11-30-2021

Trustworthy Medical Segmentation with Uncertainty Estimation

Giuseppina Carannante

Dimah Dera
The University of Texas Rio Grande Valley, dimah.dera@utrgv.edu

Nidhal C. Bouaynaya

Rasool Ghulam

Hassan M. Fathallah-Shaykh

Recommended Citation

Carannante, Giuseppina, et al. "Trustworthy Medical Segmentation with Uncertainty Estimation." arXiv preprint arXiv:2111.05978 (2021).

Trustworthy Medical Segmentation with Uncertainty Estimation

Giuseppina Carannante, *Member, IEEE,* Dimah Dera, *Member, IEEE,* Nidhal C. Bouaynaya, *Member, IEEE,* Ghulam Rasool, *Member, IEEE,* and Hassan M. Fathallah-Shaykh

***Abstract*—Deep Learning (DL) holds great promise in reshaping the healthcare systems given its precision, efficiency, and objectivity. However, the brittleness of DL models to noisy and out-of-distribution inputs is ailing their deployment in the clinic. Most systems produce point estimates without further information about model uncertainty or confidence. This paper introduces a new Bayesian deep learning framework for uncertainty quantification in segmentation neural networks, specifically encoder-decoder architectures. The proposed framework uses the first-order Taylor series approximation to propagate and learn the first two moments (mean and covariance) of the distribution of the model parameters given the training data by maximizing the evidence lower bound. The output consists of two maps: the segmented image and the uncertainty map of the segmentation. The uncertainty in the segmentation decisions is captured by the covariance matrix of the predictive distribution. We evaluate the proposed framework on medical image segmentation data from Magnetic Resonances Imaging and Computed Tomography scans. Our experiments on multiple benchmark datasets demonstrate that the proposed framework is more robust to noise and adversarial attacks as compared to state-of-the-art segmentation models. Moreover, the uncertainty map of the proposed framework associates low confidence (or equivalently high uncertainty) to patches in the test input images that are corrupted with noise, artifacts or adversarial attacks. Thus, the model can self-assess its segmentation decisions when it makes an erroneous prediction or misses part of the segmentation structures, e.g., tumor, by presenting higher values in the uncertainty map.**

***Index Terms*—Bayesian deep learning, encoder-decoder networks, reliability, segmentation, trustworthiness, uncertainty estimation.**

I. Introduction

DRIVEN by the superior performance achieved in many areas, various deep learning (DL) models have been advanced to analyze medical data, e.g., radiological images and pathology slides. Several methods have achieved, if not surpassed, prognosis parity with specialized medical personnel [1]–[4]. However, their successful deployment in clinical settings remains limited. While several autonomous algorithms are doubtlessly employed for many everyday tasks (e.g., spam filters for emails or biometrics that unlock our cellphones), there is less assertive willingness to utilize the same algorithms for risky, sensitive data, such as medical images.

The main challenge that hinders the widespread and effective use of DL in clinical settings is the lack of reliable and trustworthy predictions [5], [6]. For example, a system encountering unseen test examples, which lie outside of its training data distribution, could easily make unreasonable suggestions, and as a result, unjustifiably bias the human expert. The need for reliable models is further exacerbated owing to recent studies showing the vulnerability of DL models to adversarial inputs — perturbations that are imperceptible and would not mislead the decisions of a human observer but force a trained DL model to make erroneous predictions [7]. Specifically, some studies have demonstrated the vulnerability of medical models to adversarial perturbations [8], [9].

For the successful deployment of DL models in the real-world, e.g, a clinic, these models should provide information on the trustworthiness of their predictions. The user of these models should be aware of the level of confidence in the models' predictions. Such information could be very useful when the DL model is essentially guessing at random due to excessive noise in the input or possible adversarial attack. Unfortunately, as most DL models are inherently deterministic, a measure of confidence or uncertainty is not readily available at their output.

Estimating the confidence of a model requires a probabilistic interpretation of the model's parameters, i.e., treating model parameters as random variables endowed with a probability distribution. Through Bayesian inference, the posterior distribution of the model parameters can be found. At test time, the second moment, i.e., the variance, of the predictive distribution can serve as a measure of confidence or uncertainty in the predicted output. Several Bayesian models have been developed for the classification and regression problems [10]–[15]. Trade-offs between prediction accuracy, confidence estimation, and scalability are at the heart of these different approaches [16]. Recently, Dera *et al.* proposed a variational moments' propagation (VMP) framework that provides meaningful and scalable methods for uncertainty propagation and estimation in deep neural network (DNN) classifiers [14].

A relatively less amount of work focuses on quantifying uncertainty in pixel-level segmentation tasks using Bayesian DL models. The challenge in learning uncertainty for each pixel arises from propagating high-dimensional posterior distributions of the model's parameters through the multiple stages of non-linearities in the encoder-decoder architecture. Furthermore, the model must provide an *instantaneous* uncer-

G. Carannnate, N. C. Bouaynaya and G. Rasool are with the Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ, 08028 USA (email: carannang1@rowan.edu, bouaynaya@rowan.eduand rasool@rowan.edu.)

D. Dera is with the Department of Electrical and Computer Engineering, University of Texas Rio Grande Valley, TX, 78520 USA (email: dimah.dera@utrgv.edu).

H. M. Fathallah-Shaykh is with the Department of of Neurology, University of Alabama at Birmingham School of Medicine, Birmingham, AL 35294-1170 USA (email: hfshaykh@uabmc.edu).

tainty map at test time, i.e., simultaneously output the prediction (the segmentation decision) and corresponding pixel-level uncertainty map *without* resorting to expensive Monte Carlo estimation techniques.

In this paper, we develop a VMP framework for segmentation tasks and apply it to various medical imaging datasets. By leveraging key concepts from probability density tracking in nonlinear and non-Gaussian systems [17], [18], we propagate the first and the second moments of the posterior distribution of network parameters through the nonlinear layers of an encoder-decoder type segmentation architecture. The developed approach is tested using various medical segmentation datasets consisting of Magnetic Resonance Images (MRI) and Computed Tomography (CT) scans. The proposed VMP formulation and the derived mathematical relationships presented in the paper are applicable to various DNN architectures.

The contributions of this paper are summarized as follows:

(1) Formalize a scalable Bayesian framework that learns model confidence in the encoder-decoder segmentation networks by maximizing the evidence lower bound (ELBO). Using the first-order Taylor series approximation, we propagate (forward and backward) and learn the first two moments (mean and covariance) of the posterior distribution of the model parameters given the training data. We derive mathematical relations for all operations; thus, rendering a method that is adaptable to other models, e.g., Variational Autoencoders, and to other tasks as well;

(2) develop a Bayesian DL architecture that instantaneously outputs two maps, (1) the segmented image, and (2) the uncertainty map of the predicted segmentation. These two maps are delivered simultaneously and *without* requiring any Monte Carlo sampling at the inference time. The uncertainty map is especially valuable for integrating the proposed segmentation model in critical areas of application, e.g., disease diagnosis and surveillance;

(3) evaluate the performance of the proposed approach for various medical segmentation tasks. A thorough robustness analysis is conducted by assessing the performance of the proposed Bayesian segmentation framework when the test data is affected by random noise and adversarial attacks.

II. Related Work

Image segmentation is a fundamental problem in computer vision and application areas range from medical image analysis to scene understanding for autonomous vehicles. Several DL-based techniques have been proposed over the years to tackle pixel-level segmentation; a detailed review of all such methods is out of the scope of this paper and the reader is referred to a recent survey on the topic [19].

The success of Convolutional Neural Networks (CNNs) in classification tasks has led researchers to extend these models for performing segmentation. Fully Convolutional Networks (FCNs) represent a powerful class of CNNs that are trained end-to-end to perform pixel-wise predictions. Specifically, a standard CNN architecture was moderately modified to perform pixel-wise classification by deleting fully-connected layers, adding an upsampling step, and employing a pixel-wise loss function [20]–[24]. For example, ResNet [25] and DenseNet [26] have been extended by adding an upsampling path to perform pixel-level segmentation [21], [22], [24].

More recently, the *encoder-decoder* types of architecture have gained popularity [27], [28]. The *encoder* path reduces the dimensionality of the input by extracting low-dimensional (salient) features of the data, while the *decoder* expands the encoded features to a map that has the same size as the input to perform pixel-wise classification. Several works have focused on modifying and fine-tuning the encoder-decoder architecture to improve segmentation, e.g., by introducing skip connections, dilated convolutions, wide contest blocks or compression extraction modules [27]–[31].

Generally, the machine learning community has focused its efforts on producing accurate point estimates for the segmentation and less on investigating and improving the reliability and trustworthiness of these models. Evidently, an unreliable model is vulnerable and can jeopardize the clinical system by exposing it to possible fraud, large monetary losses, technical vulnerabilities, lawsuits, and even loss of lives [9].

In the context of semantic segmentation, estimating pixel-wise uncertainty has rarely been explored [32]–[37]. Monte Carlo Dropout (MC-Dropout) is the most popular approach as it does not require major changes in the neural network architecture [32]–[35]. The uncertainty information is obtained from the variance of multiple MC forward passes through the network at inference time. Uncertainty estimation in segmentation has also been tackled by ensemble methods [38]. After training multiple networks with random initialization, several estimates (of segmentation) are produced, and their variation is used as a measure of confidence. The concept of generating several segmentation samples was leveraged in various ways, e.g., by combining hierarchical probabilistic models with variational autoencoders or by using data augmentation at test time [36], [37].

All the above approaches from the literature for quantifying uncertainty in segmentation networks adopt a frequentist approach where the uncertainty is computed at test time from the sample variance of multiple runs through the network. The segmentation networks are not "trained" to learn uncertainty or variance as a network parameter. This gap is due to the mathematical challenges in propagating the posterior distribution or its moments, e.g., mean and variance, through the multiple (non)linear layers of a network. Building upon the work in [14], we develop a Bayesian framework that propagates (forward pass and backpropagation) the first and the second moment of the variational posterior distribution across all layers of a segmentation DL model. At test time, the uncertainty in the predicted segmentation decision is produced by the network as the covariance matrix of the predictive distribution simultaneously alongside the segmentation.

III. Variational Moment Propagation (VMP) in Segmentation

The proposed VMP framework for segmentation consists of an encoder and a decoder. In the following, we present our mathematical results for various operations performed in the encoder and decoder.

A. Mathematical Notations

Scalars are represented by lower-case letters, e.g., x, x_i. Vectors are represented by bold lower-case letters, e.g., $\mathbf{y}$. All vectors are column vectors. y_i denotes the i^{th} element of vector $\mathbf{y}$. Matrices are represented by bold upper-case letters, e.g., $\mathbf{A}$. $\mathrm{Tr}(\cdot)$ denotes the trace of a matrix, i.e., the sum of its diagonal elements. ^{T} denotes the transpose operator, and $\mathrm{vec}(\cdot)$ denotes the vectorization operator. The Hadamard product, i.e., the element-wise product, is denoted with $\odot$, while $\times$ represents matrix-matrix or matrix-vector product. Tensors with three or more dimensions are represented by curly bold upper-case letters, e.g., $\boldsymbol{\mathcal{X}}$. If x is a random variable, $\mathbb{E}[x]$ denotes the expected value of x. We use $\boldsymbol{\mathcal{W}}_e^{(k_c)}$ to represent k_c^{th} convolutional kernel of the c^{th} layer. K_c denotes the total number of kernels in layer c. The subscripts e and d represent the encoder and decoder path operations, respectively.

B. Bayesian Deep Learning and Variational Inference

In Bayesian statistics, the unknown parameters are fully characterized by their posterior distribution given the observations. In Bayesian DL, the network parameters Ω are endowed with a prior probability distribution $p(\Omega)$ and all information about the parameters is embedded in the posterior distribution $p(\Omega|\mathcal{D})$ given the (training) data $\mathcal{D} = \{\boldsymbol{\mathcal{X}}^i, \mathbf{y}^i\}_{i=1}^N$. Once the posterior is estimated, the predictive distribution, i.e., the distribution of the test data, can be derived as:

$$p(\mathbf{y}^*|\boldsymbol{\mathcal{X}}^*, \mathcal{D}) = \int p(\mathbf{y}^*|\boldsymbol{\mathcal{X}}^*, \Omega)\ p(\Omega|\mathcal{D})\ d\Omega, \tag{1}$$

where $\boldsymbol{\mathcal{X}}^*$ is the input, $\mathbf{y}^*$ is its corresponding predicted output and $p(\mathbf{y}^*|\boldsymbol{\mathcal{X}}^*, \Omega)$ is the likelihood.

Unfortunately, direct inference of the posterior is intractable due to the large parameter space and nonlinear nature of DL architectures. A popular approximation technique known as Variational Inference (VI) formulates the problem of high-dimensional posterior inference as an optimization problem [39]. The VI approach considers a simple family of distributions over the network parameters and attempts to find a distribution, called the *variational distribution* $q_{\boldsymbol{\theta}}(\Omega)$, within this family that is "close" to the *true* unknown posterior. The notion of distributional closeness is captured by the Kullback-Leibler (KL) divergence and the optimization is performed with respect to the variational distribution parameters $\boldsymbol{\theta}$:

$$\mathbf{KL}\left(q_{\boldsymbol{\theta}}(\Omega)\big|\big|p(\Omega|\mathcal{D})\right) = \int q_{\boldsymbol{\theta}}(\Omega) \log \frac{q_{\boldsymbol{\theta}}(\Omega)}{p(\Omega)p(\mathcal{D}|\Omega)} d\Omega. \tag{2}$$

By rearranging terms in (2), the well-known Evidence Lower Bound (ELBO) objective function is obtained [13]:

$$\mathcal{L}(\boldsymbol{\theta}) = -\ \mathbb{E}_{q_{\boldsymbol{\theta}}(\Omega)}\left[\log(p(\mathcal{D}|\Omega)\right] + \mathbf{KL}\left(q_{\boldsymbol{\theta}}(\Omega)\big|\big|p(\Omega)\right). \tag{3}$$

Most Bayesian DL frameworks that use the VI approach sample one set of parameters $\boldsymbol{\theta}$ and perform a deterministic forward pass and backpropagation. The second moment or the variance of the predictive distribution is obtained using MC samples at the inference time [40]. This practice is based on the assumption that the single set of sampled parameters $\boldsymbol{\theta}$ represents the variational distribution $q_{\boldsymbol{\theta}}(\Omega)$ with sufficient accuracy, which may not be the case [14].

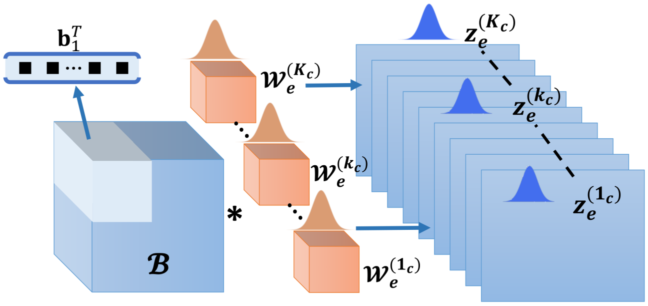
$\mathbf{b}_1^T$
$\boldsymbol{\mathcal{B}}$
$*$
$\boldsymbol{\mathcal{W}}_e^{(K_c)}$
$\boldsymbol{\mathcal{W}}_e^{(k_c)}$
$\boldsymbol{\mathcal{W}}_e^{(1_c)}$
$\mathbf{z}_e^{(K_c)}$
$\mathbf{z}_e^{(k_c)}$
$\mathbf{z}_e^{(1_c)}$

Fig. 1. An illustration of the convolution operation. We extract sub-tensors $\mathbf{b}_j$ from the input tensor $\boldsymbol{\mathcal{B}}$ having the same size of convolutional kernels $\boldsymbol{\mathcal{W}}_e^{(k_c)}$. Convolution operation is performed as a matrix-vector multiplication. The resulting feature maps $\mathbf{z}_e^{(k_c)}$ are random variables represented by the mean $\boldsymbol{\mu}_{\mathbf{z}_e^{(k_c)}}$ and the covariance $\boldsymbol{\Sigma}_{\mathbf{z}_e^{(k_c)}}$ for $k_c = 1, \cdots, K_c$.

C. Encoder Operations

We define a multivariate Gaussian distribution as a prior distribution for all convolution kernels. We assume that kernels are independent within each layer as well as across layers in both the encoder and decoder paths. The independence assumption results in a single additional parameter (i.e., variance) for each kernel, limiting the increase in the number of parameters due to the Bayesian formulation. Moreover, independent kernels help extract uncorrelated features and better explore the input space [14].

1) Convolution Between Input and Network Parameters: The convolution operation in the first layer is performed between the input data (assumed deterministic for simplicity) and network parameters (random variables). We assume that network parameters $\boldsymbol{\mathcal{W}}_e^{(k_1)}$ follow a Gaussian distribution, i.e., $\mathrm{vec}(\boldsymbol{\mathcal{W}}_e^{(k_1)}) \sim \mathcal{N}\left(\mathbf{m}_e^{(k_1)}, \boldsymbol{\Sigma}_e^{(k_1)}\right)$. We write the convolution as a matrix-vector multiplication, where $\mathbf{X}$ denotes the matrix having rows equal to the vectorized sub-tensors of the input $\boldsymbol{\mathcal{X}}$. Then, the convolution operation is expressed as $\mathbf{z}_e^{(k_1)} = \mathbf{X} \times \mathrm{vec}(\boldsymbol{\mathcal{W}}_e^{(k_1)})$, for $k_1 = 1, \cdots, K_1$. Thus, the output of the first convolutional layer follows a Gaussian distribution where the mean and covariance are given by:

$$\mathbf{z}_e^{(k_1)} \sim \mathcal{N}\left(\mathbf{X}\mathbf{m}_e^{(k_1)},\ \mathbf{X}\boldsymbol{\Sigma}_e^{(k_1)}\mathbf{X}^T\right). \tag{4}$$

2) Convolution Between Two Random Variables: We consider a generic case of convolution between two random variables. Let $\boldsymbol{\mathcal{B}}$ be the incoming input to any convolution layer, except the first layer, i.e., $c \neq 1$. The convolution operation is expressed as a matrix-vector multiplication; however, in this case both the input and the kernels are random tensors. We form $\mathbf{B}$ by vectorizing the sub-tensors of the incoming input $\boldsymbol{\mathcal{B}}$, i.e., $\mathbf{B} = [\mathbf{b}_1^T, \mathbf{b}_2^T, \cdots, \mathbf{b}_J^T]^T$, where $\mathbf{b}_j^T$ represents j^{th} row of $\mathbf{B}$. Let $\boldsymbol{\mu}_{\mathbf{b}_j}$ and $\boldsymbol{\Sigma}_{\mathbf{b}_j}$ represent the mean and covariance of $\mathbf{b}_j$. Then, the output of the convolution is formulated as $\mathbf{z}_e^{(k_c)} = \mathbf{B} \times \mathrm{vec}(\boldsymbol{\mathcal{W}}_e^{(k_c)})$ with $\mathrm{vec}(\boldsymbol{\mathcal{W}}_e^{(k_c)}) \sim \mathcal{N}\left(\mathbf{m}_e^{(k_c)}, \boldsymbol{\Sigma}_e^{(k_c)}\right)$ for $k_c = 1, \cdots, K_c$. Given that the input $\boldsymbol{\mathcal{B}}$ (known as a feature map) is independent from the subsequent layer kernels, we compute elements of the mean of $\mathbf{z}_e^{(k_c)}$ as the product of the two mean vectors, $\boldsymbol{\mu}_{\mathbf{b}_j}$ and $\mathbf{m}_e^{(k_c)}$, i.e.,

$$[\boldsymbol{\mu}_{\mathbf{z}_e^{(k_c)}}]_j = \boldsymbol{\mu}_{\mathbf{b}_j}^T \mathbf{m}_e^{(k_c)}, \quad j = 1, \cdots, J. \tag{5}$$

The elements of the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{z}_e^{(k_c)}}$ are derived as:

$$\text{Non-diagonal elements} \quad (i \neq j):$$

$$\boldsymbol{\mu}_{\mathbf{b}_i}^T \boldsymbol{\Sigma}_e^{(k_c)} \boldsymbol{\mu}_{\mathbf{b}_j}, \tag{6}$$

$$\text{Diagonal elements} \quad (i = j):$$

$$\text{Tr}\left(\boldsymbol{\Sigma}_{\mathbf{b}_i} \boldsymbol{\Sigma}_e^{(k_c)}\right) + \boldsymbol{\mu}_{\mathbf{b}_i}^T \boldsymbol{\Sigma}_e^{(k_c)} \boldsymbol{\mu}_{\mathbf{b}_j} + \mathbf{m}_e^{(k_c)T} \boldsymbol{\Sigma}_{\mathbf{b}_j} \mathbf{m}_e^{(k_c)}. \tag{7}$$

We illustrate the convolution layer in Fig. 1.

3) Nonlinear Activation Function: Convolutional layers are commonly followed by an element-wise nonlinear activation function, e.g. Rectified Linear Unit (ReLU). Let ψ denote the activation function and $\mathbf{g}_e^{(k_c)}$ denote the output of the activation function, i.e., $\mathbf{g}_e^{(k_c)} = \psi[\mathbf{z}_e^{(k_c)}]$ for $k_c = 1, \cdots, K_c$. We use the first-order Taylor series approximation to derive the mean and covariance of the random variable $\mathbf{g}_e^{(k_c)}$, i.e.,

$$\boldsymbol{\mu}_{\mathbf{g}_e^{(k_c)}} \approx \psi\big(\boldsymbol{\mu}_{\mathbf{z}_e^{(k_c)}}\big), \tag{8}$$

$$\boldsymbol{\Sigma}_{\mathbf{g}_e^{(k_c)}} \approx \boldsymbol{\Sigma}_{\mathbf{z}_e^{(k_c)}} \odot \left[\nabla\psi\big(\boldsymbol{\mu}_{\mathbf{z}_e^{(k_c)}}\big) \nabla\psi\big(\boldsymbol{\mu}_{\mathbf{z}_e^{(k_c)}}\big)^T\right], \tag{9}$$

where ∇ is the gradient with respect to $\mathbf{z}_e^{(k_c)}$.

4) Max-Pooling Operation: The max-pooling operation is often used to downsample the incoming feature map. We propagate the mean through the max-pooling layer using the classical operation of selecting the largest value from a patch in the feature map. The pooling for the covariance is achieved by only retaining the rows and columns (of the incoming covariance matrix) corresponding to the retained elements (pooled elements) of the mean vector. We write the mean and covariance as follows:

$$\boldsymbol{\mu}_{\mathbf{p}_e^{(k_c)}} = \text{pool}(\boldsymbol{\mu}_{\mathbf{g}_e^{(k_c)}}), \tag{10}$$

$$\boldsymbol{\Sigma}_{\mathbf{p}_e^{(k_c)}} = \text{co-pool}(\boldsymbol{\Sigma}_{\mathbf{g}_e^{(k_c)}}). \tag{11}$$

An encoder may consists of multiple layers of convolution operations, nonlinear activation functions, and max-pooling to get a low-dimensional representation of the input.

D. Decoder Operations

The operations in the decoder path start with the low-dimensional representation produced by the encoder. The decoder may also include multiple convolutional layers, which are performed following the mathematical relationships provided in eqs. (5)-(7).

1) Up-sampling: The up-sampling is an essential part of the decoder path that increases the resolution of the input. Using $\mathbf{g}_d^{(k_c)}$ to represent the input to the up-sampling operation and $\mathbf{u}_d^{(k_c)}$ as the output, we have:

$$\mathbf{u}_d^{(k_c)} = \text{up-sample}\left(\mathbf{g}_d^{(k_c)}\right). \tag{12}$$

The mean of $\mathbf{u}_d^{(k_c)}$ is computed by inserting 0s between two consecutive elements of the input and padding with 0s. The covariance matrix is obtained by adding rows and columns of 0s at locations corresponding to the newly added 0s in the mean.

2) Up-convolution: The up-sampling operation may produce sparse feature maps with many 0s. Generally, a 2 × 2 convolution operation is performed to get a dense high-resolution output. The mean and covariance are computed using results presented in eqs. (5)-(7).

3) Padding: The padding operation applied to the mean is the same as the classical zero-padding operation. For the covariance matrix, we add a new row and a new column for each element padded to the mean. The new elements added in the covariance matrix are all set to 0 to enforce independence and the variance (diagonal) elements are set to a user-defined small value with $\sigma_{pa} > 0$.

4) Concatenation: The features from the encoder side are generally concatenated with the corresponding features from the decoder to improve localization of various objects in the input. The feature maps from the encoder path may need to be resized or cropped before they can be concatenated with the decoder features due to the differences in the their size.

Let $\mathcal{G}_e^c$ be the c^{th} encoder feature map, and $\mathbf{g}_e^{(k_c)}$ the k_c^{th} slice from such map with mean and covariance $\boldsymbol{\mu}_{\mathbf{g}_e^{(k_c)}}$ and $\boldsymbol{\Sigma}_{\mathbf{g}_e^{(k_c)}}$, respectively. The cropped feature map is denoted with $\mathcal{G}_e^{*c}$ where k_c^{th} slice is $\mathbf{g}_e^{*(k_c)}$. For $k_c = 1, \ldots, K_c$, $\boldsymbol{\mu}_{\mathbf{g}_e^{*(k_c)}} = \text{crop}(\boldsymbol{\mu}_{\mathbf{g}_e^{(k_c)}})$ while $\boldsymbol{\Sigma}_{\mathbf{g}_e^{*(k_c)}}$ is obtained removing rows and columns from $\boldsymbol{\Sigma}_{\mathbf{g}_e^{(k_c)}}$ corresponding to the cropped elements of $\boldsymbol{\mu}_{\mathbf{g}_e^{(k_c)}}$.

The output of the concatenation operation is a feature map $\mathcal{G}_d^{*c} = \{\mathcal{G}_d^c, \mathcal{G}_e^{*c}\}$, where $\mathcal{G}_d^c$ is the c^{th} decoder feature map. The concatenation operation is done along the dimension that represents channels in the feature maps (generally the third dimension).

5) Softmax Function: Pixel-level segmentation can be considered as a dense classification problem where we assign a label to each pixel. Hence, for a multi-class problem, a softmax function ϕ is applied to the output of the last layer.

Let $\mathbf{F}$ represent the output of the last layer with mean $\boldsymbol{\mu}_{\mathbf{F}}$ and covariance $\boldsymbol{\Sigma}_{\mathbf{F}}$, and $\mathbf{Y}$ denote the output of the network after the softmax operation. We can approximate the mean $\boldsymbol{\mu}_{\mathbf{Y}}$ and covariance $\boldsymbol{\Sigma}_{\mathbf{Y}}$ using first-order Taylor series, that is:

$$\boldsymbol{\mu}_{\mathbf{Y}} \approx \phi(\boldsymbol{\mu}_{\mathbf{F}}), \tag{13}$$

$$\boldsymbol{\Sigma}_{\mathbf{Y}} \approx \boldsymbol{J}_{\phi} \boldsymbol{\Sigma}_{\mathbf{F}} \boldsymbol{J}_{\phi}^T, \tag{14}$$

where $\boldsymbol{J}_{\phi}$ is the Jacobian matrix of ϕ computed with respect to $\mathbf{F}$ evaluated at $\boldsymbol{\mu}_{\mathbf{F}}$.

The mathematical results presented above for various operations can be used to build any type of deep neural network in addition to the proposed encoder-decoder based segmentation networks.

IV. Experimental Methods

We focus on medical image segmentation for validating the efficacy of the proposed VMP framework. We employ three different datasets and compare VMP with two state-of-the-art segmentation networks, a deterministic U-Net and a Bayesian U-Net [27], [32].

TABLE I
DSC for Lungs Dataset - performance comparison of different networks under additive Gaussian noise.

	U-Net	Bayes U-Net	VMP U-Net
Noise Free	0.79	**0.82**	**0.82**
Gaussian noise added to the entire image			
SNR ≈ 35 dB	0.78	**0.82**	**0.82**
SNR ≈ 3 dB	0.15	0.19	**0.20**
Gaussian noise added to lung pixels only			
SNR ≈ 31 dB	0.79	**0.82**	**0.82**
SNR ≈ 14 dB	0.60	**0.61**	**0.61**

TABLE II
DSC for Hippocampus Dataset - performance comparison under additive Gaussian noise.

	U-Net	Bayes U-Net	VMP U-Net	U-Net	Bayes U-Net	VMP U-Net
	Anterior			**Posterior**		
Noise Free	**0.79**	**0.79**	**0.79**	**0.76**	**0.76**	0.74
Gaussian noise added to entire image						
SNR ≈ 23 dB	0.77	0.77	**0.78**	**0.74**	0.73	0.73
SNR ≈ 15 dB	0.49	0.53	**0.69**	0.54	0.43	**0.64**
SNR ≈ 8 dB	0.09	0.12	**0.36**	0.13	0.07	**0.28**
Gaussian noise added to Anterior pixels only						
SNR ≈ 23 dB	**0.77**	**0.77**	**0.77**	**0.76**	0.75	0.74
SNR ≈ 15 dB	0.51	0.55	**0.69**	0.66	0.63	**0.69**
SNR ≈ 9 dB	0.11	0.15	**0.38**	0.28	0.24	**0.43**
Gaussian noise added to Posterior pixels only						
SNR ≈ 23 dB	**0.78**	**0.78**	**0.78**	**0.76**	0.74	0.73
SNR ≈ 15 dB	0.62	0.66	**0.69**	0.55	0.45	**0.69**
SNR ≈ 9 dB	0.23	0.29	**0.51**	0.14	0.07	**0.29**

A. *Segmentation Network Architectures*

1) U-Net - The Base Segmentation Architecture: Among all architectures proposed for medical image segmentation, U-Net is the most widely used [27]. U-Net is built using the encoder-decoder structure with a contracting path and almost identical expanding path. The contracting path may consist of multiple encoder blocks, which, in turn, may include various convolution layers, max-pooling, and nonlinear activations. The expanding path consists of multiple decoder blocks, which are made of multiple layers of convolution, activation functions, up-convolution, up-sampling and padding. Additionally, there are connections between the encoder and decoder blocks that concatenate feature maps from the encoder with the corresponding feature maps of the decoder. Finally, a 1×1 convolution and softmax are applied to the decoded feature maps before calculating the cross-entropy loss function.

In this original U-Net architecture [27], the border pixels are lost due to un-padded convolution operations and the missing regions are extrapolated by mirroring. Such processing may yield erroneous results for some medical image segmentation datasets. Hence, in our settings, we apply the padding operation to increase the size of the feature maps and reconstruct the full-image at the output of the network. We include the padding operation twice in each decoder block on the expanding path. The first padding operation is performed before the concatenation and the second is performed before the second convolution in each decoder block. In our experiments, we refer to this U-Net architecture as the deterministic segmentation network.

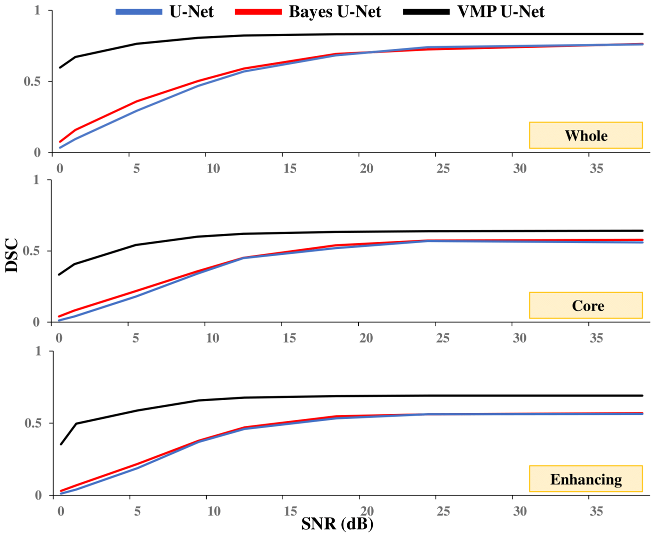

Fig. 2. We compare the performance of the three networks when various levels of Gaussian noise are added to the BraTS test data. The three subplots show DSC values for a range of SNRs for three different tumor regions, i.e., whole tumor, core, and enhancing. We note that VMP U-Net shows high performance and robust behavior, especially at low SNR values.

2) Bayes U-Net: Bayes U-Net is built with the MC-Dropout technique following the implementation of [32]. The dropout is used only in the central blocks with the probability of dropping a neuron set to $p = 0.5$. Bayes U-Net uses cross-entropy loss function. At the inference time, we use $N = 20$ MC samples.

3) VMP U-Net: VMP U-Net uses the mathematical operations presented in Sections III-C and III-D to propagate the first two moments of the variational distribution through the U-Net architecture. The output of VMP U-Net consists of a segmentation map and an uncertainty map. The former is given by the mean of the predictive distribution, while the latter is built using the variance of the predictive distribution. We use a Gaussian variational distribution and employ the ELBO loss function defined in Eq. (3). We optimize the ELBO loss function with respect to the variational parameters, i.e., the mean and covariance of the variational distribution. To reduce the computational complexity, we propagate diagonal covariance matrices in all cases.

B. *Datasets and Networks*

We use three different medical segmentation datasets, including lung CT, hippocampus MRIs and brain tumor MRIs [41]–[43]. Our experiments use only the publicly available annotated data from the respective datasets, i.e., unlabeled data is not used for training, validating or testing. The datasets are divided into training, validation and testing bins with approximately 80% selected for training, 10% for the validation and 10% for testing.

1) Lungs Dataset: The dataset includes 20 CT scans from the chest region, including the lungs. This heterogeneous dataset consists of both COVID-19 and non-COVID-19 patients. The data annotations include left lung, right lung and infections (if found). We consider a binary segmentation task for this dataset, i.e., delineating the boundaries of lungs in

TABLE III
DSC FOR BRATS DATASET - PERFORMANCE COMPARISON UNDER ADDITIVE GAUSSIAN NOISE.

	Whole			***Core***			***Enhancing***		
	U-Net	**Bayes U-Net**	**VMP U-Net**	**U-Net**	**Bayes U-Net**	**VMP U-Net**	**U-Net**	**Bayes U-Net**	**VMP U-Net**
Noise free	.77	.77	**.83**	.58	.58	**.64**	.57	.57	**.69**
Gaussian noise added to entire image									
SNR $\approx$ 24 dB	.73	.74	**.83**	.57	.57	**.64**	.56	.56	**.69**
SNR $\approx$ 18 dB	.68	.69	**.83**	.54	.55	**.63**	.54	.55	**.69**
SNR $\approx$ 9 dB	.49	.50	**.81**	.35	.36	**.60**	.38	.38	**.66**
Gaussian noise added to tumor pixels only									
SNR $\approx$ 32 dB	.70	.70	**.83**	.54	.55	**.64**	.55	.55	**.68**
SNR $\approx$ 25 dB	.62	.62	**.83**	.48	.48	**.64**	.49	.49	**.64**
SNR $\approx$ 13 dB	.21	.22	**.79**	.11	.11	**.46**	.10	.10	**.37**

TABLE IV
DSC FOR LUNGS DATASET - PERFORMANCE COMPARISON UNDER UN-TARGETED ADVERSARIAL ATTACKS.

	U-Net	**Bayes U-Net**	**VMP U-Net**
Noise Free	0.79	**0.82**	**0.82**
Un-targeted attacks generated using FGSM			
SNR $\approx$ 55 dB	0.78	**0.80**	**0.80**
SNR $\approx$ 29 dB	0.53	0.65	**0.66**
SNR $\approx$ 10 dB	0.21	0.35	**0.44**

TABLE V
DSC FOR HIPPOCAMPUS DATASET - PERFORMANCE COMPARISON UNDER VARIOUS LEVELS OF TARGETED ADVERSARIAL ATTACKS.

	Anterior			***Posterior***		
	U-Net	**Bayes U-Net**	**VMP U-Net**	**U-Net**	**Bayes U-Net**	**VMP U-Net**
Noise Free	**0.79**	**0.79**	**0.79**	**0.76**	**0.76**	0.74
Targeted adversarial attacks - source: label 1, target: label 2						
SNR $\approx$ 29 dB	0.75	**0.76**	0.75	0.60	0.60	**0.64**
SNR $\approx$ 23 dB	0.71	**0.72**	**0.72**	0.52	0.55	**0.59**
SNR $\approx$ 15 dB	0.40	0.45	**0.52**	0.19	0.27	**0.34**
Targeted adversarial attacks - source: label 2, target: label 1						
SNR $\approx$ 23 dB	0.65	**0.69**	**0.69**	0.67	**0.71**	**0.71**
SNR $\approx$ 15 dB	0.59	0.64	**0.65**	0.65	**0.70**	**0.70**
SNR $\approx$ 9 dB	0.28	0.32	**0.42**	0.39	0.40	**0.46**

the given CT images. Specifically, we assign a label of 0 to the background and 1 to lung tissue. The pre-processing steps include (1) windowing the Hounsfield units range between -1250 and 250, (2) normalizing all pixel values between 0 and 1, (3) deleting empty slices, i.e., slices that include only the label 0 corresponding to the background, and (4) cropping all images to a single size, i.e., 512×512 pixels.

The U-Net architecture used for this dataset includes 3 encoder blocks and 2 decoder blocks. The number of kernels in the encoder blocks is set to 16, 32, and 64, while to 32 and 16 in the decoder blocks. We train all three networks (i.e., U-Net, Bayes U-Net and VMP U-Net) for 50 epochs using a batch size of 10 with the Adam optimizer and the learning rate of 0.001. In VMP U-Net, we set $\sigma_{pa} = 0.05$.

2) Hippocampus Dataset: The dataset consists of 394 single-modality MRI scans. The segmentation task requires the precise delineation of two adjacent structures, i.e., anterior (label 1) and posterior (label 2). The pre-processing steps include (1) normalizing data to reduce the image bias (which is a characteristic of MRI data), (2) deleting empty slices, i.e., those that include only the label 0 corresponding to the background, and (3) padding images to have the same input size of 64×64 pixels.

For the hippocampus task, the U-Net architecture consists of 3 encoder blocks and 2 decoder blocks. The convolutional kernels are set to 32, 64, and 128 on the encoder side and 64, and 32 on the decoder side. All three networks (i.e., U-Net, Bayes U-Net and VMP U-Net) are trained with Adam optimizer for a total of 100 epochs using a batch size of 20. In VMP U-Net, we set $\sigma_{pa} = 0.02$.

3) Brain Tumor Segmentation (BraTS) Dataset: The dataset includes about 300 multi-modal (T1, T1c, T2, and FLAIR) MRI scans from 274 brain tumor patients (some patients have multiple MRI scans) [43]. The dataset is divided into two main types of tumors, low-grade gliomas (LGG) and high-grade gliomas (HGG). We focus on the more challenging HGG dataset in our experiments. The pre-processing steps include (1) normalizing data to reduce the image bias, (2) deleting images that do not include any tumor structure, and (3) cropping each image to the size of 240×240 pixels. The input data size for each sample in the dataset is $240 \times 240 \times 4$ pixels, where the last number represents the four modalities, i.e., T1, T1c, T2, and FLAIR. All three networks (U-Net, Bayes U-Net, and VMP U-Net) are trained to segment 5 different labels in the HGG MRIs, i.e., normal tissue (label 0), necrosis (label 1), edema (label 2), non-enhancing tumor (label 3), and enhancing tumor (label 4). In most clinical applications, generally, three tumor regions are considered for evaluating the results of segmentation, (1) whole tumor (labels 1, 2, 3 and 4), (2) tumor core (labels 1, 3 and 4), and (3) enhancing tumor region (label 4) [43].

We use the original U-Net architecture with 5 encoder and 4 decoder blocks [27]. The number of kernels in each encoder block is 64, 128, 256, 512, and 1024. The number of kernels used in the decoder blocks is 512, 256, 128, and 64. We set $\sigma_{pa} = 0.1$ for the VMP U-Net. All three networks are trained for 100 epochs using Adam optimizer with a learning rate of 0.001 and a batch size of 20.

C. Other Experimental Settings

We report the Dice Similarity Coefficient (DSC) as the metric to compare the performance of all three networks.

TABLE VI
DSC for BraTS Dataset - performance comparison under various levels of un-targeted and targeted adversarial attacks.

	Whole			*Core*			*Enhancing*		
	U-Net	**Bayes U-Net**	**VMP U-Net**	**U-Net**	**Bayes U-Net**	**VMP U-Net**	**U-Net**	**Bayes U-Net**	**VMP U-Net**
Noise free	.77	.77	**.83**	.58	.58	**.64**	.57	.57	**.69**
Un-targeted attacks generated using FGSM									
SNR ≈ 39 dB	.67	.68	**.82**	.46	.47	**.62**	.46	.46	**.64**
SNR ≈ 12 dB	.34	.37	**.58**	.20	.23	**.31**	.19	.20	**.30**
Targeted adversarial attacks - source: label 3, target: label 1									
SNR ≈ 44 dB	.69	.70	**.82**	.48	.50	**.61**	.48	.50	**.67**
SNR ≈ 19 dB	.34	.35	**.55**	.22	.23	**.35**	.24	.24	**.36**
Targeted adversarial attacks - source: label 1, target: label 3									
SNR ≈ 44 dB	.70	.71	**.82**	.50	.50	**.63**	.48	.49	**.67**
SNR ≈ 19 dB	.38	.38	**.56**	.24	.25	**.35**	.25	.26	**.37**
Targeted adversarial attacks - source: label 3, target: label 2									
SNR ≈ 44 dB	.70	.71	**.82**	.52	.53	**.64**	.50	.53	**.67**
SNR ≈ 19 dB	.35	.35	**.54**	.24	.24	**.36**	.24	.25	**.39**
Targeted adversarial attacks - source: label 2, target: label 3									
SNR ≈ 44 dB	.70	.71	**.82**	.51	.53	**.62**	.51	.52	**.68**
SNR ≈ 19 dB	.35	.35	**.58**	.23	.24	**.37**	.25	.26	**.38**

The choice of the loss function in segmentation can significantly impact the performance of the network [44]. For example, the loss function based on DSC may result in better performance as compared to the cross-entropy loss [44]. However, for a fair comparison, we used the same loss functions, i.e., cross-entropy, for both U-Net and Bayes U-Net.

We conduct a detailed robustness analysis of the performance of all three networks using two types of noise, i.e., Gaussian and adversarial. We compare the performance of all three networks under various levels of Gaussian noise added to the test data of all three datasets. We measure the noise level using the Signal-to-Noise Ratio (SNR) in the units of decibels (dB). For the adversarial noise, we employ the Fast Gradient Sign Method (FSGM) and generate un-targeted attacks [45]. On the other hand, we use the Projected Gradient Descent (PGD) method to generate targeted adversarial attacks [46]. We set the maximum number of iteration to 20 and use the step-size of 1. We select a *source* class and a *target* class to generate targeted attacks. The adversarial attack algorithm will try to fool the trained network into predicting pixels belonging to the *source* class as the pixels of the *target* class.

V. Results and Discussion

We report our results in three parts. First, we present the performance analysis (measured using DSC) of three networks (U-Net, Bayes U-Net, and VMP U-Net) under various levels of Gaussian noise added to the test data. Next, we analyze the same three networks under various levels of targeted and un-targeted adversarial attacks. Finally, we present our analysis of the uncertainty maps generated by the proposed VMP U-Net at the inference time.

A. *Evaluation Under Gaussian Noise*

Tables I, II, and III report DSC values for U-Net, Bayes U-Net and VMP U-Net under different levels of Gaussian noise, including the noise-free case. For each dataset, we report results for two cases, i.e., noise added to the entire input image or only to the structures that the networks are trying to segment, e.g., tumor in the BraTS dataset.

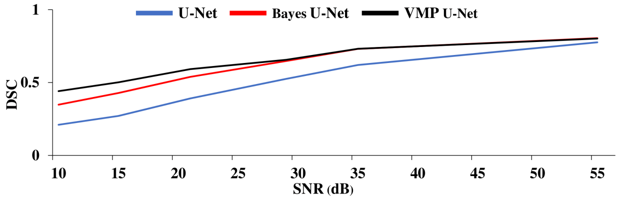

Fig. 3. DSC values for comparing the performance of three networks. Various levels of un-targeted attacks are applied to the Lungs test data. We plot the DSC vs SNR (dB) for the 3 approaches. Note the VMP method is showing a more robust behaviour when data is corrupted with higher levels of adversarial attacks.

We note that the proposed VMP U-Net generally demonstrates more robust behavior as compared to deterministic U-Net and Bayes U-Net, especially at low SNR values, i.e., high levels of noise. In Fig. 2, we plot DSCs vs. SNR for the three tumor regions. Each subplot compares the performance of the three networks for multiple levels of Gaussian noise added to the entire image.

B. *Evaluation Under Adversarial Attacks*

We present the robustness of all three networks against targeted and un-targeted adversarial attacks in Tables IV, V, and VI. The tables present DSC values for various levels of adversarial attacks, quantified using SNR. We observe that VMP U-Net shows better performance (i.e., high DSC values) as compared to the other two networks, especially for stronger attacks (i.e., low values of SNR).

In Fig. 3, we show DSC values for a range of un-targeted adversarial attacks against the lung test dataset generated using the FGSM. We show the DSC vs. SNR for the three approaches.

C. *Uncertainty Maps and Predictive Variance*

1) Uncertainty Maps: The output of VMP U-Net consists of a segmentation map (prediction) and an uncertainty map. In Fig. 4, and 5, we present representative cases selected from the hippocampus and BraTS test data. We show (a) one input modality (only FLAIR for the BraTS data), (b)

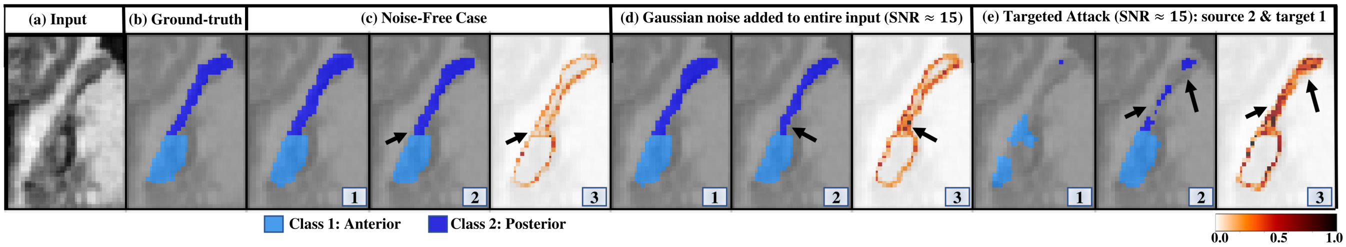
(a) Input
(b) Ground-truth
(c) Noise-Free Case
(d) Gaussian noise added to entire input (SNR ≈ 15)
(e) Targeted Attack (SNR ≈ 15): source 2 & target 1
1
2
3
1
2
3
1
2
3
Class 1: Anterior
Class 2: Posterior
0.0
0.5
1.0

Fig. 4. A representative image from the hippocampus test data. We show (a) the input image, (b) the ground-truth segmentation, (c) the noise-free case, (d) the Gaussian noise case, (d) the targeted adversarial attack. For each case, we show (1) the U-Net segmentation prediction, (2) the proposed VMP U-Net segmentation prediction, and (3) the corresponding uncertainty map. The arrows point to regions incorrectly classified by our network. We note that the corresponding pixels in the uncertainty maps reflect the low confidence by responding with higher variance values.

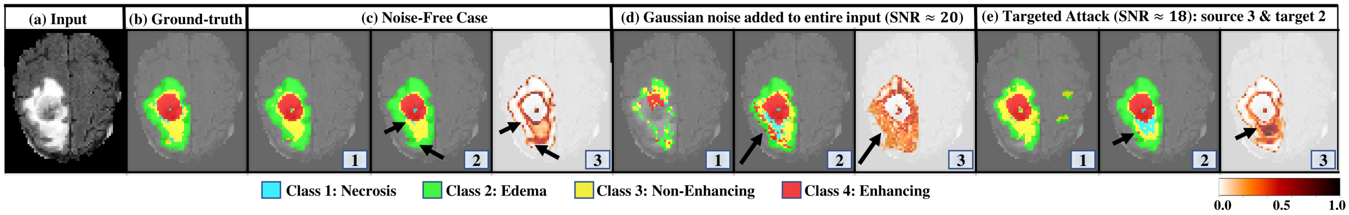

Fig. 5. A representative image from BraTS test data. We show (a) the input image (FLAIR modality only), (b) the ground-truth segmentation, (c) the noise-free case, (d) the Gaussian noise case, and (e) the targeted adversarial attack. For each case, (1) shows the U-Net segmentation prediction, (2) the segmentation prediction by VMP U-Net, and (3) the corresponding uncertainty map. The arrows point to regions incorrectly classified by the network. We note that the corresponding pixels in the uncertainty maps reflect the low confidence by responding with higher variance values.

the ground-truth label, (c) the noise-free case, (d) a Gaussian noise case and (e) a targeted attack example. For each case, we illustrate (1) the segmentation prediction obtained with U-Net, (2) the proposed VMP U-Net prediction, and (3) the corresponding VMP uncertainty map. We point to regions (pixels) incorrectly classified by the network with the arrows (sub-figure (2c), (2d), (2e)). In the uncertainty maps, we place arrows in the same spot as in the prediction figures (sub-figure (3c), (3d), (3e)). It is evident from the figure that VMP U-Net associates high uncertainty with incorrect predictions and pixels belonging to targeted regions. We also note that when the predicted segmentation is (almost) identical to the ground-truth, the model is confident in its segmentation predictions and is only uncertain at the boundary between the structures of interest (noise-free case, sub-figures (2c) and (3c)). On the other hand, when noise or adversarial attacks are detected, or when the network's segmentation predictions are incorrect, higher uncertainty values are associated with the predictions (sub-figure (2d), (3d), (2e) and (3e)). The comparison with the point-estimate approach, i.e., U-Net, demonstrates the need for the uncertainty maps. The reliability of the segmentation prediction can be assessed using the uncertainty maps.

2) Predictive Variance: We calculate the average predictive variance from uncertainty maps and plot these values against various levels of Gaussian noise in Fig. 6 and adversarial attacks in Fig. 7 for hippocampus and BraTS datasets, respectively. It is more instructive and insightful if sub-plots in both figures are interpreted from right to left, i.e., decreasing SNR or equivalently increasing noise in the test data. We note that the predictive variance monotonically increases with increasing noise (i.e., decreasing SNR) for all three sub-figures in Fig. 6 and all four sub-figures in Fig. 7. This behavior, i.e., increasing variance with increasing noise, demonstrates that the network is aware of higher noise in the input. Such information is valuable for detecting when the network may fail, and its predictions may become untrustworthy.

We report the average inference time of all three networks in Table VII. We note that VMP U-Net requires almost twice the time to process a single image at the inference time compared to the deterministic U-Net. The increased processing time is related to the propagation of the covariance information that requires performing additional operations. The Bayes U-Net takes the same time as that of a deterministic U-Net for one pass. However, it will need multiple passes to calculate the variance of the prediction. We used $N = 20$, which will lead to the inference time of 16.4 ms for each image, almost 8 times more than VMP U-Net.

D. Discussion

We build an efficient and scalable Bayesian network for segmentation, referred to as VMP U-Net, based on the encode-decoder architecture. We derive mathematical relations to train VMP U-Net for accurate segmentation and simultaneously estimate uncertainty in segmentation decisions. We use three medical datasets to test the validity of our approach. Our simulations show that VMP U-Net delivers superior robustness to noise and adversarial attacks. In the noise-free case, simple datasets, and binary task, e.g., hippocampus dataset, VMP U-Net performs equally well compared to the state-of-the-art models. However, as the noise levels increase (Gaussian or adversarial), the task or the dataset becomes more complicated, e.g., BraTS data (multiple segmentation labels and multiple

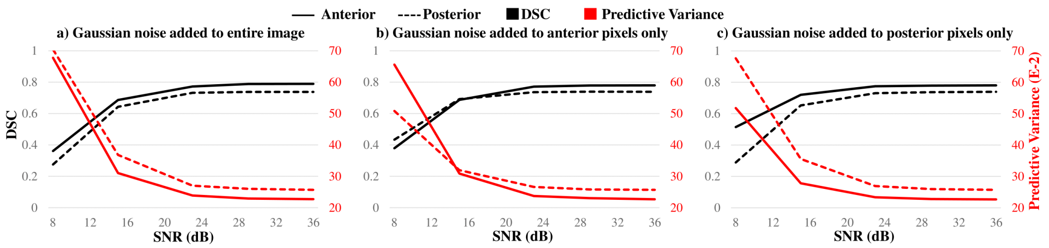

Fig. 6. Accuracy (DSCs) and average predictive variance of VMP U-Net for various levels of Gaussian noise added in the test data for hippocampus dataset. It is more instructive if each sub-figure is interpreted from right to left. The black lines indicate DSCs and red light show the average predictive variance. (a) Noise added to the entire input. (b) Noise added to the anterior pixels only. (c) Noise added to the posterior pixels only.

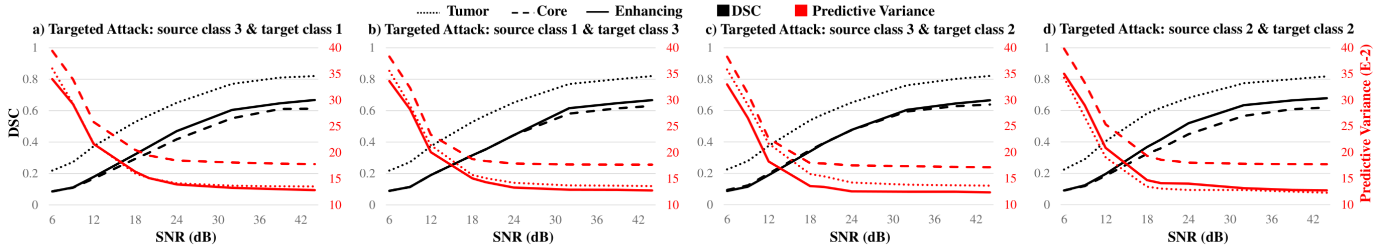

Fig. 7. Accuracy (DSCs) and average predictive variance of VMP U-Net for various levels of adversarial attacks applied to the test data for BraTS dataset. The black lines indicate DSCs and red lines show the average predictive variance. Test data is corrupted with targeted attacks: (a) source class 3 and target class 1, (b) source class 1 and target class 3, (c) source class 3 and target class 2, (d) source class 2 and target class 3.

TABLE VII
INFERENCE TIME PER IMAGE

	U-Net	Bayes U-Net	VMP U-Net
Time (ms)	0.81	0.82 N*	1.92

*N is the number of runs through the network at inference time.

modalities), VMP U-Net outperforms both U-Net and Bayes U-Net. The superior performance and robustness to noise, especially at high noise levels and complex task/data, can be attributed to the propagation of the variance information (in addition to the mean) through the network layers from input to output. In our formulation, the second moment, which represents uncertainty, is learned during the training rather than being estimated using MC runs through the network at the inference time.

VI. Conclusion

We proposed a novel Bayesian framework, VMP U-Net, to quantify uncertainty in segmentation tasks. We employed the probability density function tracking techniques used for nonlinear systems to develop a framework that propagates the mean and the covariance matrix of the variational posterior distribution across all layers of DNNs. We targeted the encoder-decoder architecture for medical image segmentation, i.e., U-Net. At test time, the uncertainty in the decision is captured by the covariance matrix of the predictive distribution, which is available at the output of the VMP U-Net along with the prediction. The predictive variance information is used to build uncertainty maps that provide crucial information about the network's self-awareness on the reliability of its prediction (i.e., segmentation). We have shown that areas incorrectly classified by our network are accompanied by higher uncertainty regions. Our results also demonstrate that the VMP configuration enhances the performance of the DNN in the presence and absence of noise or adversarial attacks. Furthermore, the uncertainty maps offer transparency that attracts the attention of supervising physicians. Nonetheless, clinical situations may arise where AI segmentation may not be under direct and continuous physician control; in such cases, it is important that the networks recognize high measures of uncertainty to generate an appropriate notification. Hence, our work paves the way for developing trustworthy and self-aware DL systems that can be safely deployed in mission-critical applications, such as healthcare.

Acknowledgment

This work was supported by the National Science Foundation awards NSF ECCS-1903466 and NSF OAC-2008690. Giuseppina Carannante is supported by the US Department of Education through a Graduate Assistance in Areas of National Need (GAANN) program Award Number P200A180055. We are also grateful to UK EPSRC support through EP/T013265/1 project NSF-EPSRC: ShiRAS. Towards Safe and Reliable Autonomy in Sensor Driven Systems, and NJ Health Foundation support through Award number PC 78-21.

References

[1] M. L. Giger, "Machine learning in medical imaging," *Journal of the American College of Radiology*, vol. 15, no. 3, pp. 512–520, 2018.

[2] D. S. Ting, Y. Liu, P. Burlina, X. Xu, N. M. Bressler, and T. Y. Wong, "Ai for medical imaging goes deep," *Nature medicine*, vol. 24, no. 5, pp. 539–540, 2018.

[3] X. Liu, L. Faes, A. U. Kale, S. K. Wagner, D. J. Fu, A. Bruynseels, T. Mahendiran, G. Moraes, M. Shamdas, C. Kern *et al.*, "A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis," *The lancet digital health*, vol. 1, no. 6, pp. e271–e297, 2019.

[4] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[5] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1321–1330.

[6] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[8] H. Hirano, A. Minagi, and K. Takemoto, "Universal adversarial attacks on deep neural networks for medical image classification," *BMC medical imaging*, vol. 21, no. 1, pp. 1–13, 2021.

[9] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, "Adversarial attacks on medical machine learning," *Science*, vol. 363, no. 6433, pp. 1287–1289, 2019.

[10] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[11] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles," in *Advances in neural information processing systems*, 2017, pp. 6402–6413.

[12] X. Lu and B. Van Roy, "Ensemble Sampling," in *Advances in neural information processing systems*, 2017, pp. 3258–3266.

[13] A. Graves, "Practical Variational Inference for Neural Networks," in *Advances in neural information processing systems*, 2011, pp. 2348–2356.

[14] D. Dera, N. C. Bouaynaya, G. Rasool, R. Shterenberg, and H. M. Fathallah-Shaykh, "Premium-cnn: Propagating uncertainty towards robust convolutional neural networks," *IEEE Transactions on Signal Processing*, vol. 69, pp. 4669–4684, 2021.

[15] G. Carannante, D. Dera, G. Rasool, N. Bouaynaya, and L. Mihaylova, "Robust Learning via Ensemble Density Propagation in Deep Neural Networks," in *IEEE International Workshop on Machine Learning for Signal Processing*. Sheffield, 2020.

[16] E. Goan and C. Fookes, "Bayesian neural networks: An introduction and survey," in *Case Studies in Applied Bayesian Data Science*. Springer, 2020, pp. 45–87.

[17] A. Doucet, A. M. Johansen *et al.*, "A tutorial on particle filtering and smoothing: Fifteen years later."

[18] N. Amor, G. Rasool, and N. C. Bouaynaya, "Constrained state estimation-a review," *arXiv preprint arXiv:1807.03463*, 2018.

[19] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[21] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," in *Deep learning and data labeling for medical applications*. Springer, 2016, pp. 179–187.

[22] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[23] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[24] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 11–19.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[26] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[28] D. E. Cahall, G. Rasool, N. C. Bouaynaya, and H. M. Fathallah-Shaykh, "Inception modules enhance brain tumor segmentation," *Frontiers in computational neuroscience*, vol. 13, p. 44, 2019.

[29] S.-T. Tran, C.-H. Cheng, T.-T. Nguyen, M.-H. Le, and D.-G. Liu, "Tmd-unet: Triple-unet with multi-scale input features and dense skip connection for medical image segmentation," in *Healthcare*, vol. 9, no. 1. Multidisciplinary Digital Publishing Institute, 2021, p. 54.

[30] M. U. Rehman, S. Cho, J. H. Kim, and K. T. Chong, "Bu-net: Brain tumor segmentation using modified u-net architecture," *Electronics*, vol. 9, no. 12, p. 2203, 2020.

[31] X. Li, W. Qian, D. Xu, and C. Liu, "Image segmentation based on improved unet," in *Journal of Physics: Conference Series*, vol. 1815, no. 1. IOP Publishing, 2021, p. 012018.

[32] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.

[33] A. G. Roy, S. Conjeti, N. Navab, C. Wachinger, A. D. N. Initiative *et al.*, "Bayesian quicknat: Model uncertainty in deep whole-brain segmentation for structure-wise quality control," *NeuroImage*, vol. 195, pp. 11–22, 2019.

[34] T. Nair, D. Precup, D. L. Arnold, and T. Arbel, "Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation," *Medical image analysis*, vol. 59, p. 101557, 2020.

[35] Y. Kwon, J.-H. Won, B. J. Kim, and M. C. Paik, "Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation," *Computational Statistics & Data Analysis*, vol. 142, p. 106816, 2020.

[36] C. F. Baumgartner, K. C. Tezcan, K. Chaitanya, A. M. Hötker, U. J. Muehlematter, K. Schawkat, A. S. Becker, O. Donati, and E. Konukoglu, "Phiseg: Capturing uncertainty in medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 119–127.

[37] G. Wang, W. Li, S. Ourselin, and T. Vercauteren, "Automatic brain tumor segmentation based on cascaded convolutional neural networks with uncertainty estimation," *Frontiers in computational neuroscience*, vol. 13, p. 56, 2019.

[38] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *arXiv preprint arXiv:1612.01474*, 2016.

[39] G. E. Hinton and D. Van Camp, "Keeping the neural networks simple by minimizing the description length of the weights," in *Proceedings of the sixth annual conference on Computational learning theory*, 1993, pp. 5–13.

[40] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight Uncertainty in Neural Networks," *arXiv preprint arXiv:1505.05424*, 2015.

[41] M. Jun, G. Cheng, W. Yixin, A. Xingle, G. Jiantao, Y. Ziqi, Z. Minqing, L. Xin, D. Xueyuan, C. Shucheng, W. Hao, M. Sen, Y. Xiaoyu, N. Ziwei, L. Chen, T. Lu, Z. Yuntao, Z. Qiongjie, D. Guoqiang, and H. Jian, "COVID-19 CT Lung and Infection Segmentation Dataset," Apr. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3757476

[42] A. L. Simpson, M. Antonelli, S. Bakas, M. Bilello, K. Farahani, B. Van Ginneken, A. Kopp-Schneider, B. A. Landman, G. Litjens, B. Menze *et al.*, "A large annotated medical image dataset for the development and evaluation of segmentation algorithms," *arXiv preprint arXiv:1902.09063*, 2019.

[43] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest *et al.*, "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE transactions on medical imaging*, vol. 34, no. 10, pp. 1993–2024, 2014.

[44] A. Mehrtash, W. M. Wells, C. M. Tempany, P. Abolmaesumi, and T. Kapur, "Confidence calibration and predictive uncertainty estimation for deep medical image segmentation," *IEEE transactions on medical imaging*, vol. 39, no. 12, pp. 3868–3878, 2020.

[45] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," *arXiv preprint arXiv:1611.02770*, 2016.

[46] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.