Imperial College London

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Combining Monte Carlo Dropout and Early Exit Ensembling

Author:
Liam Castelli

Supervisor:
Wayne Luk

Submitted in partial fulfillment of the requirements for the MSc degree in MSc Artificial Intelligence of Imperial College London

September 2022

Abstract

In this report, a novel method to improve uncertainty quantification in convolutional neural networks is tested. Monte Carlo dropout is combined with early exit ensembling, and is shown to improve both accuracy and uncertainty quantification across three different models and on Cifar100 and a medical chest x-ray dataset, ChestX-ray 14. On average, the expected calibration error was reduced by 50%, 17.7% and 16.7% for the MSDNet, VGG-19 and ResNet-18 on Cifar100 over the best tested methods from the literature. For the chest x-ray dataset, the combination models can match the best methods from the literature, while needing 55% fewer FLOPs. However, while the combination approach is found to outperform alternatives, it requires significant hyperparameter tuning to achieve optimal results which may limit its practical applicability in some domains.

Acknowledgments

Thank you to my friends and family. I would also like to acknowledge the support I received from my Supervisor Professor Wayne Luk and PhD student Hongxiang Fan over the course of the project.

Contents

Chapter 1

Introduction

1.1 Motivation

In 2021, a survey of physicians in America found that over half of them knew of a physician who had considered or attempted suicide (1). About 60% of respondents experienced feelings of burnout (1). A similar study of medical staff in Wuhan in 2020 found that nearly 63% of staff had experienced at least a mild mental health disturbance (2). The COVID-19 pandemic has exacerbated the already high demand on medical services worldwide, piling on additional work and stress on medical professionals.

Even before the pandemic, resident physicians would work shifts of over 24 consecutive hours at a time (3). This practice continued throughout the pandemic, despite growing evidence that the resulting sleep deprivation and disruption of the circadian could lead to errors, hinder performance and have long term adverse health effects (4; 5; 6). Shorter shifts instead were associated with improved well-being of the medical professional while also leading to a better level of patient care (4; 3).

Longer shifts are typically adopted to improve ratios between staff and patients (7). However, there continues to be a shortage of physicians, doctors and nurses which may be leading to the widespread adoption of longer shifts, despite the dangers (8; 9). It is estimated that the existing shortage of medical professionals will continue to get worse in the next decade in the United States (9). This will primarily be driven by population growth and the aging of the population, which will lead to increasing demands on medical services. The usage of artificial intelligence in medicine has the potential to significantly reduce this demand (9).

Machine learning algorithms have demonstrated themselves to be incredibly versatile and powerful, achieving human-level performance in complex tasks in which computers generally struggled (10; 11; 12). In the field of computer vision, the development of convolutional neural networks and vision transformers has been vital to the massive success that these algorithms have achieved, allowing them to not only classify images but identify and locate individual objects in the image (13; 14; 12).

These high performance models have also been applied to medical imaging, to identify and locate pathologies in X-rays, MRIs and CT-Scans (15; 16; 17). Medical

images are often significantly more difficult to analyze than natural images, as they can differ greatly from one another and be characterized by minor textural variations (18). However, these algorithms have continued to find success, outperforming medical specialists in a variety of disciplines (15; 19; 20).

A chest x-ray is one of the most common imaging procedures performed, as they are relatively cheap and can detect a variety of diseases (21) (22). However, interpreting these images requires extensive training and experience, and to become a radiologist typically takes over 10 years (21). The development of an effective automated system driven by the recent improvements in computer vision could significantly reduce the burden on radiology departments across the world (22).

Deep learning in chest radiography has been particularly successful, achieving results which are comparable to trained radiologists (15; 23). The high performance of these algorithms has been made possible by the multiple large datasets which are available, containing hundreds of thousands of x-rays and associated labels (22).

One of the concerns with utilizing neural networks (NNs) for medical tasks is their inability to accurately quantify the uncertainty in their predictions, which makes it difficult to ascertain the reliability of the model (24). Standard NNs have a tendency to overestimate the quality and accuracy of their prediction in comparison to other models (25). It has also been shown that even small shifts in the dataset can cause the model's estimate of the uncertainty to deteriorate (26).

While alternative probabilistic models like Bayesian NNs and deep Gaussian processes have been shown to effectively model the uncertainty of their outputs, they are often prohibitively expensive to compute or ineffective for larger data (27). Gal & Ghahramani found that Monte Carlo (MC) dropout could be used to approximate a standard NN to a Bayesian NN by introducing dropout layers and requiring multiple pass throughs of the network at inference time (28). This can help lead to better uncertainty quantification. While this slows down the model, sparsity induced by the dropout layers in the network can be leveraged to significantly speed up calculations, limiting the negative effects (25). Furthermore, the multiple passes can be parallelized relatively easily across multiple GPUs (27).

Another potential method to improve the uncertainty quantification is to introduce multiple exits. Ensembles are formed through taking the results of multiple independent networks, and averaging them to give a single result (29). This is often expensive, and a recent study has found success by instead utilizing a multi-exit architecture, treating each exit as its own individual network (30). The resulting ensemble has been shown to match the uncertainty quantification capabilities of the deep ensembles while costing significantly less (30). Furthermore, these types of networks can employ a confidence based exiting scheme, allowing the network to output a result for a given input as soon as a minimum confidence threshold on an early exit's prediction is reached. This has been shown to reduce the amount of computation required and lead to improved performance (31).

1.2 Contributions

In this report, the viability of combining MC dropout and multi-exit ensembling is explored as a method of improving the uncertainty quantification effects of the two approaches individually. This method is tested with an MSDNet, a ResNet-18 and a VGG-19 on Cifar100, a standard benchmarking image classification dataset, and then the VGG-19 is tested on ChestX-ray 14, one of the largest publicly available labeled chest x-ray datasets (32; 33).
There are four novel features introduced in this report:

(a) To the best of my knowledge, this is the first attempt at using Monte Carlo dropout and a multi-exit architecture simultaneously, on any model or any dataset.

(b) While the MSDNet has been employed on Cifar100, the aim has been to improve accuracy (34; 35). This is the first attempt to understand how well it quantifies uncertainty on this dataset, particularly when combined with other approaches like ensembling or Monte Carlo dropout (30; 27).

(c) Confidence based exiting was developed to reduce model overthinking (31). In this report, it is shown for the first time that it can also be used as a tool to improve the uncertainty quantification of multi-exit architectures.

(d) Uncertainty quantification has been the focus of multiple previous works in the medical domain, however this is the first model developed specifically to improve uncertainty quantification on the ChestX-ray 14 dataset (36; 37; 38).

The challenges this approach overcomes are detailed below:

C1: Ensembling is the most effective uncertainty quantification method, but can be prohibitively expensive (30). Multi-exit ensembling has been shown to be a cheaper alternative but can be limited by the performance of the individual exits (30).

Solution: Using MC dropout in the exit branches can improve the uncertainty quantification of the later exits by MSDNet: 91.1%, VGG-19: 75.6%, ResNet-18: 50.4%, while confidence based exiting can be employed to only use the best results of the weaker early layers. This can lead to models which improve the uncertainty quantification of the multi-exit ensembling approach by up to 85.0%, 39.0% and 12.5% for MSDNet, VGG-19 and ResNet respectively.

C2: While MC dropout is a simple and widely used method, the additional passes necessary for the technique can increase the cost of a single prediction many-fold (27; 39).

Solution: The additional cost is mitigated primarily through the use of confidence based exiting, which allows the model to exit early and avoid the additional cost of computation incurred by a full pass, and optimal dropout placement (31). By placing the dropout layers as far into the model as possible, the amount of computation which needs to be repeated in each pass is significantly reduced. This can lead to models which are better than the baseline while costing 59%, 47% and 67% respectively of the original MSDNet, VGG and ResNet.

C3: MC dropout has been shown to improve model calibration, however the introduction of significant amounts of dropout can over-regularize the network and cause

it to lose significant predictive power (40). However, without a sufficient amount of dropout the benefits of MC Dropout may be minimal on the calibration of the model.

Solution: A grid search is used to test between 12-16 dropout types for each model in the combined approach, using 3-4 different dropout locations with 4 different dropout rates. By testing a significant number of dropout configurations, the tradeoff between accuracy and ECE for each model and dataset can be better understood. This allows for the creation of models which are either more accurate, better quantify uncertainty or both than the alternative methods tested.

C4: A variety of machine learning models exist, and often domains will have domain specific architectures which are predominantly used in that particular industry. Many uncertainty quantification methods are not model or data agnostic, hence limiting their practical usefulness.

Solution: The effectiveness of the model was demonstrated on a variety of different models, including ones that are not specifically designed for multi-exit. Furthermore, the method was shown to outperform alternatives on both a standard benchmarking dataset and a chest x-ray dataset, indicating that it likely has strong generalizability and practical usefulness.

Chapter 2

Literature Review

2.1 Uncertainty in Deep Learning

Understanding how certain a model is in its prediction has become a major focus of research due to its implications in domains such as medicine and self-driving cars. If a doctor cannot trust the diagnosis made by a machine they may prefer not to use it, even if it has a high accuracy. Hence, there has been a push to develop models which are better at quantifying their own uncertainty. Abdar et al. provide a comprehensive study of a large selection of the over 2500 papers published in the last 10 years in the field (24).

Predictive uncertainty can generally be separated into uncertainty in the data, aleatoric, and uncertainty in the model, epistemic (24). While there have been attempts to quantify both types of uncertainty, aleatoric uncertainty is from the data itself and hence cannot be minimized (24). Furthermore, this type of uncertainty tends to be well estimated by machine learning, due to their use of maximum likelihood (24). Improvements in estimating predictive uncertainty largely come from better understanding the epistemic uncertainty.

Consider a supervised, multi-class classification problem, where some input x is given and the associated label y is predicted (24; 30). Each input only has a single assigned class, so y can only be one of the C total classes in the problem. A neural network is formed through the repeated use of $\sigma(Wx + b)$ at each layer, where σ is a non-linear function and x, W and b are the input, weight and bias at each layer (24). The normalized network's prediction of the probability, p_i, that the input is associated with class i can be obtained via using a softmax function as the final layer non-linear function (24):

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \tag{2.1}$$

z_i is the i^{th} element in the output of the network before the softmax layer (41). Ideally, the probabilities p obtained from the softmax function should be calibrated to correspond to the true probabilities of correctness, such that of N samples with confidence p, $N * p$ are correctly classified (42). However, it has been shown that modern networks are not well calibrated, and hence the obtained probabilities tend not to be indicative of how likely they are to be correct (42).

2.1.1 Bayesian Inference

Alternative approaches instead attempt to quantify the uncertainty faced by the models explicitly. In particular, the epistemic uncertainty can be well quantified through letting model parameters correspond to probabilistic distributions (24).

For a given model defined by $\hat{y} = f_w(x)$, where x are the model inputs, $\hat{y}$ are the model outputs, w are the parameters of the model and y is the correct associated label to the input, the model likelihood is given by $p(y|x, w)$ (24).

Following from (24), consider a model that has been trained on data x, y and is now being tested on an unseen example x^*, y^*. Inference can be performed to obtain the model likelihood of y^* given all the other data by marginalizing out the parameters w through:

$$p(y^*|x^*, x, y) = \int p(y^*|x^*, w)p(w|x, y)dw \tag{2.2}$$

The posterior function $p(w|x, y)$ can then be expressed in terms of the model likelihood of the train data through Bayes' theorem:

$$p(w|x, y) = \frac{p(y|x, w)p(w)}{p(y|x)} \tag{2.3}$$

where $p(w)$ is the prior distribution over the parameters and $p(y|x)$ is a marginal likelihood (24). Hypothetically, all three of the terms in equation 2.3 can be calculated to obtain the exact probability distribution for inference. However, for most practical implementations like a neural network, the equation used to calculate the marginal likelihood $p(y|x)$, given below, contains an intractable integral:

$$p(y|x) = \int p(y|x, w)p(w)dw \tag{2.4}$$

The integral in Equation 2.4 requires calculating the model likelihood and prior for every possible set of parameters in the parameter space. For a neural network this space is infinite, as each element of the weight and bias matrix can be any real number, making the integral impossible to calculate analytically.

One of the more common approaches to overcoming this obstacle is through variational inference. In variational inference, the difficult posterior distribution, $p(w|x, y)$, is approximated by an easier to solve distribution $q_\theta(w) \in Q$, where Q is a predefined family of distributions over the parameters and θ are the variational parameters which control the shape of a given distribution (43). The distribution which best approximates the posterior, $q^*_\theta(w)$, can be found through minimizing the Kullback-Leibler (KL) divergence, a measure of the distance between two distributions, with respect to θ (43). However, an analytical solution is again impossible as $KL(q_\theta(w)||p(w|x, y))$ can be shown to depend on the incalculable $p(y|x)$ (43).

Instead, an almost equivalent rearrangement of the above KL term is used (24; 43). This expression can also be shown to be the evidence lower bound (ELBO):

$$log(p(y|x)) \geq \mathbb{E}[log(p(y|x, w))] - KL(q_\theta(w)||p(w)) \tag{2.5}$$

It is noted that the expectation is with respect to $q_\theta(w)$, and hence maximizing the right hand side of Equation 2.5 forces $q_\theta(w)$ to be both an accurate and similar distribution to that of p (24). Through the above optimization, an approximation of the model likelihood for the new data can be found. The explicit treatment of the probabilities allows for a better quantification of the underlying model uncertainty.

2.1.2 Bayesian Machine Learning

The above strategy has been frequently used with Bayesian NNs, a type of model which places a prior distribution over the neural network's weights (27; 24). A common choice is to use a Gaussian prior, such that the weight matrices W_i at each layer i are distributed according to (28):

$$W_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{2.6}$$

However, while Bayesian NNs allow for a more mathematically rigorous treatment of the uncertainty, they are often significantly more expensive and have been shown to be less effective than other methods (27; 24).

A cheaper, more scalable alternative was developed by Gal & Ghahramani using Monte Carlo dropout (27). Dropout is a common regularization technique that randomly sets elements of a given input to zero according to some probability (44). MC dropout refers to the practice of implementing a network with dropout, but rather than turning off the dropout layers and scaling the outputs at inference time, the dropout is left on. The predictions obtained from passing the same input through the model multiple times are then averaged to produce the final prediction of the network. These multiple pass throughs of the network can be understood as a form of Monte Carlo integration, a form of numerical integration which can be used to help solve some of the intractable integrals mentioned above (27).

In (28), it is shown that a standard neural network with MC dropout is mathematically equivalent to a type of Bayesian NN with variational inference. This is the result of applying variational inference to approximate the Gaussian prior used in Equation 2.6 with a Bernoulli distribution (28). Gal & Ghahramani define the weights as:

$$W_i = \theta_i \cdot diag([z_{i,j}]_{j=1}^{K_i}) \tag{2.7}$$

where $z_{i,j}$ is the random variable associated with the respective neuron j in a fully connected layer i and θ_i are the variational parameters. These variables are distributed according to:

$$z_{i,j} \sim Bernoulli(p_i) \tag{2.8}$$

After Monte Carlo integration over the Bernoulli distributed random variables, a model which is equivalent to a network with Monte Carlo dropout is returned. See the appendix of (28) for the full proof.

MC dropout is one of the most prominent uncertainty quantification methods due to its ease of use and applicability (24). Any neural network can be made Bayesian via the addition of dropout layers. It has also been found in multiple studies to be one of the most effective uncertainty quantifiers when compared to other methods in the field (24).

Follow up studies have illustrated some of the key limitations of the approach, particularly that MC dropout may be biased, severely underestimate model uncertainty and massively increases the cost of a prediction (45; 46; 39). A variety of modifications to the standard MC dropout procedure have been suggested to improve performance, however the standard approach is still widely used (45; 47; 24)

The introduction of dropout layers can also significantly hamper the network through over-regularization. Kendall et al. tried a variety of different dropout locations, and found that implementing dropout as in (27; 28) led to poor results. Instead, using fewer but more strategically placed dropout layers allowed them to achieve better performance (40).

2.1.3 Measuring Uncertainty Quantification

Effectively assessing the uncertainty quantification of a model is an open problem and there exist many metrics each with their own strengths (42). A recently developed metric is the Expected Calibration Error (ECE), which is a measure of the difference between the confidence of a prediction and the accuracy of that prediction (48). While other metrics may more comprehensively assess the similarity of two distributions, this error is particularly valuable in machine learning contexts where decisions are often made solely using the confidence of the prediction. This metric is defined as:

$$\mathbb{E}_{\hat{P}}[|\mathbb{P}(\hat{Y} = Y|\hat{P} = p) - p|] \tag{2.9}$$

The above expectation is intractable, and is typically estimated via binning predictions and summing the average difference in each bin (49; 48). However, there is no generally applicable method for picking the parameters of the histogram, and it has been shown that the choice of where to bin and number of bins can lead to noisy and inaccurate estimates of the ECE (49).

An alternative method for estimating the ECE has been proposed which relies on kernel density estimation (49). This method was shown to outperform the histogram-based approach, regardless of the binning scheme used. In this report, the ECE is used as the primary metric for measuring the quality of the uncertainty quantification and is calculated using the kernel density estimation method presented in (49).

2.2 Ensembles

Another approach to achieve better uncertainty modeling has been the use of ensembling, combining the results of multiple individual networks to give a single final prediction. This method has seen success at both improving performance and reducing uncertainty, with optimal results being achieved by the largest ensembles (29; 26).

A theoretical justification for these results is shown by Brown et al. by expanding

on the ambiguity decomposition, which for a single point with value d is given by:

$$(f_{ens} - d)^2 = \sum_i w_i (f_i - d)^2 - \sum_i (f_i - f_{ens})^2 \tag{2.10}$$

$$f_{ens} = \sum_i w_i f_i, \tag{2.11}$$

where f_i is the estimate of the i^{th} model in the ensemble, and w_i are the normalized weightings assigned to each estimate (50). The key result from the above equation is that the ensemble model must be at least as good as the average of its components and can be better when the ambiguity term $\sum_i (f_i - f_{ens})^2$, a measure of the variability between the individual estimates f_i, is large.

Brown et al. relate the ambiguity term to the bias-variance decomposition, extending the above discussion to include results for unseen data (50). The above two terms can be understood in terms of the average bias, variance and covariance of the individual models in the ensemble according to:

$$\mathbb{E}[\frac{\sum_i (f_i - \langle d \rangle)^2}{M}] = \overline{bias}^2 + \overline{var} \tag{2.12}$$

$$\mathbb{E}[\frac{\sum_i (f_i - \overline{f})^2}{M}] = \overline{var} - \frac{\overline{var}}{M} - (1 - \frac{1}{M})\overline{covar} \tag{2.13}$$

where M is the number of models in the ensemble and $\langle d \rangle$ is the expected value of the unseen data (50).

The presence of variance in both terms indicates that the ideal ensemble must maximize both the average accuracy and the diversity of its components (50). Neither objective can be maximized independently without affecting the other (50).

2.2.1 Multi-Exit Architectures

Training and using multiple deep neural networks in an ensemble can be prohibitively expensive for practical use. An alternative approach is to use a multi-exit architecture, in which intermediary classifying layers are introduced before the final classifier. This has been proposed in multiple studies for a variety of architectures as a method of improving performance and reducing costs (51; 52; 53; 54).

These architectures can also be interpreted as an ensemble of networks of varying depth, and while weights are shared across the component networks, the variance in depth of each exit can promote sufficiently diverse results to be effective (30). Qendro et al. were able to show improvements in both accuracy and uncertainty quantification over Monte Carlo dropout networks, and comparable accuracy with better uncertainty quantification than a full deep ensemble, simply by averaging the results from each of the exits in their multi-exit model (30).

A common method to develop multi-exit architectures is to use a high performing backbone architecture and attach classifying branches to it. Common choices of backbones are the VGG, which uses small filters in their convolutional layers to build very deep robust networks, or one of its successors like a ResNet, which built on the

contributions of the VGG and introduced the idea of residual connection to allow the input to skip certain convolutional layers (55; 13; 56). Both the small filter sizes and residual connections led to significant improvements in stability and performance of deep convolutional networks (55; 13).

However, there has been significant variety in the number of different ways that the exits can be attached. Kaya et al. placed classifying branches at certain FLOP thresholds, and others have placed them after certain semantic groupings of layers, like the residual blocks in a ResNet (31; 30; 56). A more automated approach was used by Baccaerlli et al., in which a greedy algorithm was developed to optimize the placement in a given network in linear time with respect to the network depth (57). The choice of exit type has also been explored, with a variety of transformed-based and convolutional based branches being used (58; 31; 56)

(35) found that the early layers lacked the higher level features needed for accurate classification, forcing the network to develop these features earlier on and hence hurting overall performance. They proposed the Multi-Scale DenseNet (MSDNet), which maintains both high level and low level features at every layer and interconnects them to alleviate the previously identified issue. This architecture also includes built in early exits after each block. See (35) for a full breakdown of the architecture and visual representations.

Kaya et al. used early exiting to demonstrate that the majority of inputs can be correctly predicted by a fraction of the full network (31). Reducing model overthinking, which occurs when a model is able to output the correct prediction before the final convolutional layer, can save computation costs and avoid the impacts of destructive overthinking, when the additional convolutions lead to the model changing its correct prediction (31).

This reduction is accomplished through confidence based exiting, which allows the network to exit early only when the prediction confidence at the exit exceeds a threshold. They argue that this lets the network leverage the right amount of computation needed for the given sample, improving performance and saving costs (31; 59).

Training a multi-exit architecture is non-trivial, as one simultaneously wants to improve the average performance of each exit without significantly reducing the best possible performance. Attempts have been made to freeze the backbone weights and optimize only the exit branches or to optimize the architecture layer by layer, however they are usually less effective than optimizing the whole network and all exits simultaneously (60; 58; 61).

A standard weighted cross entropy loss function is often outperformed by alternatives that incorporate curriculum learning or distillation (60; 56; 34). Both curriculum learning and distillation incorporate the idea of having a better classifier guide a less powerful one, however the method in which this is done differs. Curriculum learning uses a separate network, and while this has been highly successful it is also more expensive to train due to needing multiple high performing networks (60). Distillation uses the inherent discrepancy in classifying power present in multi-exit architectures, due to their variation in depth, to have the later exits pass knowledge to the earlier exits (34).

A comprehensive comparison of distillation methods for multi-exit training was presented in (56), and they found that a bidirectional distillation method, with knowledge passing between all exits, maximized the performance of the early exits and the final classifier for a variety of ResNet and VGG based multi-exit architectures.

2.3 Combined Approach

Both MC dropout and multi-exit ensembling have been shown to be effective at accurately quantifying predictive uncertainty. However, both have their own limitations. Monte Carlo dropout has been outperformed by deep ensembles and other techniques in both accuracy and uncertainty quantification (29). Through interpreting MC dropout as an implicit ensemble, Fort et al. showed that it produces results which can be as accurate as the best performing deep ensembles, however they are significantly less diverse (29).

Furthermore, performing multiple passes through a network for evaluation increases the computation required for a single prediction many-fold, which can make the usage of MC dropout networks prohibitively expensive (39).

Early exit ensembling has been shown to be competitive with full deep network ensembles, producing comparable results at a fraction of the cost (30; 56; 31). The varying depths of the individual networks involved in an early exit ensemble likely provides sufficient diversity to the ensemble, however as discussed in Section 2.2.1 the earliest exits often perform worse due to the smaller capacity and lack of high level features.

The best ensembles require both high accuracy and diversity in their constituent models. Introducing Monte Carlo in each of the early exits should lead to better performance of each of the individual networks, improving the uncertainty quantification, as discussed in Section 2.1.2, without limiting the diversity caused by the variety of depth. In addition, through the use of confidence based exiting the improved calibration can be leveraged to reduce computation and increase accuracy by avoiding the adverse effects of overthinking.

2.4 Medical Bayesian NNs

Machine learning has been widely applied across the medical domain with significant success, however the lack of uncertainty and reliability has limited their application (24). More recent approaches have employed models which quantify uncertainty, and MC dropout has been a popular choice due to its simplicity (30). It has been applied to models across a variety of tasks like diabetic retinopathy detection, brain tumor segmentation and multiple sclerosis lesion detection (62; 63; 64; 65).

Ensembling has also been very popular, in part due to its high performance capabilities. Particularly in the field of medical image classification, ensembling approaches have outperformed alternatives, achieving extremely high classification accuracies on breast and cervical histopathology, chest x-ray and brain MRI images

(66; 67; 68; 69). However, most approaches tend to focus on the benefits to classification performance rather than the benefit to uncertainty quantification. Recent works have also been able to show that applying ensembling can lead to significant improvement in uncertainty quantification for time series data classification, medical image classification and segmentation tasks (30; 70).

The presence of multiple large publicly available datasets has made deep learning particularly viable for chest x-rays (71). The ChestX-ray 14 dataset is the most used of these datasets, with an ensembling approach having been shown to outperform a radiologist in pathology classification (23). This dataset has also been used with a few models that quantify uncertainty for sample rejection and outlier detection (72; 73; 74).

While some models have been developed on chest x-rays specifically to better quantify uncertainty, these have primarily been done on other datasets (36; 37; 38). No model designed to improve uncertainty quantification has been used on the ChestX-ray 14 dataset.

Chapter 3

Making an MSDNet Bayesian

In this chapter, the viability of Monte Carlo dropout with early exit ensembling (MC EE) as a method of improving uncertainty quantification and reducing computational cost is explored. Using the MSDNet on Cifar100, a series of MC EE models with different dropout rates and layer placements are trained and then evaluated on the performance, uncertainty quantification and computational cost of each model. Ablation studies are also performed, comparing the MC EE model with a series of MC dropout models, early exit ensembling (EE) models and the baseline, with and without confidence exiting.

The MC EE method is shown to be able to generate models which are more accurate and better calibrated than any other tested approach. The best MC EE models outperformed the most accurate alternative, an EE model, by 0.002 $\pm$ 0.001 in overall accuracy and the best calibrated alternative, an MC dropout model, by 0.012 $\pm$ 0.002 in the ECE metric. Hyperparameter tuning over the confidence threshold, number of network pass throughs and the amount of dropout is usually needed to find the optimal results, and poor choice of these parameters can lead to worse results than the baseline. Applying dropout in the exits and applying a high confidence threshold is generally found to give strong results, however this is not always guaranteed.

3.1 Method

Most models used in multi-exit architectures simply add exits onto pre-existing high performing models such as ResNets or ViTs (30; 56; 58). However, as discussed in Chapter 1, a key challenge that the MC EE method can overcome is that it is model agnostic. To demonstrate this, it is first applied to a specially designed multi-exit architecture, the Multi-Scale Dense Network (MSDNet) on a natural image dataset, Cifar100.

3.1.1 MSDNet

Multi-exit architectures have to balance maximizing the performance of the early exits with the performance of the later exits. The MSDNet is one of the few archi-

tectures designed specifically for early exiting, maintaining multiple different sized feature maps to allow each exit to have access to high-level features without compromising the power of the later exits (35). This architecture was shown to boost performance significantly when compared to a ResNet or DenseNet implementation, while generally maintaining the final layer classification accuracy (35).

While the original MSDNet authors used weighted cross entropy, a later work (34) was able to see improvement by introducing a distillation term in the loss function. The training method laid out in that paper is followed here. The model weights are initialized using the Glorot normal procedure from (75), and the model was optimized using stochastic gradient descent with an initial learning rate of 0.1. A weight decay of 10^{-4} and momentum 0.9 were used along with a batch size of 64.

Rather than the multi-step learning rate schedule adopted by (35) and (34), the learning rate is reduced when the validation accuracy plateaus. This was done to provide a more general scheduling approach which did not require optimizing hyperparameters like when and by how much to reduce the learning rate, yet could still achieve high level results. If the validation accuracy did not improve after 20 epochs, early stopping was also employed.

Initial testing supported this assertion, returning similar results to the multi-step approach. It also showed that the gradients tended to explode, and hence gradient clipping, a cheap and effective solution to exploding gradients, is used (76). Gradient clipping is where the gradient is reduced if its L2 norm exceeds some threshold and has been shown to increase the stability of training (76). Best performance was found to occur with a maximum L2 norm threshold of 2.

3.1.2 Cifar100

To investigate the viability of the approach the network is tested on Cifar100, a standard benchmark dataset. Cifar100 is a curated subset of a larger dataset scraped from the web containing photo-realistic tiny 32x32 images with a single main object (32; 77). The size and clear focus of the images have allowed models to achieve high levels of accuracy (56). The dataset has 600 images of 100 different classes, where each class corresponds to a non-abstract noun. The standard split defined in (32) is used to divide the dataset into a train and test set containing 500 and 100 images per class respectively. A further ten percent of the train set are randomly removed and added to a validation set, which is used to test for overfitting.

The images are preprocessed as in (34), normalizing all images by subtracting the mean and dividing by the standard deviation of the pixel values in the dataset. During training, the standard data augmentation of random horizontal flipping is applied with $p = 0.5$ (34).

3.1.3 Implementing MC EE

As discussed in challenge C3, dropout employed after every layer can have an extreme regularizing effect on the network, adversely affecting performance (40). However, some level of dropout is required to give the observed improvement to

uncertainty quantification. Therefore, a grid search over multiple types of dropout implementations and rates is used.

A common approach for implementing MC dropout has been to use fewer dropout layers placed after semantic groupings of layers, like an encoder/decoder unit (40; 78). This same approach is adopted in introducing dropout layers into the MSDNet, following the groupings used by the original authors of block (B), layer (L), scale (S) and exit (E). A visual representation of the various groupings is given in Figure 2 and 9 of (35). A dropout configuration of S+E means that a dropout layer is added after every scale in the network, and before the final linear classifying layer in every exit branch of the network. To limit the possible configurations only the following combinations are tested: E, B+E, L+E and S+E.

Following (25), four dropout rates are tested for each dropout configuration: 0.125, 0.25, 0.375, 0.5. The chosen rate is adopted for all dropout layers in the network. Dropout rates larger than 0.5 are not tested due to the aforementioned significant regularization provided by the dropout layers; networks with very high dropout rates are unlikely to be able to perform well, and hence due to time constraints are not explored.

3.1.4 Cutting Costs

Introducing MC dropout and multi-exiting incurs a large computational cost compared to a standard implementation, as mentioned in challenge C2. While placing the dropout layer only in the exit branch as discussed in Section 3.1.3 can mitigate large amounts of the cost, this obstacle is also overcome through the strategic choice of the number of network passes and the use of confidence based exiting.

The number of passes used for MC dropout can have significant impacts on performance and cost, and there is no standard way of choosing this hyperparameter. A very high number of passes will achieve the best possible uncertainty quantification, however this requires far more computation per prediction (40; 39). Initial testing of the impact of increased MC passes is used to find the smallest number of passes which achieve 90% of the best achievable accuracy and ECE, giving a balance between performance and cost. The details of this experiment are given in Section 3.2.4.

The set of predictions for each of the images in the test set are recorded for each exit alongside the prediction sets given by an ensemble of all the previous exits. Following from (30), the contributions of each exit in the ensemble are formed using a simple arithmetic mean.

The prediction sets from utilizing confidence thresholding to reduce model overthinking are also recorded, using both the standard and ensemble approach. Starting from the earliest exits, each instance is passed through the network and when an exit or ensemble's prediction confidence exceeds some threshold, that prediction is used as the label for that instance. The following confidence thresholds are tested, as in (31): 0.1, 0.15, 0.25, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99, 0.999.

To measure the computational cost of a particular model, the number of floating-point operations (FLOPs) per module necessary for a single forward pass are tracked

using a tool from the Facebook AI Research team (79). The number of FLOPs can be used as a measure of the amount of computation needed to produce a result (31). When confidence thresholding is employed, only the layers used to get to the exit which exceeds the threshold are counted. If ensembling, then each exit branch prior to the successful exit is also added.

For the multiple passes used in MC dropout, the minimum number of additional FLOPs necessary to get an answer is calculated. Hence, if there are only dropout layers in the exit branch, all the computation done before the dropout layer in the exit is only counted once, and then the layers after the dropout layer are repeated the necessary number of times. It is noted that the model itself does not exit early or cache results as the two above FLOP calculations would imply, however these optimizations could be implemented without changing any of the results and are not included as all possible exits and confidence thresholds are always calculated for evaluation purposes.

While there have been multiple studies which have been able to significantly reduce the computational cost of models through model compression, pruning and exploiting sparsity, they are not employed in this study (80; 81; 25). The most basic versions of the model are used to help isolate the computational effects of MC EE.

3.1.5 Ablation Studies

To establish the effectiveness of the MC EE networks as a method of overcoming challenge C1, they are compared against the standard single-exit baseline and two other possible methods used in the literature: a single-exit version with Monte Carlo dropout employed as above (28), and an early exit version without dropout (30).

By ablating various elements of the network, an attempt is made to establish what is driving the improvement in uncertainty quantification and hence evaluate whether the combined approach is better than its individual components. To ensure fair comparisons, the same hyperparameters are used where possible. The single exit models are still trained using the same loss function, however as there are no other exits, there are no distillation terms, and hence this just corresponds to standard cross entropy (56). It is noted that the performance of the single-exit MC dropout models with S+E and L+E was very poor ($< 10\%$ accuracy), and hence were not included.

3.2 Results

A total of 26 types of MSDNets were trained: 16 MC EE, 8 MC, 1 EE and the single-exit baseline. For each prediction set, the overall accuracy and ECE were calculated alongside the percentage of FLOPs compared to the single exit baseline. Each model type was trained 5 times with different random initialization and the average plus or minus one standard error of the above metrics are reported.

In addition to the above results, for each of the 130 models trained the metrics for the resulting predictions from the 11 confidence thresholds are also reported. Due to space constraints, the results of the 757 possible predictions for the test set

are reported in Appendix A. Four tables with the best accuracy, ECE and overall performance with and without considering the computational cost of all models are reported in Tables 3.1. It is noted that for the non-confidence based exiting results, the uncertainty in the FLOPs value should come from the tool used. However, this uncertainty is negligible and hence these results are reported without uncertainty in the above tables.

The models are defined by the type of dropout they use, the dropout rate used, which exit or ensemble the results are from or, if confidence exiting was used, the confidence threshold. For concision and clarity, they are reported in the following standard format: Dropout Type (Dropout Rate, Exit/Ensemble/Confidence Threshold).

Hence, a prediction set from the ensemble formed from the fifth exit of an MC EE model with a dropout rate of 0.125 and dropout only in the exits is reported as E (0.125, 5). For the same model, the prediction set obtained from confidence exiting with a threshold of 0.9 is reported as E (0.125, 0.9) if it uses the results from the standard exits or *E (0.125, 0.9)* if it uses the ensemble results. If no exit, ensemble or confidence threshold is given, as in E (0.125), this corresponds to the MC model with dropout in the exits and a dropout rate of 0.125. This format is used for all results in this report.

3.2.1 Key Findings

The MC EE models were observed to consistently outperform the alternatives both in accuracy and ECE while also reducing the number of FLOPs. The optimal choice of dropout configuration and rate could lead to an improvement in accuracy of 0.027 $\pm$ 0.001, 0.002 $\pm$ 0.001and 0.019 $\pm$ 0.002 or a decrease in ECE by 0.241 $\pm$ 0.003, 0.068 $\pm$, 0.001 and 0.012 $\pm$ 0.002 over the baseline, best EE and best MC model predictions respectively.

When optimizing for both metrics simultaneously, the best MC EE models could give similar or better accuracy and ECE simultaneously than the best of any of the MC, EE and baseline models could do on the metrics individually. This is demonstrated in Table 3.2, where the model E (0.25, Ensemble10) is shown to have a better accuracy and ECE than the best of any alternatives.

This improvement can often be done for cheaper. The best overall MC EE model, E (0.25, Ensemble5), is also better in both accuracy and ECE than the best MC models or the baseline at those metrics individually, while also only needing 60% of the FLOPs. While the best MC EE models also outperforms the best EE model, it can only do so in accuracy or ECE while reducing FLOPs.

This illustrates that on the MSDNet, the use of MC EE seems to generate higher performing and better calibrated models than alternatives in the literature, while also generally costing similar or less. However, it is important to note that of the 757 possible predictions sets, over a third of them had a much worse accuracy of under 75% of the accuracy achieved by the baseline.

While the MC EE models can get high performing results, this often requires careful tuning of the hyperparameters. In Section 3.2.2, some trends which could

	Accuracy	ECE	FLOPs
E (0.125, Ensemble10)	0.693 ± 0.001	0.037 ± 0.002	1.271
E (0.125, 0.999)	0.693 ± 0.002	0.038 ± 0.002	1.10 ± 0.02
E (0.25, Ensemble10)	0.692 ± 0.002	0.021 ± 0.001	1.271
E (0., Ensemble7)	0.691 ± 0.001	0.0883 ± 0.0003	0.875
E (0, 0.999)	0.691 ± 0.001	0.1027 ± 0.0009	0.976 ± 0.009

(a) Most Accurate

	Accuracy	ECE	FLOPs
E (0.25, Ensemble 6)	0.685 ± 0.003	0.012 ± 0.001	0.781
B+E (0.125, 3)	0.662 ± 0.003	0.0125 ± 0.0003	3.004
E (0.375, Ensemble 1)	0.612 ± 0.003	0.0127 ± 0.0006	0.164
E (0.375, 0.9)	0.680 ± 0.002	0.0135 ± 0.0009	0.888 ± 0.005
B+E (0.125, 0.95)	0.674 ± 0.003	0.014 ± 0.001	7.23 ± 0.03

(b) Lowest ECE

	Accuracy	ECE	FLOPs
E (0.25, Ensemble7)	0.689 ± 0.002	0.0129 ± 0.0009	0.881
E (0.375, Ensemble10)	0.681 ± 0.005	0.013 ± 0.001	1.271
E (0.375, 0.99)	0.681 ± 0.005	0.013 ± 0.002	1.103 ± 0.004
B+E (0.125, 6)	0.673 ± 0.005	0.013 ± 0.002	6.348
B+E (0.125, 0.95)	0.674 ± 0.003	0.014 ± 0.001	7.23 ± 0.03

(c) Best Overall

	Accuracy	ECE	FLOPs
E (0.25, Ensemble5)	0.680 ± 0.003	0.0109 ± 0.0005	0.590
E (0.375, 0.9)	0.680 ± 0.005	0.014 ± 0.002	0.888 ± 0.005
E (0.375, Ensemble8)	0.679 ± 0.006	0.013 ± 0.003	1.009
E (0.125, Ensemble5)	0.685 ± 0.005	0.025 ± 0.003	0.590
E (0.25, 0.999)	0.690 ± 0.002	0.023 ± 0.002	1.12 ± 0.01

(d) Most Efficient

Table 3.1: The best 5 tested MSDNet models according to: **(a)** Accuracy, **(b)** ECE, **(c)** Scaled Average of Accuracy and ECE, **(d)** Scaled Average of Accuracy, ECE and FLOPs.

help limit the scope of the search are discussed.

To verify the correctness of the implementation, the EE results are compared to the performance given in the original paper, (34). It is noted that (34) only gives results in top-5 accuracy, with an average accuracy across the layers of 92.7 ± 0.3 vs 90.41 ± 0.01 from the models in this paper. It is suggested that the small percentage difference may be due to the usage of a validation set rather than training with the full train dataset and a more general learning rate schedule which has not been optimized for the standard MSDNet model.

3.2.2 Trends in MC EE

Observing Tables 3.1, a few trends can be noticed. Almost all of the tables are dominated by MC EE with dropout only in the exit branches. This is within expectation, as this is the combination with the least amount of dropout, hence reducing the adverse effects of the regularization while still providing the benefit of MC dropout in better uncertainty quantification. In addition, by having dropout only in the exit branch this makes it one of the most cost efficient solutions as well, because only the final linear classification layer in each exit branch needs to be repeated in each of the passes needed for MC dropout.

It is also noted that at least a minimum level of dropout is needed to strongly reduce the ECE. In Table 3.1b of the best calibrated MSDNet models, all exit-only MC EE solutions have a dropout rate of at least 0.25, suggesting that the increased variance in predictions caused by a higher dropout rate may be needed to improve the uncertainty quantification of the network. This aligns with the previous literature's understanding that diversity is needed in a strong ensemble, which was discussed in Section 2.2.

The presence of some B+E dropout MC EE networks in Tables 3.1b and 3.1c is fairly understandable, as this is the combination with the second least amount of dropout. The increased amount of dropout layers throughout may explain why only a dropout rate of 0.125 is enough to provide the diversity needed. However, the significant cost of performing MC dropout on these types of MC EE models makes it less viable than some of the exit only dropout alternatives.

Confidence based exiting generally offers a cheaper alternative with similar results. However, for some models like *E (0.25, 0.999)* from Table 3.1d, the resulting prediction set can be worse and more expensive than a non-confidence exiting solu-

	Accuracy	FLOPs	ECE	FLOPs
Baseline	0.666 ± 0.001	1.00	0.253 ± 0.003	1.00
EE	0.691 ± 0.001	0.88	0.080 ± 0.001	0.46
MC	0.674 ± 0.002	10.00	0.024 ± 0.002	1.00
MC EE	0.693 ± 0.001	1.27	0.012 ± 0.001	0.78
E (0.25, Ensemble10)	0.692 ± 0.002	1.27	0.021 ± 0.001	1.27

Table 3.2: The best accuracy and ECE for the MSDNet on each of the tested approaches, alongside the associated FLOPs cost. For ECE and FLOPs, smaller is better.

tion, suggesting that confidence based exiting does not always lead to a performance boost. It is noted that generally a confidence threshold of 0.9 or higher is needed, which is in line with expectation; In well calibrated models like the MC EE's, a higher confidence suggests a higher likelihood of being correct and hence leading to increased performance.

In general, best performance is achieved by exit-only MC EE's using ensembling. Confidence exiting can generally lead to more cost efficient solutions, however this is not always true. There appears to be a balancing act between the amount of dropout in the network, with too little providing not enough diversity and too much limiting the predictive power of the network. This balance can be achieved on the MSDNet through either a small number of dropout layers with a higher dropout probability or more dropout layers with a small dropout probability.

3.2.3 Ablation Study

To compare the performance of the different types of MSDNets, the best accuracy and ECE of each of the different approaches are reported in Table 3.2.

The EE method is observed to produce highly accurate models which are able to reduce their ECE significantly in comparison to the baseline, however both MC EE and MC models were able to produce prediction sets with a better ECE. The EE method is also efficient, however the MC EE approach is able to produce better overall results for a similar cost.

Using MC dropout can significantly improve uncertainty quantification, however this often comes at the cost of sacrificing accuracy. While the best performing MC dropout models were comparable to the MC EE methods in terms of ECE, they are often less accurate and cost more.

3.2.4 Passes for MC Dropout

To reduce the number of hyperparameters, testing was done to pick an appropriate number of passes for MC dropout. To verify the approach, the model with the most amount of dropout (S+E with $p = 0.5$) and the least amount (E with $p = 0.125$) are evaluated to assess the convergence of the MC EE models. The results are normalized with min-max scaling and plotted in Figure 3.1. The 90% threshold is met in all models by around 10 passes, and hence this is chosen as the default number of passes to evaluate MC dropout models on.

However, the number of passes can also be used as a computation vs performance tradeoff mechanism. The additional flexibility this can provide is demonstrated in Figure 3.2, which plots the performance of the prediction sets from using between 1 and 10 passes for MC EE with dropout type B+E and a dropout rate of 0.125, plotted against the performance of the early exiting, MC and baseline prediction sets. The overall performance is taken to be the normalized average of the accuracy and ECE.

The appropriate choice of the number of passes combined with confidence exiting can be used to get the high performance of the Monte Carlo dropout models at

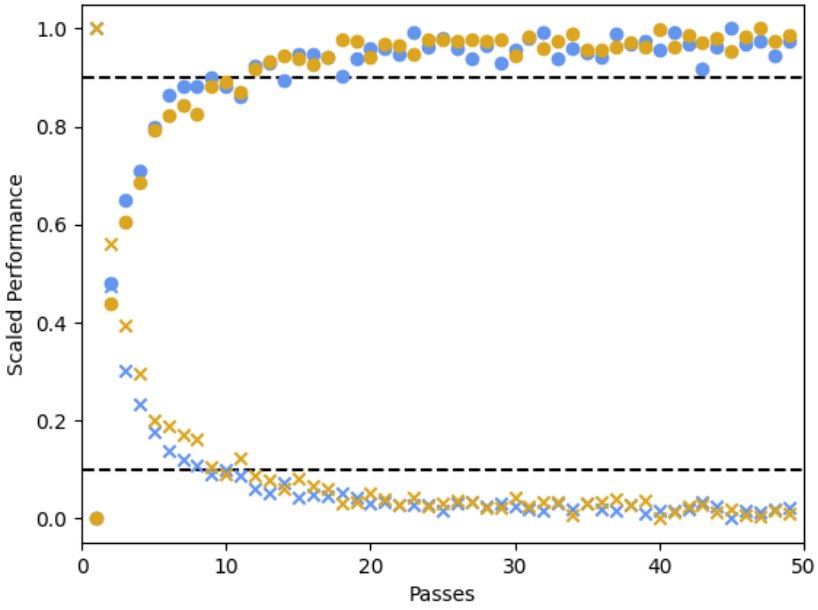


Figure 3.1: A plot of the scaled accuracy (dots) and ECE (crosses) for the models with the most (yellow) and least (blue) amount of dropout against the number of pass throughs of the network. A dotted line is plotted at the 0.1 and 0.9 performance thresholds.

a fraction of the cost, despite needing to do multiple full model run throughs.

3.2.5 Confidence Exiting vs Standard

As discussed in Section 1.1 and 2.2.1, confidence exiting has been used as a means to reduce model overthinking and limit the computational overhead incurred by using early exiting and MC dropout. This result is verified in Figure 3.3, in which the accuracy is plotted against the number of FLOPs for results predicted by each exit of the standard multi-exit architecture, the ensemble versions, and those obtained from confidence exiting. It can be observed that confidence exiting of the standard multi-exit architecture is able to yield better results for cheaper, supporting the assertion made in (31) that confidence exiting can reduce both wasteful and destructive overthinking.

However, for the ensemble predictions it is only at very high FLOPs that a possible improvement in accuracy and cost is observed. This is contrary to expectation, as not only should confidence exiting see improvements due to its effect on overthinking, but ensembles have been shown to have significantly improved ECE, which should lead to fewer mispredictions from exiting too early.

Further investigation revealed that most of the predictions were being made by the first ensemble, which consists of only the earliest exit. The predictions from this exit have the worst ECE of any of the exits or ensembles in the model, and

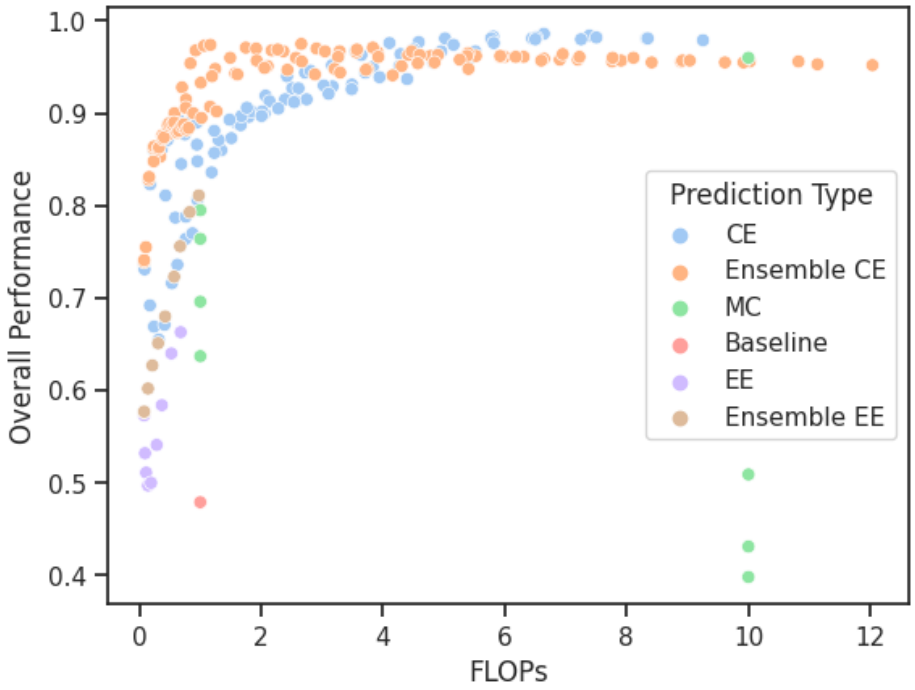


Figure 3.2: A plot of the performance of the prediction sets created by running the B+E model with a dropout of 0.125 with a different number of network pass throughs against the performances of all other prediction sets from the ablation study.

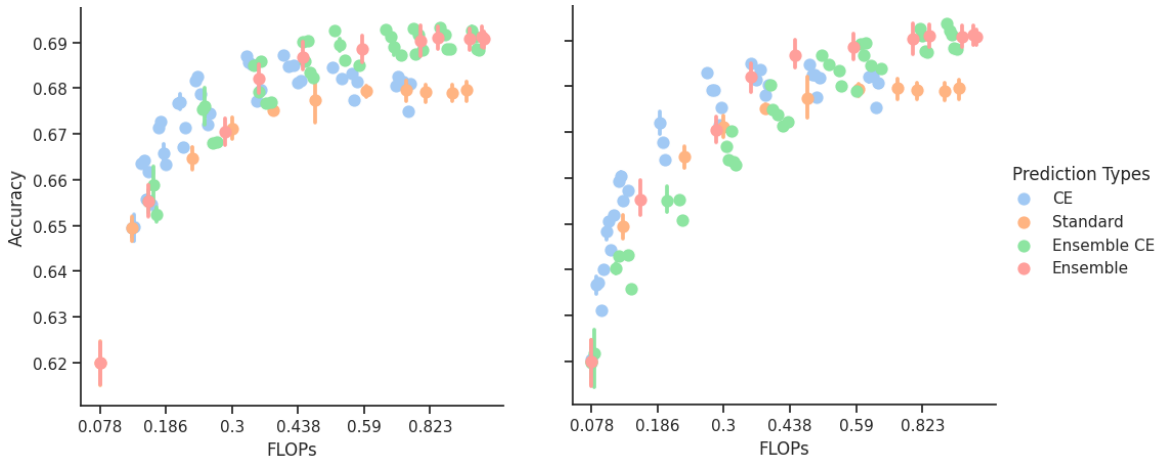
hence reliance on this exit may explain the poor performance compared to the non-confidence exiting predictions, which only use the better performing exits. This is also suggested to be one of the reasons for the poorer performance of some confidence exiting models observed in Tables 3.1.

To verify this claim Figure 3.3a is produced, which is the same as Figure 3.3b but does not include the first exit for confidence exiting with either the standard or ensemble predictions. The predictions of both become more accurate, with the ensemble confidence exiting being comparable and occasionally slightly more efficient than the standard ensembles. This supports the assertion made above that confidence exiting approaches can be limited by a poor performing exit/ensemble.

3.2.6 Dropout Types

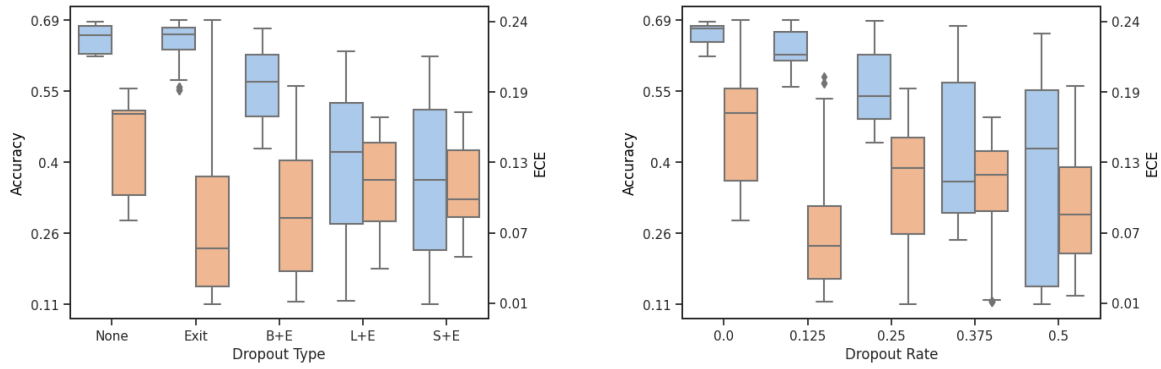
A comparison of the performances of the various dropout types and rates is performed in Figure 3.4 to further investigate the trend noted in Section 3.2.2. The accuracy and ECE are averaged across all the trials and dropout rate or dropout type respectively to create a plot of performance for each of the five possible dropout types or rates. The average accuracy is observed to drop significantly as more dropout layers are added and as the dropout rate increases, which is in agreement with the adverse regularizing effect of MC dropout observed in (40).

One interesting note is that while the average ECE is better for all dropout types compared to the baseline, it does not improve with increased dropout as one might



(a) No First Exit **(b)** Normal

Figure 3.3: Plots of the accuracy against the number of FLOPs for each exit, ensemble and confidence exiting threshold of the EE MSDNet model.



(a) Dropout Type **(b)** Dropout Rate

Figure 3.4: A boxplot of the averaged accuracy (blue) and ECE (orange) for the different dropout types **(a)** and dropout rates **(b)** of the MC EE MSDNet models.

expect. As discussed in (50; 30), a well calibrated ensemble includes diversity and accuracy. Despite the diversity introduced by the higher dropout, the lower accuracy means that the model's highest confidence prediction is often wrong, leading to an overall worse ECE. This further supports the need for hyperparameter tuning as a solution to challenge C3, as increasing dropout is not guaranteed to give you better uncertainty quantification.

3.2.7 Limitations

One of the key benefits to using a combined approach is the increased flexibility. A standard single exit MC dropout method is unable to leverage the power of confidence exiting or depth ensembling, while a standard multi-exit architecture only has a particular number of unique prediction sets that it can produce without MC

dropout. These hyperparameters can be tuned to give the solution that works best for the problem at hand, maximizing accuracy and/or ECE for a given computational budget.

However, the increased number of hyperparameters can make it prohibitive to find the best model. In addition to the multiple parameters that need to be tuned in a standard network like the batch size, learning rate, weight decay... etc., both MC dropout and multi-exit architectures have additional hyperparameters like dropout type, dropout rate and exit placement which can exponentially increase the number of choices one needs to optimize for. This can make the development and training of an appropriate effective model difficult.

Another possible limitation is with the method used to ensemble models. The ensemble method used is to form the largest ensemble possible, as larger ensembles of individual networks have been shown to produce better results (26). However, in early exit ensembles the earliest exits often perform worse due to using a smaller percentage of the network, and hence they may drag down the overall performance of the ensemble. It is suggested that using a weighted average to reduce the contribution of the early networks or a more careful treatment of the networks chosen to be in the ensemble may lead to increased performance over those presented here. Due to computational and time constraints, this is left to future works to explore.

3.3 Conclusion

The MC EE method is found to lead to improvements in both accuracy and ECE for the MSDNet on Cifar100. The hyperparameters can be adjusted to give models that are better in a particular metric while being cheaper than the best alternatives. Further investigation revealed that optimizing the number of network passes and exits used in any ensembles can likely further improve results. However, the large number of hyperparameters which need to be tuned can significantly increase the cost to train and develop MC EE models, which may make the approach less applicable for practical use.

Chapter 4

Extending to More Models

In Chapter 3, it was demonstrated that MC EE was a very effective technique to improve performance and cut costs when using the MSDNet on Cifar100. As discussed earlier, the MSDNet is an architecture designed for early exiting, however most early exiting architectures use an existing backbone and attach early exit branches. Hence, to establish whether the approach can overcome challenge C4, the ability to generalize across models and datasets, it is also important to see whether it is applicable to this other type of architecture as well.

Following the same procedure as in Chapter 3, the MC EE is tested on an early exit ResNet-18 and VGG-19, alongside various ablation studies to analyze and compare the performance of the technique.

The MC EE is again found to match or outperform the most accurate and best calibrated ResNet-18 and VGG-19 models, however the improvements are not as significant as they were with the MSDNet. The benefits of the MC EE approach are found to be primarily driven by the better uncertainty quantification of the later exits in MC EE compared to their EE counterparts, however the introduction of MC dropout in the earlier exits is found to actually worsen their accuracy and calibration. The use of confidence exiting can help leverage the improvements in the later exits while mitigating the impact of the worse earlier exits.

4.1 Method

The results obtained on MSDNet were promising, suggesting the need for further investigation. Two common networks discussed previously, the ResNet and the VGG, have been used as a backbone for a multi-exit architecture and achieved high performance on Cifar100 (56). Both networks were originally designed for ImageNet, and hence needed to be modified to be used on Cifar100

4.1.1 Models

While the training method used in (34) was used for the MSDNet due to its high efficacy, a follow up study, (56), implemented an extension which saw improved results on both a multi-exit VGG-19 and ResNet-18. Their proposed method outperformed

the method of (34) on these models, and hence is used as the training method for both the VGG and the ResNet. In particular, their best performing method, the $\alpha = 1, \beta = 0$ version with a mean squared error loss function, is used.

Multiple types of ResNets have been used with early exiting on Cifar100, however due to computational and time constraints the smallest ResNet, the ResNet-18, is used in all subsequent ResNet experimentation (56).

The original ResNet designers tested the performance on Cifar100 using a ResNet-20, a ResNet architecture designed to deal with the smaller 32x32 pixel images (13). However, it is more common to see a modified ResNet-18 than a ResNet-20 when using the Cifar100 dataset (35; 56). These modifications are primarily done by using smaller kernels and smaller strides in the convolutional layers. In this paper, the modified Resnet-18 architecture proposed in (56) is used as the backbone of the multi-exit architecture.

The VGG-19 is the only VGG network trained with the same loss function as the ResNet-18 in (56), and hence is used in this report to allow for more direct comparisons between the networks. The VGG-19 network can be used directly on the Cifar100 dataset with only small changes to the classification layer, however the exact modifications used in (56) were not mentioned and hence some additional experimentation was performed.

The original VGG-19 for the larger ImageNet calls for two fully connected layers of 25,088 and 4096 neurons respectively with dropout and RELU in between. While modifying the first fully connected layer to have 512 neurons allows the network to function on 32x32 pixel images, testing revealed that this could lead to instability and poor performance. Instead, replacing the above with a simple fully connected layer with 512 neurons was effective at lifting performance and leading to a more stable training procedure. This modification was used in this report to allow the networks to function on the Cifar100 dataset.

4.1.2 Implementing MC EE

Unlike in the MSDNet, the ResNets and VGG networks are not as cleanly defined into groupings and do not have preplaced exit branches. The approach used in this report is to consider the networks to be composed of multiple blocks of convolutions separated by pooling layers, as in (56). The early exit branches are then placed at the end of each of these blocks and each exit branch uses the minimum number of convolutions necessary to fit into a 512 neuron fully connected layer (56).

Following the same procedure as in Chapter 3, dropout layers are placed after semantic groupings, however the exact definition of each semantic grouping is different. Dropout layers placed after each block follow the new definition of block discussed above, and dropout layers placed in the exit branch are still placed just before the final connected layer. However, neither the ResNet or the VGG have a scale/layer equivalent. Hence, no scale equivalent is used and the layer configuration is defined as placing dropout after the next smallest grouping of convolutional layers. For the ResNet, this is after each residual block, and for the VGG it is after every layer in the network as in the original MC Dropout paper (27; 28).

It is noted that some exit branches, like the 4th exit from the VGG-19 network, only contain a single linear layer. Therefore, if a VGG-19 model is being trained with multiple exits and block or layer dropout, the above definitions would mean there would be two dropout layers back to back, with a dropout layer at the end of the 4th block and a dropout layer right before the linear layer in the 4th exit. To avoid this, when this occurs the dropout layer in the main architecture is removed.

4.2 Results

The same combinations of dropout types are tested: E, B+E and L+E. A total of 26 types of ResNet-18 and VGG-19 were trained: 12 MC EE, 12 MC, 1 EE and 1 baseline respectively. As before the mean and standard error over five trials of the overall accuracy, ECE, and percentage of FLOPs compared to the single exit baseline of each exit and the largest ensemble of exits is reported alongside the results of confidence exiting for the exit results and the ensemble results. The most accurate, best calibrated, best overall and most efficient are given in Tables 4.1 and 4.2. All other results are available in Appendix A.

4.2.1 Key Findings

As with the MSDNet, the MC EE method is able to generate the most accurate and the best calibrated models as shown in Tables 4.3 and 4.4.

For the VGG-19, the most accurate MC EE model is able to match the accuracy of the best possible prediction set from an EE model at 0.747 ± 0.001, while handily outperforming the baseline and best MC model by over 0.04 in overall accuracy. The best calibration error model was able to reduce the ECE by 0.150 ± 0.006, 0.010 ± 0.001 and 0.003 ± 0.002 over the baseline and the best calibrated MC and EE models respectively. Furthermore, a model like the E (0.375, 0.5) given in Table 4.1b is able to give a better accuracy and a better ECE than the baseline or any MC model, while using less than half the computational resources.

However, unlike with the MSDNet no MC EE model is able to give a better accuracy and ECE simultaneously than the best EE models can do on the metrics individually. While a better calibrated model like E (0.375, 0.5) can give a 0.007 ± 0.002 improvement in ECE over the best calibrated EE model, E (0, 0.5) and also marginally improve the accuracy by 0.002 ± 0.003, it requires more computational resources (47% vs 43% of the baseline in FLOPs). That same model can significantly reduce the computation (47% vs 98%) and the ECE (0.017 ± 0.002 vs 0.077 ± 0.002) compared to the most accurate EE model, but at the cost of a drop in accuracy of 0.015 ± 0.003.

The results are similar for the ResNet-18. Certain combinations of hyperparameters can lead to a more accurate (i.e. *E (0.125, 0.9)*), a better calibrated model (i.e. E (0.25, 0.5)) or a similar performing but cheaper model (i.e. E (0.5, 0.6)) than the best of any of the alternatives, as shown in Tables 4.2 and 4.4. However, like the VGG there was no one dominant model which outperformed all the alternatives.

	Accuracy	ECE	FLOPs
E (0.375,Ensemble3)	0.747 ± 0.001	0.1210 ± 0.0005	0.982
E (0,Ensemble3)	0.747 ± 0.002	0.077 ± 0.002	0.977
E (0.375,0.99)	0.746 ± 0.002	0.083 ± 0.001	1.038 ± 0.003
E (0,0.999)	0.746 ± 0.002	0.045 ± 0.002	1.053 ± 0.002
E (0.125,Ensemble3)	0.743 ± 0.001	0.094 ± 0.002	0.982

(a) Most Accurate

	Accuracy	ECE	FLOPs
B+E (0.25,0.95)	0.733 ± 0.002	0.015 ± 0.001	8.59 ± 0.03
B+E (0.25,4)	0.733 ± 0.002	0.015 ± 0.001	10.00
E (0.125,0.5)	0.725 ± 0.002	0.017 ± 0.001	0.45 ± 0.02
E (0.375,0.5)	0.732 ± 0.002	0.017 ± 0.002	0.47 ± 0.02
B+E (0.125,3)	0.736 ± 0.004	0.018 ± 0.002	9.05

(b) Lowest ECE

	Accuracy	ECE	FLOPs
B+E (0.25,0.95)	0.733 ± 0.002	0.015 ± 0.001	8.59 ± 0.03
B+E (0.25,4)	0.733 ± 0.002	0.015 ± 0.001	10.00
B+E (0.125,3)	0.736 ± 0.004	0.018 ± 0.002	9.05
E (0.375,0.5)	0.732 ± 0.002	0.017 ± 0.002	0.47 ± 0.02
E (0.125,0.5)	0.725 ± 0.002	0.017 ± 0.001	0.45 ± 0.02

(c) Best Overall

	Accuracy	ECE	FLOPs
E (0.375,0.5)	0.732 ± 0.002	0.017 ± 0.002	0.47 ± 0.02
E (0.125,0.5)	0.725 ± 0.002	0.017 ± 0.001	0.45 ± 0.02
E (0.25,0.5)	0.723 ± 0.003	0.02 ± 0.002	0.47 ± 0.02
E (0,0.5)	0.73 ± 0.002	0.025 ± 0.001	0.43 ± 0.02
E (0.5,0.6)	0.731 ± 0.003	0.026 ± 0.002	0.55 ± 0.01

(d) Most Efficient

Table 4.1: The best 5 tested VGG-19 models on Cifar100 according to: **(a)** Accuracy, **(b)** ECE, **(c)** Scaled Average of Accuracy and ECE, **(d)** Scaled Average of Accuracy, ECE and FLOPs.

	Accuracy	ECE	FLOPs
E (0.125,Ensemble3)	0.776 ± 0.001	0.048 ± 0.003	1.204
E (0.125,0.9)	0.776 ± 0.001	0.05 ± 0.003	1.019 ± 0.004
E (0.25,Ensemble3)	0.773 ± 0.002	0.062 ± 0.004	1.204
E (0.25,0.95)	0.773 ± 0.002	0.062 ± 0.004	1.081 ± 0.006
E (0.5,0.9)	0.773 ± 0.003	0.09 ± 0.01	1.06 ± 0.01

(a) Most Accurate

	Accuracy	ECE	FLOPs
E (0.25,0.5)	0.763 ± 0.002	0.014 ± 0.001	0.672 ± 0.003
B+E (0.125,0.7)	0.764 ± 0.002	0.016 ± 0.002	7.48 ± 0.03
E (0.375,0.5)	0.758 ± 0.002	0.016 ± 0.003	0.684 ± 0.003
L+E (0.125,0.9)	0.764 ± 0.003	0.0159 ± 0.0006	8.38 ± 0.04
E (0.5,0.6)	0.766 ± 0.004	0.016 ± 0.002	0.713 ± 0.006

(b) Lowest ECE

	Accuracy	ECE	FLOPs
E (0,Ensemble3)	0.7719 ± 0.0006	0.017 ± 0.002	1.204
E (0,0.999)	0.7719 ± 0.0006	0.017 ± 0.002	1.165 ± 0.002
E (0.125,2)	0.770 ± 0.002	0.021 ± 0.001	0.794
E (0.5,0.6)	0.766 ± 0.004	0.016 ± 0.002	0.713 ± 0.006
B+E (0.125,0.7)	0.764 ± 0.002	0.016 ± 0.002	7.48 ± 0.03

(c) Best Overall

	Accuracy	ECE	FLOPs
E (0,0.95)	0.7719 ± 0.0006	0.017 ± 0.002	1.026 ± 0.003
E (0,Ensemble3)	0.7719 ± 0.0006	0.017 ± 0.002	1.204
E (0.5,0.6)	0.766 ± 0.004	0.016 ± 0.002	0.713 ± 0.006
E (0.125,2)	0.77 ± 0.002	0.021 ± 0.001	0.794
E (0.125,0.6)	0.769 ± 0.001	0.0219 ± 0.0008	0.692 ± 0.001

(d) Most Efficient

Table 4.2: The best 5 tested ResNet models according to: **(a)** Accuracy, **(b)** ECE, **(c)** Scaled Average of Accuracy and ECE, **(d)** Scaled Average of Accuracy, ECE and FLOPs.

	Accuracy	FLOPs	ECE	FLOPs
Baseline	0.693 ± 0.002	1.00	0.165 ± 0.006	1.00
MC	0.707 ± 0.003	10.00	0.018 ± 0.002	10.00
EE	0.747 ± 0.002	0.977	0.025 ± 0.001	0.43 ± 0.05
MC EE	0.747 ± 0.001	0.982	0.015 ± 0.001	8.59 ± 0.03

Table 4.3: The best accuracy and ECE for each of the tested approaches for the VGG-19 on Cifar100, alongside the associated FLOPs cost. For ECE and FLOPs, smaller is better.

As with the MSDNet, the MC EE models of both the VGG-19 and the ResNet-18 are highly sensitive to the chosen hyperparameters. Of the 830 tested ResNet-18 and VGG-19 configurations, 152 had both a worse accuracy and ECE than the original baseline model.

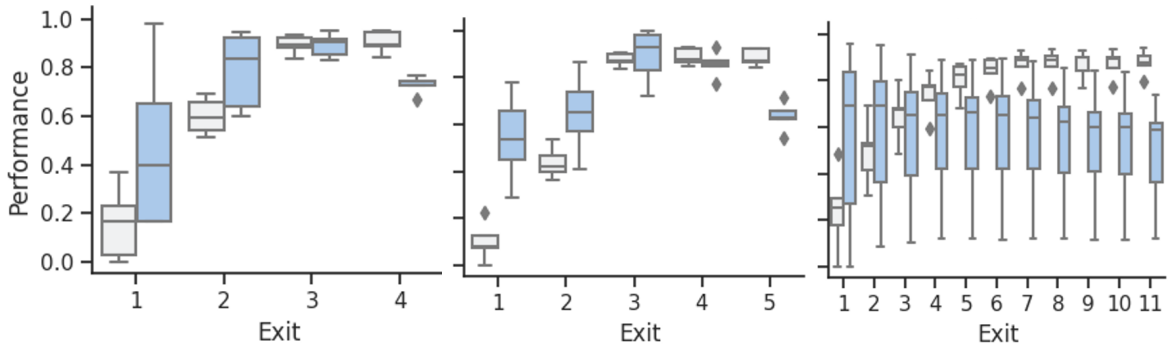
These results show that MC EE models can improve calibration over the two highest performing alternatives from the literature across a variety of architectures. However, this often comes at a tradeoff of either cost or accuracy, and careful tuning of the hyperparameters is needed to get optimal performance.

The results are validated by comparing the standard EE models for the ResNet-18 and the VGG-19 to those published in (56), which uses a similar architecture and loss. On average, the published ResNet-18 and VGG-19 results had a 1.6 ± 0.2 and a 0.9 ± 0.7 better accuracy score than those reported in this paper. The differences are likely due to the creation of a validation set and the more general learning rate reduction method used in this paper, although it is noted that (56) does not state whether the full Cifar100 training set is used.

Although the VGG-19 results are quite similar, the architecture used in this paper is likely different from that used in (56). The authors did not publish the modifications they made to the VGG-19 to get it to work with Cifar100, so it is unclear how they dealt with the dropout in the original design or where they placed their exits. Furthermore, the pooling layers were used to define the placement of the exits for their ResNet-18, however they only add three early exits to the VGG-19 compared to the four used in this paper, despite there being four pooling layers in the network.

	Accuracy	FLOPs	ECE	FLOPs
Baseline	0.752 ± 0.002	1.00	0.0840 ± 0.0008	1.00
MC	0.758 ± 0.002	1.00	0.019 ± 0.001	10.00
EE	0.7719 ± 0.0006	1.026 ± 0.003	0.017 ± 0.002	1.026 ± 0.003
MC EE	0.776 ± 0.001	1.019 ± 0.004	0.014 ± 0.001	0.672 ± 0.003

Table 4.4: The best accuracy and ECE for each of the tested approaches for the ResNet-18, alongside the associated FLOPs cost. For ECE and FLOPs, smaller is better.



(a) ResNet-18 **(b)** VGG-19 **(c)** MSDNet

Figure 4.1: A boxplot of the scaled average of accuracy and ECE for EE (white) and MC EE (blue) models for the ResNet-18 **(a)**, VGG-19 **(b)** and MSDNet **(c)** on Cifar100.

4.2.2 Trends in VGG-19 and ResNet-18

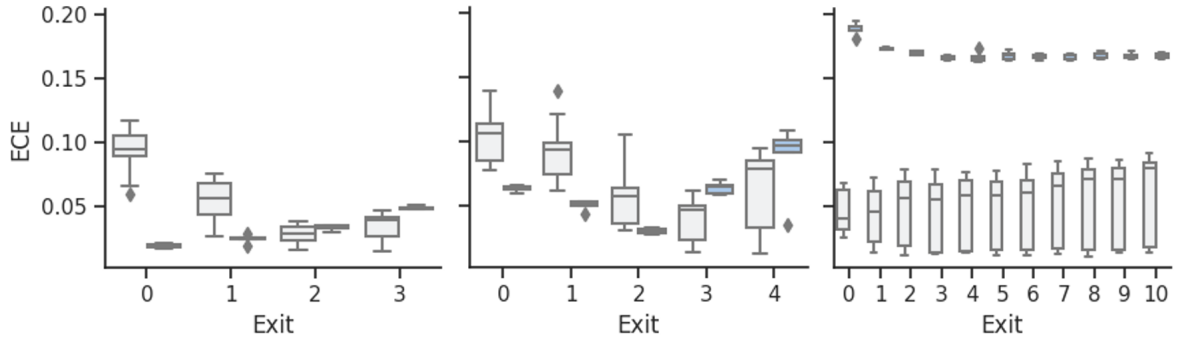
As with the MSDNet, the majority of Tables 4.1 and 4.2 are dominated by MC EE with dropout in the exit only. This type of MC EE is consistently able to create models which are accurate, calibrated and cheap. However, there are a few key differences between the trends for the MSDNet and those for the ResNet-18 and VGG-19.

For the VGG-19 and to a lesser extent the ResNet-18, the confidence thresholds in the best performing networks are lower than they were in the MSDNet. Rather than a confidence of 0.9+, the best results are instead often found around the 0.5-0.7 range.

This difference is likely due to the relative performance of each exit as one progresses through the model. The scaled average of the accuracy and ECE of the predictions made at each exit of the MC EE model with exit only dropout is plotted in Figure 4.1. It can be observed that for all model types, the later layers tend to plateau in performance. However, in both the ResNet-18 and the VGG-19 MC EE models, the final exit performance drops significantly compared to the previous exit.

A lower confidence threshold will then likely lead to better results. Confidence exiting uses the prediction at the final exit unless a confidence threshold is met at an earlier exit. If the final classifier is worse, then better performance for cheaper can be achieved by using a lower threshold to guarantee early exiting. It is noted that while performance does slightly go down on average for the later MSDNet exits, the higher number of exits means that a sufficiently large threshold is needed to prevent exiting in the earliest few exits, which may explain why the thresholds are generally higher.

One possible explanation for this drop in performance is that the implicit ensemble formed becomes less diverse for the later exits. For MC EE with dropout only in the exit, the dropout layer is a smaller part of the model in the later exits, and hence may not lead to as much change in predictions as it does for earlier exits. While this is true for most of the later exits, the reduction in diversity is likely offset by the increasing accuracy. The later exits can utilize more convolutional layers, leading to increased predictive power compared to the earlier exits.



(a) ResNet-18 **(b)** VGG-19 **(c)** MSDNet

Figure 4.2: A boxplot of the ECE for EE (white) and MC EE (blue) models for the ResNet-18 **(a)**, VGG-19 **(b)** and MSDNet **(c)** on Cifar100.

It is only in the last few exits that the accuracy may start to plateau, and hence this may explain why the highest performance in all models is usually achieved before the final layer. The drop is likely more prominent in the ResNet-18 and VGG-19 simply due to having fewer exits throughout the network, and hence the performance drop occurs more suddenly.

This is an example of the balancing act that occurs between accuracy and diversity in ensembles, which was discussed in Section 2.2.

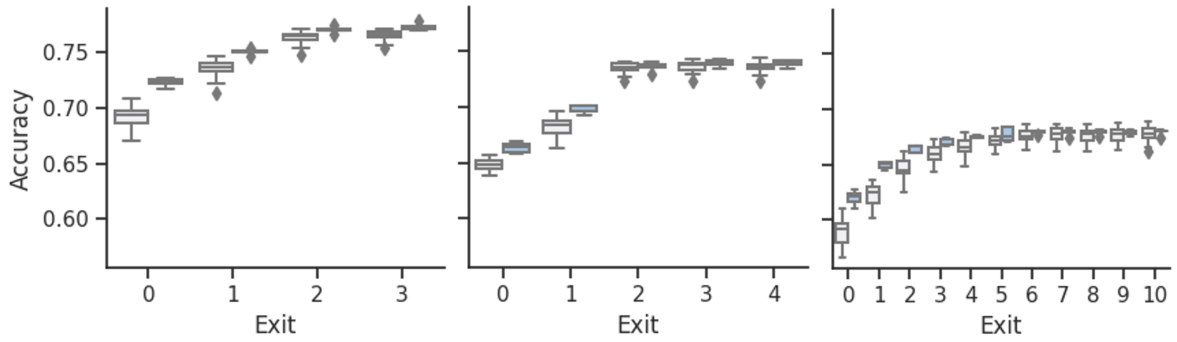
4.2.3 Why MC EE Works

It has been demonstrated that MC EE can create better calibrated models, however it is unclear why this occurs. The ablation studies performed above seem to demonstrate that most of the benefits from the combined approach come from using the early exiting models as an ensemble, which is inline with results from the literature (30). It was hypothesized in 2.3 that combining MC dropout and early exit ensembling could lead to better results due to improving the individual networks calibration and performance over the standard multi-exit models. To test this hypothesis, the individual exit performances of the five best calibrated MC EE models are plotted against the EE model in Figure 4.2.

The results for the MSDNet seem to support the hypothesis, with MC dropout drastically reducing the ECE of the results from each of the individual exits. However, this is not consistent across models. For both the ResNet-18 and the VGG-19, MC dropout seems to make the early exits worse, and then improves the later exits.

The patterns observed in Figure 4.2 are likely due to a similar phenomenon to that discussed in Section 3.2.6: As accuracy increases, the ECE is likely to go down as the highest confidence prediction is more consistently the correct answer. However, this only works to a degree, because if the confidence of prediction is much higher than the frequency with which it is correct, this negatively affects the ECE. This could explain why the ECE in the standard EE model seems to decrease for the first few layers before increasing for the later exits.

In the MC EE models, these first few layers are likely less calibrated because of



(a) ResNet-18 **(b)** VGG-19 **(c)** MSDNet

Figure 4.3: A boxplot of the accuracy for EE (white) and MC EE (blue) models for the ResNet-18 **(a)**, VGG-19 **(b)** and MSDNet **(c)** on Cifar100.

the regularizing effects of MC dropout. The accuracy at each exit of the best calibrated MC EE models are plotted against those of the EE models in Figure 4.3. Across all models, the first few exits dip in performance compared to their EE counterparts, and this difference in accuracy is likely driving the difference in ECE. However, once the accuracies are similar in the later layers, the improved uncertainty quantification induced by the MC drop method is able to reduce the ECE.

Given the above results, it may be surprising that the MC EE models are able to improve calibration as they generally seem to worsen average performance across the exits. However, the impact of the worse earlier exits can largely be avoided through confidence exiting. Appropriate selection of the confidence threshold allows the model to exit primarily in the later exits. This likely explains why the best calibrated models in Tables 4.1b and 5.2b are all MC EE models that use confidence exiting.

4.2.4 Limitations

The analysis of Figure 4.3 suggests that even for the best calibrated models, the amount of dropout used was too strong for the earlier exits, leading to poor accuracy and ECE. This is likely due to some of the choices made when implementing the MC EE. To limit the number of possible MC EE models to test, the same dropout rate was used for all layers in the network, however it is likely that the optimal model may require using a lower dropout rate in the earlier sections of the model and a higher dropout rate in the later parts. Additionally, only four rates of dropout and four types of dropout configurations per model were tested. A larger and more granular grid search is likely to give higher performing MC EE models. This choice may have prevented this report from demonstrating the full potential of the MC EE models.

The decision to use a different training method for the VGG-19 and ResNet-18 than the one used for MSDNet is also likely a limitation. While the choice helped demonstrate that the method also works for multiple training methods and meant that a literature baseline could be used to validate the results, it is difficult to disentangle the observed differences in performance across models. The drop in per-

formance observed between the MSDNet and the other models could be due to the different types of architecture or due to the different training methods, and hence it is unclear just how model agnostic the method truly is.

Another limitation is that block dropout configuration was only ever used with dropout in the exits. Both the block and exit dropout configurations introduce a similar number of dropout layers, however their placement is different. It is possible that the block configuration may be able to introduce more diversity for the same amount of regularization due to changing the feature extraction step rather than the classification one. Hence, it may have been able to provide better results than those achieved with just the exit dropout configuration.

4.3 Conclusion

Regardless of the model architecture, the MC EE method can be used to develop the best performing models across all metrics on the Cifar100 dataset. An analysis of the individual exit performances when compared to a standard multi-exit model reveals that most of the benefits of adding MC dropout to early exit architectures occur through making the later stages of the network better calibrated. Due to computational constraints, the same dropout rate was used across all dropout layers in the network. This was found to over-regularize the early exits, and it is likely that optimal tuning of these early dropout layers could lead to even better performances by the MC EE models. As with the MSDNet, poor choice of the amount of dropout or confidence threshold can lead to worse results than the standard baselines for both the ResNet-18 and the VGG-19.

Chapter 5

Extending to Harder Datasets

The combined Monte Carlo and Early Ensembling technique has proven to be effective across multiple different types of multi-exit architectures. However, all the models have been tested on the Cifar100 dataset. As mentioned in Section 3.1.2, Cifar100 is an ideal benchmarking dataset because it has small images and a clear central object. These types of images are optimal for machine learning, allowing models to achieve extremely high classification accuracies (35; 56).

However, many of the images in practical applications will not be as ideal. Particularly in the medical domain, for which accurate uncertainty quantification is a necessity, the datasets are often imbalanced and complex. In classification tasks, the manifestations of the pathologies tend to be small and difficult to discern, such that even humans need years of study to confidently make a diagnosis (18; 21). Hence, it is important that a method can overcome challenge C4 and give high performing results for harder, more realistic data.

As discussed in Section 1.1, chest x-rays are a type of medical imaging that would benefit significantly from the development of machine learning models due to their popularity and difficulty to interpret. Automated systems that effectively quantify uncertainty would be invaluable, as they could help ease the burden on medical staff. Furthermore, there are multiple large, publicly available chest x-ray datasets available which make it viable to train machine learning models in the domain.

To test whether the method presented can generalize to harder datasets than the natural images of Cifar100, it is applied using the VGG-19 model to the publicly available ChestX-ray 14 dataset for multi-classification.

The best MC EE model is found to be able to match the most calibrated alternative, an MC model, while costing only slightly more than half the amount of FLOPs per prediction. The most accurate MC EE model is also able to improve the overall accuracy score by almost 0.05 over the best alternative, an EE model. Unlike with the Cifar100 dataset, the addition of any MC dropout is found to improve the overall ECE compared to the standard multi-exit architecture. It is hypothesized that due to the smaller number of samples, more diversity is needed to generate more calibrated ensembles.

5.1 Method

To implement the VGG-19 on the ChestX-ray 14 dataset, modifications need to be made to both the dataset and the model.

5.1.1 Preparing Dataset

Obtained from the National Institute of Health in the United States, the ChestX-ray 14 dataset contains labeled chest x-rays of 14 different common thoracic pathologies, (33). The labels are extracted using Natural Language Processing techniques to provide one of the largest publicly available labeled chest x-ray datasets, with over 110,000 chest x-rays (33). However, as with most chest x-ray datasets, many of the x-rays contain multiple pathologies at once.

Therefore, models used on this dataset tend to perform multi-label classification, in which all the associated labels of an x-ray are predicted simultaneously, or one-vs-all, in which only one disease is classified at a time (15). However, a multi-classification dataset was used in the experiments run in both Chapter 3 and Chapter 4, in which each image has only one associated label belonging to the C possible classes.

Significant modification to both the loss functions and architectures would be needed to allow the previously discussed methods to be used for multi-label or binary classification, and would make it difficult to compare results across experiments. Hence, instead the procedure in (82) is followed to convert the dataset into a multi-classification one.

As there are multiple images per patient, to avoid overlap the train, validation and test sets are formed through random splitting of the 30805 patients rather than the images, as in (15). Following (82), a total of 86,271 images which contain more than one pathology, “No finding” or infrequent pathologies are removed. To prevent imbalances in the dataset, (82) removes images which contain the 6 least frequent pathologies (”Pneumonia”, ”Edema”, ”Emphysema”, ”Fibrosis”, ”Pleural Thickening”, ”Hernia”). This is also followed here. This gives a train, validation and test set size of 20695, 2508 and 2645 chest x-rays respectively.

The x-rays are quite large at 1024 x 1024 pixels, and hence they are often resized to a smaller, more manageable image size (82; 15; 33). In this report, bicubic downsampling and center cropping is used to convert the images to the standard ImageNet 224 x 224 size, as in (82). Following from (15), the images are all normalized using the mean and standard deviation of the ImageNet dataset, and random horizontal flipping is used as an augmentation for the train dataset.

5.1.2 Model

Due to time constraints, only one of the three previous models is tested on the chest x-ray dataset. The VGG-19 is chosen as it requires the least modification compared to the version used previously in Chapter 4, with only the final classifying layer being changed. The original classifier developed in (55), consisting of three fully

connected layers separated by standard dropout, is used instead of the single fully connected layer for the baseline and standard EE model experiments.

For the MC and MC EE models, when dropout is added in the exit branches, rather than adding it before the classifier as in Chapters 3 and 4, the standard dropout layers in the classifier are replaced with MC dropout, and hence are not turned off at inference time.

Given the relatively small size of the dataset, it is common to use the pre-trained ImageNet weights of a network before applying it to the new dataset (15; 33). This same procedure is applied here, although not to the final classifier. These layers were excluded for fairness of comparison, as no weights exist for the early exit classifiers.

5.1.3 Training Parameters

Due to GPU memory limitations, gradient accumulation is used as in (33). Rather than updating the network parameters after iterating through a batch of images, the network is only updated after a certain number of iterations. The gradients across each of the batches are accumulated, and hence when the network is updated it is the equivalent of using a batch size of $original\_batch\_size \times iterations\_per\_step$ (33). Based off of the results of (83), the network is updated every 16 steps and a batch size of 16 is used to give an effective batch size of 256. Preliminary investigations showed that this gave a performance improvement over the effective batch size of 80 used in (33).

The Adam optimizer is used with the same parameters as in (15), except for a smaller learning rate of 0.0005 based off of the findings of (83). The linear layers are randomly initialized between 0 and 0.01, as in (55). Gradient clipping is also used to improve the stability of training.

5.2 Results

The same combinations of dropout and dropout rate are tested for the VGG-19 as in Chapter 4. However, due to time constraints the L+E MC EE model type was not trained, and only 3 trials of each result were done. The accuracy, ECE, and FLOP percentage of all the prediction sets are reported in Appendix A.

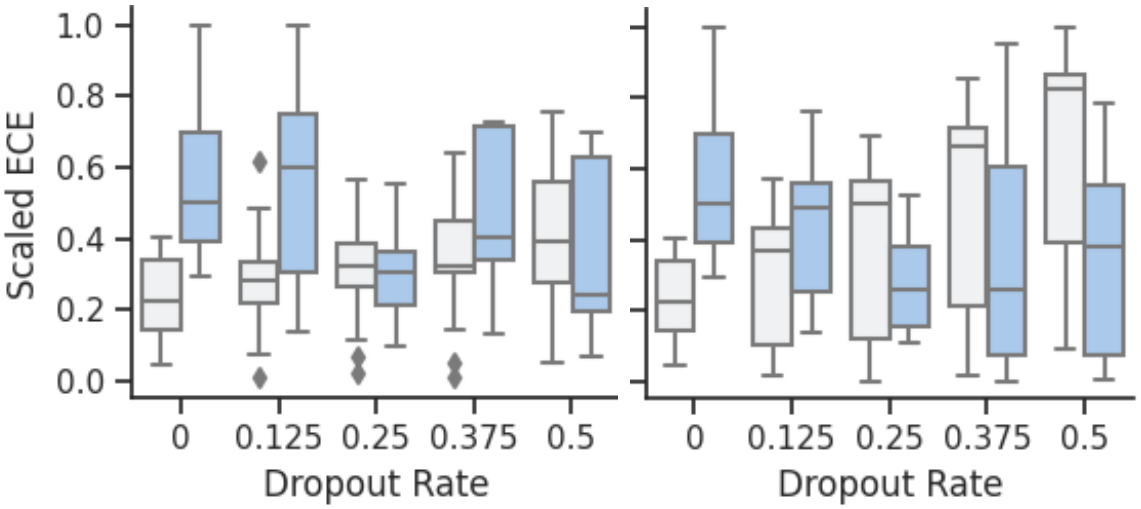
5.2.1 Key Findings

The best performance on each metric for the different model types are reported in Table 5.1, and demonstrate that MC EE models, as with the VGG-19 on Cifar100, can lead to improvements in accuracy and ECE over all other tested methods. One of the most calibrated models, B+E (0.5,0.1), is able to match the best calibrated model of all the other methods, L+E (0.25), while only costing a quarter of the computational cost and having a similar accuracy.

However, as with most of the reported results in this report, these improvements often come at a tradeoff. While the best calibrated MC EE methods are able to reduce

	Accuracy	FLOPs	ECE	FLOPs
Baseline	0.495 ± 0.008	1.00	0.08 ± 0.01	1.00
EE	0.547 ± 0.003	0.7 ± 0.2	0.05 ± 0.02	0.13
MC	0.53 ± 0.01	10.00	0.017 ± 0.003	10.00
MC EE	0.573 ± 0.006	7.6 ± 0.3	0.017 ± 0.001	5.50

Table 5.1: The best accuracy and ECE for each of the tested approaches for the VGG-19 on ChestX-ray 14, alongside the associated FLOPs cost. For ECE and FLOPs, smaller is better.



(a) Exit **(b)** Block + Exit

Figure 5.1: A boxplot of the scaled ECE of the VGG-19 on Cifar100 (white) and ChestX-ray 14 (blue) for exit only MC EE **(a)** and B+E MC EE models **(b)**.

the ECE by 0.03 ± 0.02, they cost at least 2.5 times as much as the EE or Baseline methods and reduce the overall accuracy score by at minimum 0.017 ± 0.008 .

As usual, the most efficient results are achieved by the dropout in exit only MC EE models, with models like the E (0.125, 3) able to produce more accurate and better calibrated models than the baseline while costing less. However, unlike with the VGG-19 on Cifar100, the most accurate and the best calibrated MC EE models are achieved by the B+E type MC EE models. This is explored further in the next section.

5.2.2 Trends across Datasets

The block dropout configuration MC EE models performed significantly better compared to the other MC EE models on the ChestX-ray 14 dataset. This was not observed to the same extent in the Cifar100 dataset for any of the three models tested. To investigate this trend, the ECE scores from the VGG-19 on the Chest X-ray 14 dataset and the Cifar100 dataset are plotted in Figures 5.1a and 5.1b

For the chest x-ray dataset almost any level of dropout is able to improve the

	Accuracy	ECE	FLOPs
B+E (0.25,0.7)	0.573 ± 0.0059	0.06 ± 0.02	7.6 ± 0.3
B+E (0.25,4)	0.571 ± 0.007	0.037 ± 0.008	9.34
B+E (0.125,0.9)	0.569 ± 0.003	0.09 ± 0.03	8.763 ± 0.214
B+E (0.125,4)	0.569 ± 0.003	0.08 ± 0.03	9.34
B+E (0.5,0.9)	0.568 ± 0.004	0.07 ± 0.02	9.3395 ± 0.0002

(a) Most Accurate

	Accuracy	ECE	FLOPs
B+E (0.375,2)	0.3732 ± 0.00	0.017 ± 0.001	5.50
B+E (0.5,0.1)	0.46 ± 0.04	0.017 ± 0.001	2.533 ± 0.002
L+E (0.25)	0.478 ± 0.003	0.017 ± 0.002	10.00
B+E (0.5,2)	0.373 ± 0.00	0.017 ± 0.002	5.50
B+E (0.375,0.25,	0.42 ± 0.05	0.022 ± 0.006	2.951 ± 0.001

(b) Lowest ECE

	Accuracy	ECE	FLOPs
B+E (0.25,0.7)	0.573 ± 0.006	0.06 ± 0.02	7.6 ± 0.3
B+E (0.25,4)	0.571 ± 0.007	0.037 ± 0.008	9.34
B+E (0.125,0.9)	0.569 ± 0.003	0.09 ± 0.03	8.8 ± 0.2
B+E (0.125,4)	0.569 ± 0.003	0.08 ± 0.03	9.34
B+E (0.5,0.9)	0.568 ± 0.004	0.07 ± 0.02	9.34

(c) Best Overall

	Accuracy	ECE	FLOPs
E (0.125,0.7)	0.562 ± 0.004	0.056 ± 0.009	1.01 ± 0.03
E (0.125,4)	0.561 ± 0.007	0.11 ± 0.03	1.01
E (0.25,3)	0.557 ± 0.004	0.05 ± 0.02	0.90
E (0.25,0.7)	0.554 ± 0.004	0.07 ± 0.02	0.83 ± 0.03
E (0,0.5)	0.547 ± 0.006	0.07 ± 0.03	0.6 ± 0.1

(d) Most Efficient

Table 5.2: The best 5 tested VGG-19 models on ChestX-ray 14 according to: **(a)** Accuracy, **(b)** ECE, **(c)** Scaled Average of Accuracy and ECE, **(d)** Scaled Average of Accuracy, ECE and FLOPs.

ECE, with more significant levels of dropout, like models with B+E and $p = 0.375$, giving the biggest benefit. For the Cifar100 dataset, dropout generally seems to worsen the ECE, although some outliers give highly calibrated results.

One possible explanation for this pattern is the difference in the number of possible classes. For the Cifar100 dataset, there are 100 possible classes that each image could be, compared to the 7 in the ChestX-ray 14 dataset. The smaller number of classes mean that model predictions are more likely to be similar, simply due to having fewer options to choose from. Hence, the higher diversity introduced by larger amounts of dropout may be critical in making the implicit ensemble better calibrated.

Another possible contributing factor is that the VGG-19 models used for the chest x-rays are initialized with weights from the ImageNet trained VGG-19, in which the classifier had two dropout layers. Hence, the resulting model after training may have inherited some of this robustness to dropout, further reducing the diversity in results.

The presence of dropout layers with rates of 0.5 in the standard classifier may explain why the least calibrated models came from the MC EE model with exit only dropout rates of 0.125. It is suggested that the VGG-19 model for larger images may benefit from the regularization introduced by the dropout, and hence using the lower dropout rate may actually under-regularize it, leading to worse performance.

5.2.3 Limitations

One of the largest limitations of the experiments done in this chapter is the lack of a similar result in the literature to validate against. While (82) followed a similar method, they used a ResNet-50 and did not perform end-to-end training on the chest x-ray dataset, instead freezing the weights and investigating how well the model performs with various transfer learning methods. Hence, while the models in this paper significantly outperform all those discussed in (82) by over 10%, this can only provide minor validation as end to end models should outperform transferred models.

Due to time constraints, the L+E MC EE models were not run. While these MC EE models typically performed quite poorly on the Cifar100 dataset, the L+E MC models were some of the most calibrated models tested other than a few MC EE models. This suggests that L+E MC EE models would have likely done well on this dataset, and hence not including them may be underselling the capabilities of the MC EE models.

A final limitation is that only dropout rates under 0.5 were tested. While that decision seems to be valid for the previous chapters on Cifar100, in which the higher dropout would have over-regularized the network, for the VGG-19 the original authors had to introduce regularization to get optimal performance (55). Given that the MC EE models which performed best had high levels of dropout, either through more dropout layers or higher dropout rates, it is likely that high performance could also be achieved for dropout rates over 0.5. This may have led to a further underestimate of the optimal uncertainty quantification achievable by both the MC EE models

and the MC models.

5.3 Conclusion

High performing models can be found across multiple datasets and architectures through combining MC dropout and multi-exit ensembling. However, unlike with the models on the Cifar100, a more significant level of dropout is needed to yield the best uncertainty quantification. The lack of a clear best type of MC EE model necessitates the use of hyperparameter tuning, which due to the large number of hyperparameters can be prohibitively expensive. A larger search of the hyperparameter space for the MC EE models is likely to yield even better results than those reported in this chapter.

Chapter 6

Report Conclusion

In this report, a new approach to improve the uncertainty quantification in standard CNN models through combining Monte Carlo dropout and multi-exit ensembling is tested across three models and on two datasets. These results are then compared against the best methods from the literature.

6.1 Review of Results

For all models and datasets, the best calibrated and the most accurate results were achieved by the MC EE models. Often, these improvements could be achieved for similar or cheaper cost. These results were consistent across two different early exiting architecture types and two completely different imaging datasets, suggesting that the method may be generalizable.

The multiple ways of creating an MC EE model gives the approach significant flexibility, allowing one to optimize the model to the needs of the task. The levels of dropout can be tuned to adjust the tradeoff between accuracy and calibration, and confidence thresholding can be used to reduce the computational burden.

Due to computational and time constraints, various limits were placed on the types of MC EE models tested. It is suggested that even better results than those achieved in this report can be achieved through using the trends observed here to develop a more careful treatment of the different dropout locations and rates.

However, this flexibility is a double-edged sword. Simply introducing MC dropout to a multi-exit architecture is not guaranteed to give better results. Often, poor choice of parameters can lead to worse accuracy and calibration than even the baseline models. This necessitates the need for a large amount of hyperparameter tuning, which for massive models or datasets can make the development of an effective MC EE model prohibitively expensive.

Various trends identified in this report can be used to narrow down this tuning procedure. Generally, the most cost effective models are those that only introduce dropout in the exit branches of the network. A confidence threshold of over 0.5 is usually needed for the best accuracy and the best calibration. Using a Block dropout configuration often leads to the best calibration, however comes with a significantly increased computational burden.

However, while these trends can give general guidance, there is no rule of thumb. While low levels of dropout were needed for the best results on Cifar100, the highest dropout combinations gave the optimal results on ChestX-ray 14. Ensembling was effective for the MSDNet but not as much for the ResNet or the VGG-19. Further investigation is required to narrow down the best method of optimizing MC EE networks.

6.2 Ethical Considerations

While the primary focus of this report has been on Cifar100, which does not contain images of humans, Chapter 5 uses human chest x-ray images from the ChestX-ray 14 dataset. The sensitive medical information which is contained in a chest x-ray makes it important that the ethical and legal ramifications are considered.

The chest x-ray dataset is obtained from patients at the National Institute of Health Clinical Center (33). The data has been pseudonymised through replacing the names of patients and covering up any identifying information in the chest x-ray itself, such that the individual cannot be identified (84). Furthermore, these chest x-rays have been obtained with informed consent about its potential use in research, as discussed on their website here: `https://clinicalcenter.nih.gov/participate/patientinfo/legal1.html`.

It is noted that while a recent study has been able to link individuals through their chest x-rays, to fully identify the individual required knowing which x-ray belonged to which individual, information which is not available from the dataset (84). Hence, the ChestX-ray 14 dataset is likely anonymized enough to protect the individual's privacy under the relevant regulations.

No other ethical considerations from the ethical checklist are noted.

6.3 Future Works

The findings of this report have provided a preliminary assessment of a new, highly effective uncertainty quantification method. However, more research is needed to fully understand its benefits and drawbacks.

While a broad range of dropout combinations were tested in this report, certain choices were made to limit the number of models to test. As discussed in Section 4.2.4, the choice to use the same dropout rate across all layers may limit performance, as the earlier exits tend to be hampered by the high dropout needed for the improved calibration in the later exits. MC EE models with lower levels of dropout in the earlier parts of the model are likely to give better performance, and an investigation into these types of configurations is likely to demonstrate the full capabilities of the approach.

Another area which warrants investigation is the best method to add the contributions of each exit in the ensemble. As discussed in Section 3.2.7, the choice to use the largest possible unweighted ensemble may not be appropriate for early exit

ensembling methods, as the early exits are usually worse than the later exits. Additional experimentation to figure out what weightings work best to maximize both the diversity without harming the average accuracy will likely be important to maximize the benefit of both the MC EE approach in this report and the general early exit ensembling method from the literature (30).

As discussed in Section 2.1.2, modifications have been introduced to the MC dropout method to make it a better estimator of uncertainty. These include using different types of dropout or introducing slight changes to the existing loss function. These were not employed in this report, however could potentially be used to further improve the calibration of multi-exit architectures.

Another key measure of a calibrated model is how well it maintains its calibration under a shift of dataset (30). This is particularly important in the medical imaging domain, as shifts can occur due to a variety of minor differences like different imaging machines or chosen imaging settings. It has been demonstrated that many common uncertainty quantification approaches are not robust to changes in the dataset, and hence exploring this aspect of the MC EE approach will be vital before it can be applied in many safety-critical settings (26; 30).

While the proposed approach has been shown to produce similar results for less computation than the alternatives tested, there are a variety of additional software and hardware optimizations that could be made to further reduce the cost (25; 81; 80). Investigating to what extend this approach can be sped up while maintaining performance can help clarify the practical applicability of MC EE models. To this end, a paper on an FPGA-based tool to convert models to a multi-exit bayesian equivalent is being prepared based on the results in this report.

The general nature of the approach could allow for its widespread use across a variety of domains. While the results in this report have demonstrated its effectiveness on CNN based multi-exit architectures for multi-classification, it remains to be seen whether it can generalize to other high performance multi-exit model types like those based on transformers or to binary and multilabel classification tasks (58). Evaluating the approach on a wider range of models and datasets is needed to fully understand the applicability of the approach, however the initial results show that this is a promising method which can significantly improve uncertainty quantification in machine learning models. (85) (86; 87; 88; 89)

Bibliography

[1] The Physicians Foundation. 2021 Survey of America's Physicians Covid-19 Impact Edition: A Year Later; 2021. pages 1

[2] Kang L, Ma S, Chen M, Yang J, Wang Y, Li R, et al. Impact on Mental Health and Perceptions of Psychological Care among Medical and Nursing Staff in Wuhan during the 2019 Novel Coronavirus Disease Outbreak: a Cross-sectional Study. Brain, Behavior, and Immunity. 2020 03;87. pages 1

[3] Weaver MD, Landrigan CP, Sullivan JP, O'Brien CS, Qadri S, Viyaran N, et al. National improvements in resident physician-reported patient safety after limiting first-year resident physicians' extended duration work shifts: a pooled analysis of prospective cohort studies. BMJ Quality & Safety. 2022. Available from: https://qualitysafety.bmj.com/content/early/2022/05/09/bmjqs-2021-014375. pages 1

[4] Kupperschmidt B. 12 Hour Shifts: Literature Reviewed, Wise Use Challenged. Journal of Christian nursing : a quarterly publication of Nurses Christian Fellowship. 2018 01;35:26-32. pages 1

[5] The Rosters Study Group, Rahman S, Sullivan J, Barger L, St Hilaire M, O'Brien C, et al. Extended work shifts and neurobehavioral performance in resident-physicians. Pediatrics. 2021 Mar;147(3). Publisher Copyright: © 2021 American Academy of Pediatrics. All rights reserved. pages 1

[6] Barger LK, Ayas NT, Cade BE, Cronin JW, Rosner B, Speizer FE, et al. Impact of Extended-Duration Shifts on Medical Errors, Adverse Events, and Attentional Failures. PLOS Medicine. 2006 12;3(12):1-9. Available from: https://doi.org/10.1371/journal.pmed.0030487. pages 1

[7] DWYER T, JAMIESON L, MOXHAM L, AUSTEN D, SMITH K. Evaluation of the 12-hour Shift Trial in a Regional Intensive Care Unit. Journal of Nursing Management. 2007;15(7):711-20. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2934.2006.00737.x. pages 1

[8] Rosenblatt RA, Andrilla CHA, Curtin T, Hart LG. Shortages of Medical Personnel at Community Health CentersImplications for Planned Expansion. JAMA. 2006 03;295(9):1042-9. Available from: https://doi.org/10.1001/jama.295.9.1042. pages 1

[9] Reynolds R, Chakrabarti R, Chylak D, Jones K, Iacobucci W, Dall T. The Complexities of Physician Supply and Demand: Projections From 2019 to 2034; 2021. pages 1

[10] Badia AP, Piot B, Kapturowski S, Sprechmann P, Vitvitskyi A, Guo D, et al.. Agent57: Outperforming the Atari Human Benchmark. arXiv; 2020. Available from: https://arxiv.org/abs/2003.13350. pages 1

[11] Khanjani Z, Watson G, Janeja VP. How Deep Are the Fakes? Focusing on Audio Deepfake: A Survey. arXiv; 2021. Available from: https://arxiv.org/abs/2111.14203. pages 1

[12] Jain J, Singh A, Orlov N, Huang Z, Li J, Walton S, et al. SeMask: Semantically Masked Transformers for Semantic Segmentation. CoRR. 2021;abs/2112.12782. Available from: https://arxiv.org/abs/2112.12782. pages 1

[13] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016. p. 770-8. pages 1, 10, 26

[14] Caron M, Touvron H, Misra I, Jegou H, Mairal J, Bojanowski P, et al. Emerging Properties in Self-Supervised Vision Transformers. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 2021. p. 9630-40. pages 1

[15] Rajpurkar P, Irvin J, Zhu K, Yang B, Mehta H, Duan T, et al.. CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. arXiv; 2017. Available from: https://arxiv.org/abs/1711.05225. pages 1, 2, 36, 37

[16] Montalt-Tordera J, Muthurangu V, Hauptmann A, Steeden JA. Machine learning in Magnetic Resonance Imaging: Image reconstruction. Physica Medica. 2021;83:79-87. Available from: https://www.sciencedirect.com/science/article/pii/S1120179721001095. pages 1

[17] Kadry S, Rajinikanth V, Rho S, Raja NSM, Rao VS, Thanaraj KP. Development of a Machine-Learning System to Classify Lung CT Scan Images into Normal/COVID-19 Class; 2020. pages 1

[18] Raghu M, Zhang C, Kleinberg JM, Bengio S. Transfusion: Understanding Transfer Learning with Applications to Medical Imaging. CoRR. 2019;abs/1902.07208. Available from: http://arxiv.org/abs/1902.07208. pages 2, 35

[19] Li T, Bo W, Hu C, Hong K, Liu H, Wang K, et al. Applications of Deep Learning in Fundus Images: A Review. Medical Image Analysis. 2021 01;69:101971. pages 2

[20] Esteva A, Kuprel B, Novoa R, Ko J, Swetter S, Blau H, et al. Dermatologist-level classification of skin cancer with deep neural networks. Nature. 2017 01;542. pages 2

[21] Lee S, Seo JB, Yun J, Cho YH, Vogel-Claussen J, Schiebler M, et al. Deep Learning Applications in Chest Radiography and Computed Tomography: Current State of the Art. Journal of Thoracic Imaging. 2019 03;34:1. pages 2, 35

[22] Çallı E, Sogancioglu E, van Ginneken B, van Leeuwen KG, Murphy K. Deep learning for chest X-ray analysis: A survey. Medical Image Analysis. 2021;72:102125. Available from: https://www.sciencedirect.com/science/article/pii/S1361841521001717. pages 2

[23] Rajpurkar P, Irvin J, Ball R, Zhu K, Yang B, Mehta H, et al. Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists. PLOS Medicine. 2018 11;15:e1002686. pages 2, 12

[24] Abdar M, Pourpanah F, Hussain S, Rezazadegan D, Liu L, Ghavamzadeh M, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. Information Fusion. 2021 05;76. pages 2, 5, 6, 7, 8, 11

[25] Fan H, Ferianc M, Que Z, Niu X, Rodrigues M, Luk W. Accelerating Bayesian Neural Networks via Algorithmic and Hardware Optimizations. IEEE Transactions on Parallel and Distributed Systems. 2022 01:1-1. pages 2, 15, 16, 44

[26] Ovadia Y, Fertig E, Ren J, Nado Z, Sculley D, Nowozin S, et al. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In: Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R, editors. Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc.; 2019. Available from: https://proceedings.neurips.cc/paper/2019/file/8558cb408c1d76621371888657d2eb1d-Paper.pdf. pages 2, 8, 24, 44

[27] Gal Y, Ghahramani Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48. ICML'16. JMLR.org; 2016. p. 1050–1059. pages 2, 3, 7, 8, 26

[28] Gal Y, Ghahramani Z. Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference. arXiv; 2015. Available from: https://arxiv.org/abs/1506.02158. pages 2, 7, 8, 16, 26

[29] Fort S, Hu H, Lakshminarayanan B. Deep Ensembles: A Loss Landscape Perspective. arXiv; 2019. Available from: https://arxiv.org/abs/1912.02757. pages 2, 8, 11

[30] Qendro L, Campbell A, Lio P, Mascolo C. Early Exit Ensembles for Uncertainty Quantification. In: Roy S, Pfohl S, Rocheteau E, Tadesse GA, Oala L, Falck F, et al., editors. Proceedings of Machine Learning for Health. vol. 158 of Proceedings of Machine Learning Research. PMLR; 2021. p. 181-95. Available from: https://proceedings.mlr.press/v158/qendro21a.html. pages 2, 3, 5, 9, 10, 11, 12, 13, 15, 16, 23, 32, 44

[31] Kaya Y, Hong S, Dumitras T. Shallow-Deep Networks: Understanding and Mitigating Network Overthinking. In: Chaudhuri K, Salakhutdinov R, editors. Proceedings of the 36th International Conference on Machine Learning. vol. 97 of Proceedings of Machine Learning Research. PMLR; 2019. p. 3301-10. Available from: https://proceedings.mlr.press/v97/kaya19a.html. pages 2, 3, 10, 11, 15, 16, 21

[32] Krizhevsky A. Learning Multiple Layers of Features from Tiny Images. University of Toronto. 2012 05. pages 3, 14

[33] Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM. ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 3462-71. pages 3, 36, 37, 43

[34] Phuong M, Lampert C. Distillation-Based Training for Multi-Exit Architectures. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 2019. p. 1355-64. pages 3, 10, 14, 19, 25, 26

[35] Huang G, Chen D, Li T, Wu F, van der Maaten L, Weinberger K. Multi-Scale Dense Networks for Resource Efficient Image Classification. In: International Conference on Learning Representations; 2018. Available from: https://openreview.net/forum?id=Hk2aImxAb. pages 3, 10, 14, 15, 26, 35

[36] Nwosu L, Li X, Qian L, Kim S, Dong X. Calibrated Bagging Deep Learning for Image Semantic Segmentation: A Case Study on COVID-19 Chest X-ray Image. arXiv; 2022. Available from: https://arxiv.org/abs/2206.00002. pages 3, 12

[37] Wang CS, Su FY, Lee TLM, Tsai YS, Chiang JH. CUAB: Convolutional Uncertainty Attention Block Enhanced the Chest X-ray Image Analysis. arXiv; 2021. Available from: https://arxiv.org/abs/2105.01840. pages 3, 12

[38] Ul Abideen Z, Ghafoor M, Munir K, Saqib M, Ullah A, Zia T, et al. Uncertainty Assisted Robust Tuberculosis Identification With Bayesian Convolutional Neural Networks. IEEE Access. 2020;8:22812-25. pages 3, 12

[39] Qendro L, Ha S, de Jong R, Maji P. Stochastic-Shield: A Probabilistic Approach Towards Training-Free Adversarial Defense in Quantized CNNs. In: Proceedings of the 1st Workshop on Security and Privacy for Mobile AI. MAISP'21. New

York, NY, USA: Association for Computing Machinery; 2021. p. 1–6. Available from: https://doi.org/10.1145/3469261.3469404. pages 3, 8, 11, 15

[40] Kendall A, Badrinarayanan V, Cipolla R. Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding. arXiv; 2015. Available from: https://arxiv.org/abs/1511.02680. pages 4, 8, 14, 15, 22

[41] Bishop C. Pattern Recognition and Machine Learning. Springer; 2006. Available from: https://www.microsoft.com/en-us/research/publication/pattern-recognition-machine-learning/. pages 5

[42] Guo C, Pleiss G, Sun Y, Weinberger KQ. On Calibration of Modern Neural Networks. In: Precup D, Teh YW, editors. Proceedings of the 34th International Conference on Machine Learning. vol. 70 of Proceedings of Machine Learning Research. PMLR; 2017. p. 1321-30. Available from: https://proceedings.mlr.press/v70/guo17a.html. pages 5, 8

[43] Blei DM, Kucukelbir A, McAuliffe JD. Variational Inference: A Review for Statisticians. Journal of the American Statistical Association. 2017;112(518):859-77. Available from: https://doi.org/10.1080/01621459.2017.1285773. pages 6

[44] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. J Mach Learn Res. 2014 January;15(1):1929–1958. pages 7

[45] Li Y, Gal Y. Dropout Inference in Bayesian Neural Networks with Alpha-Divergences. In: Proceedings of the 34th International Conference on Machine Learning - Volume 70. ICML'17. JMLR.org; 2017. p. 2052–2061. pages 8

[46] Folgoc LL, Baltatzis V, Desai S, Devaraj A, Ellis S, Manzanera OEM, et al.. Is MC Dropout Bayesian?. arXiv; 2021. Available from: https://arxiv.org/abs/2110.04286. pages 8

[47] Gal Y, Hron J, Kendall A. Concrete Dropout. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, et al., editors. Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc.; 2017. Available from: https://proceedings.neurips.cc/paper/2017/file/84ddfb34126fc3a48ee38d7044e87276-Paper.pdf. pages 8

[48] Pakdaman Naeini M, Cooper G, Hauskrecht M. Obtaining Well Calibrated Probabilities Using Bayesian Binning. Proceedings of the AAAI Conference on Artificial Intelligence AAAI Conference on Artificial Intelligence. 2015 04;2015:2901-7. pages 8

[49] Zhang J, Kailkhura B, Han TYJ. Mix-n-Match: Ensemble and Compositional Methods for Uncertainty Calibration in Deep Learning. In: Proceedings of the 37th International Conference on Machine Learning. ICML'20. JMLR.org; 2020. . pages 8

[50] Brown G, Wyatt J, Harris R, Yao X. Diversity creation methods: a survey and categorisation. Information Fusion. 2005;6(1):5-20. Diversity in Multiple Classifier Systems. Available from: https://www.sciencedirect.com/science/article/pii/S1566253504000375. pages 9, 23

[51] Lee CY, Xie S, Gallagher P, Zhang Z, Tu Z. Deeply-Supervised Nets. In: Lebanon G, Vishwanathan SVN, editors. Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics. vol. 38 of Proceedings of Machine Learning Research. San Diego, California, USA: PMLR; 2015. p. 562-70. Available from: https://proceedings.mlr.press/v38/lee15a.html. pages 9

[52] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015. p. 1-9. pages 9

[53] Huang F, Ash J, Langford J, Schapire R. Learning Deep ResNet Blocks Sequentially using Boosting Theory. In: Dy J, Krause A, editors. Proceedings of the 35th International Conference on Machine Learning. vol. 80 of Proceedings of Machine Learning Research. PMLR; 2018. p. 2058-67. Available from: https://proceedings.mlr.press/v80/huang18b.html. pages 9

[54] Belilovsky E, Eickenberg M, Oyallon E. Greedy Layerwise Learning Can Scale To ImageNet. In: Chaudhuri K, Salakhutdinov R, editors. Proceedings of the 36th International Conference on Machine Learning. vol. 97 of Proceedings of Machine Learning Research. PMLR; 2019. p. 583-93. Available from: https://proceedings.mlr.press/v97/belilovsky19a.html. pages 9

[55] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR. 2015;abs/1409.1556. pages 10, 36, 37, 40

[56] Lee H, Lee JS. Students are the Best Teacher: Exit-Ensemble Distillation with Multi-Exits. arXiv; 2021. Available from: https://arxiv.org/abs/2104.00299. pages 10, 11, 13, 14, 16, 25, 26, 30, 35

[57] Baccarelli E, Scardapane S, Scarpiniti M, Momenzadeh A, Uncini A. Optimized training and scalable implementation of Conditional Deep Neural Networks with early exits for Fog-supported IoT applications. Information Sciences. 2020;521:107-43. Available from: https://www.sciencedirect.com/science/article/pii/S0020025520301249. pages 10

[58] Bakhtiarnia A, Zhang Q, Iosifidis A. Multi-Exit Vision Transformer for Dynamic Inference. arXiv; 2021. Available from: https://arxiv.org/abs/2106.15183. pages 10, 13, 44

[59] Teerapittayanon S, McDanel B, Kung HT. BranchyNet: Fast inference via early exiting from deep neural networks. In: 2016 23rd International Conference on Pattern Recognition (ICPR); 2016. p. 2464-9. pages 10

[60] Bakhtiarnia A, Zhang Q, Iosifidis A. Improving the Accuracy of Early Exits in Multi-Exit Architectures via Curriculum Learning. arXiv; 2021. Available from: https://arxiv.org/abs/2104.10461. pages 10

[61] Scardapane S, Scarpiniti M, Baccarelli E, Uncini A. Why Should We Add Early Exits to Neural Networks? Cognitive Computation. 2020 jun;12(5):954-66. Available from: https://doi.org/10.1007%2Fs12559-020-09734-4. pages 10

[62] Leibig C, Allken V, Ayhan MS, Berens P, Wahl S. Leveraging uncertainty information from deep neural networks for disease detection. Scientific Reports. 2017 12;7. pages 11

[63] Wang G, Li W, Aertsen M, Deprest J, Ourselin S, Vercauteren T. Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. Neurocomputing. 2019;338:34-45. Available from: https://www.sciencedirect.com/science/article/pii/S0925231219301961. pages 11

[64] Nair T, Precup D, Arnold DL, Arbel T. Exploring uncertainty measures in deep networks for Multiple sclerosis lesion detection and segmentation. Medical Image Analysis. 2020;59:101557. Available from: https://www.sciencedirect.com/science/article/pii/S1361841519300994. pages 11

[65] Combalia M, Hueto F, Puig S, Malvehy J, Vilaplana V. Uncertainty Estimation in Deep Neural Networks for Dermoscopic Image Classification. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); 2020. p. 3211-20. pages 11

[66] Müller D, Soto-Rey I, Kramer F. An Analysis on Ensemble Learning optimized Medical Image Classification with Deep Convolutional Neural Networks. arXiv; 2022. Available from: https://arxiv.org/abs/2201.11440. pages 12

[67] Yang Y, Hu Y, Zhang X, Wang S. Two-Stage Selective Ensemble of CNN via Deep Tree Training for Medical Image Classification. IEEE Transactions on Cybernetics. 2022;52(9):9194-207. pages 12

[68] Xue D, Zhou X, Li C, Yao Y, Rahaman MM, Zhang J, et al. An Application of Transfer Learning and Ensemble Learning Techniques for Cervical Histopathology Image Classification. IEEE Access. 2020;8:104603-18. pages 12

[69] Logan R, Williams B, Silva M, Indani A, Schcolnicov N, Ganguly A. Deep Convolutional Neural Networks With Ensemble Learning and Generative Adversarial Networks for Alzheimer's Disease Image Data Classification. Frontiers in Aging Neuroscience. 2021 08;13:497. pages 12

[70] Dahal L, Kafle A, Khanal B. Uncertainty Estimation in Deep 2D Echocardiography Segmentation. arXiv:200509349 [cs]. 2020 May. ArXiv: 2005.09349. Available from: http://arxiv.org/abs/2005.09349. pages 12

[71] Çallı E, Sogancioglu E, van Ginneken B, van Leeuwen KG, Murphy K. Deep learning for chest X-ray analysis: A survey. Medical Image Analysis. 2021;72:102125. Available from: https://www.sciencedirect.com/science/article/pii/S1361841521001717. pages 12

[72] Mao Y, Xue FF, Wang R, Zhang J, Zheng WS, Liu H. Abnormality Detection in Chest X-Ray Images Using Uncertainty Prediction Autoencoders. In: Martel AL, Abolmaesumi P, Stoyanov D, Mateus D, Zuluaga MA, Zhou SK, et al., editors. Medical Image Computing and Computer Assisted Intervention – MICCAI 2020. Cham: Springer International Publishing; 2020. p. 529-38. pages 12

[73] Ghesu FC, Georgescu B, Gibson E, Guendel S, Kalra MK, Singh R, et al. Quantifying and Leveraging Classification Uncertainty for Chest Radiograph Assessment. In: Shen D, Liu T, Peters TM, Staib LH, Essert C, Zhou S, et al., editors. Medical Image Computing and Computer Assisted Intervention – MICCAI 2019. Cham: Springer International Publishing; 2019. p. 676-84. pages 12

[74] Calli E, Sogancioglu E, Scholten ET, Murphy K, van Ginneken B. Handling label noise through model confidence and uncertainty: application to chest radiograph classification. In: Mori K, Hahn HK, editors. Medical Imaging 2019: Computer-Aided Diagnosis. vol. 10950. International Society for Optics and Photonics. SPIE; 2019. p. 1095016. Available from: https://doi.org/10.1117/12.2514290. pages 12

[75] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Teh YW, Titterington M, editors. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. vol. 9 of Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR; 2010. p. 249-56. Available from: https://proceedings.mlr.press/v9/glorot10a.html. pages 14

[76] Mai VV, Johansson M. Stability and Convergence of Stochastic Gradient Clipping: Beyond Lipschitz Continuity and Smoothness. In: Meila M, Zhang T, editors. Proceedings of the 38th International Conference on Machine Learning. vol. 139 of Proceedings of Machine Learning Research. PMLR; 2021. p. 7325-35. Available from: https://proceedings.mlr.press/v139/mai21a.html. pages 14

[77] Torralba A, Fergus R, Freeman WT. 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2008;30(11):1958-70. pages 14

[78] Jungo A, McKinley R, Meier R, Knecht U, Vera Ramirez L, Pérez Beteta J, et al. In: Towards Uncertainty-Assisted Brain Tumor Segmentation and Survival Prediction; 2018. p. 474-85. pages 15

[79] fvcore library;. https://github.com/facebookresearch/fvcore/. pages 16

[80] Han S, Pool J, Tran J, Dally W. Learning both Weights and Connections for Efficient Neural Network. In: Cortes C, Lawrence N, Lee D, Sugiyama M, Garnett R, editors. Advances in Neural Information Processing Systems. vol. 28. Curran Associates, Inc.; 2015. Available from: https://proceedings.neurips.cc/paper/2015/file/ae0eb3eed39d2bcef4622b2499a05fe6-Paper.pdf. pages 16, 44

[81] Li H, Kadav A, Durdanovic I, Samet H, Graf HP. Pruning Filters for Efficient ConvNets. arXiv; 2016. Available from: https://arxiv.org/abs/1608.08710. pages 16, 44

[82] Yunhui G, Codella N, Karlinsky L, Codella J, Smith J, Saenko K, et al. In: A Broader Study of Cross-Domain Few-Shot Learning; 2020. p. 124-41. pages 36, 40

[83] Garyfallos S, Biseda B, Khan M. Improving on ChestX-ray8; 2019. Available from: https://github.com/paloukari/NIH-Chest-X-rays-Classification. pages 37

[84] Packhäuser K, Gündel S, Münster N, Syben C, Christlein V, Maier A. Deep learning-based patient re-identification is able to exploit the biometric nature of medical chest X-ray data. Scientific Reports. 2022 sep;12(1). Available from: https://doi.org/10.1038%2Fs41598-022-19045-3. pages 43

[85] Anton J, Castelli L, Tang WH, Cheung V, Outters M, Chan MF. How Well Do Self-Supervised Models Transfer to Medical Imaging?; 2022. pages 44

[86] Li D, Ling H, Kim SW, Kreis K, Barriuso A, Fidler S, et al.. BigDatasetGAN: Synthesizing ImageNet with Pixel-wise Annotations. arXiv; 2022. Available from: https://arxiv.org/abs/2201.04684. pages 44

[87] Irvin J, Rajpurkar P, Ko M, Yu Y, Ciurea-Ilcus S, Chute C, et al. CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison. CoRR. 2019;abs/1901.07031. Available from: http://arxiv.org/abs/1901.07031. pages 44

[88] Kaggle. Diabetic Retinopathy detection challenge; 2025. Available from: https://www.kaggle.com/c/diabetic-retinopathy-detection/. pages 44

[89] Pakdaman Naeini M, Cooper G, Hauskrecht M. Obtaining Well Calibrated Probabilities Using Bayesian Binning. Proceedings of the AAAI Conference on Artificial Intelligence AAAI Conference on Artificial Intelligence. 2015 04;2015:2901-7. pages 44

Appendix A

All Model Results

All results are available at this link: https://github.com/mailingliam02/MultiExit_BNNs/blob/main/All%20results.xlsx