Balancing Calibration and Performance: Stochastic Depth in Segmentation BNNs

Linghong Yao
leon.yao.18@alumni.ucl.ac.uk
Denis Hadjivelichkov
dennis.hadjivelichkov@ucl.ac.uk
Andromachi Maria Delfaki
andromachi.delfaki.23@ucl.ac.uk
Yuanchang Liu
yuanchang.liu@ucl.ac.uk
Brooks Paige
b.paige@ucl.ac.uk
Dimitrios Kanoulas
d.kanoulas@ucl.ac.uk

Department of Computer Science &
Mechanical Engineering
University College London
Gower Street, WC1E 6BT
London, UK

Abstract

In many safety-critical applications, it is critical for computer vision models to provide reliable uncertainty estimates. However, traditional Bayesian approaches often compromise between efficiency and safety. In this work, we introduce a novel implementation of stochastic depth within segmentation Bayesian Neural Networks (BNNs) that preserves performance while significantly improving uncertainty calibration. We experimentally validate our approach using an encoder-decoder model specifically tailored for real-time robotic vision tasks which demand fast and reliable decision-making under inherently uncertain conditions. Our method facilitates both safer and more effective deployment without compromises, increasing uncertainty calibration error whilst maintaining high performance.

1 Introduction

Many safety-critical applications of AI, such as robots operating in dynamic environments, require rapid and reliable processing of visual information in real-time. For instance, mobile robots often use deep convolutional neural networks (CNNs) for navigation, but CNNs with slow inference time, or overconfident predictions may result in higher chances of collisions, posing risks to both the robot and its surroundings. Having efficient segmentation models that are able to reliably predict model uncertainty is essential for timely and appropriate control responses [2].

The high expressiveness of neural networks typically comes at the cost of increased computational requirements. Modern neural networks achieve better performance by becoming

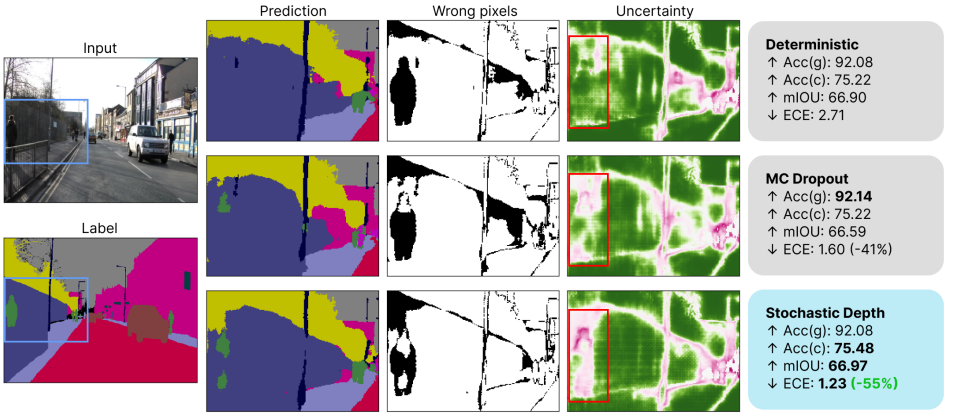

Figure 1: Comparison of a deterministic network, BNN with MC Dropout, and the proposed BNN with stochastic depth. Stochastic depth maintains high performance while achieving a lower calibration error (ECE). The wrong pixels column shows wrongly predicted pixel labels in black, highlighting where the network should be less confident. The uncertainty column shows uncertainty outputs where red pixels are uncertain and green are certain. The red boxes highlight that both deterministic and MC dropout methods are overconfident about incorrect predictions, while stochastic depth produces more locally calibrated uncertainties.

larger and deeper. The widely adopted ResNet is well known for its ability to increase expressiveness with more depth [18], and they are adopted by many popular segmentation networks, such as DeepLab [7], as the backbone encoder. The large network size and high FLOP count mean that these segmentation methods often cannot be run in real-time with the limited computational resources on mobile robots. Recent advances in vision transformers require even more parameters with upwards of 100M or more [8].

Modern neural networks also present critical safety issues, whereby they are often "confidently wrong". The overconfidence phenomenon makes it difficult to implement neural networks in safety-critical applications, where the interpretability of the network's predictions is vital for users and systems. Modern neural networks often fail in safety-critical roles due to poor uncertainty calibration [15], vulnerability to adversarial attacks [30], and inability to recognize out-of-domain inputs [36]. Networks should ideally display calibrated uncertainties, where confidence aligns with accuracy, and manage risks by invoking conservative or human-guided actions [10].

This paper addresses these two open-ended critical issues of deep learning tasks in real-time, safety-critical applications such as robotics: *efficiency* and *safety*. We first propose an efficient encoder-decoder segmentation architecture that retains a high performance-to-parameter ratio, matching state-of-the-art models on half-resolution CamVid without any pretraining or post-processing. We then propose a novel Bayesian inference technique using stochastic depth in encoder-decoder networks. Our proposed method is simple to implement in encoder-decoder architectures, easy to scale to different computational constraints, and improves the uncertainty calibration without deteriorating accuracy, as illustrated in Fig. 1.

The main contributions of this work are 1) The proposal of an efficient encoder-decoder network design with simple scaling techniques, that improves performance-to-parameter ratio compared to existing encoder-decoder methods and on par with state-of-the-art bidirec-

tional and transformer-based methods, and 2) The proposal of using stochastic depth as a novel Bayesian approximation method in such segmentation networks, performing real-time end-to-end Bayesian inference with significantly more calibrated uncertainties than its deterministic counterparts.

2 Background and Related Work

2.1 Efficient Networks

The search for more efficient CNN architectures has become an active field of research in the past few years [19], where one seeks to find networks that achieve equal performances but with fewer parameters and faster inference speed. We identify two main approaches adopted in the literature: designing new building blocks and architectures, and modifying the network scale. Several efficient building blocks have been introduced to improve the traditional convolution layer. Jaderberg et al. [24] use low-rank $1 \times k$ and $k \times 1$ convolutions. Group convolutions, by [28], restrict output channels to same-group input channels. ShuffleNet [50] uses pointwise group convolution and channel shuffling for efficiency while SqueezeNet [23] employs a fire module with 1×1 and 3×3 filters and prunes the network via deep compression [17] for reduced inference size. Howard et al. [21] introduced depthwise separable convolution blocks to build an efficient convolutional neural network called MobileNet, and MobileNetV2 [40] extended the mobile performance of MobileNet by introducing inverted residuals with linear bottlenecks. On the network scaling side, He et al. [18] varied the depth of the ResNet network to increase its expressiveness, and observed that deeper networks are capable of capturing more complex features. In [21, 40], the authors explored changing the input resolutions and the network width to trade-off accuracy with network size and complexity, while Tan and Le [41] proposed a unified framework of scaling networks by a constant compound ratio across resolution, depth, and width.

Image segmentation is a widely used problem formulation adopted for identifying and detecting objects in scenes with densely packed entities. Long et al. [31] were the first to propose that using end-to-end convolution layers alone can achieve per-pixel classification. Several architectures followed which involved a downsampling part (encoder) of the network with an upsampling part (decoder). DeconvNet [35] includes a deeper decoding network with several convolution filters at each spatial dimension upsamples using deconvolutional layers. UNet [38] is a simple and effective encoder-decoder architecture with skip connections that has also shown promising results in a wide range of domains [1, 47]. Bilinski and Prisacariu [3] explored dense decoder shortcuts, where in addition to the skip connections between encoder and decoder, skip connections are also added in the decoder to form a densely connected decoder that fuses features at different scales. Furthermore, atrous convolution-based methods [7], and more recently, bilateral segmentation networks [48] and transformer-based networks [44] have also been introduced in literature as reliable segmentation architectures. This paper specifically focuses on *efficient designs* of the encoder-decoder architecture, as variational inference methods have been predominantly validated on these frameworks [26, 49]. Our methodology involves designing an efficient decoding path, modifying the skip connections of the network, and changing the scale of the network (both width and depth). We examine the effect on both the performance of the network measured in accuracy and mIoU, as well as the effect on the number of parameters and runtime.

2.2 Bayesian Deep Learning

In prediction problems, Bayesian learning identifies two sources of uncertainty: (i) *aleatoric uncertainty* on the ambiguity of the data itself; and (ii) *epistemic uncertainty* due to errors in the learned model. Aleatoric uncertainty is irreducible with more data, and is inherent to the data itself, in practical problems it often originates from sensor errors (e.g. camera resolution). On the other hand, epistemic uncertainty is related to the learned model, either from its structure or from stochasticity in its learning process. In addition, high variability of real-world data can also cause the model to have high uncertainty in out-of-distribution data. Epistemic uncertainty could be reduced with more training data and better model designs. Bayesian approaches are advocated for intelligent systems to be "uncertainty-aware" [2] and produce uncertainty estimates that encapsulate both aleatoric and epistemic uncertainties.

Bayesian Deep Learning (BDL) integrates Bayesian principles into deep neural networks, enhancing their capability to handle uncertainties and in various learning paradigms from active learning [11] to semi-supervised learning [49]. A wide range of notable methods have been proposed, including Laplace approximation [14, 39], sampling methods with MCMC [32, 34, 45], variational inference [13], and test time augmentations [33, 43]. Most related to this work are variational inference techniques, which approximate the posterior with a simpler distribution and optimize it using evidence lower bound (ELBO) [27]. Blundell et al. further optimize the evidence lower bound directly using stochastic minibatch gradients and backpropagation [4]. Gal et al. showed stochastic regularization methods to deep neural networks can be used for approximate Bayesian inference, specifically adopting dropout techniques [10]. It has also been demonstrated that deep ensembling is a direct way of obtaining a posterior estimation with as few as 5 networks [29], and such ensembles often outperform dropout-based methods in both accuracy and calibration, but with a much higher computational cost [16]. However, ensembles pose difficult computational constraints, both in terms of inference speed and memory costs. Thus, the approach of stochastic regularization on a single network is adopted by this paper to achieve efficient Bayesian inference.

3 Methodology

3.1 Efficient Network Design

This section focuses on the encoder-decoder architecture with the aim of improving the efficiency of these models. We first design an efficient decoding path, then modify the network with the location of skip connections, and finally scale the width and depth of the network. We examine the effect on the trade-off between performance, parameters, and runtime. Finally, we combine the results of these experiments to propose a set of simple ways to achieve high-performing networks with few parameters and high inference speed.

Fig. 2 illustrates the baseline architecture used in this paper. We adopt a typical encoder-decoder structure and propose a symmetrical block design. Skip connections between the encoder and the decoder are adopted to directly transfer features from the encoder to the decoder without additional computation. We highlight that the intermediate features are held in memory during the forward pass, and can only be released when the corresponding decoder block is reached. This presents a trade-off between retaining high spatial dimension features versus computational resources, where more fine-grained features can be retained at the cost of increased inference time and memory costs.

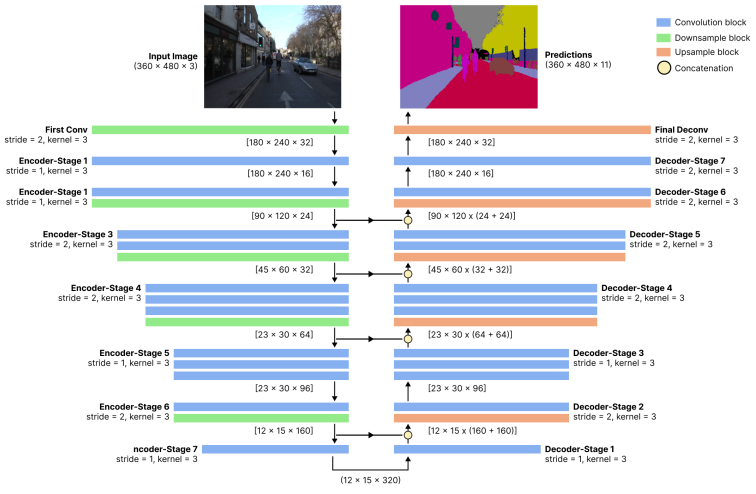

Figure 2: The efficient segmentation network template uses MobileNetV2 as the encoder, a symmetrical decoder with linear bottleneck and depthwise separable blocks, and skip connections at each scaling stage to maintain spatial information.

We chose MobileNetV2 as the backbone encoder because it is a reliable lightweight CNN with a good performance-to-parameter ratio on ImageNet [40]. On the decoder side, we experiment with three variations: 1) using standard deconvolution layers, 2) using depth-wise separable deconvolution layers, and 3) using inverted residuals with depth-wise separable deconvolutional layers, analogous to MobileNetV2's encoder blocks. Our preliminary experiments showed that with the same number of parameters, inverted residual decoders achieved the highest performance, recovering more information and achieving better performance-to-parameter ratio, thus this decoder design is adopted for the remainder of the paper. For more details of the preliminary experiments, please refer Section 3.1 of our Supplementary Material.

3.2 Bayesian Inference with Stochastic Depth

We explore variants of Bayesian Neural Networks (BNN) that use stochastic predictions at test time to approximate the predictive distribution through the posterior. One of these widely-used methods is the MC Dropout [10], which we implement as a baseline for comparison. MC Dropout has been shown as a promising method for obtaining a Bayesian posterior, especially in segmentation networks [1]. This method samples multiple subnets using dropout layers, and uses the combination of the output of the subnets as an approximation for the posterior distribution. One can interpret the averaged output as the result from *an ensemble of narrower networks*. On the other hand, recent results have shown that MC Dropout may produce worsened predictive results at the cost of improved calibration [2, 42], and this calls for alternative Bayesian techniques that achieve the same performance as deterministic inference, or improve from it. Next, we detail our method of using stochastic depth as a Bayesian framework and introduce metrics for computing the uncertainty calibration.

3.2.1 BNN with Stochastic Depth

Stochastic depth was first introduced as a regularization technique to mitigate vanishing gradients during the training process of deep networks [22] with skip connections. The idea is to use shorter networks during training by stochastically skipping a proportion of layers, and then retrieving the full network at test time. This is performed with a simple Bernoulli random variable b_ℓ for each layer, with "survival" probability $p_\ell = P(b_\ell = 1)$, followed by the forward pass $y_\ell = \text{ReLU}(b_\ell \mathcal{F}(\mathbf{x}, \{W_\ell\}) + \mathbf{x})$. A simple linear decay rule $p_\ell = 1 - \frac{\ell}{L}(1 - p_L)$ is used to effectively set the survival probability of each layer, decreasing it in deeper layers, where p_L is the survival probability of the last layer, such that the deepest parts of the network have the lowest survival probability. An interesting property of using stochastic depth at test time is that it can be interpreted as an *ensemble of shallower networks* with the same base building blocks but different depths. This presents a natural interpretation similar to MC dropout, with the added benefit that the forward pass is often faster. The number of possible network configurations grows exponentially with the number of blocks, and we can sample networks with different depths with each stochastic pass. Therefore, we can interpret Bayesian inference with stochastic depth as a Bayesian method that captures uncertainty in the structure of the network, specifically its depth.

3.2.2 Uncertainty Quantification

Consider that we perform T stochastic forward passes with any regularization technique, such as MC Dropout or stochastic depth as described above. The uncertainty of the network is commonly computed using mutual information or predictive entropy [12]. We can measure the predictive entropy or the information given in the expected softmax probabilities as $\mathbb{H}(y|x,\mathcal{D}) = -\sum_c \left(\frac{1}{T}\sum_{t=1}^{T} p(y=c|x,w_t)\right) \log\left(\frac{1}{T}\sum_{t=1}^{T} p(y=c|x,w_t)\right)$. The mutual information, on the other hand, measures the mutual dependence between the information given in the expected softmax output and the expected information in the softmax output $\mathbb{I}(y,w|x,\mathcal{D}) = \mathbb{H}(y|x,\mathcal{D}) - \mathbb{E}_{w\sim p(w|\mathcal{D})}[\mathbb{H}(y|x,\mathcal{D})]$. Mutual information can be interpreted as a measure of model uncertainty or epistemic uncertainty. It is minimized when the knowledge about the model parameter does not increase the information in the final prediction. On the other hand, predictive entropy can be interpreted as the sum of epistemic and aleatoric uncertainty.

3.2.3 Uncertainty Calibration

What does it mean for a network to produce "good" uncertainties? Our learning problem has no ground truth uncertainty, rather, we require "calibrated uncertainties". Consider a classification network with softmax outputs, where $\hat{Y}$ is the network prediction and $\hat{P}$ is the associated confidence with that prediction. We define *perfect calibration* as the confidence is equal to the true probability that $y = \hat{Y}$ given $\hat{P}$. In other words $\mathbb{P}\left(\hat{Y} = Y|\hat{P} = p\right) = p$.

In practice, we discretize the space of probability into M bins each with width $1/M$. Let B_m be the set of indices of samples with confidence score $p \in \left(\frac{m-1}{M}, \frac{m}{M}\right]$. Accuracy and confidence of bin B_m are defined respectively as $\text{acc}(B_m) = \frac{1}{|B_m|}\sum_{i\in B_m} \mathbb{I}[\hat{y}_i = y_i]$, and $\text{conf}(B_m) = \frac{1}{|B_m|}\sum_{i\in B_m} \hat{p}_i$. Two natural calibration metrics arise from this are expected calibration error (ECE) and maximum calibration error (MCE). We can also compute the ECE and MCE using discrete bins: $\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)|$, $\text{MCE} = \max_m |\text{acc}(B_m) - \text{conf}(B_m)|$.

4 Evaluation

To evaluate our method, we use the Cambridge-driving labelled Video Database (CamVid), a road-scene dataset captured from the perspective of a driving automobile [5]. Unless explicitly stated, all experiments in this paper utilize the same hyperparameters in the training process for fair comparison. We follow a similar image augmentation procedure as [20, 31], with random horizontal flipping, followed by random color jittering, and random cropping from scale $[0.7, 1.3]$. We utilize a minibatch of 10 images, trained for 200 epochs, or equivalent to around 7400 steps. Similar to [6], we use RMSProp with an initial learning rate of 0.001 with a polynomial learning rate with power 0.9. We use a weight decay of 0.0001 to help with regularization, and unless stated otherwise, we use dropout layers after each stage, with dropout probability linearly decaying from 0.1 to 0 from the deepest to the shallowest layers on both encoders and decoders. Finally, we use cross-entropy loss with no class balancing.

4.1 Efficient Designs

4.1.1 Skip Connections and Shallow Deconvolution Layers

We first focus on *efficient designs* of the network to establish a lightweight baseline for building Bayesian networks. We build upon the inverted residual decoder block described in Sec. 3.1. We experiment with shallow decoders, where only one deconvolutional block is used in the decoder at each stage. This is motivated by the hypothesis that most of the feature processing can be achieved by the encoder, and the decoding process can be low-dimensional where reasonable results can be achieved by bilinear upsampling [31]. Using shallow decoders proved to be an efficient way to reduce parameters without hurting performance, decreasing both parameter count and increasing inference time drastically in our experiments, with only minor mIoU loss, which is then fully recovered by modifying skip connection to be dense. Qualitative examples of skip connection experiments are shown in fig 3, where we empirically validated that adding skip connections between blocks that upsample features strikes the right balance between computational costs and preserving feature resolution. Additionally, The cumulative effect of each modification is included in Table 1, and for more details of each individual experiment, please refer to Section 4.1.1 our Suppl. Material.

Modification	mIoU ↑	Params ↓	Time (ms) ↓	GFLOPS ↓
Baseline	63.91	3.34M	**8.03**	2.26
+ invRes-dec	66.61	3.67M	11.47	2.24
+ shallow-dec	66.47	**2.59M**	8.99	**1.68**
+ dense-skip	**66.66**	2.70M	9.13	1.70

Table 1: Cumulative effects of different modifications made to the baseline network with symmetrical encoder-decoder structure. Notably, by combining all three techniques of using inverted residual decoders, shallow decoders and dense skip connections, we achieve higher mIoU with fewer parameters and GFLOPs, with only slightly worse inference time.

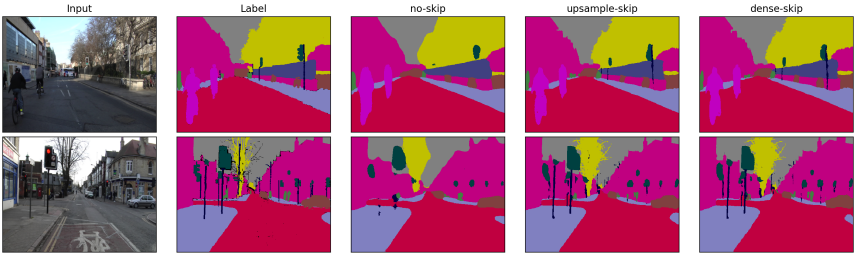

Figure 3: Qualitative results from skip connection experiments, note that increasing skip connections produces denser predictions, which helps with classes with thinner segments.

4.1.2 Network scaling

We further examine the effect of scaling the width (number of feature channels) and depth (number of blocks for each stage) of the network. We propose that for roughly the same number of parameter and inference time increases, scaling the network width improves the performance more than scaling the depth. From this observation, we present three scaled versions of our network—*lite*, *medium*, and *large*—all featuring a shallow decoder and dense skip connections. Detailed scaling experiments are included in Section 4.1.2 of our Suppl. Material, while we present our main results in Table 2. The *lite* network scales width and depth at 0.5, the *medium* at 1.0, and the *large* at 2.0 width and 1.0 depth. All networks were trained for longer using a learning rate with exponential decay. The *lite* network achieves higher mIoU than other half-resolution methods with fewer parameters. Meanwhile, the *medium* network surpasses FCDenseNet and BiSeNet in mIoU with fewer parameters. Since top-performing methods like DDRNet and RTFormer utilize pretraining and full-resolution images, it's difficult for us to fairly compare these methods to ours, and we leave it to future work to further investigate the performance of our network with those settings.

Method	Resolution	Pretrain	Encoder	Params (M)	mIoU
FCN8 [31]	half	ImageNet	VGG	134.5	57.0
DeconvNet [35]	half	ImageNet	VGG	252	48.9
SegNet [1]	half	-	-	29.5	46.4
ENet [37]	half	-	-	0.37	51.3
FCDenseNet56 [25]	half + full	-	DenseNet	1.5	58.9
BiSeNet [48]	full	ImageNet	Xception	5.8	65.6
DDRNet23 [20]	full	ImageNet	-	20.1	76.3
RTFormer [42]	full	ImageNet	ViT	16.8	82.5
Lite [ours]	half	-	MobileNetV2	0.56	66.4
Medium [ours]	half	-	MobileNetV2	2.70	71.6
Large [ours]	half	-	MobileNetV2	10.52	73.9

Table 2: Comparison to state-of-the-art methods trained and tested on half and full resolution. 'half+full' denotes trained on half resolution followed by fine-tuning on full resolution.

4.2 Bayesian Inference

4.2.1 Network Performance and Calibration

We evaluate three variants of Bayesian neural networks: MC dropout, stochastic depth, and a combination of both. The modifications are made upon the baseline with inverted residual decoder, by adding dropout layers after activation functions and stochastic depth to blocks with residual connections. Motivated by analyses from [22, 26], we decrease dropout and stochastic depth probabilities linearly from deeper to shallower layers from $[p, 0]$. We experiment with dropout and stochastic depth probabilities $p \in \{0.1, 0.3, 0.5\}$. We show the complete results of all experiments in Section 4.2.1 of Suppl. Material, and the runs with the highest performance are shown from each variation in Table 3. We show that all Bayesian methods reduce calibration error, with the combined method reducing the most. However, our results validate the finding that MC dropout sometimes worsens performance possibly due to underfitting, especially in per-class metrics Acc(c) and mIoU, respectively scoring 3.68 and 2.56 lower than the deterministic version. On the other hand, we show that stochastic depth variants do not exhibit a decrease in performance. In fact, all performance metrics are higher in stochastic depth variants compared to the deterministic network, and it still reduces ECE by more than 54%. Finally, we note that although the lowest calibration can be achieved by combining both regularization methods, it comes at the cost of significant underfitting and is thus undesirable.

Method	Acc(g) ↑	Acc(c) ↑	mIoU ↑	ECE ↓	MCE ↓
Deterministic	**92.08**	75.42	66.90	2.71	1.61
Dropout	91.87	71.74	64.34	0.70	0.16
Sd	**92.08**	**75.48**	**66.97**	1.23	0.58
Combined	91.73	70.91	63.30	**0.27**	**0.12**

Table 3: Accuracy, mIoU, and calibration results with Bayesian approximation, only results with the best-performing parameters are shown. Bayesian variants results have T=10.

4.2.2 Out-of-distribution Data

We showcase the qualitative results of using our Bayesian network (sd) on the RUGD [46], which consists of forest images notably distinct from our training data. Our Bayesian network displays high uncertainty scores, especially epistemic, on unfamiliar terrains shown in Fig. 4. The qualitative observations show that the network can robustly identify regions that are most dissimilar to the training set by predicting high epistemic uncertainties in those regions. We also observe that the network produces high confidence in regions where its predictions are indeed correct (i.e. trees, sky, and road).

5 Conclusions

This paper addresses two critical challenges: enhancing network efficiency and reliability. We optimize the standard encoder-decoder architecture using inverted residual blocks in the decoder. We further build upon this network and propose simple yet effective ways of improving the efficiency of segmentation networks through a combination of shallow decoder, dense skip connections, and prioritising scaling width over scaling depth. We proposed a

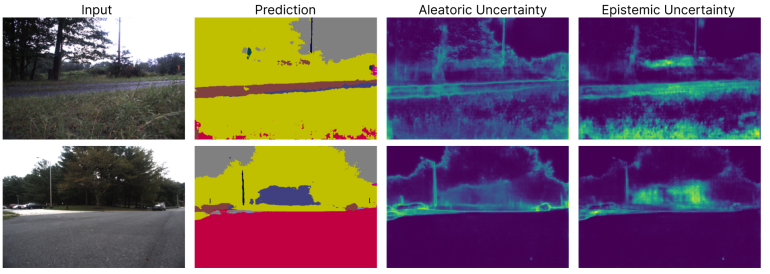

Figure 4: Qualitative results of predictions on OOD dataset RUGD. The network outputs low uncertainty on correctly identified regions, e.g. sky (grey), trees and plantation (yellow), and road (red), while predicting high epistemic uncertainty in regions that are OOD compared to CamVid, e.g. grass close to the camera in the first row, and dark regions in the second row.

novel approach of using stochastic depth in such encoder-decoder networks to build BNNs that can output calibrated uncertainties. The presented approach significantly reduces calibration error by over 50% whilst maintaining the same performance. We will direct our future research toward developing the theoretical properties of Bayesian variational inference using stochastic depth and processing stochastic forward passes more efficiently.

Acknowledgment

This work was supported by the UKRI FLF [MR/V025333/1] (RoboHike) and CDT of FAI [EP/S021566/1]. For the purpose of Open Access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[2] James O Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

[3] Piotr Bilinski and Victor Prisacariu. Dense decoder shortcut connections for single-pass semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6596–6605, 2018.

[4] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

[5] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2): 88–97, 2009.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[9] Loic Le Folgoc, Vasileios Baltatzis, Sujal Desai, Anand Devaraj, Sam Ellis, Octavio E Martinez Manzanera, Arjun Nair, Huaqi Qiu, Julia Schnabel, and Ben Glocker. Is mc dropout bayesian? *arXiv preprint arXiv:2110.04286*, 2021.

[10] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[11] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International conference on machine learning*, pages 1183–1192. PMLR, 2017.

[12] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, pages 1–77, 2023.

[13] Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

[14] Roger Grosse and James Martens. A kronecker-factored approximate fisher matrix for convolution layers. In *International Conference on Machine Learning*, pages 573–582. PMLR, 2016.

[15] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[16] Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schon. Evaluating scalable bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 318–319, 2020.

[17] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Christopher J Holder and Muhammad Shafique. On efficient real-time semantic segmentation: A survey. *arXiv preprint arXiv:2206.08605*, 2022.

[20] Yuanduo Hong, Huihui Pan, Weichao Sun, and Yisong Jia. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. *arXiv preprint arXiv:2101.06085*, 2021.

[21] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[22] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 646–661. Springer, 2016.

[23] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[24] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. *arXiv preprint arXiv:1405.3866*, 2014.

[25] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua 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*, pages 11–19, 2017.

[26] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[27] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems*, 28, 2015.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[29] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[30] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

[31] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[32] Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient mcmc. *Advances in neural information processing systems*, 28, 2015.

[33] Nikita Moshkov, Botond Mathe, Attila Kertesz-Farkas, Reka Hollandi, and Peter Horvath. Test-time augmentation for deep learning-based cell segmentation on microscopy images. *Scientific reports*, 10(1):5068, 2020.

[34] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

[35] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.

[36] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

[37] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[39] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29, 2016.

[40] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[41] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[42] Francesco Verdoja and Ville Kyrki. Notes on the behavior of mc dropout. *arXiv preprint arXiv:2008.02627*, 2020.

[43] Guotai Wang, Wenqi Li, Sébastien Ourselin, and Tom Vercauteren. Automatic brain tumor segmentation using convolutional neural networks with test-time augmentation. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part II 4*, pages 61–72. Springer, 2019.

[44] Jian Wang, Chenhui Gou, Qiman Wu, Haocheng Feng, Junyu Han, Errui Ding, and Jingdong Wang. Rtformer: Efficient design for real-time semantic segmentation with transformer. *Advances in Neural Information Processing Systems*, 35:7423–7436, 2022.

[45] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011.

[46] Maggie Wigness, Sungmin Eum, John G Rogers, David Han, and Heesung Kwon. A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments. In *International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[47] Linghong Yao, Dimitrios Kanoulas, Ze Ji, and Yuanchang Liu. Shorelinenet: An efficient deep learning approach for shoreline semantic segmentation for unmanned surface vehicles. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5403–5409. IEEE, 2021.

[48] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018.

[49] Lequan Yu, Shujun Wang, Xiaomeng Li, Chi-Wing Fu, and Pheng-Ann Heng. Uncertainty-aware self-ensembling model for semi-supervised 3d left atrium segmentation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part II 22*, pages 605–613. Springer, 2019.

[50] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.