A Survey of Transfer Learning in Uncertainty Aware Computer-Aided Diagnosis

Eugene Choi
Courant Institute of
Mathematical Sciences
New York University
eugenechoi@nyu.edu

Jeff Cui
Courant Institute of
Mathematical Sciences
New York University
zc2357@nyu.edu

Abstract

Despite the recent success of medical deep learning, two fundamental problems remain. One is that medical datasets used for supervised training are often small due to the cost of medical imaging and expert annotations. Two is that medical practitioners need models with well-calibrated prediction scores and uncertainty estimates for life-critical decisions. To address these difficulties, we examine transfer learning and uncertainty quantification methods. We find that a multi-headed density network co-trained on multiple medical imaging datasets accelerates convergence when transferred to a nodule detection task, that such a model can effectively quantify the total predictive uncertainty by modeling aleatoric and epistemic uncertainty with density predictions and the Monte Carlo dropout, and that an ensemble of these networks trained with fast gradient sign method data augmentation can detect out-of-distribution samples. All of the code used for this project can be found in the following Github repository: https://github.com/jeffacce/nyu-cv2271-final

1 Introduction

Deep learning has seen wide adoption in the medical domain. However, two core problems still remain in the domain of computer-aided diagnosis. One is that medical datasets are often small due to the cost of medical imaging and expert annotations. While recent advances in self-supervised learning seem promising, they are not accessible to many practitioners due to a lack of data and computing resources.

Two is that medical practitioners need models with well-calibrated prediction uncertainty for life-critical decisions. Computer-aided diagnosis models perform remarkably when tested against samples from the same data-generating process as the training set, but deteriorates drastically when tested against other samples. This prohibits the application in the medical domain, which requires both well-calibrated and accurate predictions, along with an accurate quantification of predictive uncertainty.

These drawbacks have hampered the mainstream deployment of deep learning models in the medical domain. Hence, we decided to study two widely applicable methods that are easily accessible to many practitioners trying to apply their machine learning skills to high-risk decision-making settings. Therefore, in this final project, we investigate the use of transfer learning and uncertainty quantification for computer-aided diagnosis models.

2 Related Work

2.1 Residual 3D U-Net

Ronneberger et al. [19] proposed the U-Net model, which has seen success in medical segmentation tasks. More recent U-Net models also use residual convolution blocks by He et al. [9] in addition to the long-range skip connections. Çiçek et al. [24] proposed using 3D convolution to segment volumetric medical imagery such as computed tomography (CT) or magnetic resonance (MR) to capture axial local context.

2.2 Transfer and multi-task learning

Transfer learning Because medical datasets are often limited in size, transfer learning from models trained on natural image datasets such as ImageNet is often used for training computer-aided diagnosis models. However, medical images are significantly different from natural images; Raghu et al. [18] makes the point that except for the bottom layer features, natural image features do not apply to medical images. Notably, they report that for the low-data regime of medical datasets, transfer learning from ImageNet helps large overparametrized models, but shows little improvement for smaller models.

Multi-task learning Another direction of active research is multi-task datasets and models in computer-aided diagnosis. While biomedical datasets are often heterogeneous in format, modalities, resolution, and class balance, their limited size (especially 3D segmentation tasks, due to the cost of image acquisition and expert annotations) makes it reasonable to coalesce different datasets into a larger dataset for model training. Large multi-task datasets such as the Medical Segmentation Decathlon [1] have been proposed to encourage generalizable models in the medical domain. There has also been work on multi-task co-training on heterogeneous 3D biomedical segmentation datasets. Chen et al. [5] proposes Med3D, training a large 3D ResNet-like backbone on 8 different tasks with 8 decoder heads to produce the segmentation map, achieving better performance in the multi-task co-training than single-task baselines. They transfer the pretrained encoder head to a pulmonary nodule classification task and show that the pretrained model converges faster and outperforms randomly initialized models.

2.3 Uncertainty Quantification

Epistemic uncertainty Many Bayesian methods have been developed to quantify uncertainty in neural networks, ranging from early works using variational inference based methods by placing a Gaussian distribution over the weights (Bayes by Backpropagation, [3]), Monte Carlo dropout as a Bayesian approximation of Gaussian processes ([6]), to more recent works using moments of stochastic gradient descent (SGD) iterates with a modified learning rate schedule to approximate a Gaussian posterior distribution over neural network weights (Stochastic Weight Averaging Gaussian, [15]) We sample from these approximate posterior distributions to perform Bayesian model averaging. These methods capture the epistemic uncertainty or the model's weight uncertainty.

Aleatoric uncertainty There have been different approaches to capture the aleatoric uncertainties of a model. One method proven effective in the medical imaging domain is the test-time augmentation method through the "image acquisition model" [23] It assumes that aleatoric uncertainty comes from the image acquisition process. Lakshminarayanan et al. [12] proposes a non-Bayesian approach to uncertainty quantification using a combination of ensemble, adversarial training and density network. Their method provides benefit such as parallelization, capturing of the aleatoric uncertainty and the improvment of the model performance. Lastly, Kendall and Gal [11] provide a more principled approach of capturing the aleatoric uncertainty using the density network with the Monte Carlo dropout method.

3 Experiment Setup

3.1 Datasets

For our experiments, we use 3 segmentation tasks for co-training and a heavily class-imbalanced classification task for transfer learning.

3.1.1 Segmentation tasks

Segmentation datasets are used for the multi-task co-training. For each task, we train the model on the union of all segmentation labels (usually denoting the whole organ along with the pathology). These are detailed below.

- The Liver Tumor Segmentation Benchmark (LiTS) 2017 by Bilic et al. [2], including 131 liver CT scans in the training set, and 70 scans in the test set. The segmentation ground truth consists of liver, and liver tumor.
- The Brain Tumor Segmentation Challenge (BraTS) 2020 by Menze et al. [16], a dataset of multimodal brain MR scans with T1, T2, T1 contrast-enhanced (T1CE), and Fluid Attenuated Inversion Recovery (FLAIR) channels, including 369 cases in the training set and 125 in the validation set. The segmentation ground truth consists of GD-enhancing tumor, the peritumoral edema, and the necrotic and non-enhancing tumor core.
- The Kidney Tumor Segmentation Challenge 2021 by Heller et al. [10], including 300 kidney CT scans in the training set. The segmentation ground truth consists of kidney, tumor, and cyst.

3.1.2 Classification task

We use the LUNA16 dataset by Setio et al. [21] for the transfer learning experiment, consisting of 888 pulmonary CT scans. Lung nodule detection systems usually have two stages: candidate generation, and false positive reduction. Candidate generation focuses on high-recall detection of possible nodule regions of interest, at the cost of returning many false positives. False positive reduction then classifies the candidates as positives and negatives. For our experiment, we choose the false positive reduction task of the challenge, using the nodule candidate coordinates prepared by the challenge organizers. In total, there are 1,557 positive candidates (actual nodule) and 753,418 negative candidates. Positive candidates are roughly 0.2% of the classification dataset.

3.1.3 Preprocessing

Isometric resampling Scans in the datasets are acquired with different voxel spacings. To normalize the aspect ratio, we resample all CT and MR scans to a voxel spacing of 1mm $\times$ 1mm $\times$ 1mm. CT and MR imagery are resampled using cubic spline interpolation. The corresponding segmentation maps are sampled with nearest neighbor interpolation.

Voxel intensity normalization For the BraTS multimodal MR images, we clip all channels to [0, 95% upper bound] intensities in the training set. For the CT datasets, voxel intensities are reported in Hounsfield scale, reporting -1000 for air and 0 for water at standard temperature and pressure. For the LUNA16 dataset, we clip the voxel intensities to [-1000, 400], corresponding to lung tissues and air. For the LiTS 2017 dataset, we clip to [-200, 200], corresponding to the liver and surrounding tissues. For the KiTS 2021 dataset, we clip to [-500, 500], corresponding to the kidney and surrounding tissues. We then linearly transform all input to [0, 1].

Candidate patch extraction For the LUNA16 false positive reduction task, nodule candidates are provided as world coordinates. We sample 48mm $\times 48$mm $\times 48$mm patches from the provided coordinates for the classifier training.

Train-validation-test split For all datasets, to ensure that validation and test sets do not leak into the training set, the splits are performed on cases rather than slices or candidates within each case. For ease of evaluation, we use only the training set with ground truth labels for the segmentation tasks: from the provided training datasets, we further split 80% for training and 20% for validation. For the

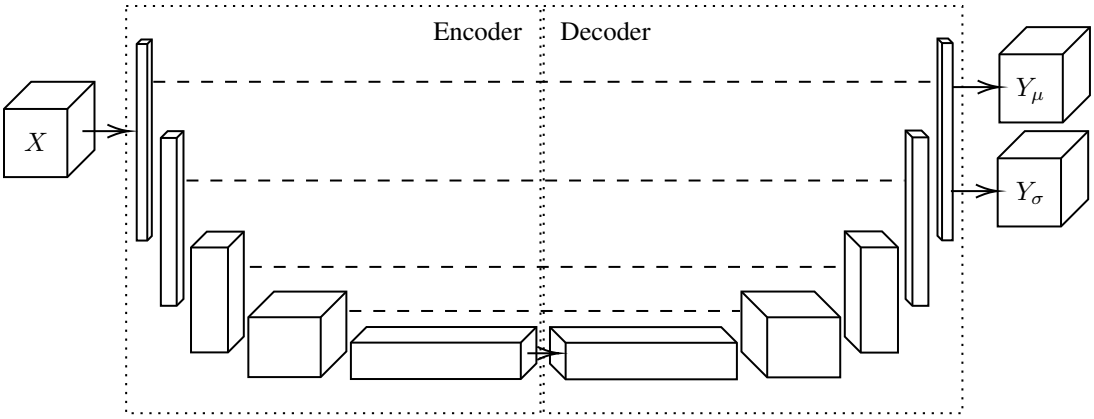


Figure 1: UNet3D density model

LUNA16 dataset, since it includes ground truth classification labels for all nodule candidates, we split 60% for training, 20% for validation, and 20% for testing the transfer learning model performance and convergence.

3.2 Multi-task segmentation training

3.2.1 Multi-headed density UNet3D model

We use a multi-headed 3D U-Net density model (Figure 1) to learn the three segmentation tasks. To achieve this, we use three decoder heads with a shared encoder trunk. Each head is trained with a different optimizer, and the encoder trunk receives gradient updates from all three decoder heads. Since BraTS has 4 MR channels and LiTS, KiTS, and LUNA16 have 1 CT channel, all CT inputs go through a convolutional layer mapping 1 channel to 4 channels before the encoder, and the encoder takes 4 channels as the input.

In order to capture the aleatoric uncertainty, we implement a density model proposed by Kendall and Gal [11] to predict a density as its output, rather than a single score. Given an input x, our network predicts both the mean ($f_W(x)$) and the variance ($\sigma^2_W(x)$). Thus, we get the logits, $z \sim \mathcal{N}(f_W(x), \sigma^2_W(x))$. During inference, we use the reparametrization trick to integrate out the Gaussian distribution.

$$z_t = f_W(x) + \sigma_W(x) \odot \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, I)$$

We get an expected softmax output after T steps of sampling:

$$\hat{P} \approx \frac{1}{T} \sum_{t=1}^{T} \text{Softmax}(z_t) = \frac{1}{T} \sum_{t=1}^{T} \exp(z_{t,c} - \log \sum_{c'} \exp(z_{t,c'})) \tag{1}$$

To see whether co-training accelerates convergence and improves performance compared to training on a single dataset, we train a baseline single-headed UNet3D density model on each of the tasks, keeping the loss function, hyperparameters, and train-validation-test splits identical.

3.2.2 Loss function

We use the Tversky loss on the each of the three segmentation decoder heads. Tversky loss [20] is defined as

$$\text{TL}(\hat{P}, G) = 1 - \frac{\hat{P}G}{\hat{P}G + \alpha\hat{P}(1-G) + \beta G(1-\hat{P})}$$

where $\hat{P}$ is the predicted probability map, G the ground truth map, and multiplications between the maps are element-wise. α, β are weights for the false positives and negatives. In computer-aided diagnosis, false negatives carry more risk than false positives. For our experiments, $\alpha = 0.3, \beta = 0.7$.

We use the AdamW optimizer [14] with a learning rate of $\alpha = 10^{-4}$, weight decay $\gamma = 10^{-2}$; both the encoder and decoder have a dropout probability of $p = 0.6$.

3.3 Transfer Learning

3.3.1 Transfer model

We transfer the encoder trunk pre-trained on the 3 segmentation datasets to the LUNA16 false positive reduction classification task, attach a classifier head with two fully connected layers, and train the model on the false positive reduction task. Positive samples are flipped randomly on x, y, z axes, and negative samples are kept as is. For each training batch, we sample 50% from the positive samples, and 50% from the negative samples, to avoid class imbalance during training. For the baseline (training from scratch), we keep the model architecture, the loss function, hyperparameters, and train-validation-test splits identical, but randomly initialize the model.

3.3.2 Loss function

Focal loss [13] for binary classification is defined as

$$\mathrm{FL}(p) = \begin{cases} -\alpha(1-p)^{\gamma}\log p & (y=1) \\ -(1-\alpha)(p)^{\gamma}\log(1-p) & (y=0) \end{cases}$$

where p is the predicted probability, y the ground truth binary label, α is the class balance weight, γ a hyperparameter for scaling losses of incorrectly classified samples. We choose $\alpha = 0.5, \gamma = 2.0$ for the transfer experiment. We use the AdamW optimizer (Loshchilov and Hutter [14]) with a learning rate of $\alpha = 10^{-4}$, weight decay $\gamma = 10^{-2}$; both the encoder trunk and classifier head have a dropout probability of $p = 0.6$.

3.4 Uncertainty Quantification

We capture both the epistemic and the aleatoric uncertainties for our project using two mainstream approaches in quantifying uncertainties in the neural network. First, we capture the epistemic by using the Monte Carlo dropout method (Gal and Ghahramani [6]). We also capture the aleatoric uncertainty using a density model (Kendall and Gal [11]), that learns to attenuate the loss at points with high uncertainty and hence capturing the heteroscedastic uncertainty. We define the following terms below, according to the work of Kendall and Gal [11]:

3.4.1 Epistemic Uncertainty

Epistemic uncertainty refers to our ignorance of the model that best explains the given data. This is also known as model uncertainty, and it arises from the fact that we do not know the optimal weight values for our neural network model. Known as the reducible uncertainty, we may theoretically lower this uncertainty by collecting more data. Here, we use and We measure the epistemic uncertainty by approximating the posterior distribution using the Monte Carlo dropout method with the dropout distribution as the variational distribution as follows.

$$\mathbb{E}_{w\sim p(w|\mathcal{D})}[f_w(\boldsymbol{x})] = \int p(w \mid \mathcal{D}) f_w(\boldsymbol{x})\,\mathrm{d}w$$

Since the true posterior $p(w \mid \mathcal{D})$ is intractable, we approximate it with a variational distribution $q_\theta(w)$.

$$\mathbb{E}_{w\sim p(w|\mathcal{D})}[f_w(\boldsymbol{x})] \approx \int q_\theta(w) f_w(\boldsymbol{x})\,\mathrm{d}w$$

Marginalizing over the variational distribution can be approximated using the Monte Carlo integration:

$$\mathbb{E}_{w\sim p(w|\mathcal{D})}[f_w(\boldsymbol{x})] \approx \hat{\mu}_{model} = \frac{1}{T}\sum_{i=1}^{T} f_{\hat{w}_i}(\boldsymbol{x}), \quad \hat{w}_{1,\ldots,T} \sim q_\theta(w)$$

with T sampled mask model weights $\hat{w}_{1,\ldots,T}$ where $q_\theta(w)$ is the dropout distribution defined as

$$q_\theta(w) = M_i \odot \mathrm{diag}\big([Z_{i,j}]_{j=1}^{K_i}\big), \text{ where } \quad Z_{i,j} \sim \mathrm{Bernoulli}(p_i)$$

and M_i is the model weights before masking. The predictive mean and variance of the resulting distribution represents model uncertainty (or epistemic uncertainty).

3.4.2 Aleatoric Uncertainty

Aleatoric uncertainty refers to the uncertainty that arises from the underlying data generating process, which may have failed to capture the information that should have been recorded. This results in observations with many possible interpretations and introduces inherent variability. Lost information cannot be recovered even with an infinite amount of data and hence is referred to as irreducible uncertainty. We measure the aleatoric uncertainty using the density network with reparametrization trick mentioned in (equation 1). Hence, $\hat{\mathbf{z}}_i \mid \mathbf{W} \sim \mathcal{N}(f_W(x), \sigma_W^2(x))$.

3.4.3 Total Variance

Assume $y = f(x) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma_{alea.}^2)$ and $f_W(x)$ is our model approximating $f(x)$.

$$\mathbb{E}[(y - f_W(x))^2] = \mathbb{E}[(y - f(x))^2] + \mathbb{E}[(f(x) - f_W(x))^2] = \sigma_{alea.}^2 + \sigma_{epis.}^2$$

Hence, we can compute the total predictive variance by decomposing it as follows:

Total predictive variance = aleatoric uncertainty + epistemic uncertainty

Since we capture both the variance term of the above Gaussian distribution with the density network and the expectation term below using the Monte Carlo sampling, our method is capable of approximating the total predictive variance, as shown below.

$$\mathrm{Var}_{q_\theta(y^*|x^*)}(y^*) \approx \frac{1}{T}\sum_{t=1}^{T} \tau(x^*) + f_W(x^*)^T f_W(x^*) - \mathbb{E}_{q_\theta(y^*|x^*)}{}^T \mathbb{E}_{q_\theta(y^*|x^*)} \tag{2}$$

3.5 Fast-gradient sign method (FGSM) data augmentation

Lakshminarayanan et al. [12] proposed a non-Bayesian method for uncertainty quantification, which uses an ensemble of M neural networks trained with an adversarial data augmentation technique. Adversarial examples refer to the samples that are visually indistinguishable to humans, but are misclassified by the neural network. This idea was first proposed by Szegedy et al. [22] and then further developed by Goodfellow et al. [7], who proposed *the fast gradient sign method (FGSM)* that generates adversarial examples. An adversarial example using the fast gradient sign method for a given input, target pair $\mathbf{x}$, y, and a loss $\mathcal{L}(\theta, \mathbf{x}, y)$ is computed as follows:

$$x' = x + \epsilon \odot \mathrm{sign}\big(\nabla_{\mathbf{x}} \mathcal{L}(\theta, \mathbf{x}, y)\big)$$

Lakshminarayanan et al. [12] attributes the deep ensemble's ability to capture the aleatoric uncertainty and detect out-of-distribution sample to the use of the density network, ensembling, and FGSM. The paper further compared their method to the Monte Carlo dropout method, which could not capture such uncertainties.

We conclude that the comparison of the two methods is ill-defined since they measure different types of uncertainty. We argue that, contrary to the authors' framework of viewing the Monte Carlo method and density network as competing approaches, combining the density network with the Monte Carlo sampling will not only lead to a more comprehensive estimation of the uncertainty (by capturing the total predictive uncertainty), but also enable the out-of-distribution sample detection to certain degree. As a result, we use FGSM to train density networks using the Monte Carlo sampling approach to test our hypothesis in the next section.

4 Results

4.1 Multi-task co-training

We compare the segmentation and uncertainty quantification performance of the multi-task co-training model and single-task baseline models. An ideal, calibrated model should produce a good segmentation, and make confident correct predictions and uncertain incorrect predictions. Therefore, we use the Dice coefficient to measure the segmentation performance of the co-training and baseline models, and we measure the uncertainty quantification performance with the 4 metrics below:

- Mean entropy of true positive (TP) and true negative (TN) voxels (lower better; TP, TN from the binary prediction; mean entropy from the predicted probability map)

- Mean entropy of false positive (FP) and false negative (FN) voxels (higher better; FP, FN from the binary prediction; mean entropy from the predicted probability map)
- Brier score (lower better; MSE between the ground truth and the predicted probability) [4]
- Expected calibration error (lower better), as suggested by Naeini et al. [17][8]

For each segmentation task, we select the best validation loss from the baseline and co-training models and run the evaluations. From the co-training experiments, we were unable to replicate the improvements over single-task baselines in Med3D by Chen et al. [5]. We note that this is perhaps due to the model size (small U-Net in our experiments compared to ResNet-152 in Med3D) and heavier regularization (for Monte Carlo dropout uncertainty quantification) used in our experiments.

Table 1: Segmentation and uncertainty quantification performance

Task	Dice	Entropy (TP+TN)	Entropy (FP+FN)	Brier	ECE
LiTS Single	0.919	1.06×10^{-2}	0.442	7.61×10^{-3}	6.65×10^{-3}
LiTS Co-train	0.918	8.24×10^{-3}	0.447	7.60×10^{-3}	6.72×10^{-3}
BraTS Single	0.830	4.28×10^{-3}	0.552	2.61×10^{-3}	2.50×10^{-3}
BraTS Co-train	0.785	8.80×10^{-3}	0.593	3.63×10^{-3}	4.42×10^{-3}
KiTS Single	0.907	4.61×10^{-3}	0.490	2.68×10^{-3}	2.41×10^{-3}
KiTS Co-train	0.892	7.24×10^{-3}	0.491	3.13×10^{-3}	2.92×10^{-3}

As seen from the figure below, the segmentation results successfully captured both the epistemic uncertainty and aleatoric uncertainty. Epistemic uncertainty map is centered in the regions of model predictions where there are regions that the model cannot discern with high confidence. Also, the aleatoric uncertainty is highlighted in the boundary regions of the segmentation map, which is the region with the most variability in the data.

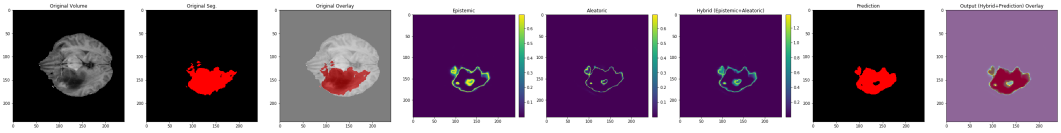


Figure 2: Brain tumor segmentation result

4.2 Out-of-distribution data detection

To see whether an ensemble of models behaves differently on an out-of-distribution dataset, we train 10 distinct models with FGSM data augmentation [12] on the LiTS segmentation task as an ensemble, and run inference on both in-distribution (LiTS) and out-of-distribution (BraTS, KiTS) datasets.

For each comparison, we sample 10 inputs from the in-distribution dataset, and 10 from the out-of-distribution datasets. All samples are from the validation sets and are previously unseen during model training. We then run inference for both samples, calculating the predictive entropy for each input. We simply average the predictions $\hat{P}$ (equation 1) of M models in the ensemble, and take the entropy:

$$\mathbb{H}(\hat{Y}) = -\sum_{c} \hat{Y}_c \log \hat{Y}_c, \quad \hat{Y} = \frac{1}{M} \sum_{m=1}^{M} \hat{P}$$

which gives us the predictive entropy for each of the 10 inputs. We then compare the two sampling distributions of predictive entropy with two-tailed Welch's t-test.

As a baseline comparison, we compare (LiTS_1, LiTS_2), two mutually exclusive samples from the LiTS validation set, as the model inputs. For in-distribution (ID) vs. out-of-distribution (OOD) comparisons, we compare (LiTS_1, BraTS), and (LiTS_1, KiTS). The comparison results are below.

Table 2: t-test results of comparing aleatoric uncertainties

M	Inputs (ID)	Inputs (OOD)	p-value
1	LiTS_1	BraTS	$\mathbf{1.67 \times 10^{-7}}$
1	LiTS_1	KiTS	0.08
1	LiTS_1	LiTS_2	0.69
5	LiTS_1	BraTS	$\mathbf{4.14 \times 10^{-9}}$
5	LiTS_1	KiTS	0.07
5	LiTS_1	LiTS_2	0.75
10	LiTS_1	BraTS	$\mathbf{4.67 \times 10^{-7}}$
10	LiTS_1	KiTS	**0.02**
10	LiTS_1	LiTS_2	0.25

We note that the mean predictive entropy is significantly different between LiTS and BraTS, which is expected as liver CT scans and multimodal brain MR scans are quite distinct from one another (figure 4). The baseline comparison yields the least significant result. Comparing the mean predictive entropy between LiTS and KiTS yields a less significant result, but we note that it is still more significant than the baseline comparison. While Lakshminarayanan et al. [12] reported results on out-of-distribution detection on natural image datasets, we hypothesize from our experiment results that their method applies to the domain of medical imagery as well.

4.3 Transfer learning

For LUNA16, we train the baseline and transfer models for 4 epochs and compare the performance of both models on the held-out 20% test set. The official evaluation metric of LUNA16 is the free receiver operating characteristic (FROC). To calculate the FROC curve, we take the mean sensitivity sampled at 7 predefined false positive rates: $\frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2, 4, 8$ false positives per scan.[21] We note that transfer learning from the co-training task accelerates convergence in the target classification task, as seen in the test set FROC below and the training loss curve in the appendix.

Table 3: LUNA16 FROC results

Model	Epoch 1	Epoch 4
Transfer	**77.58%**	**81.75%**
Baseline	74.84%	80.61%

In our experiment, we used a small 3D U-Net encoder trunk with 1.4M parameters, and observed a marginal improvement on the baseline model trained from scratch. Raghu et al. [18] concludes that except for lower-level features, natural image features from ImageNet do not transfer well to the medical domain, and that transfer learning from ImageNet does not help smaller models. From our experiment above, and the results of Chen et al. [5] and Raghu et al. [18], we hypothesize that for computer-aided diagnosis tasks, transferring from other medical image datasets could yield better performing models than transferring from natural image datasets.

5 Discussion

In this final project, we looked at transfer learning and uncertainty quantification in the medical domain. However, they come at a cost. Pretraining models for transfer learning requires large datasets, which requires collecting expensive annotations and coalescing heterogeneous datasets. Monte Carlo sampling for uncertainty quantification is compute-intensive, which is prohibitive in real-time applications, such as self-driving cars. Solving this problem will be an exciting research avenue in which we can make neural network models more applicable for deployment across broader higher-risk decision-making contexts, as these fields demand both the advantages we outlined in our study while minimizing the cost that comes with it.

References

[1] Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Bennett A Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M Summers, Bram van Ginneken, et al. The medical segmentation decathlon. *arXiv preprint arXiv:2106.05735*, 2021.

[2] Patrick Bilic, Patrick Ferdinand Christ, Eugene Vorontsov, Grzegorz Chlebus, Hao Chen, Qi Dou, Chi-Wing Fu, Xiao Han, Pheng-Ann Heng, Jürgen Hesser, Samuel Kadoury, Tomasz Konopczynski, Miao Le, Chunming Li, Xiaomeng Li, Jana Lipkovà, John Lowengrub, Hans Meine, Jan Hendrik Moltz, Chris Pal, Marie Piraud, Xiaojuan Qi, Jin Qi, Markus Rempfler, Karsten Roth, Andrea Schenk, Anjany Sekuboyina, Eugene Vorontsov, Ping Zhou, Christian Hülsemeyer, Marcel Beetz, Florian Ettlinger, Felix Gruen, Georgios Kaissis, Fabian Lohöfer, Rickmer Braren, Julian Holch, Felix Hofmann, Wieland Sommer, Volker Heinemann, Colin Jacobs, Gabriel Efrain Humpire Mamani, Bram van Ginneken, Gabriel Chartrand, An Tang, Michal Drozdzal, Avi Ben-Cohen, Eyal Klang, Marianne M. Amitai, Eli Konen, Hayit Greenspan, Johan Moreau, Alexandre Hostettler, Luc Soler, Refael Vivanti, Adi Szeskin, Naama Lev-Cohain, Jacob Sosna, Leo Joskowicz, and Bjoern H. Menze. The liver tumor segmentation benchmark (lits), 2019.

[3] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *ArXiv*, abs/1505.05424, 2015.

[4] Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78:1–3, 1950.

[5] Sihong Chen, Kai Ma, and Yefeng Zheng. Med3d: Transfer learning for 3d medical image analysis. *ArXiv*, abs/1904.00625, 2019.

[6] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation. In *International Conference on Machine Learning*, 2016.

[7] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2015.

[8] 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.

[9] 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.

[10] Nicholas Heller, Fabian Isensee, Klaus H Maier-Hein, Xiaoshuai Hou, Chunmei Xie, Fengyi Li, Yang Nan, Guangrui Mu, Zhiyong Lin, Miofei Han, et al. The state of the art in kidney and kidney tumor segmentation in contrast-enhanced ct imaging: Results of the kits19 challenge. *arXiv preprint arXiv:1912.01054*, 2019.

[11] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*, 2017.

[12] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NIPS*, 2017.

[13] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[15] Wesley Maddox, T. Garipov, Pavel Izmailov, Dmitry P. Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In *NeurIPS*, 2019.

[16] Bjoern H. Menze, András Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin S. Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, Levente Lanczi, Elizabeth R. Gerstner, Marc-André Weber, Tal Arbel, Brian B. Avants, Nicholas Ayache, Patricia Buendia, D. Louis Collins, Nicolas Cordier, Jason J. Corso, Antonio Criminisi, Tilak Das, Hervé Delingette, Çagatay Demiralp, Christopher R. Durst, Michel Dojat, Senan Doyle, Joana Festa, Florence Forbes, Ezequiel Geremia, Ben Glocker, Polina Golland, Xiaotao Guo, Andaç Hamamci, Khan M. Iftekharuddin, Rajesh Jena, Nigel M. John, Ender Konukoglu, Danial Lashkari, José Antonio Mariz, Raphael Meier, Sérgio Pereira, Doina Precup, Stephen John Price, Tammy Riklin-Raviv, Syed M. S. Reza, Michael T. Ryan, Duygu Sarikaya, Lawrence H. Schwartz, Hoo-Chang Shin, Jamie Shotton, Carlos Alberto Silva, Nuno Sousa, Nagesh K. Subbanna, Gábor Székely, Thomas J. Taylor, Owen M. Thomas, N. Tustison, Gözde B. Ünal, Flor Vasseur, Max Wintermark, Dong Hye Ye, Liang Zhao, Binsheng Zhao, Darko Zikic, Marcel Prastawa, Mauricio Reyes, and Koenraad Van Leemput. The multimodal brain tumor image segmentation benchmark (brats). *IEEE Transactions on Medical Imaging*, 34:1993–2024, 2015.

[17] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[18] Maithra Raghu, Chiyuan Zhang, Jon M. Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. In *NeurIPS*, 2019.

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[20] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In *International workshop on machine learning in medical imaging*, pages 379–387. Springer, 2017.

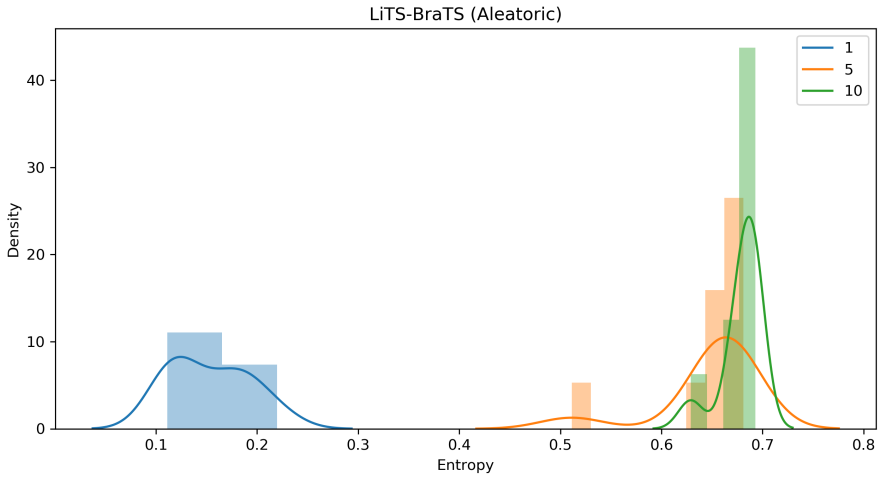
[21] Arnaud Arindra Adiyoso Setio, Alberto Traverso, Thomas de Bel, Moira S.N. Berens, Cas van den Bogaard, Piergiorgio Cerello, Hao Chen, Qi Dou, Maria Evelina Fantacci, Bram Geurts, Robbert van der Gugten, Pheng Ann Heng, Bart Jansen, Michael M.J. de Kaste, Valentin Kotov, Jack Yu-Hung Lin, Jeroen T.M.C. Manders, Alexander Sóñora-Mengana, Juan Carlos García-Naranjo, Evgenia Papavasileiou, Mathias Prokop, Marco Saletta, Cornelia M Schaefer-Prokop, Ernst T. Scholten, Luuk Scholten, Miranda M. Snoeren, Ernesto Lopez Torres, Jef Vandemeulebroucke, Nicole Walasek, Guido C.A. Zuidhof, Bram van Ginneken, and Colin Jacobs. Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: The luna16 challenge. *Medical Image Analysis*, 42:1–13, 2017. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2017.06.015. URL `https://www.sciencedirect.com/science/article/pii/S1361841517301020`.

[22] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, D. Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2014.

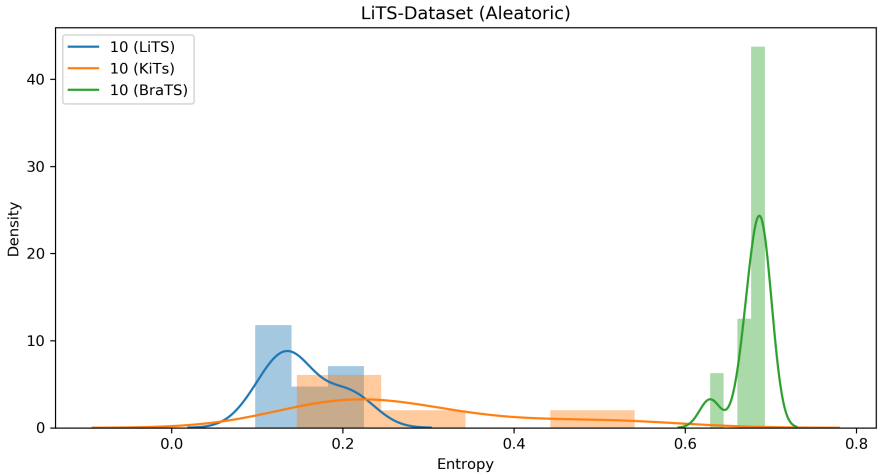
[23] Guotai Wang, Wenqi Li, Michael Aertsen, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren. Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. *Neurocomputing*, 338:34–45, 2019.

[24] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation. *ArXiv*, abs/1606.06650, 2016.

A Appendix

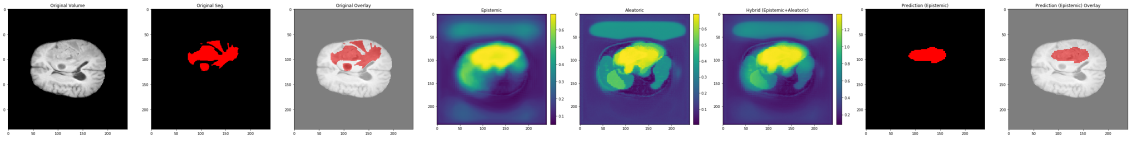


(a) Aleatoric uncertainty was measured using ensemble models trained on the LiTS train set, then tested against the BraTS dev set with size M shown on the legend.

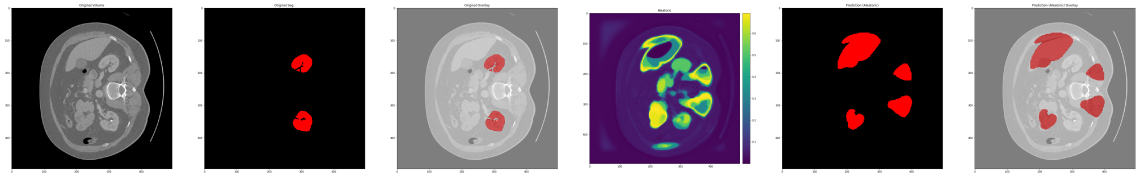


(b) Aleatoric uncertainty was measured by using an ensemble of $M = 10$ against BraTS, KiTS, and LiTS dev sets.

Figure 3: Aleatoric uncertainty was measured using an ensemble model trained on the LiTS train set against BraTS, KiTS, and LiTS dev sets. The aleatoric uncertainty from the o.o.d. samples was captured successfully.



(a) Segmentation and uncertainty (aleatoric) maps of BraTS dev-set sample



(b) Segmentation and uncertainty (aleatoric) maps of KiTS dev-set sample

Figure 4: Aleatoric uncertainty measurement for detecting out of distribution data. As can be seen in the figure above, the model trained on LiTS has much harder time predicting BraTS sample, which looks much more different from the samples from KiTS compared to its training set.

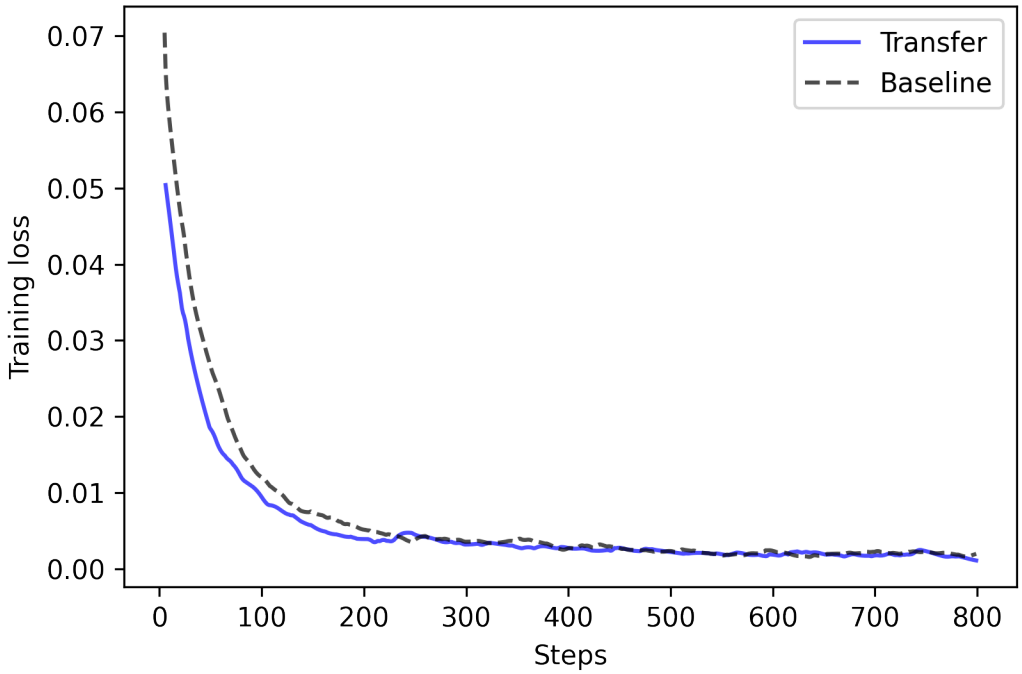


Figure 5: Training loss, LUNA16 false positive reduction task