E-20	5.3E-24	-5E-22	3.1E-21	4.6E-19	2.4E-20	-1E-25	-2E-19	1.1E-22	3.6E-19	2.4E-14	**1**
2	6.7E-16	-5E-22	3E-16	1.4E-16	-6E-15	-5E-16	2.9E-20	4.6E-16	6.8E-19	9.1E-15	1.8E-10	**2**
3	6.9E-15	3.1E-21	1.4E-16	1.7E-14	-7E-14	1.8E-15	2.8E-20	2.2E-14	3.1E-18	9.2E-14	1.3E-09	**3**
4	3.4E-13	4.6E-19	-6E-15	-7E-14	1.4E-10	8.4E-13	2.7E-17	-2E-13	2.5E-15	7.6E-12	1.2E-07	**4**
5	3.2E-14	2.4E-20	-5E-16	1.8E-15	8.4E-13	7.6E-13	1.8E-18	-5E-14	1.2E-16	4.8E-13	8.9E-09	**5**
6	2.9E-18	-1E-25	2.9E-20	2.8E-20	2.7E-17	1.8E-18	2.2E-21	-5E-18	9.5E-21	2.2E-17	5E-13	**6**
7	-7E-15	-2E-19	4.6E-16	2.2E-14	-2E-13	-5E-14	-5E-18	5.8E-12	-2E-16	1.9E-13	2.5E-08	**7**
8	1.7E-16	1.1E-22	6.8E-19	3.1E-18	2.5E-15	1.2E-16	9.5E-21	-2E-16	5.9E-18	1.8E-15	2.5E-11	**8**
9	8.3E-13	3.6E-19	9.1E-15	9.2E-14	7.6E-12	4.8E-13	2.2E-17	1.9E-13	1.8E-15	1.3E-10	1	**9**

Prediction of Deterministic CNN

8E-10	**0**
5E-13	**1**
6E-09	**2**
8E-06	**3**
5E-07	**4**
1E-08	**5**
1E-10	**6**
5E-07	**7**
2E-08	**8**
1	**9**

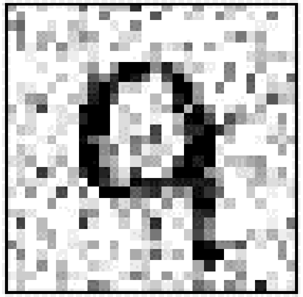
Ground Truth (yellow)
Network Prediction (green)

If the yellow block is not shown, then the network prediction and the ground truth are the same.

Self-Awareness and Robustness
Gaussian Noise

ICASSP 2022 SINGAPORE 22-27 MAY 2022

IEEE Signal Processing Society



True: 9, Pred: 9

Output Mean and Covariance Matrix of exVDP for Input Corrupted with Low level of Gaussian Noise (Correctly Classified Input)

Output Covariance Matrix

	0	1	2	3	4	5	6	7	8	9
0	0.0133	8E-07	0.0002	2E-05	0.0003	2E-05	3E-05	0.0001	0.00017	0.0029
1	8E-07	7E-08	1E-07	4E-08	5E-07	5E-08	6E-08	3E-07	2.99E-07	5E-06
2	0.0002	1E-07	0.0155	1E-05	0.0003	9E-07	2E-05	4E-05	0.000109	0.0013
3	2E-05	4E-08	1E-05	8E-05	6E-06	2E-06	1E-06	9E-06	7.25E-06	0.0002
4	0.0003	5E-07	0.0003	6E-06	0.1216	-2E-05	2E-05	0.0001	0.000373	0.0022
5	2E-05	5E-08	9E-07	2E-06	-2E-05	1E-04	8E-07	1E-05	8.07E-06	0.0003
6	3E-05	6E-08	2E-05	1E-06	2E-05	8E-07	0.0002	4E-06	1.33E-05	0.0001
7	0.0001	3E-07	4E-05	9E-06	0.0001	1E-05	4E-06	0.004	6.16E-05	0.0016
8	0.0002	3E-07	0.0001	7E-06	0.0004	8E-06	1E-05	6E-05	0.0034	0.0018
9	0.0029	5E-06	0.0013	0.0002	0.0022	0.0003	0.0001	0.0016	0.001753	0.5083

Output Prediction

0	
0.005	**0**
1E-05	**1**
0.0056	**2**
0.0004	**3**
0.0153	**4**
0.0004	**5**
0.0005	**6**
0.0028	**7**
0.0026	**8**
0.9674	**9**

Prediction of Deterministic CNN

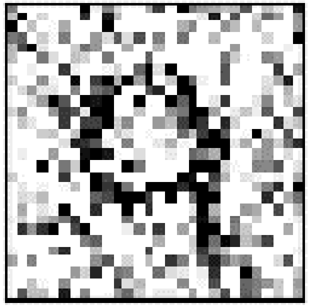
0.0009	**0**
6E-06	**1**
0.0019	**2**
0.0007	**3**
0.0234	**4**
0.0121	**5**
0.0006	**6**
0.0038	**7**
0.0037	**8**
0.9529	**9**

Ground Truth

Network Prediction

If the yellow block is not shown, then the network prediction and the ground truth are the same.

Self-Awareness and Robustness
Gaussian Noise



True: 9, Pred: 9

Output Mean and Covariance Matrix of exVDP for Input Corrupted with High level Gaussian Noise (Correctly Classified Input)

Prediction of Deterministic CNN

1.1E-11	**0**
3.9E-16	**1**
0.00083	**2**
3.1E-08	**3**
0.02638	**4**
0.00825	**5**
3.2E-18	**6**
8.5E-06	**7**
3.8E-06	**8**
0.96454	**9**

Output Covariance Matrix

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	0.72614	7.3E-06	0.02063	0.00011	0.00301	0.00041	5.4E-05	0.02911	0.00164	0.10533
1	7.3E-06	1.4E-06	5.3E-06	4.6E-08	4.9E-07	4.9E-07	2.9E-08	1.4E-05	1.2E-06	4.7E-05
2	0.02063	5.3E-06	6.30839	0.00028	0.00323	-2E-05	0.0001	0.04422	0.00198	0.1513
3	0.00011	4.6E-08	0.00028	0.00012	2.7E-06	6E-06	3.2E-07	0.00017	5.1E-06	0.00115
4	0.00301	4.9E-07	0.00323	2.7E-06	0.05826	-0.0001	9.8E-06	0.00286	0.00048	0.01931
5	0.00041	4.9E-07	-2E-05	6E-06	-0.0001	0.00451	1.1E-06	0.00222	1.7E-05	0.00518
6	5.4E-05	2.9E-08	0.0001	3.2E-07	9.8E-06	1.1E-06	1.2E-05	7E-05	5.2E-06	0.00024
7	0.02911	1.4E-05	0.04422	0.00017	0.00286	0.00222	7E-05	10.2073	0.00447	0.2822
8	0.00164	1.2E-06	0.00198	5.1E-06	0.00048	1.7E-05	5.2E-06	0.00447	0.00649	0.01209
9	0.10533	4.7E-05	0.1513	0.00115	0.01931	0.00518	0.00024	0.2822	0.01209	29.1693

Output Prediction

0	
0.03217	**0**
4.3E-05	**1**
0.10437	**2**
0.00042	**3**
0.00889	**4**
0.00248	**5**
0.00013	**6**
0.1372	**7**
0.00301	**8**
0.71129	**9**

Ground Truth
Network Prediction

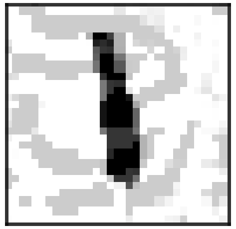
If the yellow block is not shown, then the network prediction and the ground truth are the same.

22-27 MAY 2022
ICASSP 2022
SINGAPORE

IEEE Signal Processing Society

Self-Awareness and Robustness Adversarial Noise

Example: Adversarial Noise (the targeted attack class is digit "3")



True: 1, Pred: 3

Prediction of Deterministic CNN

2E-13	0
0.0238	1
8E-08	2
0.9762	3
7E-10	4
5E-08	5
5E-10	6
1E-09	7
8E-06	8
2E-13	9

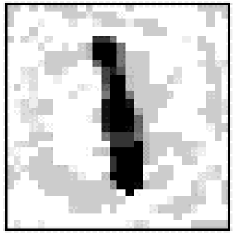
exVDP

Output Covariance Matrix

	0	1	2	3	4	5	6	7	8	9
0	2.8E-12	2E-07	1.1E-09	1.4E-07	2.3E-10	8E-11	4.2E-11	1.1E-09	1.8E-09	2.1E-11
1	2E-07	1.39853	0.00093	0.11747	0.0003	8.4E-05	4.7E-05	0.0006	0.00154	2E-05
2	1.1E-09	0.00093	7.9E-05	0.00079	1.3E-06	3.5E-07	2.9E-07	7.7E-07	6.6E-06	1.1E-07
3	1.4E-07	0.11747	0.00079	1.09673	0.00017	6.9E-05	3.8E-05	0.00068	0.00088	1.4E-05
4	2.3E-10	0.0003	1.3E-06	0.00017	4.4E-06	1.1E-07	1.1E-07	1.3E-06	2.2E-06	2.9E-08
5	8E-11	8.4E-05	3.5E-07	6.9E-05	1.1E-07	5.7E-07	1.8E-08	1.1E-07	7.6E-07	8.7E-09
6	4.2E-11	4.7E-05	2.9E-07	3.8E-05	1.1E-07	1.8E-08	1E-07	1.8E-07	3.7E-07	4.5E-09
7	1.1E-09	0.0006	7.7E-07	0.00068	1.3E-06	1.1E-07	1.8E-07	4.4E-05	6.9E-06	1E-07
8	1.8E-09	0.00154	6.6E-06	0.00088	2.2E-06	7.6E-07	3.7E-07	6.9E-06	0.00018	2E-07
9	2.1E-11	2E-05	1.1E-07	1.4E-05	2.9E-08	8.7E-09	4.5E-09	1E-07	2E-07	1.7E-08

Output Prediction

2E-07	0
0.1529	1
0.0011	2
0.8427	3
0.0002	4
9E-05	5
3E-05	6
0.0011	7
0.0018	8
2E-05	9



True: 1, Pred: 3

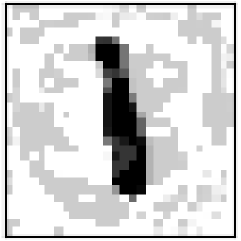
unVDP

Output Covariance Matrix

	0	1	2	3	4	5	6	7	8	9
0	4.35E-07	-0.00033	2.68E-07	0.000318	2.91E-06	2.88E-07	1.17E-06	7.16E-07	1.02E-05	6.49E-09
1	-0.00033	19.50457	-0.00061	-19.4682	-0.00653	-0.00066	-0.00265	-0.0016	-0.02399	-1.5E-05
2	2.68E-07	-0.00061	1.42E-06	0.000578	5.29E-06	5.18E-07	2.12E-06	1.29E-06	1.85E-05	1.18E-08
3	0.000318	-19.4682	0.000578	19.43496	0.006104	0.000625	0.002506	0.001518	0.021562	1.44E-05
4	2.91E-06	-0.00653	5.29E-06	0.006104	0.000171	5.64E-06	2.33E-05	1.39E-05	0.000202	1.29E-07
5	2.88E-07	-0.00066	5.18E-07	0.000625	5.64E-06	1.62E-06	2.27E-06	1.37E-06	1.99E-05	1.27E-08
6	1.17E-06	-0.00265	2.12E-06	0.002506	2.33E-05	2.27E-06	2.72E-05	5.58E-06	8.1E-05	5.14E-08
7	7.16E-07	-0.0016	1.29E-06	0.001518	1.39E-05	1.37E-06	5.58E-06	9.99E-06	4.86E-05	3.12E-08
8	1.02E-05	-0.02399	1.85E-05	0.021562	0.000202	1.99E-05	8.1E-05	4.86E-05	0.002048	4.51E-07
9	6.49E-09	-1.5E-05	1.18E-08	1.44E-05	1.29E-07	1.27E-08	5.14E-08	3.12E-08	4.51E-07	8.38E-10

Output Prediction

0	
4.1E-05	0
0.59416	1
7.5E-05	2
0.40145	3
0.00082	4
8E-05	5
0.00033	6
0.0002	7
0.00285	8
1.8E-06	9

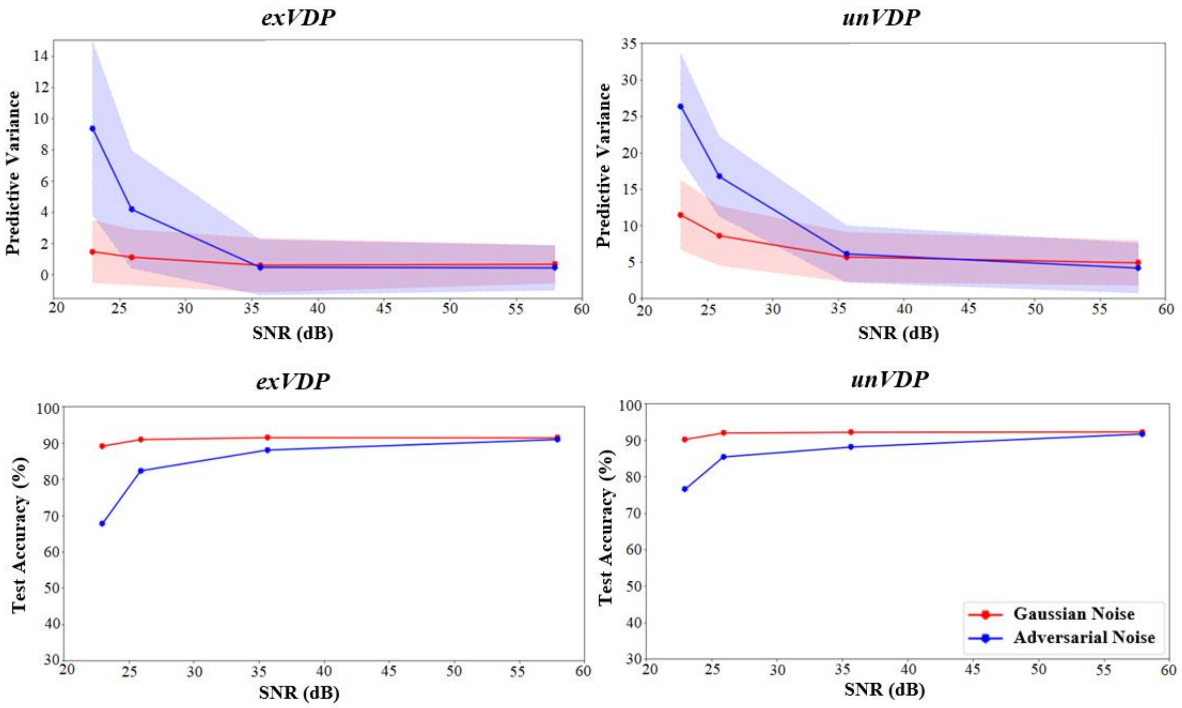


True: 1, Pred: 1

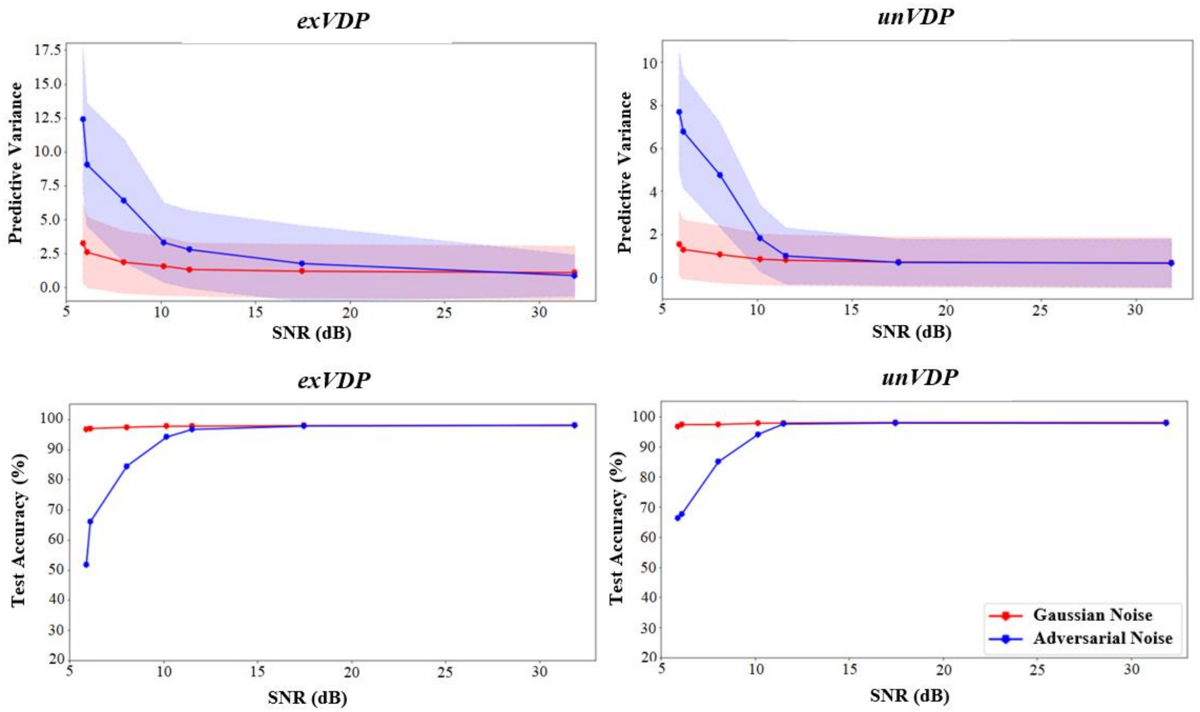
Ground Truth (yellow)
Network Prediction (green)

Self-Awareness and Robustness Analysis of the Output Variance

CIFAR-10 Dataset

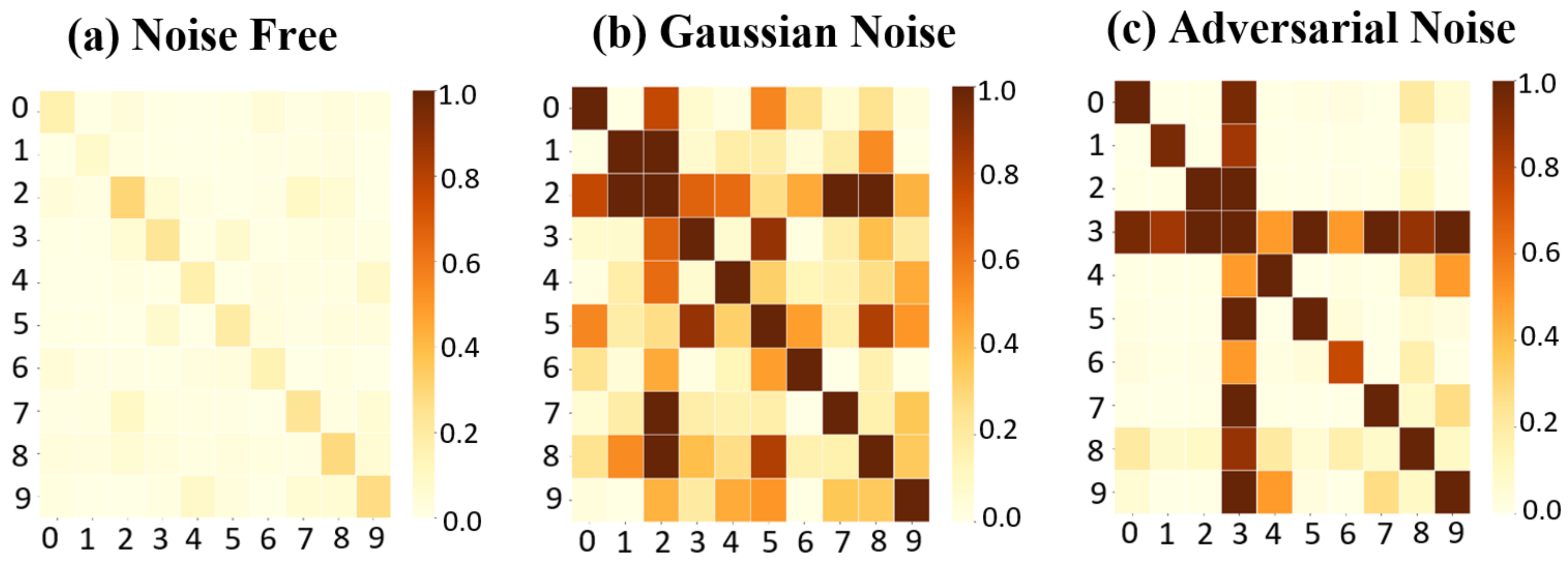


MNIST Dataset



Self-Awareness and Robustness
Analysis of the Output Covariance Matrix

- The heat-maps represent the average output covariance matrices of the *exVDP* model on MNIST dataset for three cases: (a) noise-free, (b) Gaussian noise, and (c) adversarial noise.
- The average test accuracies for the three cases were 97.8%, 84.6%, *and* 84.4%, respectively.
- Each pixel of the heat-map is a normalized average of absolute value of the covariance for all 10,000 test examples.
- The targeted adversarial examples were generated to fool the model into predicting digit "3" [16].

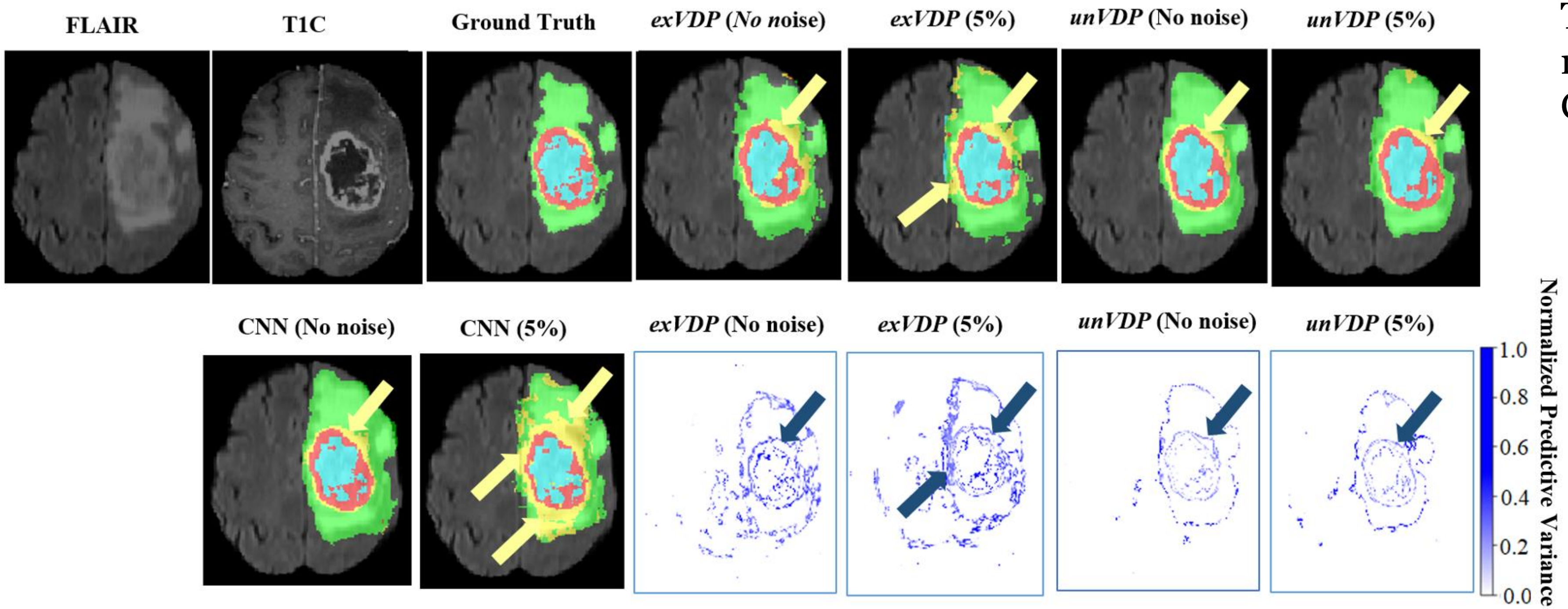


Application to Brain Tumor Segmentation in MRI Images

- We evaluate the performance of proposed *exVDP* and *unVDP* models on High Grade Glioma (HGG) brain tumor segmentation task using Brain Tumor Segmentation Challenge (BraTS) 2015 dataset [17].
- The uncertainty map will allow physicians to quickly review the segmentation results and, if needed, make corrections of tumor boundaries in the regions where the uncertainty is high.
- We evaluated the models before and after adding Gaussian noise or targeted adversarial attack (targeted class is class 3, i.e., “non-enhancing tumor”).

Method	Tumor Regions	Noise level		
		Zero (No noise)	Adversarial 5%	Gaussian 5%
unVDP	Complete	**85.3%**	**81.7%**	**83.0%**
	Core	81.9%	78.7%	80.7%
	Enhancing	83.7%	75.4%	81.7%
exVDP	Complete	**80.8%**	**77.4%**	**80.6%**
	Core	74.6%	72.6%	74.5%
	Enhancing	74.0%	69.8%	73.9%
CNN	Complete	78.0%	43.4%	66.9%
	Core	65.0%	47.1%	51.9%
	Enhancing	75.0%	43.9%	55.7%

The evaluation of the segmentation results was done using Dice Similarity Coefficient (DSC).



References

1. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proceedings of the 25th International Conference on Neural Information Processing Systems,(NIPS), 2012, pp. 1097–1105.
2. R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR). IEEE Computer Society, 2014, pp. 580–587.
3. E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 4, pp. 640–651, 2017.
4. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, S. Watanabe, T. Yoshimura, and W. Zhang, “A comparative study on transformer vs RNN in speech applications,” in2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), 2019, pp. 449–456.
5. K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, “Deep reinforcement learning: A brief survey,” IEEE Signal Processing Magazine, vol. 34, no. 6, pp. 26–38, 2017.
6. C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in Proceedings of the 34th International Conference on Machine Learning, ICML, 2017, pp. 1321–1330.
7. S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, “Universal adversarial perturbations,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR). IEEE Computer Society, 2017, pp. 86–94.
8. C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, “Weight uncertainty in neural networks,” in Proceedings of the 32nd International Conference on International Conference on Machine Learning, (ICML),vol. 37, 2015, pp. 1613–1622.
9. Y. Gal and Z. Ghahramani, “Bayesian convolutional neural networks with Bernoulli approximate variational inference,” in Proceedings of 4thInternational Conference on Learning Representations, (ICLR) work-shop track, 2016.
10. K. Shridhar, F. Laumann, A. Llopart Maurin, and M. Liwicki, “Bayesian convolutional neural networks,”arXiv preprint arXiv:1806.05978, 2018.
11. C. Louizos and M. Welling, “Structured and efficient variational deep learning with matrix Gaussian posteriors,” in Proceedings of the 33rdInternational Conference on International Conference on Machine Learning, (ICML), 2016, pp. 1708–1716.
12. W. Roth and F. Pernkopf, “Variational inference in neural networks using an approximate closed-form objective,” in Neural Information Processing Systems, (NIPS) workshop, 2016.
13. C. Louizos and M. Welling, “Multiplicative normalizing flows for variational Bayesian neural networks,” in Proceedings of the 34thInternational Conference on Machine Learning, (ICML), vol. 70, 2017,pp. 2218–2227.
14. A. Wu, S. Nowozin, E. Meeds, R. E. Turner, J. M. Hernandez-Lobato, and A. L. Gaunt, “Deterministic variational inference for robust Bayesian neural networks,” in Proceedings of 7th International Conference on Learning Representations, (ICLR), 2019.
15. S. J. Julier and J. K. Uhlmann, “A new extension of the Kalman filter to nonlinear systems,” in In Proceedings of SPIE - The International Society for Optical Engineering, vol. 3068, February 1999.
16. Y. Liu, X. Chen, C. Liu, and D. Song, “Delving into transferable adversarial examples and black-box attacks,” in Proceedings of 5thInternational Conference on Learning Representations, (ICLR), 2017.
17. B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani,J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiestet al., “The multimodal brain tumor image segmentation benchmark (BRATS),”IEEE Transactions on Medical Imaging, vol. 34, no. 10, pp. 1993–2024,2014.

ICASSP 2022 SINGAPORE

22-27 MAY 2022

IEEE Signal Processing Society

Acknowledgment

The authors would like to thank the financial support of

- NSF ECCS 1903466
- NSF CCF-1527822
- NSF DUE-1610911
- NSF CRII: RI-2153413

National Science Foundation
WHERE DISCOVERIES BEGIN

- Lockheed Martin Inc.

LOCKHEED MARTIN

- ACM SIGHPC/Intel Computational & Data Science Fellowship

sighpc | intel

ACM SIGHPC/Intel Computational Data and Science Fellows