Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding

Alex Kendall Vijay Badrinarayanan Roberto Cipolla
University of Cambridge
agk34, vb292, rc10001 @cam.ac.uk

Abstract

We present a novel deep learning framework for probabilistic pixel-wise semantic segmentation, which we term Bayesian SegNet. Pixel-wise semantic segmentation is an important step for visual scene understanding. It is a complex task requiring knowledge of support relationships and contextual information, as well as visual appearance. Our contribution is a practical system which is able to predict pixel-wise class labels with a measure of model uncertainty. We achieve this by Monte Carlo sampling with dropout at test time to generate a posterior distribution of pixel class labels. We show this Bayesian neural network provides a significant performance improvement in segmentation, with no additional parameterisation. We set a new benchmark with state-of-the-art performance on both the indoor SUN Scene Understanding and outdoor CamVid driving scenes datasets. Bayesian SegNet also performs competitively on Pascal VOC 2012 object segmentation challenge.

1. Introduction

Semantic segmentation requires an understanding of an image at a pixel level and is an important tool for scene understanding. It is a difficult problem as scenes often vary significantly in pose and appearance. However it is an important problem as it can be used to infer scene geometry and object support relationships. This has wide ranging applications from robotic interaction to autonomous driving.

Previous approaches to scene understanding used low level visual features [30]. We are now beginning to see the emergence of machine learning techniques for this problem [29, 23]. In particular deep learning [23] has set the benchmark on many popular datasets [9, 6]. However none of these methods produce a probabilistic segmentation or a measure of model uncertainty.

An important step in applying the output of a scene understanding system is knowing the confidence with which we can trust the semantic segmentation output. For instance, a system on an autonomous vehicle may segment an object as a pedestrian. But it is desirable to know the model uncertainty with respect to other classes such as street sign or cyclist as this can have a strong effect on behavioural decisions.

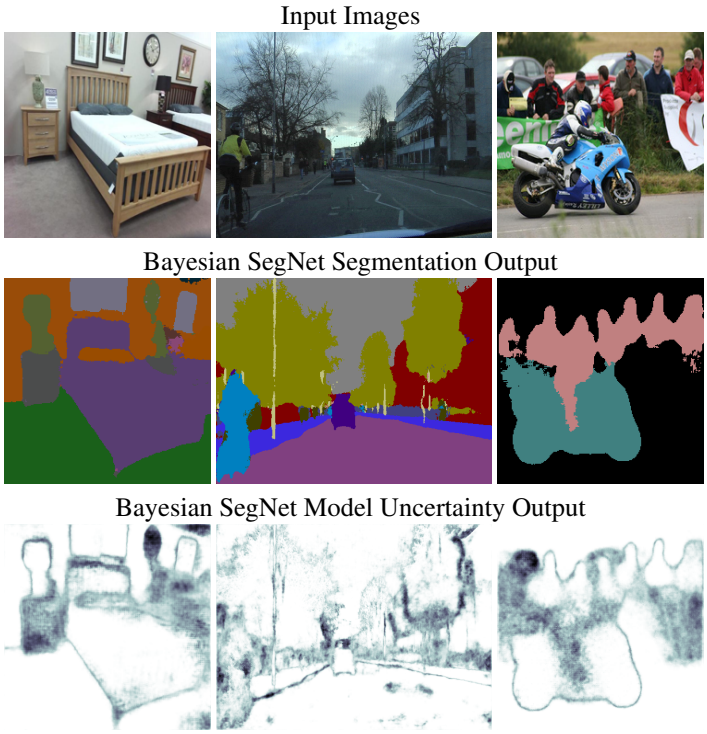


Figure 1: **Bayesian SegNet.** These examples show the performance of Bayesian SegNet on popular segmentation and scene understanding benchmarks: SUN [33] (left), CamVid [3] (center column) and Pascal VOC [9] (right). The system takes an RGB image as input (top), and outputs a semantic segmentation (middle row) and model uncertainty estimate, averaged across all classes (bottom row). We observe higher model uncertainty at object boundaries and with visually difficult objects. Our web demo and full source code are publicly available at mi.eng.cam.ac.uk/projects/segnet/

The main contribution of this paper is extending deep convolutional encoder-decoder neural network architectures [2] to Bayesian convolutional neural networks which can produce a probabilistic output [11]. We propose Bayesian SegNet, a probabilistic deep convolutional neural network framework for pixel-wise semantic segmentation. We use dropout at test time which allows us to approximate the posterior distribution by sampling from the Bernoulli distribution across the network's weights. This is achieved with no additional parameterisation.

We show that our Bayesian SegNet outputs a measure of model uncertainty. This measure can be used to understand with what confidence we can trust image segmentations and to determine to what degree of specificity we can assign a semantic label. For example, can we say that the label is a truck, or simply a moving vehicle? We qualitatively show that the model uncertainty reflects the visual ambiguity in images.

Finally, we present results which show that this probabilistic approach also increases the performance of the core segmentation engine. We set the best performing benchmark on prominent scene understanding datasets, CamVid Road Scenes [3] and SUN RGB-D Indoor Scene Understanding [33]. Additionally we obtain a competitive result on the Pascal VOC 2012 benchmark [9].

2. Related Work

Semantic pixel labelling was initially approached with TextonBoost [30], TextonForest [28] and Random Forest Based Classifiers [29]. We are now seeing the emergence of deep learning architectures for pixel wise segmentation, following its success in object recognition for a whole image [19]. Architectures such as SegNet [2] and Fully Convolutional Networks (FCN) [23] have been proposed, which we refer to as the *core segmentation engine*. FCN is trained using stochastic gradient descent with a stage-wise training scheme. SegNet was the first architecture proposed that can be trained end-to-end in one step, due to its lower parameterisation.

We have also seen methods which improve on these core segmentation engine architectures by adding post processing tools. HyperColumn [14] and DeConvNet [25] use region proposals to bootstrap their *core segmentation engine*. DeepLab [5] post-processes with conditional random fields (CRFs) and CRF-RNN [40] use recurrent neural networks. These methods improve performance by smoothing the output and ensuring label consistency. However none of these proposed segmentation methods generate a probabilistic output.

Neural networks which model uncertainty are known as Bayesian neural networks [7, 24]. They offer a probabilistic interpretation of deep learning models by inferring distributions over the networks weights. They are often computationally very expensive, increasing the number of model parameters without increasing model capacity significantly. Performing inference in Bayesian neural networks is a difficult task, and approximations to the model posterior are often used, such as variational inference [12].

On the other hand, the already significant parameterization of convolutional network architectures leaves them particularly susceptible to over-fitting without large amounts of training data. A technique known as *dropout* is commonly used as a regularizer in convolutional neural networks to prevent overfitting and co-adaption of features [34]. During training with stochastic gradient descent, *dropout* randomly removes units within a network. By doing this it samples from a number of thinned networks with reduced width. At test time, standard dropout approximates the effect of averaging the predictions of all these thinnned networks by using the weights of the unthinned network. This is referred to as *weight averaging*.

Gal and Ghahramani [11] have cast dropout as approximate Bayesian inference over the network's weights. [10] shows that dropout can be used at test time to impose a Bernoulli distribution over the convolutional net filter's weights, without requiring any additional model parameters. This is achieved by sampling the network with randomly dropped out units at test time. We can consider these as Monte Carlo samples obtained from the posterior distribution over models. This technique has seen success in modelling uncertainty for camera relocalisation [17]. Here we apply it to pixel-wise semantic segmentation.

We note that the probability distribution from Monte Carlo sampling is significantly different to the *'probabilities'* obtained from a softmax classifier. The softmax function approximates relative probabilities between the class labels, but not an overall measure of the model's uncertainty [11].

3. SegNet Architecture

We briefly review the SegNet architecture [2] which we modify to produce Bayesian SegNet. SegNet is a deep convolutional encoder decoder architecture which consists of a sequence of non-linear processing layers (encoders) and a corresponding set of decoders followed by a pixel-wise classifier. Typically, each encoder consists of one or more convolutional layers with batch normalisation and a ReLU non-linearity, followed by non-overlapping max-pooling and sub-sampling. The sparse encoding due to the pooling process is upsampled in the decoder using the max-pooling indices in the encoding sequence. This has the important advantage of retaining class boundary details in the segmented images and also reducing the total number of model parameters. The model is trained end to end using stochastic gradient descent.

We take both SegNet [2] and a smaller variant termed

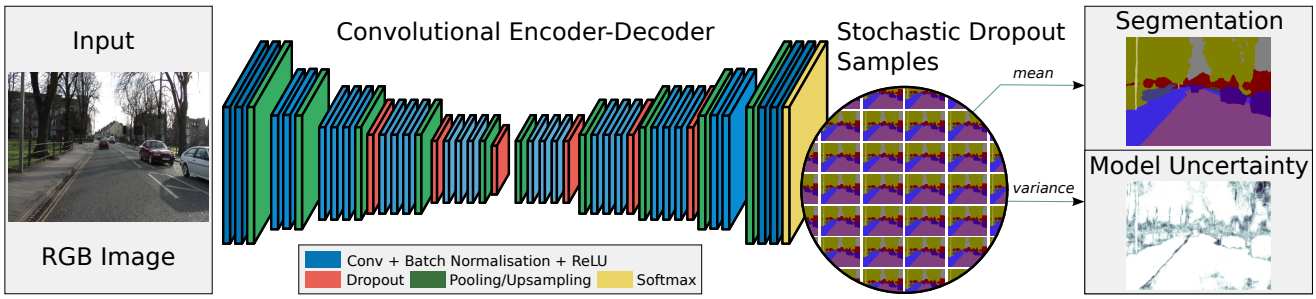


Figure 2: **A schematic of the Bayesian SegNet architecture.** This diagram shows the entire pipeline for the system which is trained end-to-end in one step with stochastic gradient descent. The encoders are based on the 13 convolutional layers of the VGG-16 network [32], with the decoder placing them in reverse. The probabilistic output is obtained from Monte Carlo samples of the model with dropout at test time. We take the variance of these softmax samples as the model uncertainty for each class.

SegNet-Basic [1] as our base models. SegNet's encoder is based on the 13 convolutional layers of the VGG-16 network [32] followed by 13 corresponding decoders. SegNet-Basic is a much smaller network with only four layers each for the encoder and decoder with a constant feature size of 64. We use SegNet-Basic as a smaller model for our analysis since it conceptually mimics the larger architecture.

4. Bayesian SegNet

The technique we use to form a probabilistic encoder-decoder architecture is dropout [34], which we use as approximate inference in a Bayesian neural network [10]. We can therefore consider using dropout as a way of getting samples from the posterior distribution of models. Gal and Ghahramani [10] link this technique to variational inference in Bayesian convolutional neural networks with Bernoulli distributions over the network's weights. We leverage this method to perform probabilistic inference over our segmentation model, giving rise to Bayesian SegNet.

For Bayesian SegNet we are interested in finding the posterior distribution over the convolutional weights, $\mathbf{W}$, given our observed training data $\mathbf{X}$ and labels $\mathbf{Y}$.

$$p(\mathbf{W} \mid \mathbf{X}, \mathbf{Y}) \tag{1}$$

In general, this posterior distribution is not tractable, therefore we need to approximate the distribution of these weights [7]. Here we use variational inference to approximate it [12]. This technique allows us to learn the distribution over the network's weights, $q(\mathbf{W})$, by minimising the Kullback-Leibler (KL) divergence between this approximating distribution and the full posterior;

$$\mathrm{KL}(q(\mathbf{W}) \,||\, p(\mathbf{W} \mid \mathbf{X}, \mathbf{Y})). \tag{2}$$

Here, the approximating variational distribution $q(\mathbf{W_i})$ for every $K \times K$ dimensional convolutional layer i, with units j, is defined as:

$$\begin{aligned} \mathbf{b}_{i,j} &\sim \text{Bernoulli}(p_i) \text{ for } j = 1, ..., K_i, \\ \mathbf{W_i} &= \mathbf{M}_i \text{diag}(b_i), \end{aligned} \tag{3}$$

with b_i vectors of Bernoulli distributed random variables and variational parameters $\mathbf{M_i}$ we obtain the approximate model of the Gaussian process in [10]. The dropout probabilities, p_i, could be optimised. However we fix them to the standard probability of dropping a connection as 50%, i.e. $p_i = 0.5$ [34].

In [10] it was shown that minimising the cross entropy loss objective function has the effect of minimising the Kullback-Leibler divergence term. Therefore training the network with stochastic gradient descent will encourage the model to learn a distribution of weights which explains the data well while preventing over-fitting.

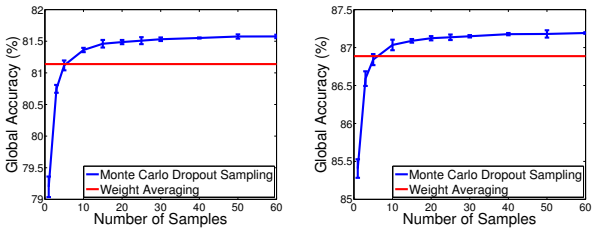
We train the model with dropout and sample the posterior distribution over the weights at test time using dropout to obtain the posterior distribution of softmax class probabilities. We take the *mean of these samples for our segmentation prediction* and use the *variance to output model uncertainty for each class*. We take the mean of the per class variance measurements as an overall measure of model uncertainty. We also explored using the *variation ratio* as a measure of uncertainty (i.e. the percentage of samples which agree with the class prediction) however we found this to produce a more binary measure of model uncertainty. Fig. 2 shows a schematic of the segmentation prediction and model uncertainty estimate process.

4.1. Probabilistic Variants

A fully Bayesian network should be trained with dropout after every convolutional layer. However we found in practice that this was too strong a regulariser, causing the network to learn very slowly. We therefore explored a number of variants that have different configurations of Bayesian

Probabilistic Variants	Weight Averaging			Monte Carlo Sampling			Training Fit		
	G	C	I/U	G	C	I/U	G	C	I/U
No Dropout	82.9	62.4	46.4	n/a	n/a	n/a	94.7	**96.2**	**92.7**
Dropout Encoder	80.6	68.9	53.4	81.6	69.4	54.0	90.6	92.5	86.3
Dropout Decoder	82.4	64.5	48.8	82.6	62.4	46.1	94.6	96.0	92.4
Dropout Enc-Dec	79.9	69.0	54.2	79.8	68.8	54.0	88.9	89.0	80.6
Dropout Central Enc-Dec	81.1	**70.6**	**55.7**	81.6	**70.6**	**55.8**	90.4	92.3	85.9
Dropout Center	82.9	68.9	53.1	82.7	68.9	53.2	93.3	95.4	91.2
Dropout Classifier	**84.2**	62.6	46.9	**84.2**	62.6	46.8	**94.9**	96.0	92.3

Table 1: **Architecture Variants for SegNet-Basic on the CamVid dataset** [3]. We compare the performance of weight averaging against 50 Monte Carlo samples. We quantify performance with three metrics; global accuracy (G), class average accuracy (C) and intersection over union (I/U). Results are shown as percentages (%). We observe that dropping out every encoder and decoder is too strong a regulariser and results in a lower training fit. The optimal result across all classes is when only the central encoder and decoders are dropped out.



(a) SegNet Basic (b) SegNet

Figure 3: **Global segmentation accuracy against number of Monte Carlo samples for both SegNet and SegNet-Basic.** Results averaged over 5 trials, with two standard deviation error bars, are shown for the CamVid dataset. This shows that Monte Carlo sampling outperforms the weight averaging technique after approximately 6 samples. Monte Carlo sampling converges after around 40 samples with no further significant improvement beyond this point.

or deterministic encoder and decoder units. We note that an encoder unit contains one or more convolutional layers followed by a max pooling layer. A decoder unit contains one or more convolutional layers followed by an upsampling layer. The variants are as follows:

- **Bayesian Encoder**. In this variant we insert dropout after each encoder unit.
- **Bayesian Decoder**. In this variant we insert dropout after each decoder unit.
- **Bayesian Encoder-Decoder**. In this variant we insert dropout after each encoder and decoder unit.
- **Bayesian Center**. In this variant we insert dropout after the deepest encoder, between the encoder and decoder stage.
- **Bayesian Central Four Encoder-Decoder**. In this variant we insert dropout after the central four encoder and decoder units.
- **Bayesian Classifier**. In this variant we insert dropout after the last decoder unit, before the classifier.

For analysis we use the smaller eight layer SegNet-Basic architecture [2] and test these Bayesian variants on the CamVid dataset [3]. We observe qualitatively that all four variants produce similar looking model uncertainty output. That is, they are uncertain near the border of segmentations and with visually ambiguous objects, such as cyclist and pedestrian classes. However, Table 1 shows a difference in quantitative segmentation performance.

We observe using dropout after all the encoder and decoder units results in a lower training fit and poorer test performance as it is too strong a regulariser on the model. We find that dropping out half of the encoder or decoder units is the optimal configuration. The best configuration is dropping out the deepest half of the encoder and decoder units. We therefore benchmark our Bayesian SegNet results on the Central Enc-Dec variant. For the full 26 layer Bayesian SegNet, we add dropout to the central six encoders and decoders. This is illustrated in Fig. 2.

In the lower layers of convolutional networks basic features are extracted, such as edges and corners [38]. These results show that applying Bayesian weights to these layers does not result in a better performance. We believe this is because these low level features are consistent across the distribution of models because they are better modelled with deterministic weights. However, the higher level features that are formed in the deeper layers, such as shape and contextual relationships, are more effectively modelled with Bayesian weights.

4.2. Comparing Weight Averaging and Monte Carlo Dropout Sampling

Monte Carlo dropout sampling qualitatively allows us to understand the model uncertainty of the result. However, for segmentation, we also want to understand the quantitative difference between sampling with dropout and using the weight averaging technique proposed by [34]. Weight averaging proposes to remove dropout at test time and scale the weights proportionally to the dropout percentage. Fig. 3 shows that Monte Carlo sampling with dropout performs better than weight averaging after approximately 6 samples. We also observe no additional performance improvement beyond approximately 40 samples. Therefore the weight averaging technique produces poorer segmentation results, in terms of global accuracy, in addition to being unable to provide a measure of model uncertainty. However, sampling comes at the expense of inference time, but when computed in parallel on a GPU this cost can be reduced for practical applications.

Method	Building	Tree	Sky	Car	Sign-Symbol	Road	Pedestrian	Fence	Column-Pole	Side-walk	Bicyclist	Class avg.	Global avg.	Mean I/U
SfM+Appearance [4]	46.2	61.9	89.7	68.6	42.9	89.5	53.6	46.6	0.7	60.5	22.5	53.0	69.1	n/a
Boosting [35]	61.9	67.3	91.1	71.1	58.5	92.9	49.5	37.6	25.8	77.8	24.7	59.8	76.4	n/a
Dense Depth Maps [39]	85.3	57.3	95.4	69.2	46.5	**98.5**	23.8	44.3	22.0	38.1	28.7	55.4	82.1	n/a
Structured Random Forests [18]	n/a											51.4	72.5	n/a
Neural Decision Forests [27]	n/a											56.1	82.1	n/a
Local Label Descriptors [37]	80.7	61.5	88.8	16.4	n/a	98.0	1.09	0.05	4.13	12.4	0.07	36.3	73.6	n/a
Super Parsing [36]	87.0	67.1	96.9	62.7	30.1	95.9	14.7	17.9	1.7	70.0	19.4	51.2	83.3	n/a
Boosting + pairwise CRF [35]	70.7	70.8	94.7	74.4	55.9	94.1	45.7	37.2	13.0	79.3	23.1	59.9	79.8	n/a
Boosting+Higher order [35]	84.5	72.6	**97.5**	72.7	34.1	95.3	34.2	45.7	8.1	77.6	28.5	59.2	83.8	n/a
Boosting+Detectors+CRF [20]	81.5	76.6	96.2	78.7	40.2	93.9	43.0	47.6	14.3	81.5	33.9	62.5	83.8	n/a
SegNet-Basic (layer-wise training [1])	75.0	84.6	91.2	82.7	36.9	93.3	55.0	37.5	44.8	74.1	16.0	62.9	84.3	n/a
SegNet-Basic [2]	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 [2]	**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
Bayesian SegNet Models in this work:														
Bayesian SegNet-Basic	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
Bayesian SegNet	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**

Table 2: **Quantitative results on CamVid** [3] consisting of 11 road scene categories. Bayesian SegNet outperforms all other methods, including those using depth, video and CRF's. Particularly noteworthy are the significant improvements in accuracy for the smaller classes.

4.3. Training and Inference

Following [2] we train SegNet with median frequency class balancing using the formula proposed by Eigen and Fergus [8]. We use batch normalisation layers after every convolutional layer [15]. We compute batch normalisation statistics across the training dataset and use these at test time. We experimented with computing these statistics while using dropout sampling. However we found computing them with weight averaging produced better results experimentally.

We implement Bayesian SegNet using the Caffe library [16] and release the source code and trained models for public evaluation.[1] We train the whole system end-to-end using stochastic gradient descent with a base learning rate of 0.001 and weight decay parameter equal to 0.0005. We train the network until convergence when we observe no further reduction in training loss.

5. Experiments

We quantify the performance of Bayesian SegNet on three different benchmarks using our Caffe implementation. Through this process we demonstrate the efficacy of Bayesian SegNet for a wide variety of scene segmentation tasks which have practical applications. CamVid [3] is a road scene understanding dataset which has applications for autonomous driving. SUN RGB-D [33] is a very challenging and large dataset of indoor scenes which is important for domestic robotics. Finally, Pascal VOC 2012 [9] is a RGB dataset for object segmentation.

[1]Our web demo and full source code is publicly available at mi.eng.cam.ac.uk/projects/segnet/

5.1. CamVid

CamVid is a road scene understanding dataset with 367 training images and 233 testing images of day and dusk scenes [3]. The challenge is to segment 11 classes such as road, building, cars, pedestrians, signs, poles, side-walk etc. We resize images to 360x480 pixels for training and testing of our system.

Table 2 shows our results and compares them to previous benchmarks. Bayesian SegNet obtains the highest overall class average and intersection over union score by a significant margin. We set a new benchmark on 7 out of the 11 classes. Qualitative results can be viewed in Fig. 4.

5.2. Scene Understanding (SUN)

SUN RGB-D [33] is a very challenging and large dataset of indoor scenes with 5285 training and 5050 testing images. The images are captured by different sensors and hence come in various resolutions. The task is to segment 37 indoor scene classes including wall, floor, ceiling, table, chair, sofa etc. This task is difficult because object classes come in various shapes, sizes and in different poses with frequent partial occlusions. These factors make this one of the hardest segmentation challenges. For our model, we resize the input images for training and testing to 224x224 pixels. Note that we only use RGB input to our system. Using the depth modality would necessitate architectural modifications and careful post-processing to fill-in missing depth measurements. This is beyond the scope of this paper.

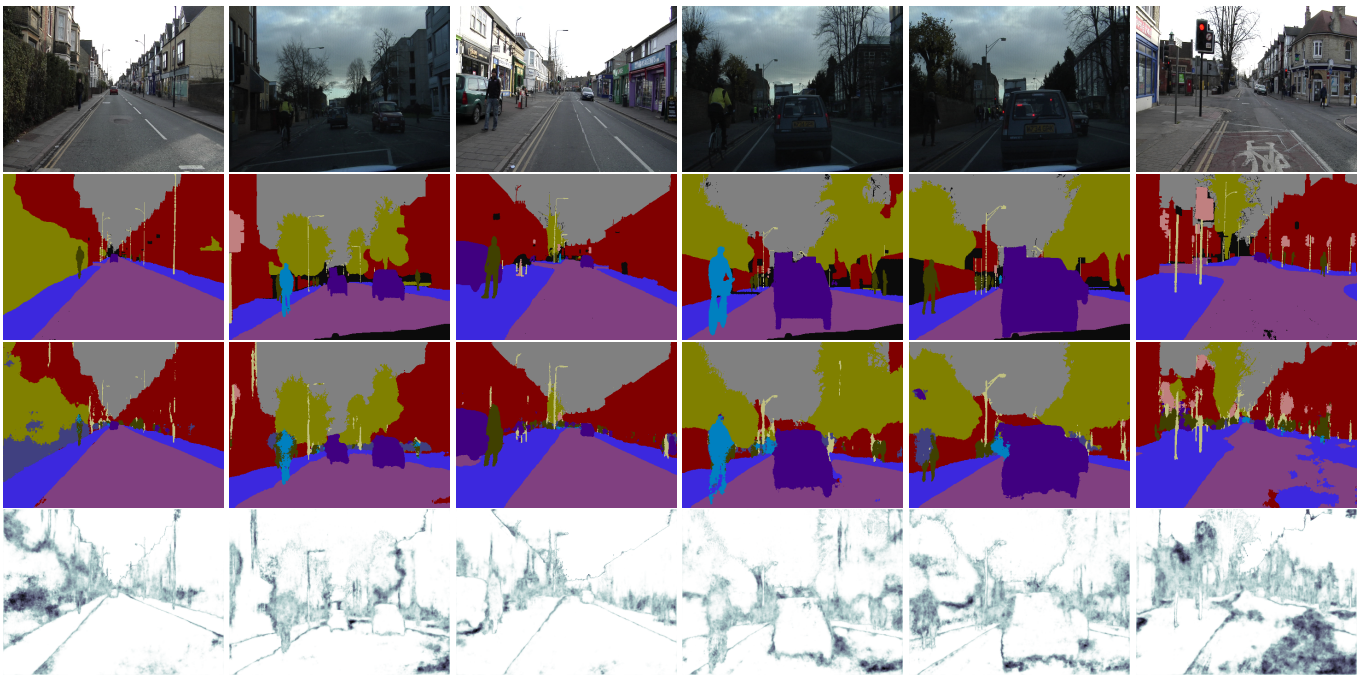


Figure 4: **Bayesian SegNet results on CamVid road scene understanding dataset [3].** The top row is the input image, with the ground truth shown in the second row. The third row shows Bayesian SegNet's segmentation prediction, with overall model uncertainty, averaged across all classes, in the bottom row (with darker colours indicating more uncertain predictions). In general, we observe high quality segmentation, especially on more difficult classes such as poles, people and cyclists. Where SegNet produces an incorrect class label we often observe a high model uncertainty.

Figure 5: **Bayesian SegNet results on the SUN RGB-D indoor scene understanding dataset [33].** The top row is the input image, with the ground truth shown in the second row. The third row shows Bayesian SegNet's segmentation prediction, with overall model uncertainty, averaged across all classes, in the bottom row (with darker colours indicating more uncertain predictions). Bayesian SegNet uses only RGB input and is able to accurately segment 37 classes in this challenging dataset. Note that often parts of an image do not have ground truth labels and these are shown in black colour.

Table 3 shows our results on this dataset compared to previous methods. Bayesian SegNet outperforms all previous benchmarks, including those which use depth modality. We also note that an earlier benchmark dataset, NYUv2 [31], is included as part of this dataset, and Table 4 shows our evaluation on this subset. Qualitative results can be viewed in Fig. 5.

5.3. Pascal VOC

The Pascal VOC12 segmentation challenge [9] consists of segmenting a 20 salient object classes from a widely varying background class. For our model, we resize the input images for training and testing to 224x224 pixels. We train on the 12031 training images and 1456 testing images, with scores computed remotely on a test server. Table 6 shows our results compared to previous methods, with qualitative results in Fig. 6.

This dataset is unlike the segmentation for scene understanding benchmarks described earlier which require learning both classes and their spatial context. A number of techniques have been proposed based on this challenge which are increasingly more accurate and complex [2]. Our efforts in this benchmarking experiment have not been diverted towards attaining the top rank by either using multi-stage training [23], other datasets for pre-training such as MS-COCO [21], training and inference aids such as object proposals [25] or post-processing using CRF based methods [5, 40]. Although these supporting techniques clearly have value towards increasing the performance it unfortunately does not reveal the true performance of the deep architecture which is the *core segmentation engine*. It however does indicate that some of the large deep networks are difficult to train end-to-end on this task even with pre-trained encoder weights. Therefore, to encourage more controlled benchmarking, we trained Bayesian SegNet end-to-end without other aids and report this performance.

5.4. Qualitative Results

Fig. 4 shows segmentations and model uncertainty results from Bayesian SegNet on CamVid Road Scenes [3]. Fig. 5 shows SUN RGB-D Indoor Scene Understanding [33] results and Fig. 6 has Pascal VOC [9] results. Additional per-class qualitative results are presented in the supplementary material. These figures show the qualitative performance of Bayesian SegNet. We observe that segmentation predictions are smooth, with a sharp segmentation around object boundaries. These results also show that when the model predicts an incorrect label, the model uncertainty is generally very high. More generally, we observe that a high model uncertainty is predominantly caused by three situations.

[2]See the full leader board at http://host.robots.ox.ac.uk:8080/leaderboard

Method	G	C	I/U
RGB			
Liu *et al.* [22]	n/a	9.3	n/a
SegNet [2]	70.3	35.6	22.1
Bayesian SegNet	**71.2**	**45.9**	**30.7**
RGB-D			
Liu *et al.* [22]	n/a	10.0	n/a
Ren et. al [26]	n/a	36.3	n/a

Table 3: **SUN Indoor Scene Understanding.** Quantitative comparison on the SUN RGB-D dataset [33] which consists of 5050 test images of indoor scenes with 37 classes. SegNet RGB based predictions have a high global accuracy and out-perform all previous benchmarks, including those which use depth modality.

Method	G	C	I/U
RGB			
FCN-32s RGB [23]	60.0	42.2	29.2
SegNet [2]	66.1	36.0	23.6
Bayesian SegNet	**68.0**	45.8	32.4
RGB-D			
Gupta et al. [13]	60.3	-	28.6
FCN-32s RGB-D [23]	61.5	42.4	30.5
Eigen et al. [8]	65.6	45.1	-
RGB-HHA			
FCN-16s RGB-HHA [23]	65.4	**46.1**	**34.0**

Table 4: **NYU v2.** Results for the NYUv2 RGB-D dataset [31] which consists of 654 test images. Bayesian SegNet is the top performing RGB method, also outperforming all RGB-D methods.

Method	Parameters (Million)	Inference Time (ms)	Pascal VOC 2012 [9]
DeepLab [5]	134.5+	n/a	58
FCN-8 [23] (multi-stage training)	134.5	210	62.2
Hypercolumns [14] (object region proposals)	134.5+	n/a	62.6
DeconvNet [25] (object region proposals)	276.7	92 ($\times$ 50)	**69.6**
CRF-RNN [40] (multi-stage training)	134.5+	n/a	**69.6**
SegNet-Basic [2]	**1.2**	**36**	-
SegNet [2]	29.45	48	59.1
Bayesian SegNet-Basic	**1.2**	310	-
Bayesian SegNet	29.45	470	60.0

Table 6: **Pascal VOC12 dataset [9] results.** We compare to competing architectures with the least supporting training and inference techniques. However, since they are not trained end-to-end like SegNet and use aids such as object proposals, we have added corresponding qualifying comments. Many of the models are approximately the same size as FCN. In comparison, Bayesian SegNet is considerably smaller but achieves a competitive accuracy without these training or inference aids.

Firstly, at class boundaries the model often displays a high level of uncertainty. This reflects the ambiguity surrounding the definition of defining where these labels transition. The Pascal results clearly illustrated this in Fig. 6.

Secondly, objects which are visually difficult to identify

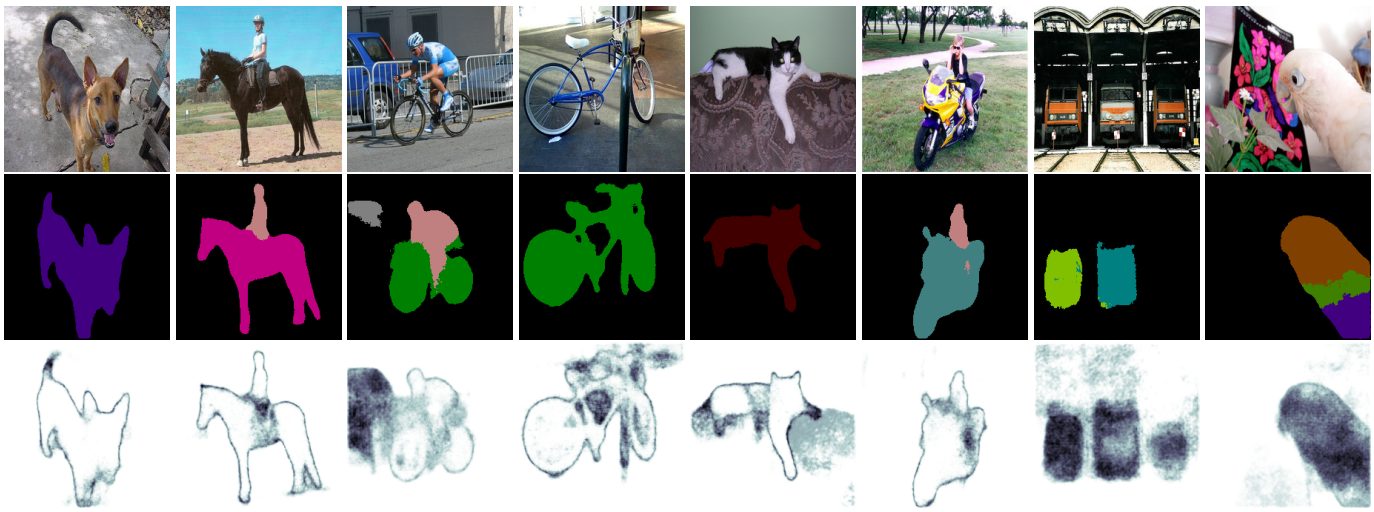


Figure 6: **Bayesian SegNet results on the Pascal VOC 2012 dataset [9].** The top row is the input image. The middle row shows Bayesian SegNet's segmentation prediction, with overall model uncertainty averaged across all classes in the bottom row (darker colours indicating more uncertain predictions). Ground truth is not publicly available for these test images.

	Wall	Floor	Cabinet	Bed	Chair	Sofa	Table	Door	Window	Bookshelf	Picture	Counter	Blinds	Desk	Shelves	Curtain	Dresser	Pillow	Mirror	Floor Mat	Clothes	Ceiling	Books	Fridge	TV	Paper	Towel	Shower curtain	Box	Whiteboard	Person	Night stand	Toilet	Sink	Lamp	Bathtub	Bag
Liu et. al (RGB) [22]	38.9	47.2	18.8	21.5	17.2	13.4	20.4	6.8	11.0	9.6	6.1	2.6	3.6	7.3	1.2	6.9	2.4	2.6	6.2	0.0	1.3	39.1	5.9	7.1	1.4	1.5	2.2	0.0	0.7	10.4	0.0	1.5	12.3	14.8	1.3	0.9	1.1
Ren et. al (RGB-D) [26]	43.2	78.6	26.2	42.5	33.2	40.6	34.3	33.2	43.6	23.1	57.2	31.8	42.3	12.1	**18.4**	59.1	31.4	49.5	24.8	**5.6**	**27.0**	**84.5**	35.7	**24.2**	36.5	26.8	**19.2**	**9.0**	11.7	51.4	**35.7**	25.0	64.1	53.0	44.2	**47.0**	18.6
SegNet [2]	**86.6**	**92.0**	52.4	**68.4**	**76.0**	54.3	59.3	37.4	53.8	29.2	49.7	32.5	31.2	17.8	5.3	53.2	28.8	36.5	29.6	0.0	14.4	67.7	32.4	10.2	18.3	19.2	11.5	0.0	8.9	38.7	4.9	22.6	55.6	52.7	27.9	45.1	8.1
Bayesian SegNet	80.2	90.9	**58.9**	64.8	**76.0**	**58.6**	**62.6**	**47.7**	**66.4**	**31.2**	**63.6**	**33.8**	**46.7**	**19.7**	16.2	**67.0**	**42.3**	**57.1**	**39.1**	0.1	24.4	84.0	**48.7**	21.3	**49.5**	**30.6**	18.8	0.1	**24.1**	**56.8**	17.9	**42.9**	**73.0**	**66.2**	**48.8**	45.1	**24.1**

Table 5: Class accuracy of Bayesian SegNet predictions for the 37 indoor scene classes in the **SUN RGB-D benchmark dataset** [33]. Bayesian SegNet sets a new benchmark in 25 of these classes.

often appear uncertain to the model. This is often the case when objects are occluded or at a distance from the camera.

The third situation causing model uncertainty is when the object appears visually ambiguous to the model. As an example, cyclists in the CamVid results (Fig. 4) are visually similar to pedestrians, and the model often displays uncertainty around them. We observe similar results with visually similar classes in SUN (Fig. 5) such as chair and sofa, or bench and table. In Pascal this is often observed between cat and dog, or train and bus classes.

5.5. Real Time Performance

Table 6 shows that SegNet and Bayesian SegNet maintains a far lower parameterisation than its competitors. Monte Carlo sampling requires additional inference time, however if model uncertainty is not required, then the weight averaging technique can be used to remove the need for sampling (Fig. 3 shows the performance drop is modest). Inference time would then be identical to the SegNet model which can be run in real time on a GPU.

6. Conclusion

We have presented Bayesian SegNet, the first probabilistic framework for semantic segmentation using deep learning, which outputs a measure of model uncertainty for each class. Bayesian SegNet's qualitative results show that the model is uncertain at object boundaries and with difficult and visually ambiguous objects. Bayesian SegNet obtains the highest performing result on CamVid road scenes and SUN RGB-D indoor scene understanding datasets. We show that the segmentation model can be run in real time on a GPU. For future work we intend to explore how video data can improve our model's scene understanding performance.

References

[1] V. Badrinarayanan, A. Handa, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*, 2015. 3, 5

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015. 2, 4, 5, 7, 8

[3] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009. 1, 2, 4, 5, 6, 7

[4] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *Computer Vision–ECCV 2008*, pages 44–57. Springer, 2008. 5

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014. 2, 7

[6] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. *arXiv preprint arXiv:1301.3572*, 2013. 1

[7] J. Denker and Y. Lecun. Transforming neural-net output levels to probability distributions. In *Advances in Neural Information Processing Systems 3*. Citeseer, 1991. 2, 3

[8] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *arXiv preprint arXiv:1411.4734*, 2014. 5, 7

[9] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 1, 2, 5, 7, 8

[10] Y. Gal and Z. Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv:1506.02158*, 2015. 2, 3

[11] Y. Gal and Z. Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *arXiv:1506.02142*, 2015. 2

[12] A. Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems*, pages 2348–2356, 2011. 2, 3

[13] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Computer Vision–ECCV 2014*, pages 345–360. Springer, 2014. 7

[14] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. *arXiv preprint arXiv:1411.5752*, 2014. 2, 7

[15] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 5

[16] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 5

[17] A. Kendall and R. Cipolla. Modelling uncertainty in deep learning for camera relocalization. *arXiv preprint arXiv:1509.05909*, 2015. 2

[18] P. Kontschieder, S. Rota Buló, H. Bischof, and M. Pelillo. Structured class-labels in random forests for semantic image labelling. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2190–2197. IEEE, 2011. 5

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

[20] L. Ladickỳ, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr. What, where and how many? combining object detectors and crfs. In *Computer Vision–ECCV 2010*, pages 424–437. Springer, 2010. 5

[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*, pages 740–755. Springer, 2014. 7

[22] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. In *Computer Vision–ECCV 2008*, pages 28–42. Springer, 2008. 7, 8

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1411.4038*, 2014. 1, 2, 7

[24] D. J. MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992. 2

[25] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. *arXiv preprint arXiv:1505.04366*, 2015. 2, 7

[26] X. Ren, L. Bo, and D. Fox. Rgb-(d) scene labeling: Features and algorithms. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2759–2766. IEEE, 2012. 7, 8

[27] S. Rota Bulo and P. Kontschieder. Neural decision forests for semantic image labelling. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 81–88. IEEE, 2014. 5

[28] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Computer vision and pattern recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 2

[29] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013. 1, 2

[30] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23, 2009. 1, 2

[31] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012*, pages 746–760. Springer, 2012. 7

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[33] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 567–576, 2015. 1, 2, 5, 6, 7, 8

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 2, 3, 4

[35] P. Sturgess, K. Alahari, L. Ladicky, and P. H. Torr. Combining appearance and structure from motion features for road scene understanding. In *BMVC*, volume 1, page 6, 2009. 5

[36] J. Tighe and S. Lazebnik. Superparsing. *International Journal of Computer Vision*, 101(2):329–349, 2013. 5

[37] Y. Yang, Z. Li, L. Zhang, C. Murphy, J. Ver Hoeve, and H. Jiang. Local label descriptor for example based semantic image labeling. In *Computer Vision–ECCV 2012*, pages 361–375. Springer, 2012. 5

[38] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014. 4

[39] C. Zhang, L. Wang, and R. Yang. Semantic segmentation of urban scenes using dense depth maps. In *Computer Vision–ECCV 2010*, pages 708–721. Springer, 2010. 5

[40] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. *arXiv preprint arXiv:1502.03240*, 2015. 2, 7