This is a repository copy of *Leveraging uncertainty in adversarial learning to improve deep learning based segmentation*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/152068/

Version: Accepted Version

Proceedings Paper:
Javed, M. and Mihaylova, L. orcid.org/0000-0001-5856-2223 (2019) Leveraging uncertainty in adversarial learning to improve deep learning based segmentation. In: Proceedings of IEEE 13th Symposium Sensor Data Fusion. 13th Symposium Sensor Data Fusion, 15-17 Oct 2019, Bonn, Germany. Institute of Electrical and Electronics Engineers (IEEE) . ISBN 9781728150864

https://doi.org/10.1109/SDF.2019.8916632

© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works. Reproduced in accordance with the publisher's self-archiving policy.

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Leveraging Uncertainty in Adversarial Learning to Improve Deep Learning Based Segmentation

Mahed Javed
Department of Automatic Control & Systems Engineering
University of Sheffield
Sheffield, United Kingdom
mjaved1@sheffield.ac.uk

Lyudmila Mihaylova
Department of Automatic Control & Systems Engineering
University of Sheffield
Sheffield, United Kingdom
l.s.mihaylova@sheffield.ac.uk

***Abstract*—This paper proposes a new framework that combines Bayesian SegNet with adversarial learning to obtain high-quality segmented objects of interest. The proposed architecture takes in the form of two discriminator networks that are trained separately. The first network discriminates between segmentation maps coming either from the SegNet or the ground truth. The second network discriminates between the model uncertainty obtained from SegNet and an ideal solution that does not include uncertainty. The process is very similar to the fusion of sensor information for better decision making. Uncertainty is considered as a measure of mistakes. Hence, learning from it will help improve the performance of neural networks. Our results show that we obtain higher accuracies compared to Bayesian SegNet. Training is performed on a small-sized dataset called CamVid and a large-sized dataset Sun RGB-D. The paper shows that dealing with uncertainties is beneficial for decision making in neural networks, especially in applications with highly uncertain environments. Examples include self-driving cars and medical imaging in cancer treatment.**

***Index Terms*—segmentation, adversarial learning, deep neural networks, Bayesian SegNet, epistemic uncertainty**

I. Introduction

Recently, deep learning (DL) has emerged as a prominent technology in the use of modern-day artificial intelligence (AI) applications. Deep neural networks (DNNs) can be seen widely used in applications such as medical imaging [1], [2] autonomous driving [3], [4], [5], machine translation [6] and weather forecasting [7] etc. When focusing on the area of segmentation, a popular task in generic scene understanding, we find that DNNs have celebrated a wide variety of contributions. Some of these are in the form of architectures, such as SegNet [8], PSPNet [9], U-Net [10] and Deep Lab [11], that have shown to achieve higher performance than their non-deep learning counterparts. Segmentation is a difficult task since it requires classifying each pixel of an image (or images) into an instance (or a category) corresponding to an object.

Despite its popularity, deep learning frameworks are yet to be trained to deal with highly uncertain environments. Majority of the performance of deep learning is blindly assumed to be accurate. There have been two specific incidents that outline the failure of DNNs in uncertain environments; firstly a fatality caused by self-driving cars due to false perception and error in classification results [12], secondly a failure in an image classification system that incorrectly identified human examples with animals which lead to a concern of racial discrimination [13].

To address such challenges, the deep learning community has adopted Bayesian models. Such models are capable of highlighting the confidence of their predictions in the form of uncertainties. One specific example in segmentation is the Bayesian SegNet [14]. Bayesian modelling allows encoding for two types of uncertainty; aleatoric and epistemic [15]. Aleatoric represents uncertainty present in the dataset/observations e.g. sensor noises. Epistemic, or model uncertainty, represent noise in the model. The general rule of thumb in dealing with the two is that aleatoric uncertainty can be made redundant if the dataset is small and epistemic uncertainty can be reduced if more dataset is collected [15]. The two objectives are clearly contradictory and to the best of our knowledge, there is no DNN system available that can deal with epistemic uncertainty in the presence of small dataset.

A. Contributions

In this paper, we propose a novel framework as an example of an add-on architecture that improves the performance of a DNN system (Bayesian SegNet) by teaching it to reduce its model uncertainty without the aid of additional dataset. We first build a simplified version of Bayesian SegNet [14] as a toy example and use dropout layers to output uncertainty [16]. We then build two discriminator networks that train independently; the quality critic (QC) and the uncertainty critic (UC). QC focuses on penalizing SegNet's weights if SegNet produces segmented output maps that are different to ground truth samples. UC, on the other hand, penalizes the weights if SegNet fails to reduce its uncertainty. The framework is analogous to a simple case of two sensors fusing information to overcome uncertainty in the operating environment [17]. We train the models on CamVid [18], a small-sized dataset on outdoor scenes. Our main aim is to confirm if there exists a relation between dealing with uncertainty and increased performance without feeding additional data. The novelties of our paper are as follows:

- Firstly, we build a novel DNN based model that can learn to adapt to model uncertainty without the need of the user to increase dataset size.

- Second, the proposed DNN network achieves performance accuracy higher than its more uncertain, frequentist counterpart model.

The rest of the paper is organized as follows. Section II discusses related methods from the literature on adversarial learning and Bayesian methods. Section III presents the proposed framework, called AdvSegNet. Section IV presents results and finally, Section V provides the summary.

II. Related Work

A. Bayesian Methods in Deep Learning

Bayesian deep learning is a probabilistic paradigm that views DNNs differently from their usual frequentist treatment. The idea originated in the '90s in the works of Radford Neal on Gaussian processes and Bayesian neural networks (BNNs) (see e.g. [19]). Bayesian methods were, to some extent, overshadowed by the recent overwhelming success of their frequentist counterpart which proved to be superior in terms of computational load. However, some of the deep learning approaches are very sensitive to pixel errors [20]. Even a slight modification to a single-pixel can lead to erroneous classification prediction in DNNs. Additionally, considering the previously highlighted incidences and the recent development of faster and more computationally light approximations to Bayesian models [16], [21], [22], [23], there has been a sudden rebirth in the interest of the field.

Bayesian neural networks often require extrinsic modification to network layers and are also difficult to scale. One popular example is BNNs that use variational inference (VI) [24] as an approximated means to obtain posterior distribution. Such methods require alteration of the DNN architecture which can add complexity and sacrifice test accuracy. Additionally, the number of model parameters can increase [22], without increasing model capacity. The uncertainties arising in such methods cover the entire hypothesis space of the DNN's weights [25] or some of the subspaces. Though there are other numerous approximations and workarounds [21], [22], [23], the simplest of which is dropout's uncertainty [16].

Dropout's uncertainty sidesteps complexity introduced by BNNs and other extrinsic methods by utilizing the dropout layers within a DNN model. These layers randomly inhibit activation of nodes in the previous layers within a user assigned probability range [26]. Furthermore, it has been proved in [16] that "dropout applied before every weight layer, is mathematically equivalent to an approximation to the probabilistic deep Gaussian process". Therefore, obtaining a variance from these dropout samples is mathematically equivalent to obtaining uncertainty from variational inference in Gaussian processes. In a hypothetical sense, this theoretical framework is shadowing footsteps of Bayesian inference in deep Gaussian processes, but, within frequentist domains. This allows DNNs to benefit from the best of both worlds. Furthermore, since dropout layers can be sequentially added to numerous diverse architectures, it is also true that uncertainties scale much more easily than in BNN architectures.

The dropout's uncertainty can be characterised as follows. Consider an output of a DNN model as $f^W(x)$ with weight distribution W which takes in a data sample x from the data distribution X. The model inference (i.e. posterior probability) can be defined as $p(W|X,Y)$ and the likelihood as $p(y|f^W(x))$. Here, Y represents the ground truth distribution of which y is a sample of. Using Monte Carlo integration, we can obtain the epistemic uncertainty, (3), from the variance of the output of the softmax layer, (1), as an approximation.

$$p(\hat{y}|x,X,Y) \approx \frac{1}{T}\sum_{t=1}^{T} \text{Softmax}(f^{W_t}(x)). \tag{1}$$

The role of the softmax layer is to 'squash' the inputs to probabilities. Here, $\hat{y}$ is the predicted sample that must resemble the label sample y and W_t is the t^{th} sampled model weight, where $t = 1,\ldots,T$ and T is the total number of sample runs. Obtaining the mean or the expected value of $\hat{y}$, $\mathbb{E}(\hat{y})$, and the variance, $\text{Var}(\hat{y})$, from the sample runs allows us to calculate the prediction and the model uncertainty. This is shown in (2) and (3) respectively:

$$\mathbb{E}(\hat{y}) \approx \frac{1}{T}\sum_{t=1}^{T} f^{W_t}(x), \tag{2}$$

$$\text{Var}(\hat{y}) \approx \frac{1}{T}\sum_{t=1}^{T} f^{W_t}(x)^T f^{W_t}(x) - \mathbb{E}(\hat{y})^T\mathbb{E}(\hat{y}). \tag{3}$$

The choice of the number of sample runs depends on the user. Increasing sample runs does give a more accurate representation of both prediction and uncertainty, however, at the price of increased computational cost. An ideal value would be one that balances both. For the experimentation in the paper, T is set to 30. We also discuss later how this affects the performance of the two critics that use adversarial learning to improve the architecture.

B. Adversarial Learning in Segmentation

Adversarial learning is a form of unsupervised learning in which the learning system is challenged by an adversary (called the discriminator). It penalizes the system for producing fake or undesired outcomes (labelled '0' by the discriminator) as opposed to real or desired outcomes (labelled '1'). In deep learning terminology, the words discriminator and critic are used interchangeably. Learning is accomplished when the learning system successfully manages to confuse the discriminator so that it predicts 0.5 [27].

Much like Bayesian deep learning, adversarial learning has also given a sudden boost to the progress of deep learning, specifically in image generation where generative adversarial networks (GANs) [27] are popular. The problem of image generation is challenging as it involves the construction of pixel-rich information, on a higher dimension, which is based on lower-dimensional feature information encoded within the deep hidden layers of neural networks.

Before the rise of GANs, the issue of image generation was addressed with undirected graphical models such as deep

Boltzmann machines (DBMs) [28]. Such methods involve making inferences from potential functions which capture the interactions within the models that have intractable gradients. Also before GANs, graphical models such as deep belief networks (DBNs) [29] and noise-contrasting estimation [30] were used. These involve learned probability densities to be specified explicitly. With such methods, training with back-propagation is impossible.

Alternatively, GANs use adversarial learning to learn intractable real distributions by side-stepping the complex inference methods by having a discriminator as a guiding principle. This network, $D(x)$, then discriminates whether the produced sample x comes from the generator distribution $p_z(z)$ or the real distribution $p_{data}(x)$. Here, z represents the generator sample. In doing so, a mini-max game is established where we train D to maximize the probability of assigning the correct labels to both training and generator samples. We then train the generator G to minimize $log(1 - D(G(z))$. The following two-player mini-max game with value function $V(G, D)$ is formed:

$$\begin{aligned} min_G max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] \\ +\mathbb{E}_{z\sim p_z(z)}[log(1 - D(G(z)))]. \end{aligned} \tag{4}$$

For object segmentation, GANs have led to the development of supervised and semi/weakly-supervised learning algorithms that either improve the segmentation performance or assist in producing labelled examples. Few notable examples are from the work of Wei et al. [31] in semi/weakly-supervised and Luc et al. [32] in a supervised setting. Both Wei et al. and Luc et al. treat the segmentation network as a generator network and propose a coupling of adversarial loss with standard cross-entropy (see [33] for the definition of cross-entropy). Furthermore, they propose a fully convolutional discriminator that learns ground truth label maps from probability maps of segmentation predictions. The only difference between their approaches is that Luc et al. focuses more on the label quality and uses the adversarial framework as a supervisor for improving the accuracy of the segmentation network, while Wei et al. propose a semi-supervised setting where the prior framework adds additional input images without labels, thus increasing segmentation accuracy. This also avoids manually constructing more dataset samples.

Our work follows a similar approach to Luc et al. except that we utilize Bayesian methods to output uncertainty and then use adversarial learning to teach the network to learn to deal with uncertainty. Specifically, we use two discriminators; one penalizes segmentation network for output labels that differ from ground truth (QC) and the other penalizes if there is uncertainty in the prediction (UC). We discuss our method in more detail in the following section.

III. The Proposed Framework: AdvSegNet

A. Network Layers

The proposed adversarial form of SegNet (AdvSegNet) architecture, as shown in Figure 1, is a simplified version of the original Bayesian SegNet [14]. The differences can be listed as follows. The input receptive field is reduced to 128x128x3 (height, width, depth). This is done to simulate an attack. The encoder consists of a series of convolutional layers and 2x2 sized pooling layers inserted after every duo of convolutional layers. The initial kernel size of the convolutional layers is set to 3 with 64 number of filters. The number of filters is doubled after every duo. Additionally, dropout layers are inserted after the 3^{rd}, 4^{th} and the 5^{th} pooling layer. The probability of dropout is set to 0.5 for each dropout layer. Convolutional layers convolute the features of an image and hierarchically learn them, starting from simple features in the earlier layers and more complex ones in the later layers. Pooling layers downsample these features.

Batch normalization layers are added to SegNet after every convolutional layer. These layers scale and adjust the activations of network layers and help in the stabilizing of the training process. These are followed by rectified linear units (ReLU) [34] that introduce the non-linearities in the network.

The decoder architecture follows a similar style. However, it uses upsampling layers that increase the window size back to 128x128. Dropout layers are added before the start of the decoder and before the 1^{st} and the 2^{nd} upsampling layer. This is then passed through a softmax layer. The softmax layer 'squashes' the logits of AdvSegNet to probabilities of class predictions. Then, the architecture is run several times to output dropout samples.

B. Uncertainty & Quality Critics

After the dropout samples are obtained from the decoder, the sample mean and the sample variance is calculated:

$$\mu_n \approx \frac{1}{T}\sum_{t=1}^{T} g_{n,t}, \tag{5}$$

$$\sigma_n \approx \frac{1}{T}\sum_{t=1}^{T} g_{n,t}^T g_{n,t} - \mu_n^T \mu_n. \tag{6}$$

We denote the output of the decoder to be $g_{n,t}$ for the t^{th} run on the n^{th} input image x_n. Here, $n = 1, \ldots, N$ and N is the number of training images. We then define the mean of the dropout samples as μ_n and the model uncertainty (variance) as σ_n both obtained from (2) and (3). The mean is fed to QC and the model uncertainty to UC.

We define QC as a discriminator and represent its logits as d_n^{QC}. QC learns to map the ground truth sample y_n to '1' (real) and dropout samples μ_n to '0' (fake). We show this as $D_{QC} : \{y_n, \mu_n\} \rightarrow \{0, 1\}$. Then, we define UC as a discriminator that learns to map the perfect solution (no uncertainty), σ_p, to '1' (real) and the uncertainty coming from SegNet, σ_n, to '0' (fake). We represents its logits as d_n^{UC}. This can be shown as $D_{UC} : \{\sigma_p, \sigma_n\} \rightarrow \{0, 1\}$. The perfect solution is considered to be a blank white image of dimensions 128x128.

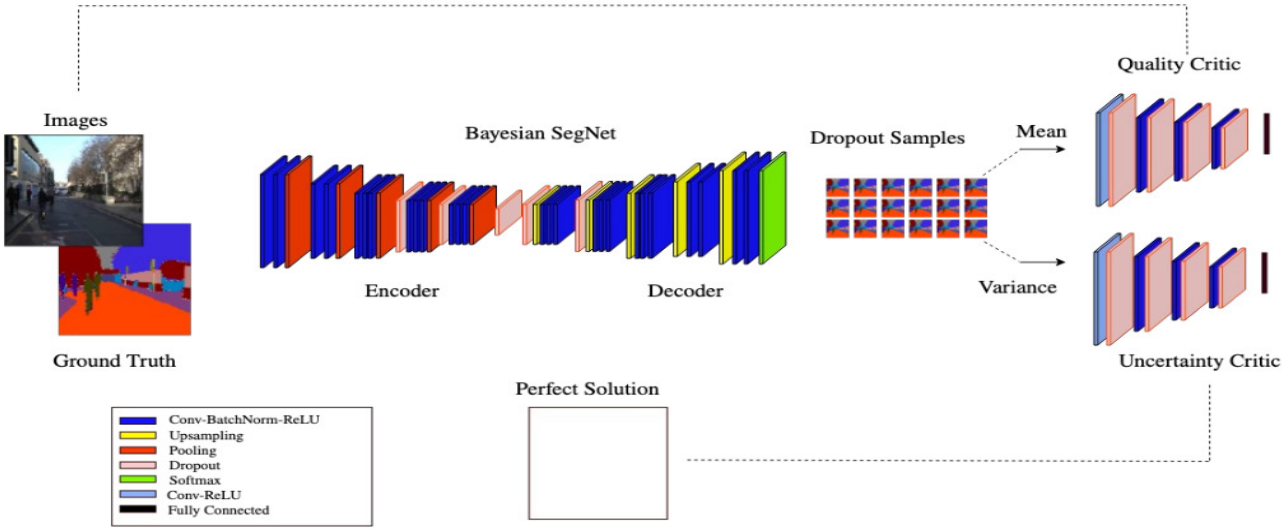

Fig. 1. The proposed architectural framework of adversarial SegNet (AdvSegNet) during train time. Image adapted from [14]

C. Loss Functions

To define loss functions, we follow a similar style to Luc et al. [32] with the term SegNet loss denoting both cross-entropy (based on SegNet's logits) and the adversarial loss (based on critic's logits). The SegNet loss is shown below.

$$\mathcal{L}_{SEG} = -x_n \log(\mu_n) + \tfrac{1}{2}\mathbb{E}[d_n^{QC} - 1]^2 + \tfrac{1}{2}\mathbb{E}[d_n^{UC} - 1]^2 \tag{7}$$

The cross-entropy term encourages SegNet to produce labels that match ground truth. The first term in the square brackets in (7) defines the adversarial loss from QC and the term in the second square brackets from UC. These losses both encourage SegNet to: a) improve quality of segmented label outputs, b) learn to deal with uncertainty. Furthermore, the individual loss functions of the critics that ensure that both QC and UC discriminate effectively are calculated in (8) and (9).

$$\mathcal{L}_{QC} = \tfrac{1}{2}\mathbb{E}[d_n^{QC} - 1]^2 + \tfrac{1}{2}\mathbb{E}[d_n^{QC}]^2 \tag{8}$$

$$\mathcal{L}_{UC} = \tfrac{1}{2}\mathbb{E}[d_n^{UC} - 1]^2 + \tfrac{1}{2}\mathbb{E}[d_n^{UC}]^2 \tag{9}$$

The adversarial losses assosciated to SegNet in (7) and the adversarial loss functions in (8) and (9) are trained separately. They differ from the vanilla GAN loss in (4) and are specifically used in Least Squares (LSQ) GAN [35]. LSQ GAN based losses are chosen in our experiment over the traditional loss because they provide stability in terms of training. The traditional GAN loss suffers from vanishing gradient problem [36] and is proven to output images of low quality [37].

D. Optimization & Training

In our experiment, we use Adam optimizers [38] to train both the SegNet and the two discriminators. We train the discriminators at a lower learning rate than SegNet (see Algorithm 1) in order to ensure stable training, as reviewed in [39]. The training regime adopted for training the discriminators is inspired from [36]. Here, for each episode, the number of optimization steps taken by both discriminators is denoted by t_{critic}. After that the SegNet takes an optimization step. The default value of t_{critic} is set to 5. However, for episodes less than 25 and on the 500^{th} episode, t_{critic} is set to 25. Furthermore, after the discriminators take an optimization step, their weights (i.e. θ_{QC} and θ_{UC}) are clipped to values in between -0.01 and 0.01. This is done in order to stabilize the training process [36]. The weights of SegNet, represented as θ_g, are not clipped in this experiment.

IV. Experimental Results & Discussion

A. Experiment

We test our proposed framework on small-sized dataset CamVid [18]. CamVid is an outdoor road scene understanding dataset of both day and evening scenes taken from camera-rigged automobile. It has a total of 367 training images and 233 validation images. The segmented classes amount to 12 and consist of common outdoor objects such as road, cars, buildings, signs and poles. An extensive experiment is also performed on the dataset SUN RGB-D [40]. This dataset is very challenging as it consists of 5285 training and 5050 testing images of indoor scenes that come in various shapes and sizes. SUN RGB-D consist of 37 classes of common indoor objects e.g. laptop, chair, door, bed, kitchen utensils e.t.c.

To measure the performance of our framework, we use the accuracy and the mean intersection over union metrics (mIoU). The accuracy is measured by computing the frequency of predictions that match the labels. The mIoU is obtained by first obtaining IoU for each of the classes and then taking an average.

We first take a pretrained Bayesian SegNet and then simulate a perturbation method as a simple hack attempt.

Method	Building	Tree	Sky	Car	Sign-Symbol	Road	Pedestrian	Fence	Column-Pole	Side-Walk	Bicyclist	ClassAvg	GlobalAvg	MeanIU
SegNet-Basic [8]	80.6	72.0	93.0	78.5	21.0	94.0	62.5	31.4	36.6	74.0	42.5	62.3	82.8	46.3
SegNet [8]	88.0	87.3	92.3	80.0	29.5	97.6	57.2	49.4	27.8	84.8	30.7	65.9	88.6	50.2
BayesianSegNet-Basic [14]	75.1	68.8	91.4	77.7	52.0	92.5	71.5	44.9	52.9	79.1	69.6	70.5	81.6	55.8
BayesianSegNet [14]	80.4	85.5	90.1	86.4	67.9	93.8	73.8	64.5	50.8	91.7	54.6	76.3	86.9	63.1
DeepLab [11]	81.5	74.6	89.0	82.2	42.3	92.2	48.4	27.2	14.3	75.4	50.1	60.7	89.7	54.7
BayesianSegNet-128x128	**11.4**	**93.8**	**88.0**	**81.7**	**47.1**	**90.8**	**22.8**	**5.5**	**3.3**	**86.0**	**38.0**	**55.7**	**63.4**	**30.0**
AdvSegNet	**77.9**	**95.8**	**98.1**	**75.5**	**36.6**	**95.8**	**52.5**	**77.6**	**6.2**	**92.6**	**84.6**	**72.1**	**87.1**	**52.8**

Fig. 2. Comparison of accuracies of state-of-art and SegNet family included those made in this experiment highlighted as bold

Though there are many forms of such attacks in the literature of machine learning security [20], [41], we use a simple reduction of the receptive field. Here, the input size of the Bayesian SegNet is reduced from 360x480 to 128x128. We then continue to train the networks to 2000 episodes, once with the discriminators and once without. The results obtained are presented in Figure 3. The green dashed line sets the Bayesian SegNet performance benchmark on CamVid. The classical training method involves simple cross-entropy loss without considering the discriminators (blue line). The adversarial training method involves the use of AdvSegNet (red line) on CamVid. A separate experiment is performed on Sun RGB-D but rather trained from scratch (black line).

Furthermore, we observe the evolution of our losses and plot them in Figure 4. The main objective of our research is to compare the performance of Bayesian SegNet with AdvSegNet. To achieve this, we run a separate experiment with SegNet using similar hyperparameter to those chosen for AdvSegNet (see Algorithm 1). We then compare the two frameworks side by side and with the state-of-the-art segmentation including DeepLab [11] and original SegNet [8] in Figure 2. Finally, the segmented label maps obtained both from classical training and AdvSegNet are shown in Figure 6 (E, F) and the respective uncertainties from both models in Figure 6 (G, H) for test samples A) and B).

B. Discussion & Future Works

Bayesian SegNet trained under classical training method takes a moment to adapt to the changed receptive field of 128x128 but begins to improve its performance after 1000^{th} episode on CamVid. AdvSegNet, on the other hand, adapts much faster and can achieve accuracies higher than pretrained Bayesian SegNet. Training on Sun RGB-D dataset is more challenging as AdvSegNet struggles to improve performance from 300th to 1800th episode. A separate experiment to test the sensitivity of validation accuracy on various learning rate configurations for the Sun RGB-D dataset is shown in Figure 5. Here we see that increasing learning rate sequentially for both SegNet and the discriminators leads to more unsteady performance but the overall accuracy obtained is higher than lower learning rate configurations. One of the major issues with dropout uncertainty is that it is not calibrated well [16] for categories. As the number of categories increases (e.g. in large datasets), this issue becomes more prevalent. In the future, we would like to test our method on more diverse architectures and introduce calibration to adapt to large and complex datasets.

Moving to Figure 4 we observe that both losses concerning QC and UC following a similar trend and decrease at a steeper rate than the SegNet loss. This strengthens our hypothesis which states that the two are related since learning to deal with uncertainty aids the improvement of the network's performance.

Considering the comparison of AdvSegNet with the state-of-the-art network DeepLab and Bayesian SegNet in Figure 2, we find that we obtain performance closely similar to Bayesian SegNet in terms of global average accuracy. In the majority of per-class accuracies, we do topple both the networks, but lose performance in the classes Sign-Symbol and Column-Pole. This is further evident in Figure 6 F that the AdvSegNet fails to detect the two poles present in the ground truth label D). However, the performance of AdvSegNet is achieved with receptive field less than half of those employed for Bayesian SegNet and Deep Lab, making it much easier and faster to train. On our system of GPU cluster (NVIDIA K80) provided by the University, the wall clock time to train AdvSegNet was three hours only.

A further key observation to notice is that in Figure 6 G and H, we find that the "cloud" of uncertainty around a mixture of classes Building, Fence and Trees is much higher in classical training of Bayesian SegNet as opposed to AdvSegNet. We believe this might have been the cause for the erroneous prediction of a Fence, despite being labelled Building in the ground truth. The same situation appears in the right-hand

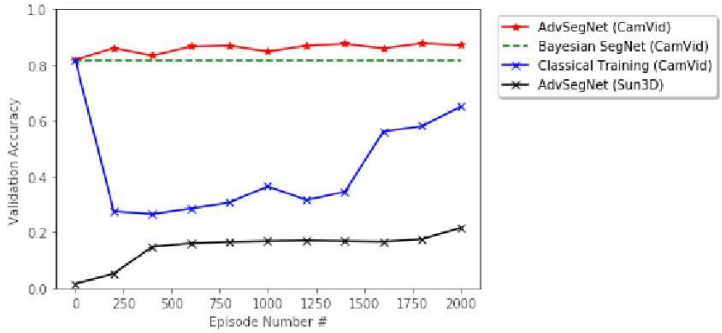

Fig. 3. Results for training pretrained Bayesian SegNet for 2000 episodes in both scenerios of using AdvSegNet and without in classical training

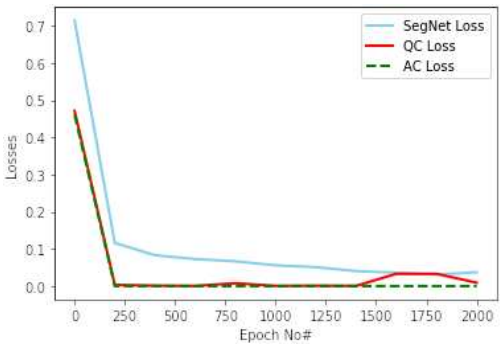

Fig. 4. The evolution of loss functions for AdvSegNet against number of episodes

corner for the test sample of AdvSegNet, however since the uncertainty is less cloudy in AdvSegNet, it is more successful in isolating the Tree class from the Fence. This is perhaps a very important test result obtained from our experiment and in the future, we would like to study in-depth more the relationship of reduced uncertainty and less erroneous prediction. We would also like to explore further ways to obtain uncertainty and experiment with different Bayesian architectures such as BNNs and Gaussian processes.

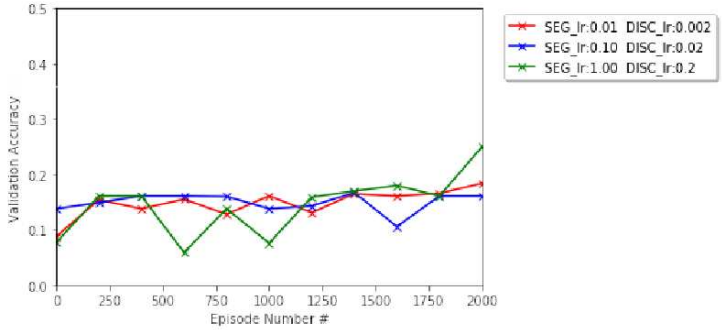

Fig. 5. Validation accuracy comparison for various learning rate configurations of AdvSegNet on Sun RGB-D dataset. The term SEG lr corresponds to learning rate associated with SegNet and DISC lr for the learning rate of the discriminators

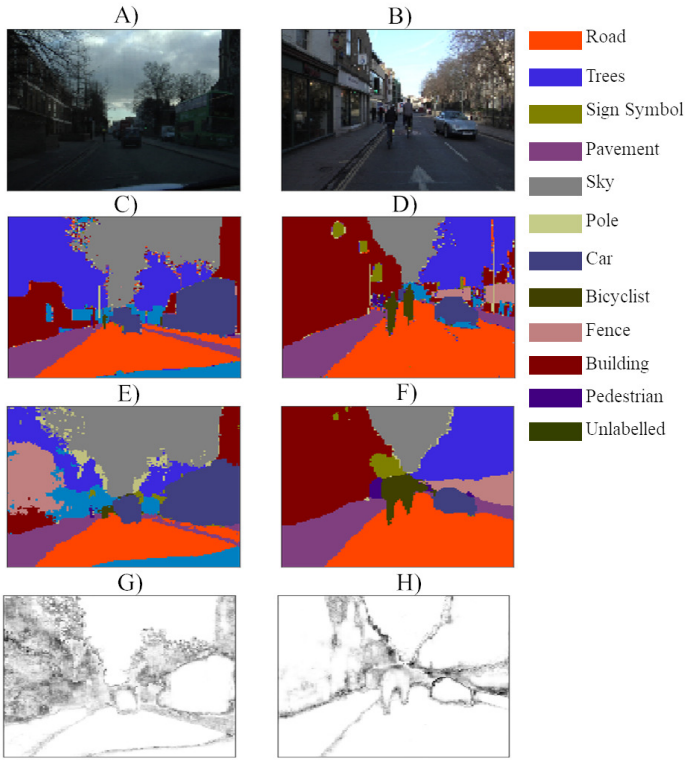

Fig. 6. Qualitative results from both classical training and AdvSegNet training. The figures on the left column (A,C,E,G) represent classical training and AdvSegNet on the right (B,D,F,H). The first row represents input images (A,B), the second row ground truths (C,D), the third row segmented output labels (E,F) and the final row model uncertainty outputs (G,H)

Algorithm 1: AdvSegNet, our proposed algorithm. All experiments in our paper use the following default arguments. batch size $= 1$, $t_{critic} = 5$ or 25, $\gamma_{seg} = 0.005$, $\gamma_{disc} = 0.0005$, $c = 0.01$, $epochs = 2000$

Require: t_{critic} : critic episodes, γ_{seg}: AdvSegNet learning rate, γ_{disc}: critic learning rate, $clip$: clip parameter for critic weights, $epochs$: training episodes for AdvSegNet

Do initialization

for $epoch = 0, ..., epochs$ **do**
  if $epoch < 25$ *or* $epoch = 500$ **then**
    $t_{critic} = 25$
  else $t_{critic} = 5$
  Sample an image batch x_n from the dataset of N training samples
  for $epoch = 0, ..., t_{critic}$ **do**
    Obtain mean prediction: μ_n from (5)
    Compute loss $\mathcal{L}_{QC}$ from (8)
    Update loss: $\theta_{QC} \leftarrow \theta_{QC} + \gamma_{disc} \cdot Adam\ (\theta_{QC},\ \mathcal{L}_{QC})$
    Clip weights: $\theta_{QC} \leftarrow clip\ (\theta_{QC}, -c, c)$
    end
  for $t = 0, ..., t_{critic}$ **do**
    Obtain uncertainty: σ_n from (6)
    Compute loss $\mathcal{L}_{UC}$ from (9)
    $\theta_{UC} \leftarrow \theta_{UC} + \gamma_{disc} \cdot Adam\ (\theta_{QC},\ \mathcal{L}_{UC})$
    $\theta_{UC} \leftarrow clip\ (\theta_{UC}, -c, c)$
    end
  Compute SegNet loss $\mathcal{L}_{SEG}$ from (7)
  Update loss: $\theta_g \leftarrow \theta_g + \gamma_{seg} \cdot Adam\ (\theta_g,\ \mathcal{L}_{SEG})$
  end

V. Summary

This paper proposes a deep learning framework called AdvSegNet, that improves the performance of Bayesian SegNet by teaching it to reduce its model uncertainty without the aid of additional dataset. The developed add-on architecture includes two discriminators called quality critic and uncertainty critic. The performance of the AdvSegNet architecture is evaluated and validated over CamVid dataset. The discriminators are trained independently. They penalize the segmentation network based on the quality of the label map outputs and uncertainty. We show that dealing with epistemic uncertainty is directly linked to the increase in performance. Improved performance of the Bayesian SegNet approach is demonstrated on segmentation by characterising the uncertainty in the model using dropout. We compare our architecture with the state-of-the-art DeepLab and Bayesian SegNet. We find our performance to be similar to Bayesian SegNet with less than half of the receptive field and our results show that AdvSegNet achieves strong results in majority of the classes while poor in some. More importantly, our results form an interesting relationship between reduced model uncertainty and lesser erroneous prediction, which will form basis of our future research work.

References

[1] R. Kälviäinen and H. Uusitalo, "Diaretdb1 diabetic retinopathy database and evaluation protocol," in *Medical Image Understanding and Analysis*, vol. 2007, p. 61, Citeseer, 2007.

[2] G.-Q. Wei, K. Arbter, and G. Hirzinger, "Automatic tracking of laparoscopic instruments by color coding," in *CVRMed-MRCAS'97*, pp. 357–366, Springer, 1997.

[3] A. Ess, T. Mueller, H. Grabner, and L. V. Gool, "Segmentation based urban traffic scene understanding," in *BMVC*, vol. 1, p. 2, 2009.

[4] A. Geiger, "Are we ready for autonomous driving? the Kitti vision benchmark suite," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'12)*, pp. 3354–3361, IEEE Computer Society, 2012.

[5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," pp. 3213–3223, 06 2016.

[6] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands deep in deep learning for hand pose estimation," *CoRR*, vol. abs/1502.06807, 2015.

[7] K. Abhishek, M. Singh, S. Ghosh, and A. Anand, "Weather forecasting model using artificial neural network," *Procedia Technology*, vol. 4, pp. 311–318, 2012.

[8] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," vol. 39, no. 12, pp. 2481–2495, 2017.

[9] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *CoRR*, pp. 2881–2890, 2017.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, eds.), pp. 234–241, Springer International Publishing, 2015.

[11] L. 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, pp. 834–848, April 2018.

[12] T. Team, "A tragic loss." https://www.tesla.com/en-GB/blog/tragic-loss, Jun 2016. [Accessed: 30-03-2019].

[13] J. Guynn, "Google photos labeled black people 'gorillas'." https://eu.usatoday.com/story/tech/2015/07/01/google-apologizes-after-photos-identify-black-people-as-gorillas/29567465/. [Accessed: 30-03-2019].

[14] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *CoRR*, vol. abs/1511.02680, 2015.

[15] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?," *CoRR*, vol. abs/1703.04977, 2017.

[16] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. 33nd International Conf. on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pp. 1050–1059, 2016.

[17] W. A. Abdulhafiz and A. Khamis, "Handling data uncertainty and inconsistency using multisensor data fusion," *Adv. in Artif. Intell.*, vol. 2013, pp. 11:11–11:11, Jan. 2013.

[18] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

[19] R. Neal, "Bayesian learning for neural networks [Phd thesis]," *Toronto, Ontario, Canada: Department of Computer Science, University of Toronto*, 1995.

[20] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *CoRR*, vol. abs/1710.08864, 2017.

[21] D. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *arXiv e-prints*, Jan 2014.

[22] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with bernoulli approximate variational inference," *CoRR*, vol. abs/1506.02158, 2015.

[23] F. Laumann, K. Shridhar, and A. L. Maurin, "Bayesian convolutional neural networks," *CoRR*, vol. abs/1806.05978, 2018.

[24] S. Sun, C. Chen, and L. Carin, "Learning structured weight uncertainty in bayesian neural networks," in *Proc. 20th International Conf. on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, pp. 1283–1292, 2017.

[25] "Will Bayesian deep learning be the next big thing? or is Bayesian modelling dead? [Seminal Talk]." Advances in Neural Information Processing Systems 29th Annual Conf. on Neural Information Processing Systems 2016, NeurIPS 2016 [Accessed: 2019-04-10].

[26] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[27] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. 27th Advances in Neural Information Processing (NeurIPs 2014)*, pp. 2672–2680, 2014.

[28] R. Salakhutdinov and G. Hinton, "Deep boltzmann machines," in *Proc. 12th International Conf. on Artificial Intelligence and Statistics (AISTATS 2009)* (D. van Dyk and M. Welling, eds.), vol. 5, pp. 448–455, PMLR, 16–18 Apr 2009.

[29] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[30] M. Gutmann and A. Hyvrinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proc. 13th International Conf. on Artificial Intelligence and Statistics (AISTATS 2010)* (Y. W. Teh and M. Titterington, eds.), vol. 9, pp. 297–304, PMLR, 13 May 2010.

[31] W. Hung, Y. Tsai, Y. Liou, Y. Lin, and M. Yang, "Adversarial learning for semi-supervised semantic segmentation," *CoRR*, vol. abs/1802.07934, 2018.

[32] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," *CoRR*, vol. abs/1611.08408, 2016.

[33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[35] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, and Z. Wang, "Multi-class generative adversarial networks with the L2 loss function," *CoRR*, vol. abs/1611.04076, 2016.

[36] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. 34th International Conf. on Machine Learning*

(ICML 2017) (D. Precup and Y. W. Teh, eds.), vol. 70, pp. 214–223, PMLR, 06 Aug 2017.

[37] Y. Choi, M. J. Choi, M. Kim, J. W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," *CoRR*, vol. abs/1711.09020, 2017.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Published as a Conf. paper at the 3rd International Conf. for Learning Representations, San Diego, 2015.

[39] L. Mescheder, S. Nowozin, and A. Geiger, "The numerics of GANs," *CoRR*, vol. abs/1705.10461, 2017.

[40] S. Song, S. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," pp. 567–576, 06 2015.

[41] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conf. on Learning Representations*, 2015.