# Efficient Uncertainty Estimation in Semantic Segmentation via Distillation

Christopher J. Holder Muhammad Shafique
New York University Abu Dhabi
Abu Dhabi, United Arab Emirates
{chris.holder, ms12713}@nyu.edu

## Abstract

*Deep neural networks typically make predictions with little regard for the probability that a prediction might be incorrect. Attempts to address this often involve input data undergoing multiple forward passes, either of multiple models or of multiple configurations of a single model, and consensus among outputs is used as a measure of confidence. This can be computationally expensive, as the time taken to process a single input sample increases linearly with the number of output samples being generated, an important consideration in real-time scenarios such as autonomous driving, and so we propose Uncertainty Distillation as a more efficient method for quantifying prediction uncertainty. Inspired by the concept of Knowledge Distillation, whereby the performance of a compact model is improved by training it to mimic the outputs of a larger model, we train a compact model to mimic the output distribution of a large ensemble of models, such that for each output there is a prediction and a predicted level of uncertainty for that prediction. We apply Uncertainty Distillation in the context of a semantic segmentation task for autonomous vehicle scene understanding and demonstrate a capability to reliably predict pixelwise uncertainty over the resultant class probability map. We also show that the aggregate pixel uncertainty across an image can be used as a metric for reliable detection of out-of-distribution data.*

## 1. Introduction

Deep neural networks have come to dominate the field of machine learning, surpassing prior approaches in a multitude of tasks across domains including natural language processing and computer vision. Despite their unrivalled performance in many of these tasks, neural networks typically suffer from a lack of interpretability, posing major challenges when it comes to analysing what exactly a model has learned, how a given input maps to a subsequent output or assigning any meaningful measure of confidence to outputs. This is an especially poignant problem when neural networks are deployed in safety critical systems, such as autonomous vehicles, where undetected errors have the potential to result in loss of life.

Many proposed solutions to the problem of quantifying the uncertainty of model outputs require the computation of a distribution of outputs. Some methods by which this distribution can be computed include Bayesian neural networks [1], whereby model parameters are learned as distributions rather than fixed points, Monte Carlo Dropout [2], in which model parameters are randomly set to zero during each inference pass, Variational Autoencoders [3], wherein a sample's latent representation is encoded as a distribution rather than a point within the feature space, and Ensembles [4], which comprise multiple models each trained to perform the same task but with slightly different parameters due to the stochasticity of the training process. In each of these cases, sampling from a stochastic process provides a distribution of outputs from a single input sample, and the parameters of this distribution can be used to compute some measure of uncertainty, however this sampling process is computationally expensive, due to many forward passes being required to capture an adequate sample size.

In this work we present a novel approach for the efficient computation of uncertainty, which we term 'Uncertainty Distillation'. It has been shown that through the paradigm of Knowledge Distillation, a compact student model can learn to produce outputs that closely match those of a larger teacher model, boosting its performance beyond that achieved with conventional supervised learning. In the case of Uncertainty Distillation, the teacher is an ensemble and the student learns to output a distribution that closely matches that of the outputs produced by the models that comprise the teacher ensemble. In the context of semantic segmentation for autonomous driving, we demonstrate that our approach results in a compact model capable of reliably predicting the uncertainty of its own predictions.

This work makes the following contributions:

1. A novel method for the distillation of the uncertainty quantification capability of deep ensemble networks into a single compact model.

2. Generation of pixelwise uncertainty maps in the context of semantic segmentation from a single pass of a single model.

3. Robust detection of out-of-distribution (OOD)

samples via aggregation of uncertainty values across whole images.

## 2. Related Work

### 2.1. Uncertainty Quantification

Most neural networks addressing classification problems, including semantic segmentation, use the softmax function to give a set of output probabilities for each possible class, however it has been shown [5] that these probabilities do not reliably capture any meaningful measure of confidence or uncertainty in a model's predictions. There is a significant body of work attempting to address this problem of quantifying the uncertainty of predictions in a variety of ways.

Much prior work towards quantifying the uncertainty of neural network outputs relies on computationally expensive sampling processes. Bayesian neural networks [1] learn a distribution over their parameters, which can be sampled from at inference time to create potentially infinite ensembles. Several challenges exist in training such models, particularly scalability and selection of a suitable prior, and there have been many works attempting to address these or to otherwise approximate a true Bayesian neural network. In [6], the concept of Bayes by Backprop is introduced, in which parameter uncertainty is computed during gradient updates. Probabilistic backpropagation is proposed in [7], with the posterior over model weights approximated using a product of Gaussians. A Bayesian neural network can be approximated using Monte Carlo Dropout [2], in which a randomly selected set of model parameters are ignored on each pass creating a huge number of potential model permutations from which to sample. Bayesian SegNet [8] applies this approach to semantic segmentation, generating pixelwise uncertainty values that can subsequently be used to improve segmentation accuracy. Another method for making a neural network non-deterministic is that of the Variational Autoencoder [9], which learns to encode an input sample as a distribution in latent feature space, which can subsequently be infinitely sampled from to generate a distribution of outputs. Ensemble models can be considered an approximation of a Bayesian neural network, with the distribution of component model outputs used to compute uncertainty, as in [4]. In all these cases, uncertainty is derived from the distribution of outputs over multiple passes of the same input sample, with the level of consensus achieved across outputs taken as a measure of confidence. This is computationally expensive, and so we propose shifting this sampling process from inference to training time, resulting in a single efficient deterministic model capable of predicting the uncertainty of its own predictions in a single pass.

Work towards quantifying uncertainty without sampling include [10] and [11], in which a model learns to predict the parameters of a Dirichlet distribution over outputs, however training such models is challenging and requires a well-defined prior. In [12] a model learns to represent regression outputs in the form of a normal inverse-gamma distribution, from which aleatoric and epistemic uncertainty can be modelled, although the technique is not applicable to classification problems. In [13] a variational Dirichlet framework is proposed, with entropy of the learned posterior used to quantify uncertainty, however OOD training data is required.

*In this work, we address the main limitation of sampling-based techniques, that is their computational expense, by distilling the uncertainty quantification capability of an ensemble into a single model*. Our approach is relatively straightforward to train, does not require OOD training data, and has the potential to be adapted for any application of a neural network.

### 2.2. Knowledge Distillation

The concept of compressing the knowledge of a large ensemble into a single compact neural network was first proposed in [14], taking the logits of a teacher model as targets for a student. Building on this work, [15] presented the approach now commonly known as Knowledge Distillation, which replaces the logit targets with high temperature softmax targets. In scenarios where the computational requirements of a large model are not available, knowledge distillation has demonstrated better results than those of a small model trained using standard ground truth labels.

Later work has built upon this idea via training the student to also match intermediate feature maps [16] or attention maps [17] to those of the teacher, or by applying adversarial learning [18], whereby a discriminator network learns to classify outputs as those of the teacher or student, and the student aims to produce outputs indistinguishable from those of its teacher.

Several works have applied knowledge distillation in the context of semantic segmentation: In [19] adversarial loss is used so that the student learns to output segmentation maps that match those of the teacher; In [20] the student is trained to match the teacher's latent representation; In [21] the teacher is trained to predict depth as well as segmentation, with the resulting depth-aware embedding used to train the student for segmentation alone.

In this work we build on the concept of knowledge distillation to transfer not just predictions from teacher to student, but the distributions of those predictions over a teacher ensemble so that the student learns to quantify the uncertainty of its own predictions.

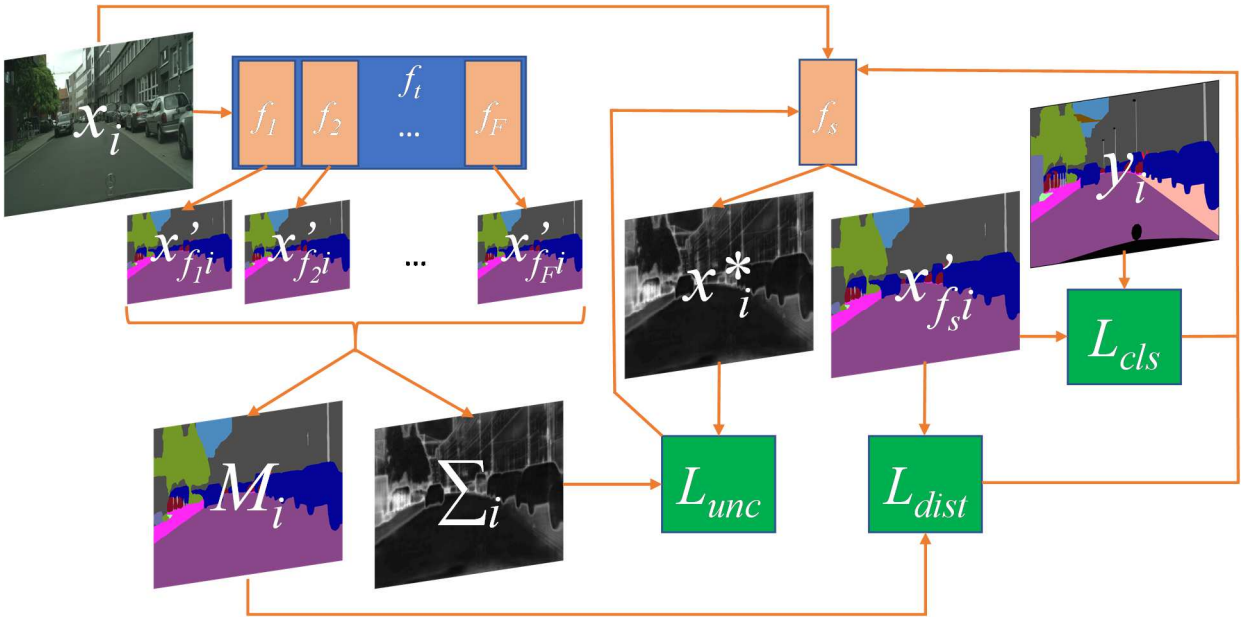


Figure 1. The process for training the student model in Uncertainty Distillation. For input $x_i$, ensemble models $f_1$ *to* $f_F$ comprising teacher $f_t$ output probability maps $x'_{f1i}$ to $x'_{fFi}$, the elementwise mean and standard deviation of which populate $M_i$ and $\sum_i$ respectively. Student model $f_s$ outputs class probability map $x'_{fsi}$, which is compared with ground truth segmentation $y_i$ to compute classification loss $L_{cls}$ and to $M_i$ to compute distillation loss $L_{dist}$, and uncertainty map $x^*_i$, which is compared to $\sum_i$ to compute uncertainty loss $L_{unc}$. These losses are then combined to optimise $f_s$ such that it learns to approximate the output distribution of $f_t$.

## 3. Uncertainty Distillation

In a typical supervised learning scenario, we aim to learn a function $f$ that minimises some loss function $L(f(x_i),y_i)$ over a dataset $D$ comprising $N$ input samples $x_i$ each paired with a ground truth target $y_i$:

$$\min_f \frac{\sum_{i=1}^{N} L(f(x_i),y_i)}{N} \quad (1)$$

In the case of a convolutional neural network (CNN) performing semantic segmentation of images, $x_i$ is an RGB input image of dimensions $3 \times H \times W$, $f$ is a CNN parameterised by a set of weights $w$ that outputs a class probability map $x'_i$ of dimensions $C \times H \times W$, $C$ being the number of classes present in $D$, $y_i$ is the ground truth segmentation map pertaining to $x_i$, and $L$ is some measure of similarity between $x'_i$ and $y_i$, commonly cross entropy. At each training step, the gradient of $L(x'_i, y_i)$ is computed with respect to $w$, which are adjusted accordingly by some learning scheme.

When Knowledge Distillation is applied, a teacher model $f_t$ is trained via the method described above, and the outputs from this model are used to train a student model $f_s$. The softmax function is applied to the final layer of $f_t$ so that outputs $x'$ are within the range (0, 1), with a single tunable parameter, temperature $T$, determining the smoothness of the resulting distribution:

$$x'_{i(c,v,u)} = \frac{\exp(\tilde{x}_{i(c,v,u)}/T)}{\sum_{j=1}^{C} \exp(\tilde{x}_{i(j,v,u)}/T)} \quad (2)$$

Where $c \in \{1,\ldots C\}$, $v \in \{1,\ldots H\}$ and $u \in \{1,\ldots W\}$ denote the location of an element within our class probability map, and $\tilde{x}_i$ is the logits class probability map at the final layer before the softmax function is applied.

Subsequently $f_s$, also with a softmax function at its final layer, is trained to minimise the difference between its own outputs and those of $f_t$, $L_{dist}(x'_{f_s i}, x'_{f_t i})$, over each sample $x_i$ in dataset $D$. As this is no longer a pure classification problem, distillation loss $L_{dist}$ can be some distance measure such as mean squared error or Kullback-Leibler (KL) divergence [22].

An additional classification loss function, $L_{cls}(x'_{f_s i}, y_i)$, may be used to compare $f_s$ outputs with ground truth labels, usually with a lesser weighting than $L_{dist}$.

We build upon this with our proposed concept of Uncertainty Distillation, the training process for which is described in Figure 1. Teacher $f_t$ is an ensemble comprising $F$ trained models. For a given input $x_i$, $x'_{f_j i}$ is computed for each model $f_j$, and the mean and standard deviation across the outputs of all models are calculated at each element in the class probability map to generate mean map $M_i$ and standard deviation map $\Sigma_i$, each of dimensions $C \times H \times W$:

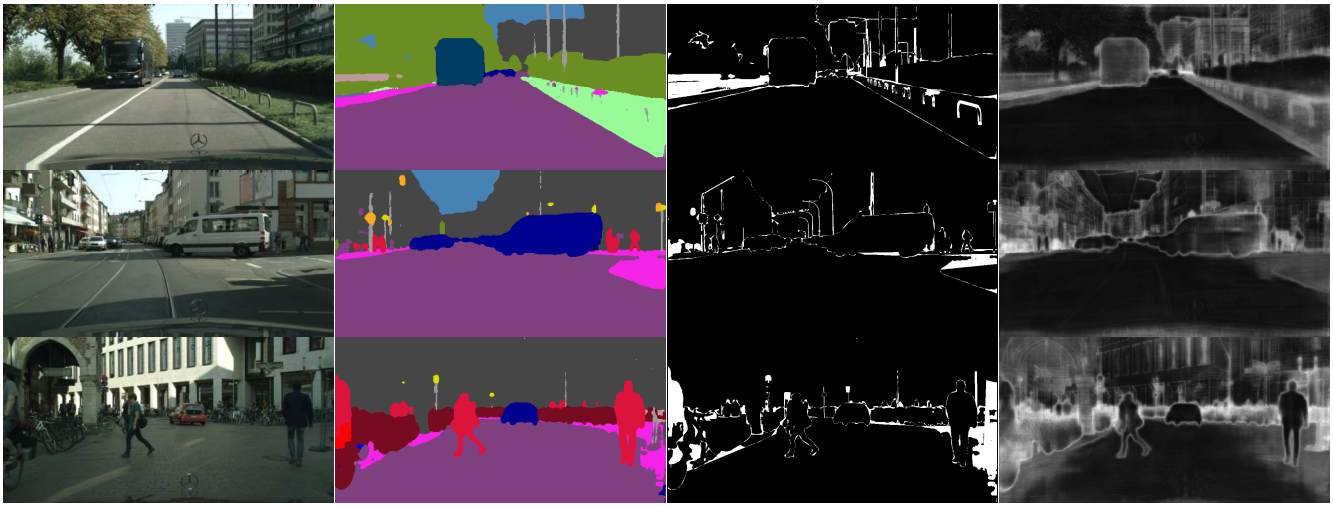

Figure 2. Example Images from the Cityscapes test set with corresponding (left to right) segmentation output of our trained student model, binary accuracy map (white pixels are incorrectly labelled), uncertainty map output by our student model.

$$M_{i(c,v,u)} = \frac{\sum_{j=1}^{F} x'_{f_j i(c,v,u)}}{F}, \qquad (3)$$

$$\boldsymbol{\Sigma}_{i(c,v,u)} = \sqrt{\frac{\sum_{j=1}^{F}(x'_{f_j i(c,v,u)} - M_{i(c,v,u)})^2}{F}}$$

Student model $f_s$ is modified such that the output layer is replaced by a prediction head and an uncertainty head, which respectively output prediction map $x'_{f_s i}$ and uncertainty map $x^*_i$, for each input image $x_i$. Each head comprises a 3 × 3 conv-BN-ReLU block with 256 output channels, followed by a 1 × 1 convolution with $C$ output channels, and a sigmoid function bounds uncertainty outputs between 0 and 1, while the softmax function (2) is applied to prediction outputs. Both outputs are of dimensions $C \times H \times W$, giving a predicted mean and standard deviation describing a distribution for each class at each pixel.

Three loss functions are combined to optimise $f_s$ during training: Distillation loss $L_{Dist}$, Classification loss $L_{cls}$, and uncertainty loss $L_{unc}$, with their respective contributions weighted by parameters α and β, as shown in equation 4. We determine α and β empirically, with values of 0.1 and 1.1 found to give the best results in our experiments.

$$L_{Total} = T^2 L_{dist} + \alpha L_{cls} + \beta L_{unc} \qquad (4)$$

$L_{dist}$ for input sample $x_i$ is the KL divergence between prediction head output $x'_{f_s i}$ and mean map $M_i$:

$$KL(X,Y) = \frac{\sum_{j=1}^{C\times H\times W} Y_j \cdot (\log Y_j - \log X_j)}{N}, \qquad (5)$$

$$L_{dist} = KL\left(x'_{f_s i}, M_i\right)$$

$L_{cls}$ is the cross-entropy between prediction head output $x'_{f_s i}$ and ground truth segmentation labels $y_i$:

$$CE(p,c) = -\log\left(\frac{\exp(p[c])}{\sum_j^C \exp(p[j])}\right), \qquad (6)$$

$$L_{cls} = \frac{\sum_v^H \sum_u^W CE(x'_{f_s i(v,u)}, y_{i(v,u)})}{H \times W}$$

Where $p$ is an array of $C$ values denoting the output probability of each class being present at pixel ($v$, $u$), and $c$ is the ground truth label at pixel ($v$, $u$). $L_{unc}$ is computed as the mean squared error between the uncertainty head output $x^*_i$ and standard deviation map $\Sigma_i$:

$$L_{unc} = \frac{\sum_{v=1}^{H}\sum_{u=1}^{W}\sum_{c=1}^{C}(x^*_{i(c,v,u)} - \Sigma_{i(c,v,u)})^2}{C\times H\times W} \qquad (7)$$

## 4. Experimental Setup

For our experiments, we use a Deeplab v3+ [23], a widely used semantic segmentation model that has demonstrated state of the art results on many datasets, with a mobilenet [24] backbone, chosen for its compactness and suitability for real time applications. We use the Cityscapes dataset [25], a common benchmark for autonomous vehicle scene understanding tasks, which consists of 2975 training images and 500 validation images which we use for testing as the ground truth labels of the official test set are not available. Each RGB image has dimensions of 2048 × 1024 and each pixel is assigned one of 19 class labels in the corresponding ground truth. Ideally we would have 2 separate training sets – one to train $f_t$, and a second unseen set from which to extract the outputs of $f_t$ for training $f_s$ – however due to the small number of samples in the Cityscapes dataset we perform both steps with the same

training dataset.

Teacher ensemble $f_t$ can be an infinite virtual ensemble generated by sampling from a single Bayesian model, however in our implementation $f_t$ is an ensemble of 25 identically initialised Deeplab models with variance introduced via the stochasticity of the training process. This configuration was chosen due to several unsolved challenges in the training of Bayesian neural networks and for the deterministic nature of such an ensemble once training is completed. In training our ensemble, we use different learning rates, optimisation algorithms and data augmentation techniques, to increase the variability between individual models.

Our student model $f_s$ is a single Deeplab model, identical to those used in $f_t$, and is pretrained to convergence with ground truth labels and a standard output layer before this is replaced with our prediction and uncertainty heads and training continues via Uncertainty Distillation. For comparison, we take a single model from our ensemble to act as a baseline for segmentation performance.

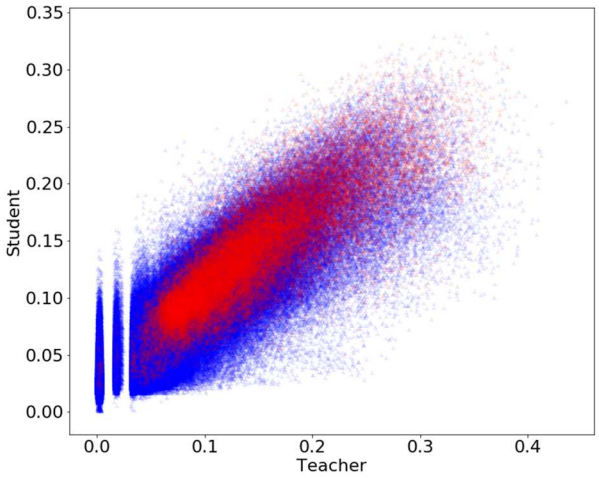


Figure 3. Per Pixel standard deviation predicted by our trained student model compared with that of the teacher ensemble. Red points denote pixels that are incorrectly labelled by the student, blue correctly labelled.

# 5. Results

## 5.1. Semantic Segmentation Performance

Table 1 lists the mean intersection over union (IoU) for each class across the test dataset for a standard Deeplab model (baseline), our ensemble of 25 such models (teacher), and a model that has been trained via Uncertainty Distillation (student), as well as the mean uncertainty of pixels predicted to belong each class as predicted by the teacher and the student. Overall, these results demonstrate that while the student does not match the teacher in segmentation performance, it does surpass the baseline, as we would expect in a knowledge distillation scenario. In particular, the distillation process appears to improve results for some of the harder classes such as Motorbike and Traffic Light. Predicted uncertainty correlates well with performance across classes, with harder classes demonstrating greater uncertainty, from both teacher and student.

## 5.2. Pixelwise Uncertainty Quantification

Figure 2 displays example images from the test dataset alongside corresponding predicted segmentation, binary accuracy map of incorrect pixels within predicted segmentation, and predicted uncertainty. The uncertainty map $U$ is computed as the mean predicted uncertainty across all classes at each pixel:

$$U_{i(v,u)} = \frac{\sum_{c=1}^{C} x^*_{i(c,v,u)}}{C} \quad (8)$$

We can see that those pixels assigned an incorrect label (white in the binary accuracy map) generally have a high predicted uncertainty.

Figure 3 plots pixelwise uncertainty predicted by our student model (y axis) against that computed from the outputs of our teacher ensemble (x axis) across the test set.

| | *Road* | *Sidewalk* | *Building* | *Wall* | *Fence* | *Pole* | *Tr. Light* | *Tr. Sign* | *Vegetation* | *Terrain* | *Sky* | *Person* | *Rider* | *Car* | *Truck* | *Bus* | *Train* | *Motorbike* | *Bicycle* | *Mean* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline IoU | 0.976 | 0.814 | 0.901 | 0.415 | 0.544 | 0.55 | 0.482 | 0.698 | 0.904 | 0.569 | 0.927 | 0.739 | 0.44 | 0.927 | 0.654 | 0.686 | 0.622 | 0.289 | 0.706 | 0.676 |
| Teacher IoU | 0.961 | 0.794 | 0.914 | 0.432 | 0.563 | 0.584 | 0.62 | 0.73 | 0.917 | 0.596 | 0.937 | 0.782 | 0.553 | 0.935 | 0.668 | 0.793 | 0.677 | 0.534 | 0.743 | 0.723 |
| Student IoU | 0.964 | 0.772 | 0.9 | 0.426 | 0.547 | 0.476 | 0.511 | 0.652 | 0.904 | 0.564 | 0.919 | 0.739 | 0.498 | 0.921 | 0.617 | 0.723 | 0.625 | 0.493 | 0.689 | 0.681 |
| Teacher Uncertainty | 0.029 | 0.097 | 0.055 | 0.21 | 0.147 | 0.1 | 0.128 | 0.108 | 0.028 | 0.129 | 0.03 | 0.068 | 0.125 | 0.03 | 0.176 | 0.155 | 0.257 | 0.165 | 0.082 | 0.111 |
| Student Uncertainty | 0.032 | 0.086 | 0.077 | 0.155 | 0.141 | 0.133 | 0.135 | 0.111 | 0.055 | 0.127 | 0.046 | 0.097 | 0.127 | 0.046 | 0.108 | 0.100 | 0.127 | 0.135 | 0.130 | 0.104 |

Table 1. Per-class results - mean intersection over union and mean predicted uncertainty. Red denotes lower IoU, higher uncertainty

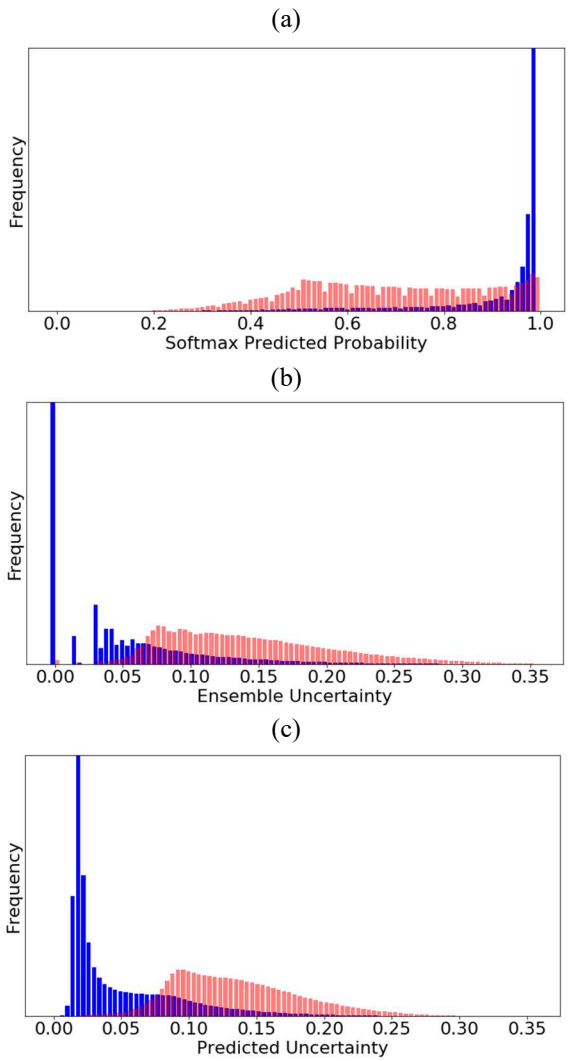


Figure 4. Histograms displaying the frequency of output uncertainty values for correctly (blue) and incorrectly (red) labelled pixels. (a) shows softmax predicted probabilities of the baseline model, (b) plots uncertainty of teacher ensemble outputs, (c) plots uncertainty predictions of our student model.

Each blue point represents a pixel that was assigned the correct class label by the student, and each red point represents a pixel that was assigned an incorrect label. We can see a clear correlation between prediction and target, demonstrating that the student has learned to approximate the uncertainty in the predictions of the ensemble. It can also be observed that the ratio of red to blue points increases as both target and predicted value increase, showing that standard deviation of ensemble predictions can be a useful predictor of output uncertainty.

Figure 4 plots normalised histograms of uncertainty values computed for test set pixels that are correctly (blue) and incorrectly labelled (red), taken from ensemble outputs (b) and student model outputs (c). For comparison, (a) plots softmax class probability from the outputs of the baseline model. Student predicted uncertainty demonstrates a similar distribution to the ensemble uncertainty used to train it. While a minority of correctly labelled pixels are assigned relatively high uncertainty, the distribution of incorrectly labelled pixels tails off quite steeply towards 0, meaning that very few incorrectly labelled pixels are assigned an uncertainty of below 0.05. With softmax probabilities, the correctly labelled histogram is a long-tail distribution with most pixels near to 1, however the incorrectly labelled histogram appears almost to resemble a uniform distribution, with a very slight peak also near to 1, suggesting softmax is a poor predictor of uncertainty. This demonstrates that our measure of uncertainty is significantly more meaningful than softmax predicted probability.

### 5.3. Out of Distribution Detection

We investigate the capability of a model trained via Uncertainty Distillation to detect out of distribution samples, that is data that is outside of the distribution of a models training data. To do this, we test our model, trained only using the standard Cityscapes training set, using three further datasets: Cityscapes Foggy [27] and Cityscapes Rain [28], which modify the standard Cityscapes images with synthetic weather effects, and the Audi Autonomous Driving Dataset (A2D2) [29], which shares a similar labelling scheme to Cityscapes but features different scenarios and visual properties.

Table 2 shows the mean class intersection over union of our student model's predictions on each of these datasets, demonstrating a significant drop in performance compared to the in-distribution Cityscapes test set. Figure 5 shows examples from each of these datasets. We can see that the binary accuracy maps show a significantly higher number of incorrectly labelled pixels in these OOD scenarios, and while the uncertainty maps appear to correlate less with pixel accuracy than they do for in-distribution data, overall uncertainty values are higher.

To detect out of distribution samples, we assign a single uncertainty value to the whole image with the goal of setting a threshold above which an image can be considered OOD. To assign an image uncertainty value to image $x_i$, we consider taking the minimum, mean, median and maximum across all pixel uncertainty values in $U_i$, with the ROC curve for each plotted in Figure 6. Of these metrics, median offers the best discriminative performance, while maximum

| Dataset | Mean IoU |
|---|---|
| Cityscapes | 0.681 |
| CS Foggy | 0.589 |
| CS rain | 0.348 |
| A2D2 | 0.344 |

Table 2. Segmentation performance over in- and out- of distribution datasets

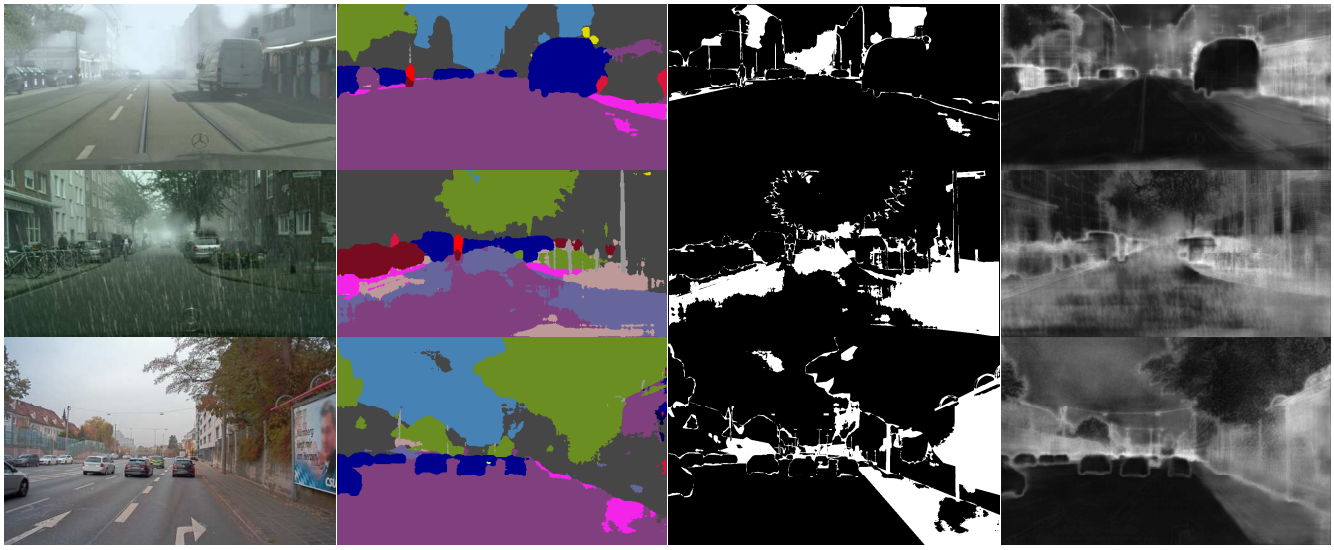

Figure 5. Example Images from (top to bottom) CS Foggy, CS Rain and A2D2 test sets, with corresponding (left to right) segmentation output of our trained student model, binary accuracy map (white pixels are incorrectly labelled), uncertainty map output by our student model.

gives the opposite of the expected result, with OOD images generally having a lower maximum pixel uncertainty than in-distribution images. We hypothesise that this may be because the model is less confident in its own predicted uncertainty when faced with an OOD image, and so is less likely to assign high uncertainty values to noisy or otherwise difficult image regions than when faced with an in-distribution image. We find that the formula $\frac{median+min}{max}$, in blue on the ROC curve, offers the best discriminative capability for detecting OOD samples.

Figure 7 plots the distribution of image uncertainty values, assigned via the above formula, for the four datasets tested. We can see that CS Rain and A2D2, both of which demonstrate a significant drop in performance, feature distributions far from that of the in-distribution data of Cityscapes, while CS Foggy, for which the performance drop was less severe, has more overlap with Cityscapes. Overall, this suggests that our image uncertainty value is a good metric for detecting OOD samples.

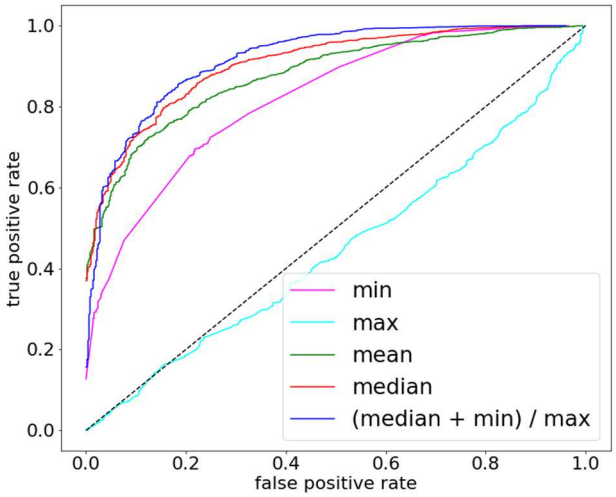


Figure 6. ROC curves comparing different statistical metrics for assigning an image uncertainty value. True positives are correctly identified OOD samples, false positives are in-distribution images incorrectly identified as OOD.

## 6. Discussion

In this work we have proposed a novel method for training a semantic segmentation model such that it can

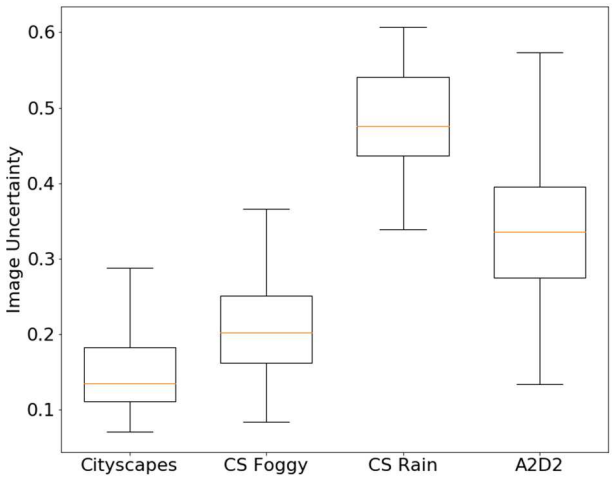


Figure 7. Image uncertainty values computed for all images in each dataset. Cityscapes is the only dataset considered to be in-distribution.

predict the uncertainty of its own predictions, via distillation from an ensemble model. Our approach demonstrates several advantages over existing work: The ensemble training process is relatively straightforward, albeit time consuming; Uncertainty predictions are made in a single pass with marginal additional computation required over the baseline model, in contrast to prior techniques that require computationally expensive sampling; And no out-of-distribution data is required during training.

We have demonstrated that predicted pixel level uncertainty values are a good predictor of incorrectly labelled pixels, and that an image level uncertainty value can be computed that is a robust discriminator for detecting out of distribution input samples.

In future work we plan to compare different teacher models, such as Bayesian Neural Networks from which an infinite ensemble can be generated, and explore the use of Uncertainty Distillation for other tasks such as depth estimation and object detection.

## References


[1] D. J. MacKay, "A practical bayesian framework for backpropagation networks," Neural Computation, vol. 4, no. 3, pp. 448-472, 1992.

[2] Y. Gal, and Z. Ghahramani, "Dropout as a bayesian approximation: representing model uncertainty in deep learning," ICML, pp. 1050-1059, 2016.

[3] J. Walker, C. Doersch, A. Gupta, and M. Hebert, "An uncertain future: forecasting from static images using variational autoencoders," ECCV, pp. 835-851, 2016.

[4] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," ArXiv:1612.01474, 2016.

[5] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," ArXiv:1806.01768, 2018.

[6] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," ICML, pp. 1613-1622, 2015.

[7] J. M. Hernández-Lobato, and R. Adams, "Probabilistic backpropagation for scalable learning of bayesian neural networks," ICML, pp. 1861-1869, 2015.

[8] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," ArXiv:1511.02680, 2015.

[9] D. P. Kingma, and M. Welling, "Auto-encoding variational bayes," ArXiv:1312.6114, 2013.

[10] T. Tsiligkaridis, "Information robust dirichlet networks for predictive uncertainty estimation," ArXiv:1910.04819, 2019.

[11] A. Amini, W. Schwarting, A. Soleimany, and D. Rus, "Deep evidential regression," ArXiv:1910.02600, 2019.

[12] W. Chen, Y. Shen, H. Jin, and W. Wang, "A variational dirichlet framework for out-of-distribution detection," ArXiv:1811.07308, 2018.

[13] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," ACM Int. Conf. On Knowledge Discovery And Data Mining, pp. 535-541, 2006.

[14] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," ArXiv:1503.02531, 2015.

[15] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: hints for thin deep nets," ArXiv:1412.6550, 2014.

[16] N. Komodakis, and S. Zagoruyko, "Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer," ICLR, 2017.

[17] X. Wang, R. Zhang, Y. Sun, and J. Qi, "KDGAN: knowledge distillation with generative adversarial networks.," NeurIPS, pp. 783-794, 2018.

[18] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang, "Structured knowledge distillation for semantic segmentation," CVPR, pp. 2604-2613, 2019.

[19] T. He, C. Shen, Z. Tian, D. Gong, C. Sun, and Y. Yan, "Knowledge adaptation for efficient semantic segmentation," CVPR, pp. 578-587, 2019.

[20] J. Jiao, Y. Wei, Z. Jie, H. Shi, R. W. Lau, and T. S. Huang, "Geometry-aware distillation for indoor semantic segmentation," in Proc. Of The IEEE/CVF Conf. On Computer Vision And Pattern Recognition, pp. 2869-2878, 2019.

[21] S. Kullback, and R. A. Leibler, "On information and sufficiency," Annals Of Mathematical Statistics, vol. 22, no. 1, pp. 79-86, 1951.

[22] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," ECCV, pp. 801-818, 2018.

[23] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: efficient convolutional neural networks for mobile vision applications," ArXiv:1704.04861, 2017.

[24] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset," in CVPR Workshop On The Future Of Datasets In Vision, vol. 2, 2015.

[25] D. P. Kingma, and J. Ba, "Adam: a method for stochastic optimization," ArXiv:1412.6980, 2014.

[26] C. Sakaridis, D. Dai, and L. Van Gool, "Semantic foggy scene understanding with synthetic data," Int. Journal Of Computer Vision, vol. 126, no. 9, pp. 973-992, 2018.

[27] X. Hu, C. Fu, L. Zhu, and P. Heng, "Depth-attentional features for single-image rain removal," CVPR, pp. 8022-8031, 2019.

[28] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, and S. Dorn, "A2d2: audi autonomous driving dataset," ArXiv:2004.06320, 2020.