arXiv:1807.00502v1 [cs.CV] 2 Jul 2018

Leveraging Uncertainty Estimates for Predicting Segmentation Quality

Terrance DeVries
University of Guelph and Vector Institute
terrance@uoguelph.ca

Graham W. Taylor
University of Guelph and Vector Institute
Canadian Institute for Advanced Research
gwtaylor@uoguelph.ca

Abstract

The use of deep learning for medical imaging has seen tremendous growth in the research community. One reason for the slow uptake of these systems in the clinical setting is that they are complex, opaque and tend to fail silently. Outside of the medical imaging domain, the machine learning community has recently proposed several techniques for quantifying model uncertainty (i.e. a model knowing when it has failed). This is important in practical settings, as we can refer such cases to manual inspection or correction by humans. In this paper, we aim to bring these recent results on estimating uncertainty to bear on two important outputs in deep learning-based segmentation. The first is producing spatial uncertainty maps, from which a clinician can observe where and why a system thinks it is failing. The second is quantifying an image-level prediction of failure, which is useful for isolating specific cases and removing them from automated pipelines. We also show that reasoning about spatial uncertainty, the first output, is a useful intermediate representation for generating segmentation quality predictions, the second output. We propose a two-stage architecture for producing these measures of uncertainty, which can accommodate any deep learning-based medical segmentation pipeline.

1. Introduction

In recent years, the use of deep learning for medical imaging tasks has increased in prevalence, with these powerful algorithms being applied to a wide variety of medical imaging applications, from metastasis detection for breast cancer [19], to improving reconstruction for medical resonance imaging [22]. In some cases, deep learning has even matched or exceeded human performance, such as on the tasks of skin lesion classification [5] and identifying diabetic retinopathy [7].

Unfortunately, despite the recent successes reported in the literature, we have yet to see the widespread adoption of deep learning in clinical settings. One possible reason for this delay could be the lack of suitable uncertainty estimates [18]. Current neural network-based models are often incapable of indicating when their predictions may be faulty, and as a result they fail silently, without any indication that a mistake has been made. This behaviour is worrying for applications that rely on accurate uncertainty estimates for decision making, such as those that a medical professional might encounter when diagnosing a patient based on the results of a predictive model. Proper uncertainty estimates would allow those reviewing the predictions to act accordingly in order to prevent undesirable outcomes [1].

One area where this is a problem is the task of segmenting medical images. If the result of an automated segmentation is poor, we would like to refer the case to a qualified human for follow-up. Furthermore, given the spatial nature of the task, it would be useful to provide the human with a map of the model's uncertainty so that they can better understand where and why the model failed, and perhaps take the uncertainty estimates into consideration when manually correcting the segmentation.

In this work, we consider the specific task of segmenting skin lesions. However, we propose a framework that is general enough to support a variety of medical segmentation tasks. We propose learning spatial uncertainty maps for each segmentation, which can then be used to improve our prediction of the quality of the segmentation, and we demonstrate that this yields an improved performance over alternative techniques that use deep learning to predict segmentation quality. Based on our finding that per-pixel uncertainty is a useful intermediate representation for predicting image-level segmentation quality we compare several contemporary uncertainty estimation methods to assess their relative merits.

The key contribution of this work is unifying two pursuits that have to-date remained disparate: uncertainty estimation in deep neural networks and predicting image-level segmentation quality. Our method is easy to deploy in practice, as it is modular and agnostic to the specific deep-learning architecture or uncertainty estimation technique. Secondary contributions are the extension of Learning Con-

fidence Estimates [4] to pixel-level, rather than scalar output and the empirical finding that at least three recently-proposed methods for quantifying uncertainty can aid almost equally well in predicting segmentation quality.

2. Related Work

Our work attempts to unite two research areas that are of interest to the machine learning and computer vision community: uncertainty estimation and predicting segmentation quality. Here we provide a brief overview of relevant recent work in the respective areas.

2.1. Uncertainty Estimation

Uncertainty estimates are useful in the context of deployed machine learning systems as they have been shown to be capable of detecting when a neural network is likely to make an incorrect prediction, or when an input may be out-of-distribution.

Traditionally, much of the work done on uncertainty estimation techniques is inspired by Bayesian statistics. A classic example is the Bayesian Neural Network (BNN) [24], which attempts to learn a distribution over each of the network's weight parameters. Such a network would be able to produce a distribution over the output for any given input, thereby naturally producing uncertainty estimates. Unfortunately, Bayesian inference is computationally intractable for these models in practice, so much effort has been put into developing approximations of Bayesian neural networks that are easier to train.

Recent efforts in this area include Monte-Carlo Dropout [6], Multiplicative Normalizing Flows [20], and Stochastic Batch Normalization [2]. These methods have been shown to be capable of producing uncertainty estimates, although with varying degrees of success. The main disadvantage with these BNN approximations is that they require sampling in order to generate the output distributions. As such, uncertainty estimates are often time-consuming or resource-intensive to produce, often requiring 10 to 100 forward passes through a neural network in order to produce useful uncertainty estimates at inference time.

An alternative to BNNs is Deep Ensembles [16], which proposes a frequentist approach to the problem of uncertainty estimation by training many models and observing the variance in their predictions. However, this technique is still quite resource intensive, as it requires inference from multiple models in order to produce the uncertainty estimate.

A promising alternative to sampling-based methods is to instead have the neural network *learn* what its uncertainty should be for any given input, as demonstrated in [12] and [4]. These methods are more computationally efficient compared to other techniques, and thus better suited when computational resources are limited or when real-time inference is required.

2.2. Segmentation Quality Prediction

When applying uncertainty estimates to the task of semantic segmentation, a number of works have proposed ways to produce spatial *uncertainty maps*, which visualize a model's confidence in its predictions for each pixel in the image. In most cases, uncertain regions are likely to be misclassified, and the uncertainty maps allow one to see which parts of the image are likely to be problematic [10, 11, 12]. This feature gives the model some amount of interpretability, and provides the end user with more information with which they can decide whether the final segmentation is to be trusted or how it should be modified (e.g. in a human-in-the-loop setting).

However, if the semantic segmentation model is part of a larger automated pipeline, pixel-level uncertainty estimates are not as useful, as perfectly acceptable segmentations can still contain some uncertainty. In this case, it is more useful to create a model that can predict the quality of the segmentation at the whole-image level. Previous efforts have attempted to learn the quality of segmentations from hand-crafted image or segmentation features [15, 29], but these approaches are limited by the expressiveness of their respective hand-crafted features. They are also limited in their transferability across different medical imaging modalities.

Contemporary approaches have exploited the powerful feature learning capabilities of deep learning. Recently, adversarial training has been used to improve the performance of convolutional segmentation networks by having an auxiliary discriminator network predict the quality of the segmentation (i.e. whether or not the predicted segmentation is discernible from a ground truth segmentation) [21]. The segmentation network then uses this information to improve its predictions and produce more realistic looking segmentations. This technique has previously been demonstrated to improve segmentation quality in medical imaging tasks such as prostate cancer or brain MRI segmentation [14, 23]. Adversarial training works well to improve segmentation quality, but the quality estimation network has limited utility as the outputs don't have any human interpretable meaning associated with them beyond whether the segmentation looks realistic or not.

As a solution to the interpretability issue, methods have been proposed which attempt to predict segmentation quality in terms of metrics that are more meaningful to humans, such as Jaccard index or Dice coefficient. An example of this is QualityNet [9], which learns a direct mapping between a masked input image and its corresponding segmentation quality via a convolutional neural network (CNN). Another interesting approach is Reverse Classification Accuracy (RCA) [27], which evaluates segmentation quality by training a reverse classifier on a predicted segmentation

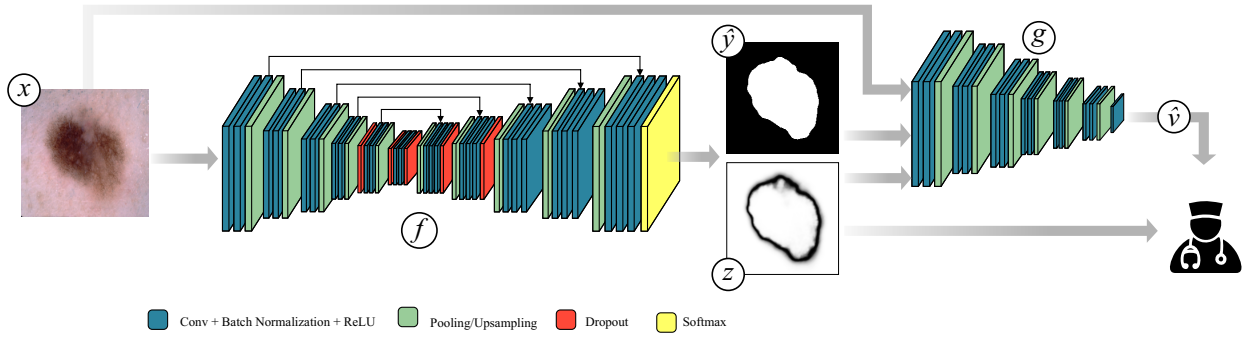


Figure 1: System diagram for our proposed pipeline. A semantic segmentation network f takes an input image x and produces a segmentation prediction $\hat{y}$ and an uncertainty map z. A segmentation quality network g then receives as input x, $\hat{y}$, and z to produce a quality estimate $\hat{v}$. The uncertainty map can be used to interpret the segmentation network's output, while the quality estimate can be used to automatically reject, or alert the user to poor segmentations. This diagram depict a system which utilizes a maximum softmax probability uncertainty map. Other uncertainty estimation methods may have small architectural differences.

from a new image, and then evaluating on a set of reference images that have ground truths available. While these techniques work well in their respective settings, none of them exploit uncertainty information, which could be used to improve the accuracy of the segmentation quality prediction.

3. Estimating Segmentation Quality with Uncertainty Information

In order to leverage uncertainty information in our predictions of segmentation quality, we first train a neural network-based semantic segmentation model f (as shown in Figure 1). The segmentation network takes in some image x, and produces two outputs: class prediction logits ρ and corresponding uncertainty (or confidence) estimates z:

$$\rho, z = f(x). \tag{1}$$

As uncertainty is estimated per pixel, we refer to z as an uncertainty map. In this formulation, z may either be calculated using the original outputs of f, or the network can produce the z directly. To obtain the final segmentation prediction $\hat{y}$ we take the argmax of the prediction logits or the class prediction probabilities:

$$\hat{y} = \text{argmax}(\rho). \tag{2}$$

A second network g is then trained to predict the quality of the segmentation $\hat{v}$, given the original input image x, as well as the predicted segmentation mask $\hat{y}$ and uncertainty map z from f:

$$\hat{v} = g(x, \hat{y}, z). \tag{3}$$

Under our framework, the segmentation quality measurement can be any segmentation-based evaluation metric, or even multiple metrics predicted simultaneously. To obtain the true segmentation quality labels v to train g, we evaluate the segmentation predictions from f using the ground truths from the training set. The training set for g can be the same one as used to train f, or a separate holdout set, or a combination of the two. In the case that f performs very well on the training set, a holdout set may be necessary, as the lack of examples of poor segmentations will bias g towards always predicting that the segmentation is good.

There have been many methods proposed for attaining uncertainty or confidence estimates from neural networks, but for our experiments we consider four methods: maximum softmax probability, Monte-Carlo Dropout, heteroscedastic classifier neural networks, and learned confidence estimates. We selected these based on their simplicity to implement as well as their diversity.

3.1. Maximum Softmax Probability

The first method we evaluate is the maximum softmax probability, which was demonstrated by [8] to be surprisingly effective at the tasks of misclassification and out-of-distribution detection. The softmax probability can be obtained from any classification neural network for free, making it an appealing choice for confidence estimates. To calculate the maximum softmax probability we simply calculate the maximum across the class dimension of the softmax output from the network f:

$$z = \max(\text{Softmax}(\rho)). \tag{4}$$

For segmentation, this is done per output pixel in order to obtain an uncertainty map that is of the same resolution as the input image.

3.2. Monte-Carlo Dropout

The second uncertainty estimation method we consider is Monte-Carlo dropout (MC-dropout) [6], which has previously seen success in the field of medical imaging [17, 28]. MC-dropout approximates a BNN by sampling from a neural network trained with dropout [26] at inference time in order to produce a distribution over the outputs. This approach is very simple to implement in practice, and as many modern neural network architectures already leverage dropout for regularization purposes, uncertainty estimates can often be attained without any changes to the architecture or training paradigm. MC-dropout models epistemic uncertainty, which is the uncertainty associated with the model parameters, such that increasing the amount of training data tends to decrease the epistemic uncertainty associated with the model.

Following the approach used for Bayesian SegNet [11, 12], we apply dropout with $p = 0.5$ after each central convolutional block of our U-Net architecture. During test time we sample from the segmentation network T times (we use $T = 20$) and then calculate the average softmax probability over all of the samples in order to approximate Monte Carlo integration:

$$p = \frac{1}{T}\sum_{t=1}^{T}\text{Softmax}(\rho_t). \tag{5}$$

Model uncertainty z is estimated by calculating the entropy of the averaged probability vector across the class dimension:

$$z = -\sum_{c=1}^{C} p_c \log p_c. \tag{6}$$

3.3. Heteroscedastic Classifier Neural Network

The third uncertainty estimation technique we evaluate is one which attempts to model aleatoric uncertainty, which is the uncertainty present in the data itself, such as from noisy labels or measurements. To model aleatoric uncertainty, [12] introduce the heteroscedastic classifier neural network, which we will refer to as HCNN. In this method, uncertainty estimates are learned by the network, rather than being calculated post-hoc as with MC-dropout. The HCNN produces two outputs via two separate output branches: class prediction logits, and a variance estimate which represents model uncertainty. Again, in the case of segmentation, these two quantities are computed per output pixel. During training, Gaussian noise with magnitude equal to the variance estimate is sampled and added to the probability logits, which are used to calculate the training loss as usual:

$$p = \frac{1}{T}\sum_{t=1}^{T}\text{Softmax}(\rho + z\epsilon_t), \quad \epsilon_t \sim \mathcal{N}(0, 1). \tag{7}$$

In our experiments we set T = 100. We also apply a softplus function to the output of the variance estimation branch in order to ensure that it is non-negative.

3.4. Learned Confidence Estimates

The final technique we evaluate is Learned Confidence Estimates (LCE), which was introduced by [4]. This method is similar to HCNN in that the network produces two separate outputs: prediction probabilities and a confidence estimate. Confidence estimates are motivated by interpolating between the predicted probability distribution and the target distribution during training, where the degree of interpolation is proportional to the confidence estimate:

$$p = z \cdot \text{Softmax}(\rho) + (1 - z) \cdot \text{Onehot}(y). \tag{8}$$

In this formulation, low confidence estimates are pushed towards the correct answer, while high confidence estimates remain unchanged. To prevent the model from always producing low confidence estimates, a log penalty on the confidence estimate is added to the loss function. As a result, the network can reduce its overall training loss if it correctly infers which samples it is likely to predict incorrectly.

4. Experiments

To evaluate our method we apply it to the problem of skin lesion segmentation, as this application has received a fair amount of attention from the deep learning community [5]. Specifically, we work with the ISIC 2017 dataset [3], which consists of 2,750 dermoscopic images in three official dataset splits: 2,000 training images, 150 validation images, and 600 test images. Each image depicts a skin lesion from one of three different classes: melanoma, seborrheic keratosis, and benign nevi. Additionally, each image has an accompanying expert-labeled binary segmentation mask. For our experiments, we resize all images and ground truth masks to 224×224 pixels.

For our semantic segmentation network f, we adopt a U-Net style model architecture [25]. To facilitate MC-dropout we apply dropout with $p = 0.5$ to the central layers of the encoder and decoder, as in Bayesian SegNet [11]. Each model is trained for 120 epochs using batches of 16 images, and the Adam optimizer [13] with a learning rate of 0.001. Images are randomly flipped and rotated at 90 degree intervals for data augmentation. For each uncertainty estimation method we train five models with random parameter initializations so that we can observe variance in performance. We find that all segmentation networks score within the range of 0.73 ± 0.02 Jaccard index, which is competitive with single-model performance for this task.

Our segmentation quality prediction network g is a VGG-style CNN, which is trained to predict the Jaccard index of any given segmentation prediction given the original image, predicted segmentation mask, and confidence.

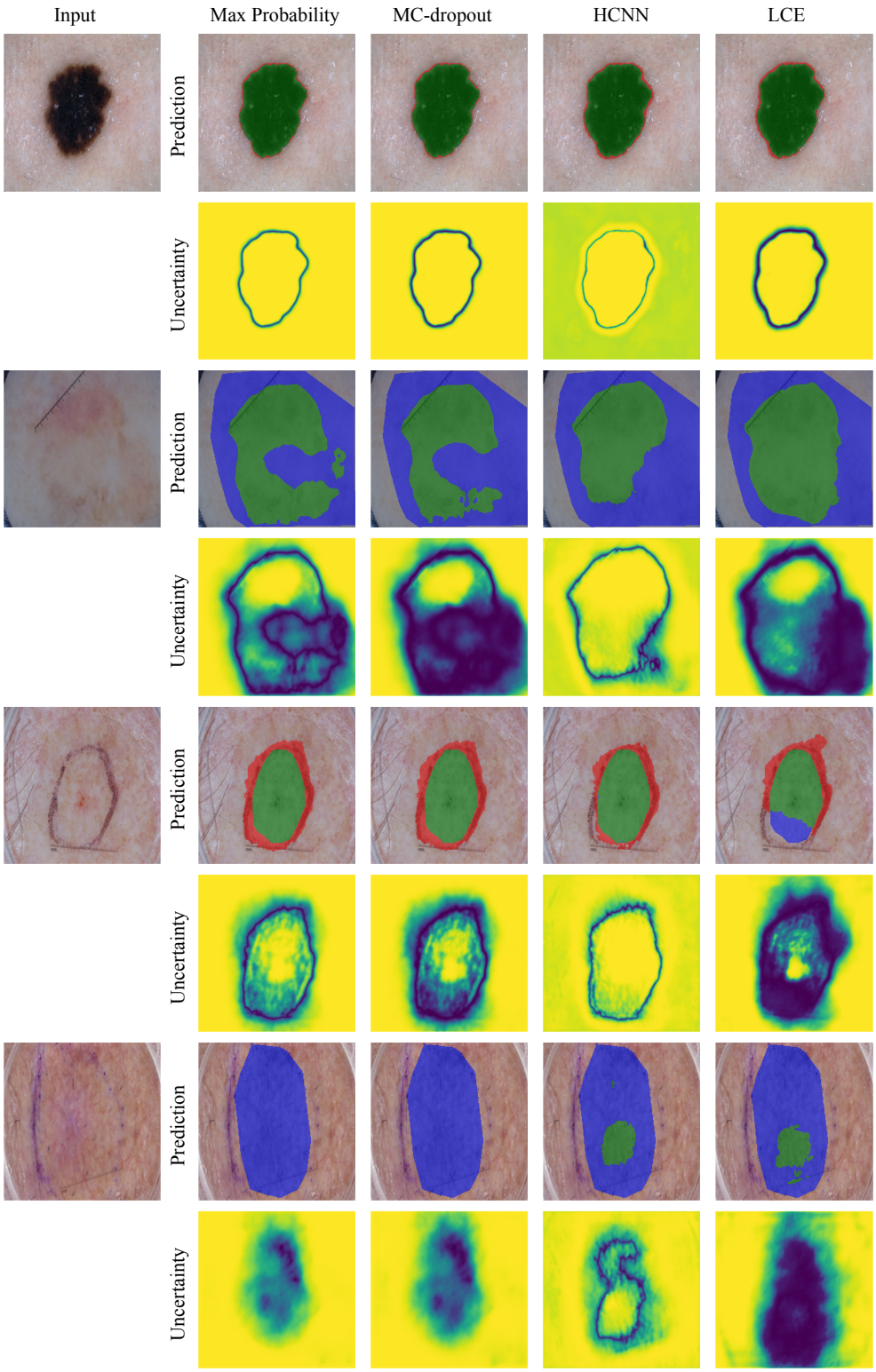


Figure 2: Segmentation predictions (even rows) and uncertainty maps (odd rows) for four different uncertainty estimation methods. In the segmentation predictions, green = true positive, red = false positive, and blue = false negative. In the uncertainty maps yellow = low uncertainty and purple = high uncertainty. Best viewed in colour.

Table 1: Comparison of different Jaccard index estimation methods. ↓ indicates that a lower score is better, while ↑ indicates that a higher score is better. All values except for RMSE are percentages. Results averaged over 5 runs with parameters randomly initialized.

Method	RMSE ↓	Detection Error ↓	AUROC ↑	AUPR-Pass ↑	AUPR-Fail ↑
RCA	0.438 ± 0.007	43.8 ± 1.0	53.7 ± 1.4	74.4 ± 1.4	30.7 ± 0.9
QualityNet	0.213 ± 0.009	25.7 ± 2.7	80.9 ± 3.1	89.0 ± 2.3	69.1 ± 4.7
No Uncertainty	0.198 ± 0.011	27.3 ± 3.3	79.8 ± 3.8	88.5 ± 1.9	66.4 ± 7.9
Max Probability	$\mathbf{0.168} \pm 0.014$	$\mathbf{18.4} \pm 3.0$	$\mathbf{88.4} \pm 2.2$	$\mathbf{93.2} \pm 1.6$	$\mathbf{80.5} \pm 3.2$
MC-dropout	$\mathbf{0.163} \pm 0.010$	$\mathbf{18.8} \pm 1.4$	$\mathbf{88.1} \pm 0.8$	$\mathbf{93.5} \pm 1.3$	$\mathbf{78.1} \pm 3.0$
HCNN	0.196 ± 0.023	21.3 ± 1.8	85.5 ± 1.5	91.6 ± 1.4	76.2 ± 4.5
LCE	$\mathbf{0.167} \pm 0.019$	$\mathbf{19.3} \pm 1.1$	$\mathbf{88.3} \pm 1.4$	$\mathbf{93.6} \pm 1.5$	$\mathbf{79.1} \pm 3.9$

We train our quality prediction network for 30 epochs using batches of 16 images, and the Adam optimizer with a learning rate of 0.001. As with our segmentation network, we apply flipping and rotating transforms for data augmentation.

For comparison, we also train models using two other neural network-based segmentation quality prediction methods: Reverse Classification Accuracy (RCA) [27] and QualityNet [9]. As each of these approaches have their own architectural and optimization-based hyper-parameters, we have kept these the same as our technique, where applicable.

4.1. Uncertainty Maps

To compare the quality of the uncertainty maps from each of the different uncertainty (and confidence) estimation techniques, we visualize them in Figure 2. In cases where the predicted segmentation is very close to the ground truth segmentation, we find that all techniques act similarly, outputting a tight ring along the segmentation borders. This is what we would expect in such a situation. The more interesting observation is how the uncertainty maps react when the predicted segmentation is very poor. We find that in general, maximum softmax probabilty, MC-dropout, and LCE all display high uncertainty in regions that are segmented incorrectly. Conversely, HCNN usually outputs a small band of low uncertainty around its prediction, but does not highlight other areas that may be incorrect. This output is less useful for identifying failed segmentations, which agrees with our findings in §4.2.

4.2. Segmentation Quality Prediction

We use a variety of metrics to evaluate how well our models can predict the quality of segmentations: RMSE, detection error, AUROC, and AUPR; each of which is defined below.

RMSE: Measures Root Mean Squared Error, which is the difference between the predicted Jaccard index and the true Jaccard index. Predictions that are further from the true value are penalized more heavily in this metric. RMSE is calculated as $\sqrt{\frac{\sum_{t=1}^{n}(\hat{v}_t - v_t)^2}{n}}$ where t indexes the test examples, and n is the total number of test examples.

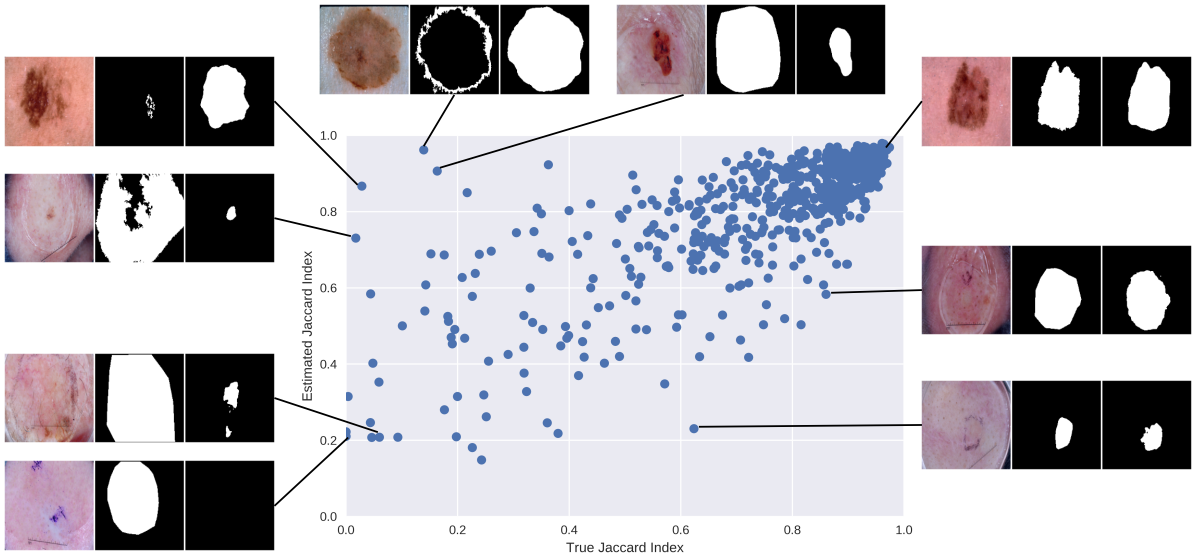
For practical applications (e.g. human-in-the-loop) we may also want to measure how well our model can detect *failed* segmentations. For the ISIC 2017 dataset, a Jaccard index of below 0.7 is considered to be a failed segmentation [3]. To evaluate how well our model can detect these failures, we threshold the true Jaccard index labels at 0.7 to obtain binary labels, which we can use to calculate detection error, AUROC, and AUPR.

Detection Error: Measures the minimum possible misclassification probability over all possible thresholds δ when detecting segmentation failures, as defined by $\min_\delta \{0.5\, P_{\text{pass}}(f(x) \leq \delta) + 0.5\, P_{\text{fail}}(f(x) > \delta)\}$. Here, we equally weight P_{pass} and P_{fail} as if they have the same probability of appearing in the test set.

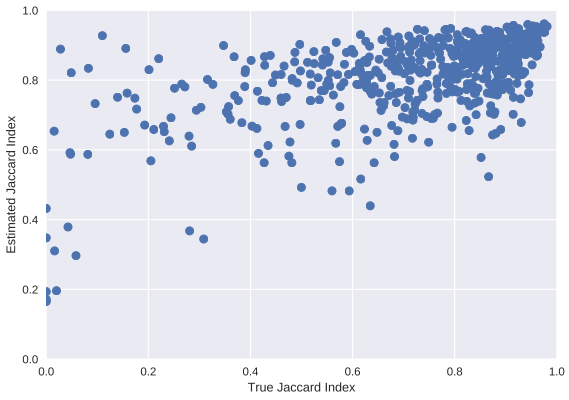
AUROC: Measures the Area Under the Receiver Operating Characteristic curve. The Receiver Operating Characteristic (ROC) curve plots the relationship between true positive rate and false positive rate. The area under the ROC curve can be interpreted as the probability that a correctly segmented image will have a higher quality estimate than a failed segmentation.

AUPR: Measures the Area Under the Precision-Recall (AUPR) curve, which is calculated by plotting precision versus recall. In our tests, AUPR-Pass indicates that acceptable segmentations are used as the positive class, and AUPR-Fail indicates that failed segmentations are used as the positive class. We evaluate both metrics so that we can see if our model is biased towards either class.

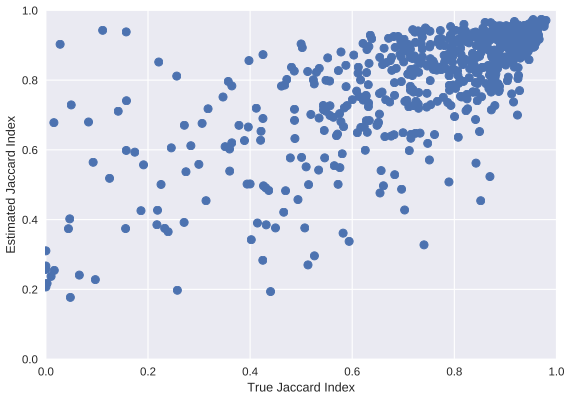
In Table 1, we present the results of our quantitative evaluation of different segmentation quality estimation methods. These are organized by two groupings: 1) recent baselines RCA and QualityNet, which leverage CNNs to predict



(a) LCE

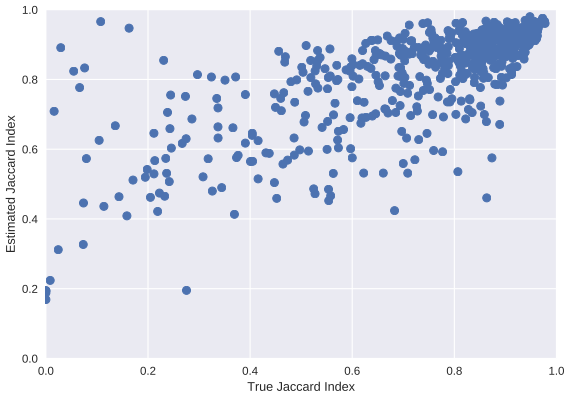


(b) No uncertainty



(c) Max probability

(d) MC-dropout



(e) HCNN

Figure 3: Scatter plots of true Jaccard index versus the estimated Jaccard index from segmentation quality estimation networks trained with different uncertainty estimate techniques. In a) we annotate some points of interest with their associated image, ground truth, and the predicted segmentation respectively. We note that most false positives (upper left corner) are caused by corrupted labels or lazy annotations.

segmentation quality and 2) variants of our two-stage CNN approach. The variants of our approach apply different quantitative measures of uncertainty, described above. We also include a method which does not explicitly generate uncertainty as an intermediate representation, i.e. g receives only $(x, \hat{y})$ as input rather than $(x, \hat{y}, z)$. This is slightly different than QualityNet, in which g receives $(x \odot \hat{y})$, where $\odot$ is an element-wise matrix multiplication.

We find that treating uncertainty explicitly improves performance significantly compared to the case with no uncertainty information, reducing RMSE by up to 0.03 points, and detection error by up to 8 percentage points. However, surprisingly the particular method by which uncertainty is captured does not have a large effect, at least in this setting. Maximum softmax probability, MC-dropout, and LCE all produce similar improvements in performance. It was expected that MC-dropout or LCE would outperform maximum softmax probability since they have been shown to surpass softmax probability in tasks such as out-of-distribution detection [4], but this was not the case for this particular dataset. Of the uncertainty estimation techniques, HCNN improved performance the least; only 5 points of detection error and no improvement on RMSE. This agrees with [12], which indicates that aleatoric uncertainty which HCNN aims to capture, is a poor choice for detecting model failures since it mainly models noise in the data itself.

We find that our implementation of QualityNet performs roughly equal to our *no uncertainty* baseline, which is expected given how similar the implementations are. Unfortunately, RCA performs very poorly; only slightly better than random. This is likely because the algorithm was designed to work on datasets of registered images with very little variation between them, such as MRI scans of internal organs. In these datasets each of the objects to be segmented are extremely similar in shape and location. In contrast, the skin lesions from the ISIC 2017 dataset appear with a wide variety of colours, textures, shapes, sizes, and locations, making them very difficult for modern image registration techniques to succeed.

In Figure 3 we plot the true versus the predicted Jaccard index for each of the different uncertainty estimates we tested. We observe that max probability, MC-dropout, and LCE are all better at identifying poor quality segmentations (lower left corner) compared to HCNN or the no uncertainty baseline. Additionally, we note that the majority of false positives (poor quality segmentations that are rated highly) are caused by either corrupted labels or lazy annotations, as shown in Figure 3a.

Interestingly, segmentation quality estimates rarely fall below 0.2 for any method. This is likely caused by the rarity of poor quality segmentations in the dataset used to train the segmentation quality estimation network, since it was trained on the same dataset as the original segmentation network. While it is probable that using a separate held-out dataset would result in a greater number of poor quality segmentation examples, and therefore better performance from the segmentation quality estimation network, we do not explore this option due to the small size of the ISIC 2017 dataset.

5. Conclusion

In this work, we investigated techniques which aid a human operator, such as a clinician, interact with a deep learning-based automated segmentation pipeline. We showed how uncertainty could be derived at the pixel- and image-level within a single end-to-end framework. We demonstrated our method qualitatively and quantitatively on the task of skin lesion segmentation. Though a neural network trained to predict segmentation quality has the capacity to measure uncertainty internally, we showed that making spatial uncertainty explicit aided in predicting a measure of segmentation quality, the Jaccard index. Moreover, we demonstrated that several recent methods for quantifying uncertainty worked well in this setting. In the future, we plan on extending our analysis to other medical segmentation problems, and even tasks outside segmentation that could benefit from a human-in-the-loop. We also used simple, standard losses for our segmentation model. Recent techniques that aim to optimize application-specific metrics like the Jaccard index would likely improve overall performance.

References

[1] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[2] A. Atanov, A. Ashukha, D. Molchanov, K. Neklyudov, and D. Vetrov. Uncertainty estimation via stochastic batch normalization. *International Conference on Learning Representation (ICLR)*, 2018.

[3] N. C. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1710.05006*, 2017.

[4] T. DeVries and G. W. Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

[5] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115, 2017.

[6] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In

international conference on machine learning, pages 1050–1059, 2016.

[7] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22):2402–2410, 2016.

[8] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *International Conference on Learning Representation (ICLR)*, 2016.

[9] C. Huang, Q. Wu, and F. Meng. Qualitynet: Segmentation quality evaluation with deep convolutional networks. In *Visual Communications and Image Processing (VCIP), 2016*, pages 1–4. IEEE, 2016.

[10] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2016 IEEE Conference on*, pages 680–688. IEEE, 2016.

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

[12] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, pages 5580–5590, 2017.

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representation (ICLR)*, 2015.

[14] S. Kohl, D. Bonekamp, H.-P. Schlemmer, K. Yaqubi, M. Hohenfellner, B. Hadaschik, J.-P. Radtke, and K. Maier-Hein. Adversarial networks for the detection of aggressive prostate cancer. *arXiv preprint arXiv:1702.08014*, 2017.

[15] T. Kohlberger, V. Singh, C. Alvino, C. Bahlmann, and L. Grady. Evaluating segmentation error without ground truth. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 528–536. Springer, 2012.

[16] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6405–6416, 2017.

[17] C. Leibig, V. Allken, M. S. Ayhan, P. Berens, and S. Wahl. Leveraging uncertainty information from deep neural networks for disease detection. *Scientific reports*, 7(1):17816, 2017.

[18] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[19] Y. Liu, K. Gadepalli, M. Norouzi, G. E. Dahl, T. Kohlberger, A. Boyko, S. Venugopalan, A. Timofeev, P. Q. Nelson, G. S. Corrado, J. D. Hipp, L. Peng, and M. C. Stumpe. Detecting cancer metastases on gigapixel pathology images. Mar. 2017.

[20] C. Louizos and M. Welling. Multiplicative normalizing flows for variational bayesian neural networks. *International Conference on Machine Learning (ICML)*, 2017.

[21] P. Luc, C. Couprie, S. Chintala, and J. Verbeek. Semantic segmentation using adversarial networks. *NIPS Workshop on Adversarial Training*, 2016.

[22] M. Mardani, E. Gong, J. Y. Cheng, S. Vasanawala, G. Zaharchuk, M. Alley, N. Thakur, S. Han, W. Dally, J. M. Pauly, and L. Xing. Deep generative adversarial networks for compressed sensing automates MRI. May 2017.

[23] P. Moeskops, M. Veta, M. W. Lafarge, K. A. Eppenhof, and J. P. Pluim. Adversarial training and dilated convolutions for brain mri segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 56–64. Springer, 2017.

[24] R. M. Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

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

[26] 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.

[27] V. V. Valindria, I. Lavdas, W. Bai, K. Kamnitsas, E. O. Aboagye, A. G. Rockall, D. Rueckert, and B. Glocker. Reverse classification accuracy: predicting segmentation performance in the absence of ground truth. *IEEE transactions on medical imaging*, 36(8):1597–1606, 2017.

[28] X. Yang, R. Kwitt, and M. Niethammer. Fast predictive image registration. In *Deep Learning and Data Labeling for Medical Applications*, pages 48–57. Springer, 2016.

[29] H. Zhang, S. Cholleti, S. A. Goldman, and J. E. Fritts. Meta-evaluation of image segmentation using machine learning. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1138–1145. IEEE, 2006.