

Uncertainty Quantification in Chest X-Ray Image Classification using Bayesian Deep Neural Networks

Yumin Liu¹ and Claire Zhao² and Jonathan Rubin³

Abstract. Deep neural networks (DNNs) have proven their effectiveness on numerous tasks. However, research into the reliability of DNNs falls behind their successful applications and remains to be further investigated. In addition to prediction, it is also important to evaluate how confident a DNN is about its predictions, especially when those predictions are being used within medical applications. In this paper, we quantify the uncertainty of DNNs for the task of Chest X-Ray (CXR) image classification. We investigate uncertainties of several commonly used DNN architectures including ResNet, ResNeXt, DenseNet and SENet. We then propose an uncertainty-based evaluation strategy that retains subsets of held-out test data ordered via uncertainty quantification. We analyze the impact of this strategy on the classifier performance. In addition, we also examine the impact of setting uncertainty thresholds on the performance. Results show that utilizing uncertainty information may improve DNN performance for some metrics and observations.

1 INTRODUCTION

Neural networks have been very successful in many fields such as natural language processing [41, 23], computer vision [18, 8], speech recognition [15, 5], machine translation [6], control system [36], auto driving [4] and so on. However, there is much less research available on how reliable neural network predictions are. A common criticism of neural networks is that they are a black box that can perform very well for many tasks, yet lacking interpretability. On the other hand, it is very important to ensure the reliability of a system involved in high risk fields, including stock-market analysis, self-driving cars and medical imaging [28]. As the rapid development of machine learning and artificial intelligence especially deep learning, they are getting more and more applications in health areas including disease diagnosis [9, 10], drug discovery [25, 30] and medical imaging [7, 16, 33]. Rather than just being told a final result by a machine learning algorithm, shareholders (doctors, physicians, radiologists, etc) would like to know how “confident” a neural network model is, so that they can take different actions according to different confidence levels. For example, in a medical image classification scenario, a neural network model is applied to detect whether a patient has a certain type of lung pathology by classifying his/her chest X-ray images. An ideal situation would be that physicians can trust the result of the neural network, if it is highly confident (low uncertainty) about its prediction. On the contrary, if the neural network gives a prediction with low confidence (or high uncertainty), then the prediction could not be trusted and the patient’s scan should be

further examined by a radiologist. Applying this mechanism is beneficial since there are lots of X-ray images everyday but there are limited radiologist resources. It can help prioritize X-ray images for radiologists to examine, require more attention to low confidence instances and support treatment recommendations for highly confident instances.

Neural network-based deep learning algorithms are also getting popular for medical X-ray image processing [27, 1, 35]. It is necessary to examine the uncertainty of neural network models in medical X-ray image processing. The confidence of a prediction by a machine learning method can be measured by the uncertainty of the method outputs. A typical way to estimate uncertainty is through Bayesian learning [2], which regards the parameters of methods as random variables and attempts to get the posterior distribution of the parameters during training while marginalizing out the parameters to get the distribution of the prediction during inference. Bayesian learning is well developed in traditional non-neural network machine learning framework [2]

2 RELATED WORKS

In recent years Bayesian learning and estimation of prediction uncertainty have gained more and more attention in neural networks context due to the wide application of deep neural networks in many areas [11, 3, 12, 13, 22, 14, 32, 24, 40, 26, 12, 31, 32].

The authors in [3] introduced a method called “Bayes By Backprop” to learn the posterior distribution on the weights of neural networks and get weight uncertainty. Essentially this method assumes the weights come from a multivariate Gaussian distribution and updates the mean and covariance of the Gaussian instead of the weight samples during training. During inference the network weights are drawn from the learned distribution. This method is mathematically grounded, backpropagation-compatible and can learn the distribution of network weights directly, but it cannot utilize pre-trained model and has to build the corresponding model for every neural network architecture. [13] reformulated dropout in neural networks as approximate Bayesian inference in deep Gaussian processes and thus can estimate uncertainty in neural networks with dropout layers. This method requires dropout layers applied before every weight layer. During inference, the dropout layers with random 0-1s drawn from Bernoulli distribution mask out some weights and only use a subset of the weights learned during training phase to make a prediction. In [22], the authors further proposed that there are two types of uncertainties and they showed the benefits of explicitly formulating these two uncertainties separately. The first type is called *aleatoric* uncertainty (or data uncertainty), which is due to the noise in the data and cannot be eliminated, while the other type is called *epistemic* uncer-

¹ Northeastern University, USA, email: yuminliu@ece.neu.edu

² Philips Research North America, USA, email: claire.zhao@philips.com

³ Philips Research North America, USA, email: jonathan.rubin@philips.com

tainty (or model uncertainty), which accounts for uncertainty in the model and can be eliminated given enough data. The network architectures have to be modified to add extra outputs in order to model these uncertainties. [24] adopted this typing of uncertainty, but modified the formulation of aleatoric and epistemic uncertainty to avoid the requirement of extra outputs.

[26] proposed a method called ‘‘Stochastic Weight Averaging Gaussian (SWAG)’’ to approximate the posterior distribution over the weights of neural networks as a Gaussian distribution by utilizing information in Stochastic Gradient Descent (SGD). This method has an advantage in that it can be applied to almost all existing neural networks without modifying their original architectures and can directly leverage pre-trained models. [34] also decomposed predictive uncertainty in deep learning into two components and modeled them separately. They shown that quantifying the uncertainty can help to improve the predictive performance in medical image super resolution. [39] investigated the relationship between uncertain labels in CheXpert [21] and Chest X-ray14 [37] data sets and the estimated uncertainty for corresponding instances using Bayesian neural network and suggested that utilizing uncertain labels helped prevent over-confident for ambiguous instances.

Despite the above works in Bayesian deep neural network learning and uncertainty quantification, there are few works on evaluating the effects of uncertainty-based evaluation strategies for medical image classification. To the best of our knowledge, we are the first to apply uncertainty quantification strategies for chest X-ray image classification using deep neural networks and evaluate their impacts on performances. The main contributions of this paper are:

- We apply uncertainty quantification to five deep neural network models for chest X-ray image classification and analyze their performances.
- We investigate the impact that uncertainty information has on classification task performance by evaluating subsets of held-out test data ordered via uncertainty quantification.

3 METHOD

In this section, we will introduce the basic ideas of Bayesian Neural Networks and one of its approximations – SWAG [26], which is used in this paper. We also describe the uncertainty quantification method used in this paper.

3.1 Bayesian Neural Network

In the ordinary deterministic neural networks, we get point estimation of the network weights \mathbf{w} which are regarded as fixed values and will not be changed after training. During inference, for each input \mathbf{x}_i we get one deterministic prediction $p(y_i|\mathbf{x}_i) = p(y_i|\mathbf{x}_i, \mathbf{w})$ without getting the uncertainty information.

In the Bayesian neural network settings, in addition to the target prediction, we also want to get the uncertainty for the prediction. To do so we regard the neural network weights as random variables that subject to some form of distribution and try to estimate the posterior distribution of the network weights given the training data during training. We then integrated out the weights and get the distribution over the prediction during inference. From the prediction distribution we can further calculate the prediction output and corresponding uncertainty. More specifically, let $D = \{(\mathbf{X}, \mathbf{Y})\}$ and \mathbf{w} be the training data and weights of a neural network, respectively. The ordinary deterministic neural network methods try to get a

point estimate of \mathbf{w} by either maximum likelihood estimator (MLE) $\mathbf{w}^* = \arg \max_{\mathbf{w}} p(D|\mathbf{w})$ or maximum a posterior (MAP): $\mathbf{w}^* = \arg \max_{\mathbf{w}} p(\mathbf{w}|D)$ where $p(\mathbf{w}|D) = \frac{p(\mathbf{w})p(D|\mathbf{w})}{p(D)} \propto p(\mathbf{w})p(D|\mathbf{w})$. The \mathbf{w}^* are fixed after training and used for inference for the new data. In Bayesian learning, we estimate the posterior distribution $p(\mathbf{w}|D)$ during training and marginalize out \mathbf{w} during the inference to get a probability distribution of the prediction.

$$p(\mathbf{y}|\mathbf{x}, D) = \mathbb{E}_{\mathbf{w} \sim p(\mathbf{w}|D)}[p(\mathbf{y}|\mathbf{x}, \mathbf{w})] = \int p(\mathbf{y}|\mathbf{x}, \mathbf{w})p(\mathbf{w}|D)d\mathbf{w} \quad (1)$$

After getting the $p(\mathbf{y}|\mathbf{x})$, we can calculate the statistical moments of the predicted variable and regard the first and second moment (i.e., mean and variance) as the prediction and uncertainty, respectively.

However, in practice there are two major difficulties. The first one is that $p(D) = \int p(\mathbf{w})p(D|\mathbf{w})d\mathbf{w}$ is usually intractable and thus we cannot get exact $p(\mathbf{w}|D)$. The second lies in that Eq. (1) is also usually intractable for neural networks. One common approach to deal with the first difficulty is to use a simpler form of distribution $q(\mathbf{w}|\theta)$ with hyperparameters θ to approximate $p(\mathbf{w}|D)$ by minimizing the Kullback-Leibler (KL) divergence between $q(\mathbf{w}|\theta)$ and $p(\mathbf{w}|D)$. This turns the problem into an easier optimization problem:

$$\begin{aligned} \theta^* &= \arg \min_{\theta} KL[q(\mathbf{w}|\theta)||p(\mathbf{w}|D)] \\ &= \arg \min_{\theta} \int q(\mathbf{w}|\theta) \log \frac{q(\mathbf{w}|\theta)}{p(\mathbf{w}|D)} d\mathbf{w} \end{aligned} \quad (2)$$

For the second difficulty, the usual approach is to use sampling to estimate Eq. (1), and it becomes

$$p(\mathbf{y}|\mathbf{x}) \approx \mathbb{E}_{\mathbf{w} \sim q(\mathbf{w}|\theta^*)}[p(\mathbf{y}|\mathbf{x}, \mathbf{w})] \approx \frac{1}{T} \sum_{i=1}^T p(\mathbf{y}|\mathbf{x}, \mathbf{w}^{(i)}) \quad (3)$$

where $\mathbf{w}^{(i)} \sim q(\mathbf{w}|\theta^*)$.

People had proposed different methods to approximate the posterior $p(\mathbf{w}|\theta)$ or to get the samples of \mathbf{w} [26, 3, 12, 13].

3.2 Stochastic Weight Averaging Gaussian (SWAG)

The basic idea of SWAG [26] is to regard the weights of the neural networks as random variables and get their statistical moments through training with SGD. Then use these moments to fit a multivariate Gaussian to get the posterior distribution of the weights. After the original training process in which we get the optimal weights, we continue to train the model using the same training data with SGD and get T samples of the weights $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_t, \dots, \mathbf{w}_T$. The mean of those samples is $\bar{\mathbf{w}} = \frac{1}{T} \sum_{t=1}^T \mathbf{w}_t$. The mean of the square is $\overline{\mathbf{w}^2} = \frac{1}{T} \sum_{t=1}^T \mathbf{w}_t^2$ and we define a diagonal matrix $\Sigma_{diag} = diag(\overline{\mathbf{w}^2} - \bar{\mathbf{w}}^2)$ and a deviation matrix $\mathbf{R} = [\mathbf{R}_1, \dots, \mathbf{R}_t, \dots, \mathbf{R}_T]$ whose columns $\mathbf{R}_t = \mathbf{w}_t - \bar{\mathbf{w}}_t$, where $\bar{\mathbf{w}}_t$ is the running average of the first t weights samples $\bar{\mathbf{w}}_t = \frac{1}{t} \sum_{j=1}^t \mathbf{w}_j$. In the original paper, the authors used the last K columns of \mathbf{R} to get the low rank approximation of \mathbf{R} . The K-rank approximation is $\hat{\mathbf{R}} = [\mathbf{R}_{T-K+1}, \dots, \mathbf{R}_T]$. Then the mean and covariance matrix for the fitted Gaussian are given by:

$$\mathbf{w}_{SWA} = \bar{\mathbf{w}} \quad (4)$$

$$\Sigma_{SWA} = \frac{1}{2} \Sigma_{diag} + \frac{1}{2(K-1)} \hat{\mathbf{R}} \hat{\mathbf{R}}^T \quad (5)$$

During inference, for each input (image) \mathbf{x}_i , sample the weights from the Gaussian $\mathbf{w}_s \sim N(\mathbf{w}_{SWA}, \Sigma_{SWA})$ then update the batch norm statistics by performing one epoch of forward pass, and then

the sample prediction is given by $p(\hat{y}_{is}|\mathbf{x}_i) = p(y_i|\mathbf{x}_i, \mathbf{w}_s)$. Repeat the procedure for S times and we get S predictions $\hat{y}_{i1}, \hat{y}_{i2}, \dots, \hat{y}_{is}, \dots, \hat{y}_{iS}$ for the same input \mathbf{x}_i . By using these S predictions we can get the final prediction and uncertainty. For regression problem, the final prediction will be $\hat{y}_i = \frac{1}{S} \sum_{s=1}^S \hat{y}_{is}$.

3.3 Uncertainty Quantification

Some methods had been proposed to quantify the uncertainty in classification [24, 22]. Here we adopt the method proposed by [24] since it does not require extra output and does not need to modify the network architectures.

For a classification problem, suppose there are C classes, denote $\mathbf{p}_s \triangleq [p_{s1}, p_{s2}, \dots, p_{sc}] = p(\mathbf{y}|\mathbf{x}, \theta_s)$, $s \in \{1, 2, \dots, S\}$ as the softmax (or sigmoid in binary case if $C = 2$) output of the neural network for a same repeated input \mathbf{x} for S times, then the predicted ‘‘probability’’ is the average of those S sample outputs $\bar{\mathbf{p}} = \frac{1}{S} \sum_{s=1}^S \mathbf{p}_s$. The predicted class label index is $\hat{\mathbf{y}} = \arg \max_c \bar{\mathbf{p}}$. The aleatoric uncertainty U_a and the epistemic uncertainty U_e are $U_a = \frac{1}{S} \sum_{s=1}^S [\text{diag}(\mathbf{p}_s) - \mathbf{p}_s \mathbf{p}_s^T]$, $U_e = \frac{1}{S} \sum_{s=1}^S (\mathbf{p}_s - \bar{\mathbf{p}})(\mathbf{p}_s - \bar{\mathbf{p}})^T$. The total uncertainty is $U_{total} = U_a + U_e$. For binary classification, the sigmoid output is a scalar and the uncertainty equations are reduced to

$$U_a = \frac{1}{S} \sum_{s=1}^S p_s(1 - p_s) \quad (6)$$

$$U_e = \frac{1}{S} \sum_{s=1}^S (p_s - \bar{p})^2 \quad (7)$$

where $\bar{p} = \frac{1}{S} \sum_{s=1}^S p_s$ and $p_s = p(y = 1|\mathbf{x}, \theta_s) = 1 - p(y = 0|\mathbf{x}, \theta_s)$. The predicted label is:

$$\hat{y} = \begin{cases} 1 & \bar{p} \geq 0.5 \\ 0 & \bar{p} < 0.5 \end{cases} \quad (8)$$

In this way, we can get uncertainties for all the instances.

3.4 Transfer Learning

Transfer learning is a widely used technique to help improve performance for deep neural networks in image classification. Here we can also benefit from transfer learning by loading pre-trained neural network models trained by ImageNet (<http://image-net.org>) dataset. The SWAG method has one advantageous characteristic that it does not require to modify any architecture of the original neural networks and therefore we can fully utilize pre-trained models trained by ImageNet dataset to speed up training process and get better predictions. In the initialization stage, we download the pre-trained model parameters and use them to initialize our models to be trained.

3.5 Procedure

Basically we follow the method in [26] to approximate the Bayesian neural network and the formulas in [24] to quantify uncertainty of the models. The overall algorithm for SWAG and uncertainty quantification is shown in Algorithm 1. We initialize the model with corresponding pre-trained model, and then fine-tune it by training using chest X-ray images and observation labels. After that we perform SWAG algorithm by continuing training using Stochastic Gradient Descent for T epochs and calculate statistics $\bar{\mathbf{w}}$, $\overline{\mathbf{w}^2}$, Σ_{diag} and $\hat{\mathbf{R}}$,

Algorithm 1 Uncertainty Quantification

- 1: **Input:**
 $D = \{(\mathbf{X}, \mathbf{Y})\} / \mathbf{X}_i$: training / evaluating chest X-ray images and corresponding observation labels
 - 2: **Initialization:**
load pre-trained neural network (NN) models by ImageNet
 - 3: **Training:**
Fine-tune NN models using cheXpert dataset
 - 4: **Perform SWAG:**
Continue training with SGD
 - i) train NN models using SGD for some epochs with D
 - ii) save statistics of the weights for those epochs
 - iii) calculate \mathbf{w}_{SWA} and Σ_{SWA} using Eq. 4 and 5
 - vi) fit a Gaussian using \mathbf{w}_{SWA} as mean and Σ_{SWA} as covariance**Prediction**
for s from 1 to S
draw weights $\mathbf{w}_s \sim N(\mathbf{w}_{SWA}|\Sigma_{SWA})$
update batch norm statistics using D
 $p(y_{is}|\mathbf{X}_i) = p(y_{is}|\mathbf{X}_i, \mathbf{w}_s)$
end for
 - 5: **Calculate Outputs:**
 $\bar{p}(y_i|\mathbf{X}_i) = \frac{1}{S} \sum_{s=1}^S p(y_{is}|\mathbf{X}_i)$
Calculate \hat{y}_i , U_a and U_e using Eq. (8), (6) and (7).
 $U_{total} = U_a + U_e$
 - 6: **Return:**
 $\hat{y}_i, U_a, U_e, U_{total}$
-

from which we can get \mathbf{w}_{SWA} and Σ_{SWA} using Eq. 4 and 5. Then we fit a multivariate Gaussian using \mathbf{w}_{SWA} as mean and Σ_{SWA} as covariance and get an approximated distribution for the neural network weights. When doing a prediction, an input chest X-ray image is repeatedly fed into the network for S times, each time with a new set of weights sampled from the Gaussian distribution. The S output probabilities are used to calculate the final predicted label \hat{y}_i and uncertainty $U_{total} = U_a + U_e$. It is worthwhile to note that, after drawing sample weights the network batch norm statistics need to be updated for the models that use batch normalization. It can be achieved by running one epoch with partial or full training set D . More detailed justification for the necessity was given in the original paper [26].

4 DATASET

We perform experiments using the CheXpert data set [21]. CheXpert is a large chest X-ray dataset released by researchers at Stanford University. This dataset consists of 224,316 chest radiographs of 65,240 patients. Each data instance contains a chest X-ray image and a vector label describing the presence of 14 observations (pathologies) as positive, negative, or uncertain. The labels were extracted from radiology reports using natural language processing approaches. For our experiments we focus on 5 observations, namely Cardiomegaly, Edema, Atelectasis, Consolidation and Pleural Effusion. As [21] had pointed out, these 5 observations were selected based on their clinical importance and prevalence in this dataset. In their experiment they also used these 5 observations to evaluate the labeling approaches. A sample image for each observation is shown in Figure 1.

The original dataset consists of training set and validation set and we do not have access to test set. The labels for the training set were generated by automated rule-based labeler which extract informa-

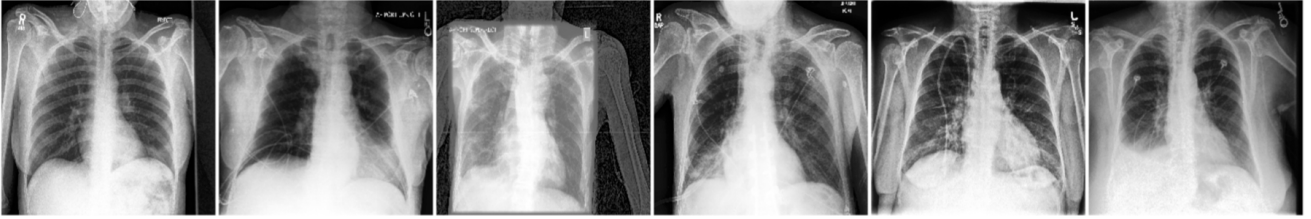


Figure 1: Sample image for each observation. From left to right: no finding (all negative), cardiomegaly, edema, consolidation, atelectasis and pleural effusion

tion from radiology reports. This was done by the Stanford research group who released the dataset. There are three possible values for the label of an instance for a given observation, i.e., 1, 0 and -1 . 1 means the observation is positive (or exists), 0 means negative (or not exists), and -1 means not certain about whether the observation exists. The labels for the validation set were determined by the majority vote from three board-certified radiologists and only contains positive (1) or negative (0) values. The original paper [21] investigated several different ways to deal with the uncertain labels (-1), such as regarding them as positive (1), negative (0), the same with the majority class, or a separate class. They found out that for different observations, the optimal ways to deal with the uncertain labels are different, and they gave the replacement for 5 observations mentioned above. Based on the results from [21] and for simplicity, we replace the uncertain labels with 0 or 1 for different observations.

Specifically, the uncertain labels of cardiomegaly, consolidation and pleural effusion are replaced with 0, while edema and atelectasis with 1. Therefore the problem becomes a multi-label binary image classification problem. The predicted result is a five dimensional vector with element value being 1 or 0, where 1 means that the network predicts existence for the corresponding observation while 0 means the network predicts not existence of the corresponding observation. We follow the official training set / validation set split given by the data set provider. After removing invalid instances, we get a total number of 223,414 instances for training and 234 instances for validation. We first initialize the neural network’s parameters with corresponding downloaded pre-trained model parameters, and then train the neural network using the training set and test their performance on the validation set. We will use the original training set as the training set and original validation set as the evaluation set in our experiments.

In Figure 2 we show the patient statistics of the 5 observations after replacing the uncertain labels in the training set. The prevalence is the ratio of the number of positive instances over the total number of instances. From the figure we can see that all five observations are imbalance as the prevalence being under 50%. Besides, there is a gap in the prevalence for the training and evaluation sets in all observations, which will probably affect the performance of the neural network models.

5 EXPERIMENT

In this section, we perform experiments and present the investigation results of uncertainty quantification and strategy on five different neural network models using *PyTorch* implementation. These neural networks are DenseNet [20] with 121 layers (denote as *DenseNet121*), DenseNet with 201 layers (denote as *DenseNet201*), ResNet [17] with 152 layers (denote as *ResNet152*), ResNeXt [38] with 101 layers (denote as *ResNeXt101*) and Squeeze-and-Excitation network [19] with 154 layers (denote as *SENet154*). ResNet uses

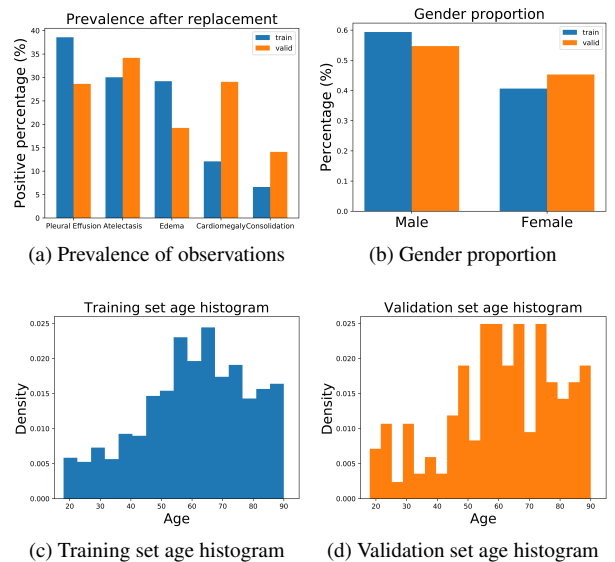


Figure 2: Patient statistics

skip connections to mitigate the gradient vanishment problem and was the winner of ILSVRC 2015 [29] and COCO 2015 (<http://cocodataset.org>) competition. ResNeXt is a variant of ResNet and won the 2nd place in ILSVRC 2016 classification task. DenseNet further utilizes the concept of skip connections by connecting previous layer output to all its subsequent layers and forming “dense” skip connections. DenseNet further alleviates vanishing gradient problem, reduce number of parameters and reuses intermediate features, and is widely used since it was proposed. SENet uses squeeze-and-excitation block to model image channel interdependencies and won the ILSVRC 2017 competition for classification task.

All networks are trained as binary classifiers for multi-label classification instead of training separate models for each class.

The pipeline of the experiment is shown in Figure 4. We use *PyTorch* implementation. The neural network models and pre-trained parameters are from torchvision (except *SENet154* which is from *pretrainedmodels*, <https://github.com/Cadene/pretrained-models.pytorch>).

In our experiment we set the number of sample weights $T = 5$, the number of columns of the deviation matrix $K = 10$ and the number of repeated prediction samples $S = 10$. During training, we use Adam optimizer with weight decay regularizer and *ReduceLROnPlateau* learning rate scheduler. The the initial learning rate is 1×10^{-5} and weight decay coefficient is 0.005. The maximum number of fine-tuning epoch is 50 epochs. The original chest X-ray images are resized and randomly cropped to 256×256 (except for *SENet154* which has a fixed input size 224×224). We stop fine-

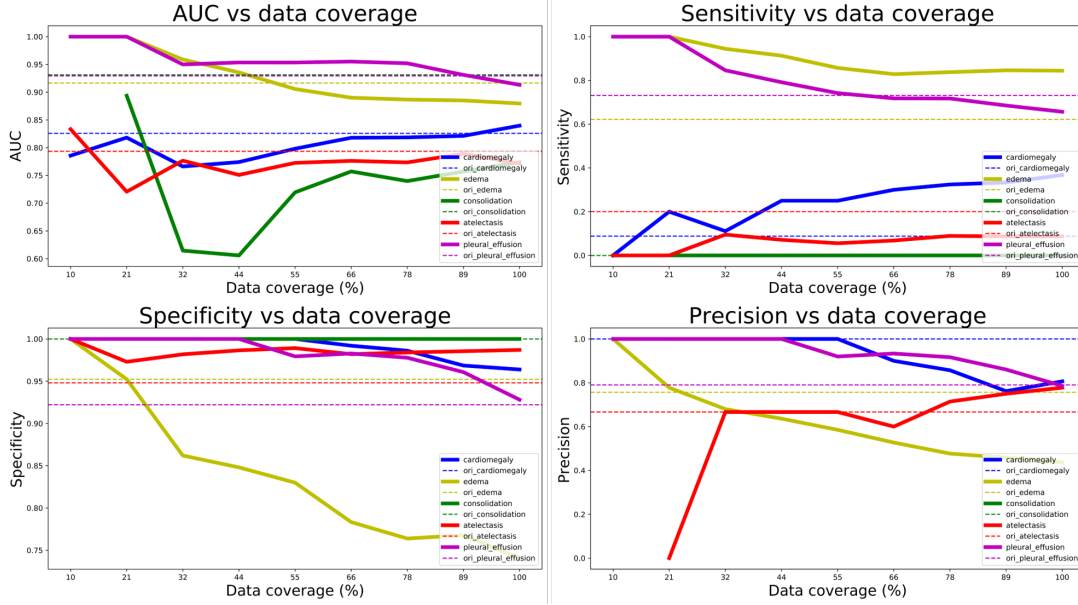


Figure 3: Comparison of performance between original deterministic network and Bayesian neural network with uncertainty strategy. The neural network is DenseNet with 201 layers.

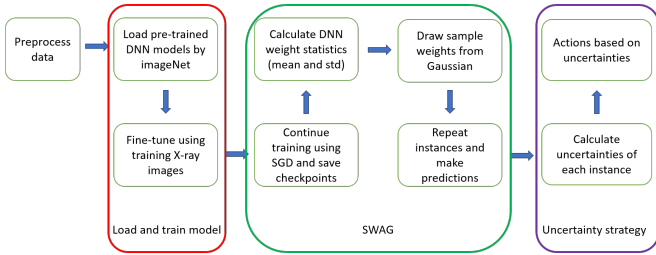


Figure 4: Pipeline of the experiment

tuning the model when the AUC (explained below) does not increase for consecutive 10 epochs and save the model with the best AUC as the optimal trained model.

We use four metrics to evaluate the network classification performance: Area under curve (AUC), Sensitivity, Specificity and Precision. Those metrics are widely used for machine learning and medicine community. The AUC is often used to measure the quality of a classifier and is defined as the area under the Receiver Operating Characteristic (ROC) curve which plots the sensitivity against the false positive rate. The sensitivity (or true positive rate or recall) is defined as the ratio of the number of correctly predicted positive instances over the number of total positive instances. The specificity is defined as the ratio of the number of correctly predicted negative instances over the total number of negative instances. And the precision is defined as the ratio of the number of correctly predicted positive instances over the number of instances that are predicted as positive.

5.1 Without Strategy

First we compare the AUC of the original ordinary deterministic neural networks with the AUC corresponding neural networks after performing SWAG but before applying any uncertainty strategies. The results are shown in Table 1. The ‘‘Average’’ column is the average over all 5 observations. The bold font indicates better performance. For edema and pleural effusion, the original neural network performs

better than SWAG for most of the networks. For cardiomegaly, consolidation and atelectasis, the performances are mixed. This maybe because edema and pleural effusion are harder to detect and more sensitive to network weights perturbation. On the whole the SWAG algorithm does not outperform the original neural network. These might be accountable because SWAG uses a Gaussian to approximate the distribution over the optimal weights and then draws sample weights from the approximated Gaussian distribution, and may deviate from the optimal weights if the approximation is inaccurate. Therefore we need to adopt some strategy to prevent the performance from deterioration. The benefit lies in that we can get the uncertainty estimation for each prediction while keeping similar or even better prediction results.

Table 1: Original AUC vs SWAG AUC

Networks	AUC	Average		Cardiomegaly		Edema		Consolidation		Atelectasis		Pleural Effusion	
		Original	SWAG	Original	SWAG	Original	SWAG	Original	SWAG	Original	SWAG	Original	SWAG
Resnet152	0.8831	0.8786	0.8376	0.8149	0.9123	0.8713	0.8927	0.9234	0.8543	0.8537	0.9184	0.9298	
ResNext101	0.8807	0.8726	0.8013	0.8339	0.9212	0.8748	0.9250	0.9311	0.8246	0.8162	0.9314	0.9071	
Sfnet154	0.8794	0.8695	0.8203	0.8040	0.9195	0.8702	0.9216	0.9187	0.8056	0.8553	0.9301	0.8992	
Densenet121	0.8842	0.8942	0.8436	0.8752	0.9264	0.8940	0.9139	0.9512	0.8153	0.8489	0.9220	0.9016	
Densenet201	0.8793	0.8356	0.8259	0.8397	0.9165	0.8796	0.9313	0.7739	0.7936	0.7714	0.9294	0.9132	

5.2 With Coverage Strategy

Next we utilize the uncertainty quantification information to determine if the performances can be improved. One strategy is to sort instances according to uncertainty in an ascending order, and then take those instances with less uncertainty into consideration and discard the rest. In clinical practice, the discarded instances could be flagged for further evaluation by a physician.

Ideally we would expect a decreasing trend for the metrics when data coverage increase as shown in Figure 6. The horizontal axis ‘‘Data coverage’’ is the percentage of instances being considered. For example, a data coverage of 20% means that only the top twenty percent of the least uncertain (or the most confident) instances are taken into consideration and the rest are discarded.

Figure 3 shows the comparison of performances with regard to the four metrics (AUC, sensitivity, specificity and precision) between

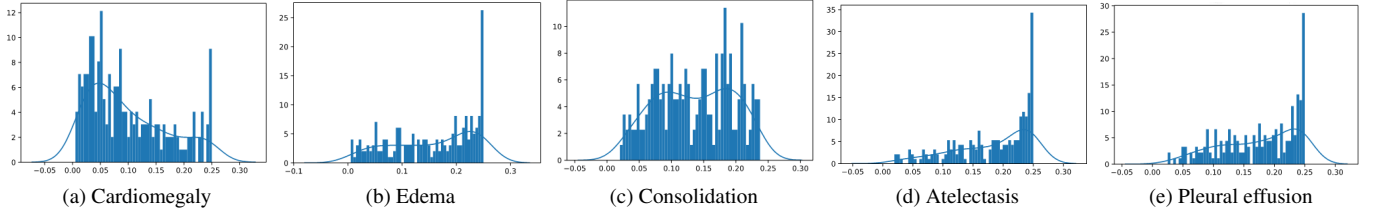


Figure 5: Estimated total uncertainty (aleatoric + epistemic) histogram for each observation

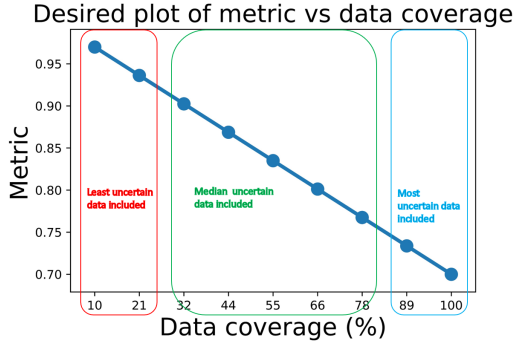


Figure 6: Expected ideal performance. The metric decreases as data coverage increases.

the original deterministic networks and Bayesian neural networks with uncertainty strategy. The solid lines are the Bayesian neural network with uncertainty strategy, while the dashed lines are the original ordinary deterministic networks without any uncertainty strategy. Different colors represent different observations.

From Figure 3 we can see that for edema and pleural effusion, the AUC decreases as the coverage increases, and are above the corresponding original AUC until around 45% and 90% coverage, respectively. This means that applying the uncertainty strategy can improve AUC for these two observations. The highest AUC gain can be 8% and 6% for edema and pleural effusion, respectively. We also observe similar trend in sensitivity, specificity and precision for both edema and pleural effusion. Three observations (cardiomegaly, atelectasis and consolidation) have low sensitivity as most of the predictions are negative. On the contrary the specificity is high.

The highest gains for applying the uncertainty strategy are shown in the Table 2. The effect of the uncertainty strategy over the five ob-

Table 2: Performance gain for edema and pleural effusion. The values are the absolute and relative gains

Gain (% Gain)	AUC	Sensitivity	Specificity	Precision
Edema	0.0835(9.11%)	0.3778(60.71%)	0.0476(5.00%)	0.2432(32.14%)
Pleural effusion	0.0706(7.60%)	0.2687(36.73%)	0.0778(8.44%)	0.2097(26.53%)

servations with the model DenseNet201 can be summarized as in the Table 3. The symbols \checkmark , \times , \circ and $-$ represents *helpful*, *not helpful*, *mixed behavior* and *missing value*, respectively. For edema and pleural effusion, applying uncertainty strategy is beneficial for improving all four metrics. However, for other observations, it does not show benefits or only limited benefits for some metrics. The reason why it show varied behavior may be interesting and needs further investigation. Similarly, we summarize the effect of applying uncertainty strategy for different neural network architectures and the results are shown in Table 4 to Table 7. From the tables we can see that applying

Table 3: Effect of uncertainty strategy for DenseNet201.

Densenet201	AUC	Sens.	Spec.	Prec.
Cardiomegaly	\times	\times	\checkmark	\circ
Edema	\checkmark	\checkmark	\checkmark	\checkmark
Consolidation	\times	\times	\circ	$-$
Atelectasis	\circ	\times	\circ	\times
Pleural effusion	\checkmark	\checkmark	\checkmark	\checkmark

\checkmark : helpful; \times : not helpful; \circ : mixed behavior; $-$: missing value

uncertainty strategy will help to improve some performance metrics for all four neural network models.

Table 4: Effect of uncertainty strategy for different networks

ResNet152	AUC	Sens.	Spec.	Prec.
Cardiomegaly	\checkmark	\times	$-$	$-$
Edema	\times	\times	\checkmark	\checkmark
Consolidation	\times	\times	$-$	$-$
Atelectasis	\times	\times	\checkmark	\checkmark
Pleural effusion	\checkmark	\checkmark	\checkmark	\checkmark

\checkmark : helpful; \times : not helpful; \circ : mixed behavior; $-$: missing value

Table 5: Effect of uncertainty strategy for different networks

SENet154	AUC	Sens.	Spec.	Prec.
Cardiomegaly	\checkmark	$-$	$-$	$-$
Edema	\times	\times	\checkmark	\times
Consolidation	\checkmark	$-$	$-$	$-$
Atelectasis	\circ	$-$	\times	$-$
Pleural effusion	\checkmark	\times	\checkmark	\checkmark

\checkmark : helpful; \times : not helpful; \circ : mixed behavior; $-$: missing value

Despite that for some observations (e.g., pleural effusion), several metrics performance benefit a lot from applying the uncertainty strategy, we should also notice that the strategy does not help to improve performance for some other observations with regard to these metrics, and in some cases even degrade the performance. The reasons behind might be varied and needs more investigation. For example, this may be that the neural network weight distribution approximated by the SWAG algorithm does not capture the true distribution, or even the uncertainty quantification formulas are inappropriate.

5.3 With Absolute Threshold Strategy

We also plot the total uncertainty distribution for each observation, as shown in Figure 5. From the figure we can see that for cardiomegaly, the estimated uncertainty tends to be smaller, while for edema, atelectasis and pleural effusion, the proportion of larger estimated uncertainty is higher. Consolidation has a relatively even distribution for estimated uncertainty. This suggest that edema, atelectasis and

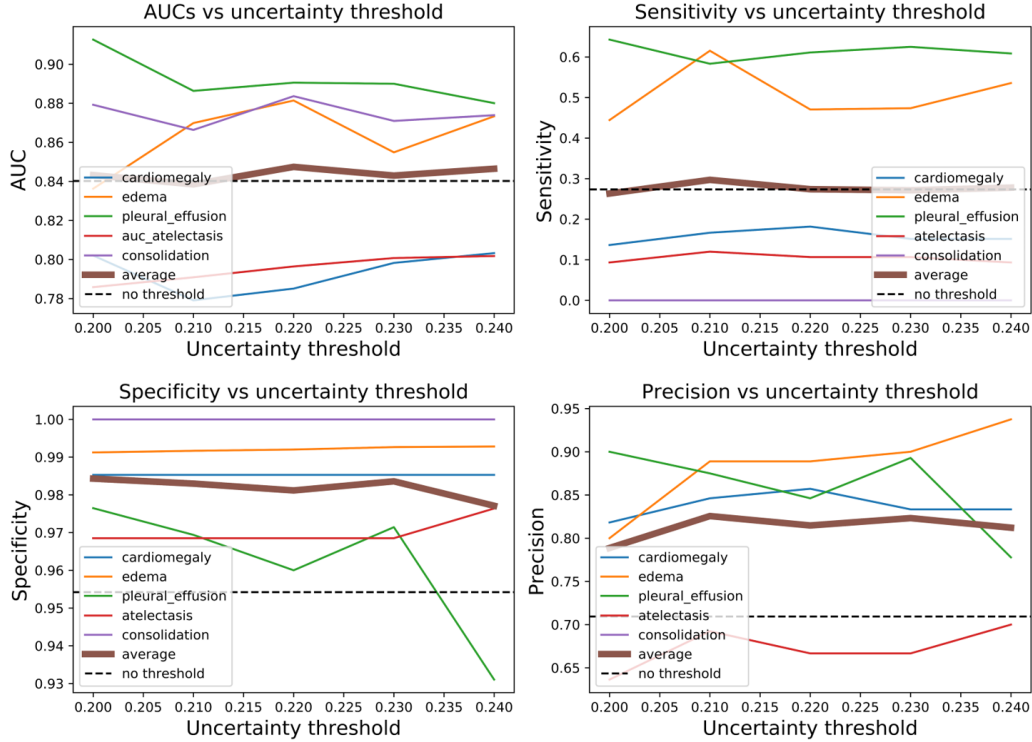


Figure 7: Comparison of performance between original deterministic network and Bayesian neural network with uncertainty threshold.

Table 6: Effect of uncertainty strategy for different networks

ResNext101	AUC	Sens.	Spec.	Prec.
Cardiomegaly	✓	×	×	×
Edema	×	×	✓	✓
Consolidation	✓	×	✓	-
Atelectasis	○	✓	×	×
Pleural effusion	✓	✓	✓	✓

✓: helpful; ×: not helpful; ○: mixed behavior; -: missing value

pleural effusion are more prone to be affected by setting an uncertainty threshold. Combining this finding with the results in Table 2, we set thresholds for both edema and pleural effusion to check the influence on metric performance. We only consider the instances whose estimated uncertainty is smaller than the threshold to compute the performance metrics. We vary the threshold from 0.2 to 0.24 by a step of 0.01 and the results are shown in Figure 7. The black dashed line is the average metric values of the original deterministic neural network, while the solid color thin lines are metric values for each observation, and the thick brown line is the average metric values of all five observation after applying threshold only to edema and pleural effusion. Comparing the thick brown line with the dash black line, we can see that the average specificity and precision have been improved while the average AUC and sensitivity roughly keep the same. This means that applying uncertainty threshold to edema and pleural effusion is beneficial.

6 CONCLUSION

Table 7: Effect of uncertainty strategy for different networks

DenseNet121	AUC	Sens.	Spec.	Prec.
Cardiomegaly	×	×	-	○
Edema	×	×	✓	✓
Consolidation	✓	-	-	-
Atelectasis	✓	○	✓	✓
Pleural effusion	✓	✓	✓	✓

✓: helpful; ×: not helpful; ○: mixed behavior; -: missing value

In this paper we investigate uncertainty quantification in medical image classification using Bayesian deep neural networks. We train five different deep neural network models on the CheXpert X-ray image data for five clinical observations and quantify the model uncertainty. Then we analyze the performance of the network for situations with and without applying uncertainty strategy. The results show that the uncertainty quantification and strategy improve several performance metrics for some observations. This suggests that uncertainty quantification is helpful in medical image classification using neural networks. However, the results also show that in some cases the strategy is not helpful, or can even deteriorate the performance. Further analysis may be needed to examine this phenomenon.

REFERENCES

- [1] Yaniv Bar, Idit Diamant, Lior Wolf, Sivan Lieberman, Eli Konen, and Hayit Greenspan, 'Chest pathology detection using deep learning with non-medical training', in *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, pp. 294–297. IEEE, (2015).
- [2] Christopher M Bishop, *Pattern recognition and machine learning*, springer, 2006.
- [3] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra, 'Weight uncertainty in neural networks', *arXiv preprint arXiv:1505.05424*, (2015).
- [4] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Praseon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al., 'End to end learning for self-driving cars', *arXiv preprint arXiv:1604.07316*, (2016).
- [5] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, 'Listen, attend and spell: A neural network for large vocabulary conversational speech recognition', in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964. IEEE, (2016).
- [6] Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio, 'A character-level decoder without explicit segmentation for neural machine translation', *arXiv preprint arXiv:1603.06147*, (2016).
- [7] Marleen de Bruijne. Machine learning approaches in medical image analysis: From detection to diagnosis, 2016.
- [8] Chao Dong, Chen Change Loy, and Xiaoou Tang, 'Accelerating the super-resolution convolutional neural network', in *European conference on computer vision*, pp. 391–407. Springer, (2016).
- [9] Meherwar Fatima and Maruf Pasha, 'Survey of machine learning algorithms for disease diagnostic', *Journal of Intelligent Learning Systems and Applications*, **9**(01), 1, (2017).
- [10] Konstantinos P Ferentinis, 'Deep learning models for plant disease detection and diagnosis', *Computers and Electronics in Agriculture*, **145**, 311–318, (2018).
- [11] Yarin Gal, *Uncertainty in deep learning*, Ph.D. dissertation, PhD thesis, University of Cambridge, 2016.
- [12] Yarin Gal and Zoubin Ghahramani, 'Bayesian convolutional neural networks with bernoulli approximate variational inference', *arXiv preprint arXiv:1506.02158*, (2015).
- [13] Yarin Gal and Zoubin Ghahramani, 'Dropout as a bayesian approximation: Representing model uncertainty in deep learning', in *international conference on machine learning*, pp. 1050–1059, (2016).
- [14] Yarin Gal, Riashat Islam, and Zoubin Ghahramani, 'Deep bayesian active learning with image data', in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1183–1192. JMLR. org, (2017).
- [15] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, 'Speech recognition with deep recurrent neural networks', in *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649. IEEE, (2013).
- [16] Hayit Greenspan, Bram Van Ginneken, and Ronald M Summers, 'Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique', *IEEE Transactions on Medical Imaging*, **35**(5), 1153–1159, (2016).
- [17] 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*, pp. 770–778, (2016).
- [18] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, 'Mobilenets: Efficient convolutional neural networks for mobile vision applications', *arXiv preprint arXiv:1704.04861*, (2017).
- [19] Jie Hu, Li Shen, and Gang Sun, 'Squeeze-and-excitation networks', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, (2018).
- [20] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, 'Densely connected convolutional networks', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, (2017).
- [21] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silvana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghighi, Robyn Ball, Katie Shpanskaya, et al., 'Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison', *arXiv preprint arXiv:1901.07031*, (2019).
- [22] Alex Kendall and Yarin Gal, 'What uncertainties do we need in bayesian deep learning for computer vision?', in *Advances in neural information processing systems*, pp. 5574–5584, (2017).
- [23] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher, 'Ask me anything: Dynamic memory networks for natural language processing', in *International conference on machine learning*, pp. 1378–1387, (2016).
- [24] Yongchan Kwon, Joong-Ho Won, Beom Joon Kim, and Myunghee Cho Paik, 'Uncertainty quantification using bayesian neural networks in classification: Application to ischemic stroke lesion segmentation', *Medical Imaging with Deep Learning, 2018*, (2018).
- [25] Antonio Lavecchia, 'Machine-learning approaches in drug discovery: methods and applications', *Drug discovery today*, **20**(3), 318–331, (2015).
- [26] Wesley Maddox, Timur Garipov, Pavel Izmailov, Dmitry Vetrov, and Andrew Gordon Wilson, 'A simple baseline for bayesian uncertainty in deep learning', *arXiv preprint arXiv:1902.02476*, (2019).
- [27] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al., 'Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning', *arXiv preprint arXiv:1711.05225*, (2017).
- [28] Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib, 'Deep learning for medical image processing: Overview, challenges and the future', in *Classification in BioApps*, 323–350, Springer, (2018).
- [29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, 'ImageNet Large Scale Visual Recognition Challenge', *International Journal of Computer Vision (IJCV)*, **115**(3), 211–252, (2015).
- [30] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik, 'Inverse molecular design using machine learning: Generative models for matter engineering', *Science*, **361**(6400), 360–365, (2018).
- [31] Peter Schulam and Suchi Saria, 'Can you trust this prediction? auditing pointwise reliability after learning', in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1022–1031, (2019).
- [32] Murat Sensoy, Lance Kaplan, and Melih Kandemir, 'Evidential deep learning to quantify classification uncertainty', in *Advances in Neural Information Processing Systems*, pp. 3179–3189, (2018).
- [33] Kenji Suzuki, 'Overview of deep learning in medical imaging', *Radiological physics and technology*, **10**(3), 257–273, (2017).
- [34] Ryutaro Tanno, Daniel Worrall, Enrico Kaden, Aurobrata Ghosh, Francesco Grussu, Alberto Bizzi, Stamatios N Sotiropoulos, Antonio Criminisi, and Daniel C Alexander, 'Uncertainty quantification in deep learning for safer neuroimage enhancement', *arXiv preprint arXiv:1907.13418*, (2019).
- [35] Demetri Terzopoulos et al., 'Semi-supervised multi-task learning with chest x-ray images', in *International Workshop on Machine Learning in Medical Imaging*, pp. 151–159. Springer, (2019).
- [36] Huanqing Wang, Peter Xiaoping Liu, Shuai Li, and Ding Wang, 'Adaptive neural output-feedback control for a class of nonlinear triangular nonlinear systems with unmodeled dynamics', *IEEE transactions on neural networks and learning systems*, **29**(8), 3658–3668, (2017).
- [37] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers, 'Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097–2106, (2017).
- [38] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He, 'Aggregated residual transformations for deep neural networks', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, (2017).
- [39] Hao-Yu Yang, Junling Yang, Yue Pan, Kunlin Cao, Qi Song, Feng Gao, and Youbing Yin, 'Learn to be uncertain: Leveraging uncertain labels in chest x-rays with bayesian neural networks', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 5–8, (2019).
- [40] Jiayu Yao, Weiwei Pan, Soumya Ghosh, and Finale Doshi-Velez, 'Quality of uncertainty quantification for bayesian neural network inference', *arXiv preprint arXiv:1906.09686*, (2019).
- [41] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria, 'Recent trends in deep learning based natural language processing', *IEEE Computational Intelligence Magazine*, **13**(3), 55–75, (2018).