CONTINUAL AND INCREMENTAL LEARNING IN COMPUTER-AIDED DIAGNOSIS SYSTEMS

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computer Science and Engineering by Research

by

PRATHYUSHA AKUNDI
2018701014
prathyusha.akundi@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
April 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled " CONTINUAL AND INCREMENTAL LEARNING IN COMPUTER-AIDED DIAGNOSIS SYSTEMS " by PRATHYUSHA AKUNDI, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Prof. JAYANTHI SIVASWAMY

To my parents,

who stood supportive of my decision to pursue higher studies.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Prof. Jayanthi Sivaswamy, for providing immense guidance and support to my research. My most significant motivational factors were her passion, love, sincerity, and respect for the Medical Image Processing field. As one of the renowned researchers, she always believed and taught her students that Medical Image Processing is not merely about feeding data to a Deep Learning Model. Thus, she ensured that we acknowledged the intricacies of Medical Imaging and incorporated a sense of responsibility into our research proposals. Also, her take on how to deal with rejections, rectify one's mistakes, and stand up for one's beliefs helped me in my research and personal life.

I feel fortunate to be a part of this fantastic group of researchers from MIP and primarily grateful to my seniors Sukesh Adiga, Abhinav Dhere, and Alphin Thottupattu for constantly guiding me through the research journey. Thank you, Soumyasis Gun, for keeping me on my toes and Mythri Venkatesh for making me realize that I am not alone.

Research is no fun without friends; thank you, Sowmya, Kiruthika, Mythri, Ruchi, and Tanu, for making this journey easy and turning it into a colorful bliss. I cherish our memories together, including our late-night fun, never-ending conversations, cooking our special one-pot pasta, and much more. A special thanks to our hostel and mess staff for making my first ever hostel experience comfortable and secure.

Abstract

Deep Neural Networks (DNNs) have shown remarkable performance in a broad range of computer vision tasks, including in the medical domain. With the advent of DNNs, the medical community has witnessed significant developments in segmentation, classification, and detection. But this success comes with a cost of heavy reliance on the abundance of data. Medical data, however, is often highly limited in volume and quality due to sparsity of patient contact, variability in medical care, and privacy concerns. Hence, to train large networks we seek data from different sources. In such a scenario, it is of interest to design a model that learns continuously and adapts to datasets or tasks as and when they are available.

However, one of the important steps to achieve such a never-ending learning process is to overcome Catastrophic Forgetting (CF) of previously seen data or tasks. CF refers to the significant degradation in performance on the old task/dataset. To avoid confusion, we call a training regime Continual Learning (CL) when CAD systems have to handle a sequence of datasets collected over time from different sites with different imaging parameters/populations. Similarly, Incremental Learning (IL) is when CAD systems have to learn new classes as and when new annotations are made available. The work described in this thesis address core aspects of both CL & IL and has been compared against the state-of-the-art methods.

In this thesis, we assume that access to the data belonging to previously trained datasets or tasks is not available which makes both CL and IL processes even more challenging. We start with developing a CL system that learns sequentially on different datasets and handles CF using the Uncertainty mechanism. The system consists of an ensemble of models which are trained or finetuned on each dataset and considers the prediction from the model which has the least uncertainty. We then investigate a new way to tackle CF in CL by manifold learning, inspired by the defense mechanisms against adversarial attacks. Our method uses a 'Reformer' which is essentially a denoising autoencoder that 'reforms' or brings the data from all the datasets together towards a common manifold. These reformed samples are then passed to the network to learn the desired task.

Towards IL, we propose a novel approach that ensures that a model remembers the causal factor behind the decisions on the old classes, while incrementally learning new classes. We introduce a common auxiliary task during the course of incremental training, whose hidden representations are shared across all the classification heads. All the experiments for both CL and IL are conducted on multiple datasets and have shown significant performance over the state-of-the-art methods.

Contents

List of Figures

List of Tables

Glossary

CAD Computer Aided Diagnosis

IL Incremental Learning

CL Continual Learning

CF Catastrophic Forgetting

EWC Elastic Weight Consolidation

SI Synaptic Intelligence

LwF Learning without Forgetting

LwM Learning without Memorizing

MER Meta Experience Replay

GEM Gradient Episodic Memory

MAS Memory Aware Synapses

iCARL Incremental Classifier and Representation Learning

PNN Progressive Neural Networks

DEN Dynamically Expandable Networks

PC Progress and Compress Networks

BWT Backward Transfer

FWT Forward Transfer

Chapter 1

Introduction

Medical imaging is a process of visualizing the human body's interior to monitor and diagnose medical conditions. There are various medical imaging techniques, also referred to as modalities, that use a broad spectrum of Electromagnetic waves for image acquisition: Radio frequency in MRI; Visible range in Endoscopy, Optical Coherence Tomography (OCT), and fundus photography; Sound in Ultrasound Scans; X-rays in radiography, CT Scans; Gamma-rays in Nuclear SPECT, PET imaging as shown fig 1.1. Computer-Aided Diagnosis (CAD) systems assist medical practitioners in interpreting these medical images and swiftly making decisions based on the analysis.

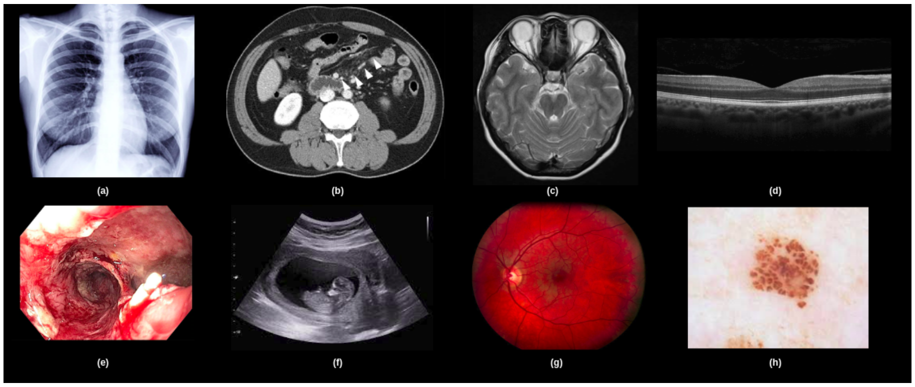


Figure 1.1: Sample Medical Images: (a) Chest X-Ray (b) CT scan of abdomen (c) MRI scan of Brain (d) OCT image (e) Endoscope image of colon (f) Ultrasound (g) Fundus image (h) Dermoscopic image

With the advent of Deep Neural Networks (DNNs), the medical research community has witnessed significant developments in segmentation, classification, and detection, paving the way to better CAD systems. Despite these rapid technical advancements, only a few CAD systems have successfully deployed on clinical premises, and part of the reason is the lack of robustness in the CAD systems. The following are some of the crucial factors that affect the performance of CAD systems.

1. **Lack of data:** Medical data is often highly limited in volume and quality due to the sparsity of patient contact, variability in medical care, and privacy concerns [8].

2. **Data from different sources:** To accumulate more data, we rely on multiple sources for image acquisition, and hence, the differences in distribution among sources due to changes in acquisition systems, the subject's demography, the quality/resolution, etc., are inevitable.

3. **Expert annotation:** Developing a medical dataset requires significant human effort, including acquisition and annotation. Hence, the process is usually slow and can take up to several months.

4. **Purging policies:** Medical data is subjected to multiple privacy regulations, and hence the data needs to be purged after the stipulated amount of time. Consequently, we can no longer access old data for joint training or enhancing the neural network model.

To address the above issues and to facilitate the early deployment of CAD systems, developing a model that learns and improves on the data as and when it is available is desirable. This is the problem addressed in this thesis.

1.1 Continual And Incremental Learning

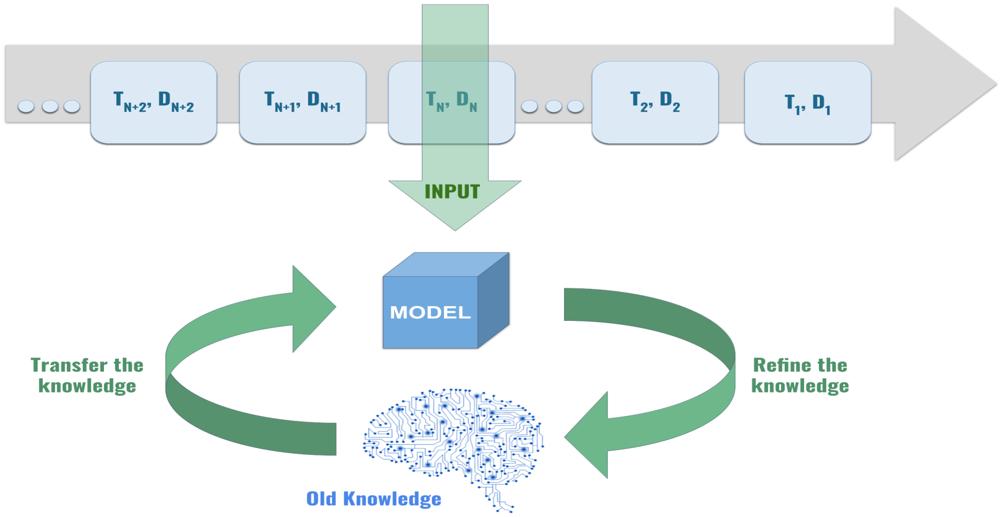


Figure 1.2: Illustration of a lifelong learning system that uses old knowledge to learn the new tasks, while maintaining the old knowledge without forgetting

The ability of the model to use the knowledge acquired from previous tasks in learning new tasks without losing the previously acquired knowledge is commonly referred to as Lifelong Learning. As illustrated in Figure 1.2, with an incoming stream of new data with a task T_n and a dataset D_n, the model should leverage the knowledge gained from old tasks and datasets without forgetting. We identified two main branches of Lifelong learning:

1. **Continual Learning (CL)**: It refers to learning from a sequence of datasets that differ only in distributions but not the tasks. The changes in the distributions, for example, a change in the scanner, may not be discernible to humans, but a neural network will see a big change via small aggregated changes. In a CL setting, given two tasks T_i and T_j along with their corresponding datasets D_i and D_j, we have $T_i = T_j$ and $D_i \neq D_j$

2. **Incremental Learning (IL)**: It refers to learning new classes from the dataset without forgetting old classes. In this case, the dataset distribution may or may not change, but the tasks are different, i.e., $T_i \neq T_j$.

1.1.1 Catastrophic Forgetting

The Continual and Incremental Learning paradigms suffer from a phenomenon called Catastrophic Forgetting (CF) [35] which is performance degradation of old tasks when the model is adapted to learn new tasks. When a neural network model undergoes training, the weights of the model change such that it is optimal to the new task. However, these new weights may no longer be optimal for the old tasks, which leads to forgetting. Mitigating CF is the primary goal of any CL/IL framework.

1.1.2 The Stability - Plasticity tradeoff

A model is said to be stable when the decisions on the old tasks do not change despite being subjected to CL/IL on new tasks. Very high stability conditions lead to zero CF and, at the same time, zero learning on new tasks. On the contrary, a highly plastic model learns and adapts to new tasks very well, leading to very high CF on the old tasks.

An ideal solution is to have a trade-off between stability and plasticity of the model such that the model learns new tasks and remembers the knowledge acquired from old tasks.

1.2 Approaches

The existing approaches that deal with CF can be broadly divided into three categories: (i) Regularization-based methods, (ii) Memory-based methods, and (iii) Structure-based methods.

- **Regularization-based methods**: As the name implies, these methods regularize the change in parameters of the model during CL/IL, which are essential for the model to do well on the old experiences. EWC [26] and SI [54] identify these important parameters by computing the Fisher Importance matrix after every task; any change to the parameters with high importance is penalized. LWF [30] algorithm penalizes the model when there is a change in the model's probabilities

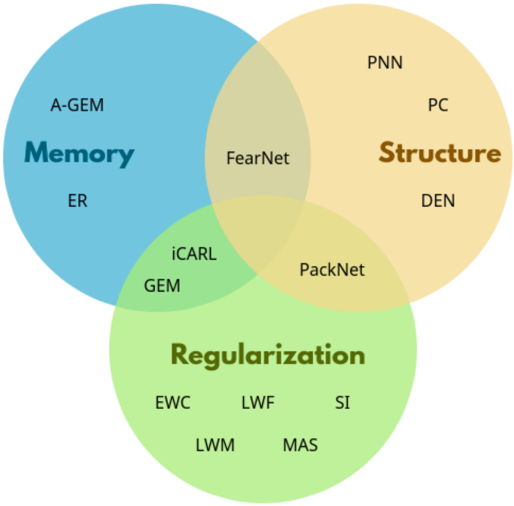


Figure 1.3: Venn diagram of popular CL/IL methods based on their approaches: EWC [26], SI [54], LWF [30], LWM [9], MAS [2], GEM [32], A-GEM [7], MER [44], iCARL [43], PNN [46], DEN [53], PC [47], PackNet [34], FearNet [21]

on old tasks. Enforcing that model gives the same output on old tasks even after learning new tasks makes the model learn not to change parameters critical to the old tasks. LWM [9] is similar to LWF but also ensures that the class attention maps of the old tasks remain the same after learning new tasks, thereby mitigating CF even further. Regularization methods can help alleviate CF only to some extent as the stability-plasticity tradeoff breaks when the number of new tasks to be learned increases. These methods are highly suitable in scenarios where access to data from old tasks is unavailable and with a fixed learning capacity.

- **Memory-based methods**: These methods alleviate CF by storing a few samples from previous experience, either explicitly or implicitly. In the explicit memory-based methods, raw samples from old tasks are stored in the memory, which is used for rehearsal during CL/IL. In the implicit memory-based methods, samples from old tasks are generated by networks like GAN [17]s, autoencoders, etc., to do pseudo-rehearsal. Some notable state-of-the-art techniques in this category include GEM [32], A-GEM [7], MER [44], iCARL [43], etc. These methods reduce CF effectively but require an additional memory buffer

- **Structure-based methods**: The structure-based methods (PNN [46], DEN [53], PC [47]) increase the model's capacity by attempting to modularize the neural network models. Task-specific modules are created for each task, and the CF is mitigated by storing and freezing the modules belonging to old tasks. Structure-based methods have shown remarkable performance in mitigating CF and can be used in cases where large tasks are to be learned, or dynamic growth of model size is tolerable.

Due to various data privacy and purging policies, we cannot hold access to medical data for longer times. Hence this thesis mainly focuses on the scenario where the data belonging to the old tasks are *unavailable* while training on new tasks. The contributions made are presented in different chapters of the thesis as described below.

A crucial part of intelligence is not to act/decide when one is uncertain. We build on this idea in Chapter 2 and propose a Continual Learning system based on an ensemble of models to reject the predictions done by models with high uncertainty and thereby routing the image to the correct model which was trained on a similar input distribution to that of an unseen image. The models in the ensemble are built with Bayesian Neural Networks, as they facilitate the derivation of uncertainty. At the test time, the output of the model which has the least uncertainty is chosen as the final prediction for an unseen image.

In Chapter 3, a novel approach is proposed to address CF in computer aided diagnosis (CAD) system design in the medical domain. CAD systems often need to handle a sequence of datasets collected over time from different sites with different imaging parameters/populations. The solution we propose is to move samples from all the datasets closer to a common manifold via a reformer at the front end of a CAD system. The utility of this approach is demonstrated on two common tasks, namely segmentation and classification, using publicly available datasets.

Chapter 4 deals with the class incremental learning (IL): learn new classes as and when new data or annotations are made available and old data is no longer accessible. We propose a novel approach that ensures that a model remembers the causal factor behind the decisions on the old classes, while incrementally learning new classes. We introduce a common auxiliary task during the course of incremental training, whose hidden representations are shared across all the classification heads. Since the hidden representation is no longer task-specific, it leads to a significant reduction in CF.

Chapter 2

A Continual Learning system based on Uncertainty using Bayesian Neural Networks

2.1 Introduction

Learning typically involves three stages: 1. Encoding, which is how the information is learned, 2. Storing or maintaining that information over time, and 3. Retrieval which is accessing the learned information. While successful remembering involves all the three stages, the key stage is retrieval since inability to retrieve renders information useless however well it is learnt and stored. Two types of errors occur should any of the three stages fail: Forgetting and Misremembering [36], both being the reasons for CF.

We propose a method that uses an ensemble of models to overcome the above limitations by using uncertainty as a measure, to route the incoming data to the model which was trained on a similar input distribution to that of the given data. Intuitively, humans express uncertainty about the things they forget and the crucial aspect of intelligence is not to act/decide when uncertain, which is not possible in deep neural networks (DNN). This is because most of the DNN output single point predictions rather than a distribution of predictions. In order to calculate uncertainty in DNNs, we propose to use Bayesian Neural Networks (BNN) that learn probability distributions over parameter space. An ensemble of BNN models can be created with each model trained on a separate dataset with a varied distribution. During inference, the uncertainty measure can be used to reject the predictions done by models with high uncertainty. This is in effect will allow for a model whose training dataset had a similar distribution to that of the test image to be chosen for the final prediction. The ensemble therefore represents a collection of old and new knowledge which can be efficiently retrieved using the uncertainty measure. We show that by following the steps of Learning, Storing and Retrieving knowledge efficiently, CF can be avoided. We next present some background on BNN, uncertainty followed by details of the proposed system.

2.2 Bayesian Neural Networks

A neural network can be viewed as a deterministic model $p(\mathbf{y}|\mathbf{x}, \omega)$ that assigns a probability for each possible output $\mathbf{y}$ given the weights ω. These weights are learned by maximum likelihood estimation (MLE) i.e. $\omega^* = argmax(log(P(\mathcal{D}|\omega)))$. This results in point-estimates, thereby ignoring any uncertainty present in those weights. Bayesian neural networks are the extension of standard neural networks with a posterior distribution $p(\omega|\mathcal{D})$ over the network weights ω, given the training data $\mathcal{D}$. Assuming a prior distribution $p(\omega)$ over the weights and bias, the posterior distribution is given as follows:

$$p(\omega|\mathcal{D}) = \frac{p(\mathcal{D}|\omega)p(\omega)}{p(\mathcal{D})} = \frac{\prod_{i=1}^{N} p\left(y_i|x_i, \omega\right) p(\omega)}{p(\mathcal{D})} \tag{2.1}$$

The predictive distribution of unseen data $\hat{x}$ is calculated as:

$$P(\hat{\mathbf{y}}|\hat{\mathbf{x}}) = \mathbf{E}_{P(\omega|\mathcal{D})}[P(\hat{\mathbf{y}}|\hat{\mathbf{x}}, \omega)] \tag{2.2}$$

The above equation is intractable as the expectation over the posterior distribution of weights is equivalent to using an uncountably infinite number of neural networks [5]. Several solutions have been proposed to estimate the approximate inference like Markov Chain Monte Carlo sampling [28], Variational Inference [24], Bayes By Backpropagation [5]. In our experiments, we utilized Bayesian CNNs from Tensorflow Probability [10] that use a stochastic variational inference [25] on the distribution integrating over kernel and bias in the CNN layer.

2.3 Uncertainty

A DNN framework such as a regression model estimates the mapping from the input f(x) to continuous output variable y and classification models output a probability vector, where elements of the vector represent the probability for each of the output classes. This probability vector is often misinterpreted as the model's confidence because a model can be highly uncertain even when giving a high softmax output for a particular class.

BNN framework allows for a distribution over model parameters/weights ω. Thus a new set of features is formed each time a prediction $\hat{y}$ is made on unseen data x^*. The variance of this predictive distribution is a measure of the model's confidence about its prediction on the given data. It has been shown [48] that the variance can be formulated as the sum of types of uncertainty: Aleatoric uncertainty and Epistemic uncertainty.

Aleatoric uncertainty captures the noise which is inherent in observations. This can be due to acquisition systems which cannot be reduced even if we add more data from the same source. Aleatoric uncertainty can be further divided into homoscedastic uncertainty and heteroscedastic uncertainty. Homoscedastic uncertainty assumes identical observation noise for each input x, hence it stays constant. On the other hand, Heteroscedastic uncertainty depends on inputs to the model, with some inputs having higher uncertainty than others [22]. Epistemic uncertainty captures variance in the model's parameters, which accounts for our ignorance about how we generated the model. This uncertainty can be reduced with more data collection. Together, these uncertainties can help to capture the confidence of a model on different inputs which can be interpreted as a model's way of expressing “I don't know”.

To quantify these uncertainties, we followed the procedure described in [27]. The proposed method calculates uncertainty based on variability and is computationally less expensive compared to other methods [22], [48]. The uncertainty is defined as:

$$Uncertainty = \underbrace{\frac{1}{T}\sum_{t=1}^{T} \operatorname{diag}\left(\hat{p}_t\right) - \hat{p}_t^2}_{\text{aleatoric}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T}\left(\hat{p}_t - \bar{p}\right)^2}_{\text{epistemic}} \tag{2.3}$$

Where T is the number of samples, $\bar{p} = \sum_{t=1}^{T} \hat{p}_t / T$, $\hat{p}_t = p\left(\hat{\omega}\right) = \operatorname{Softmax}\left\{f^{\omega}\left(x^*\right)\right\}$ and $f^{\omega}\left(x^*\right)$ denotes the pre-activated output of BNN which is a function of parameter ω

2.4 Proposed Method

In a continual learning (CL) system, we have a collection of datasets $\mathbf{D} = (D_1, D_2, \ldots, D_n)$ that arrive sequentially such that there may be a change in distribution from one dataset to another. The assumption is that access to old datasets is not there once training is done on them and there is no prior information on the distribution of the upcoming datasets.

We propose creating an ensemble of BNN models, each trained on different distributions of datasets. It is computationally undesirable to add a new model to the ensemble each time a new dataset arrives. Also, one of the key aspects of CL is the forward transfer of knowledge gained from the old experience to the new one. Hence, creating a new model without utilizing knowledge from previous experiences is unreasonable. At the same time, choosing a random model from the ensemble to initialize the new model does not offer any guaranteed advantages, because the random model might have been trained on a completely different distribution from the upcoming dataset.

To address the above issues, we propose Algorithm 1 where uncertainty is used as a measure to identify/retrieve the model in the ensemble whose training data distribution is close to the upcoming dataset's distribution. We use this as the base-model (m_B) for transfer learning and at the end of the training on the new dataset, if the difference between the uncertainties of m_B and the newly trained model is less than a threshold t, then it implies that the new dataset has a similar distribution to that of m_B's training data. Hence, we remove m_B from the Ensemble (as it is redundant) and add the new model to the Ensemble. If the difference in uncertainties is higher than t, then it is safe to assume that the new dataset has a very different distribution when compared to earlier ones, and hence this model is added to the Ensemble. In this way, we eliminate the need to create new model for every dataset.

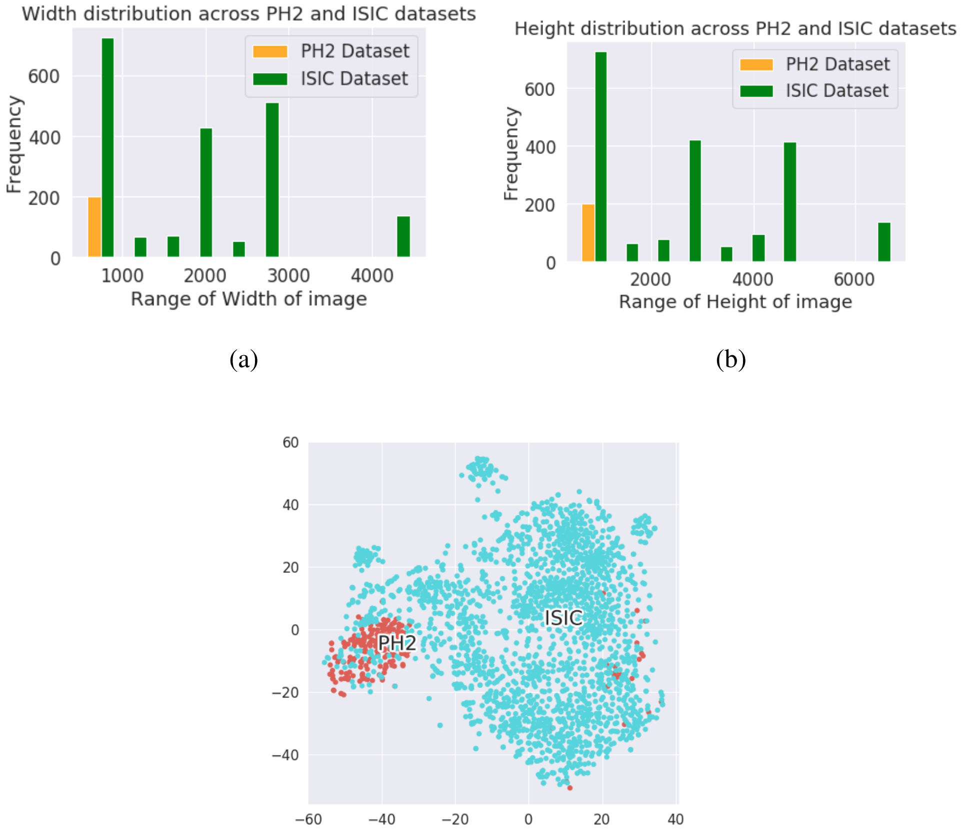
During test time, we select a model from the Ensemble which has the least uncertainty on the unseen data; as described in the algorithm 2. In both the algorithms 1 and 2, the uncertainty measure is used

Algorithm 1 Training procedure for Continual Learning

```
Input: Datasets $\mathcal{D} = (D_1, D_2, \ldots, D_n)$ where $D_i = (x, y)$, Threshold $t$
$Ensemble \leftarrow \phi$ for each $D_i$ in $\mathcal{D}$ do
    if $Ensemble$ is empty then
        Train model $m$ on $D_i$ ; // Since it is the first model, train from scratch
        $Ensemble.add(m)$
    else
        $uncertaintyList \leftarrow \phi$ for each model $m_j$ in $Ensemble$ do
            Calculate uncertainty u of model $m_j$ on the dataset $D_i$ using equation 2.3
            $uncertaintyList.add(u)$
        end
        $m_b \leftarrow Ensemble[argmin_u(uncertaintyList)]$ ; // Select a model with least uncertainty
        Initialize new model $m_n$ with $m_b$ for Transfer Learning
        Train $m_n$ on $D_i$
        Calculate uncertainty $u_n$ of model $m_n$ on the dataset $D_i$ using equation 2.3
        if $|u_n - u_b| \leq t$ then
            $Ensemble.remove(m_b)$ ; // Remove redundant model
        end
        $Ensemble.add(m_n)$
    end
end
```

Algorithm 2 Test algorithm

```
Input: $Ensemble$, Unseen data $x^*$
Output: $y^*$ as the final prediction on $x^*$
$uncertaintyList \leftarrow \phi$
for each model $m_i$ in $Ensemble$ do
    Calculate uncertainty u of model $m_i$ on the data $x^*$ using equation 2.3
    $uncertaintyList.add(u)$
end
$m_k \leftarrow Ensemble[argmin_u(uncertaintyList)]$  $y^* = m_k.predict(x^*)$
```



(a)

(b)

(c)

Figure 2.1: Variability across distributions in the PH2 and ISIC datasets. (a) Width and (b) height distributions; (c) TSNE plot, with red and blue indicating PH2 and ISIC images.

to route the data to the right model for transfer learning and final predictions respectively. Since the Ensemble has both old and new models, all trained on different distributions, we need not worry about the stability-plasticity dilemma.

2.5 Results

The proposed system was used to develop solutions for three major problems: Classification of 2D images (Malaria Disease), Segmentation on 2D images (Skin Lesion) and Segmentation on 3D images (ventricles from neuro images/MRI). The above choice of experiments is done to illustrate that our approach to CL is generic and not restricted to just one set of problems. In all the experiments, to evaluate the process of CL, we calculate and report a metric for a model on a dataset X, and again after transfer learning on the dataset Y. The change in performance on X is an indicator of CF or the effectiveness of CL and hence is also reported. It is to be noted that the results reported for our system (last row of all Tables) is that of an ensemble.

2.5.1 Skin Lesion Segmentation

The public data sets ISIC 2017 [4] and PH2 [38] were used for Skin lesion analysis towards melanoma detection. The ISIC dataset, with 2000 training images and 150 test images, is sourced from multiple clinical centers internationally and acquired from a variety of devices within each center. The dataset contains images of resolutions ranging from 152x152x3 to 4500x7000. Thus, the dataset itself contains a good mix of possible distributions. The dermoscopic images in the PH2 dataset were obtained at the Dermatology Service of Hospital Pedro Hispano (Matosinhos, Portugal) under the same conditions through the Tuebinger Mole Analyzer system using a magnification of 20x. They are 8-bit RGB color images with a resolution of 768x560 pixels. It is split into 200 training and 150 test sets.

The data description of the two datasets indicate that they vary in terms of resolution which is apparent in the image width/height distributions shown in Fig. 2.1) (a) and (b). From the TSNE plot in Fig. 2.1) (c), we can see that the two datasets form two different clusters in latent space. Apart from these differences, there may be some imperceptible differences among the images which can potentially lead to drastic changes in weights during training.

In our experiments, in order to mimic real-world scenarios, where we get multiple sets of data each with the same or different distributions we do the following. We start with the ISIC dataset which is large and split it randomly into 2 parts ISIC-1 and ISIC-2. Then set $\mathcal{D} = \{D_1, D_2, D_3\}$ where D_1, D_2, D_3 are taken to be ISIC-1, PH2 and ISIC-2 respectively. Training is done in the following order: D_1, D_2, D_3 datasets. Each of datasets in $\mathcal{D}$ are again split into 80/20% for training/testing. A simple U-NET model with the middle order replaced with BCNNs [48] was used for training. The threshold t was set to be 10 for the algorithm 1 to determine if the model is to be retained in the ensemble. We will next explain how the algorithm proceeds to create the ensemble, in detail.

Initially, after training on D_1, we have a single model m_1 in Ensemble. Later when D_2 arrives, a new model m_2 is created by taking m_1 and doing a transfer learning on D_2. At the end of training, since both D_1 and D_2 have different distributions, we will have two models in the ensemble: $\{m_1, m_2\}$. When D_3 arrives it will have a similar distribution to D_1. Hence, m_1 is chosen as a base model again for transfer learning by the algorithm 1 and m_3 is created. At the end of transfer learning, change in uncertainty of m_1 and m_3 will be less than threshold, t and hence m_1 is discarded from ensemble and m_3 is added. So, finally the ensemble contains $\{m_2, m_3\}$. The performance of the proposed system was tested and compared with other approaches to CL. The results on the respective test sets are given in Table **??**. The results show that our method has performed superior to LWF and EWC and on par with MER techniques in terms of CL. As mentioned earlier, MER has the additional baggage of storing random samples in memory which is undesirable.

Methods	**ISIC-1**	**PH2**	**Change on ISIC-1**	**ISIC-2**	**Change on ISIC-1**	**Change on PH2**
FT	91.16	91.69	-14.00	91.24	-8	-11.00
LWF	91.16	91.93	-8.93	91.11	**+0.23**	-5.27
EWC	91.16	66.86	-33.63	58.21	-34.46	-9.14
MER	91.16	91.75	-1.80	91.03	+0.15	-2.98
Ours	90.12	92.24	**-0.06**	90.06	0.00	**+0.06**

Table 2.1: Consolidated report of **Dice scores** (in %) and percentage change in the same during Continual Learning on ISIC-1, PH2 and ISIC-2 datasets (in the same order) evaluated using Fine Tuning (FT), LWF, MER and ours

2.5.2 Malaria Disease Classification

The second problem in our study was malaria disease classification. Two public datasets of microscopy images were chosen: 1. Thin blood smear slide images from the repository developed as a part of Malaria Screener research activity [42], hosted in NLM (National Library of Medicine) and 2. Data collected by the Medical School of the University of Alabama at Birmingham [11]

The NLM dataset is based on a collection of 200 Giemsa-stained thin blood smear slides from 150 P. falciparum-infected and 50 healthy patients. A smartphone's camera was used to acquire images of slides for each microscopic field of view. The dataset has a total of 27,558 cell images with equal instances of parasitized and uninfected cells. The Alabama dataset has RBC images acquired from whole slide images (WSI) with 100x magnification. [11], reports that morphological transform was done to separate cells and resize into 50x50 dimensions. This dataset has a total of 1034 infected and 1531 uninfected images.

Both these datasets have very different images due to variation in resolution, background, staining color etc. (see Fig. 2.3). Thus, they form two different clusters in the latent space indicating differences in distribution, as depicted in Figure 2.2.

Both the Alabama and NLM datasets were split into two equal parts randomly such that $\mathcal{D} = \{D_1, D_2, D_3, D_4\}$ corresponds to A1, N1, A2 and N2, and training was done in the same order. At the end of training, the goal is to have minimal forgetting and the number of models in the ensemble should be limited to two since we are ideally using datasets from only two different distributions. We used a simple 3 layer BCNN network for training these datasets and each of these datasets are divided again into 80% of train and 20% of test sets.

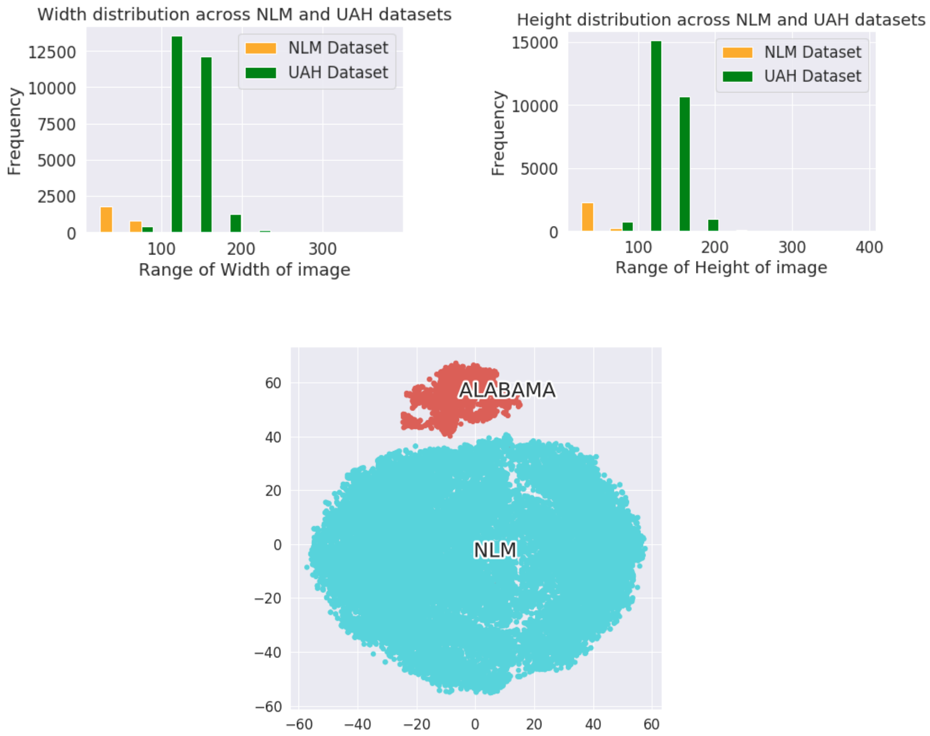


Figure 2.2: Variability across distributions in the Alabama and NLM datasets. (a) Width and (b) height distributions; (c) TSNE plot, with red and blue indicating Alabama and NLM images.

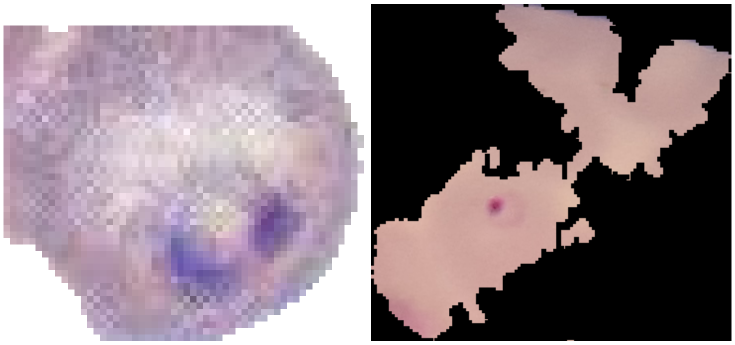


Figure 2.3: (a) and (b) show the sample images from Alabama and NLM Datasets respectively

By following algorithm 1, as and when data D_i arrives we create a new model m_i that is fine-tuned with a base model m_b. The base model can be discarded if the change in uncertainty is less than threshold t, which is 10 in this experiment. At the end of training, we have $\{m_2, m_4\}$ in the ensemble.

Results from Table 2.2 show that our method has performed consistently well across all datasets with graceful forgetting. It is to be noted that the change in accuracy for A2 set after training on N2 using Fine Tuning method is **0%** because the model here failed to converge, consequently, there won't be any change in accuracy. Also, using the regularization methods EWC and LWF, we can see that model immediately forgets the old knowledge after getting trained on a different distribution, and consequently performs well on the datasets which has a similar distribution. This is the case of stability-plasticity trade-off where plasticity won for both LWF and EWC in all phases of training.

Methods	**A1**	**N1**	**Change on A1**	**A2**	**Change on A1**	**Change on N1**	**N2**	**Change on A1**	**Change on N1**	**Change on A2**
FT	96.88	94.60	-54.66	97.81	+0.32	-48.11	49.94	+0.32	-48.11	**0**
LWF	97.5	93.9	-43.61	95.01	-0.31	-47.14	95.40	-28.73	+0.31	-36.72
EWC	96.88	92.60	-33.65	95.63	-1.62	-46.99	92.39	-53.72	+0.530	-56.03
MER	97.50	88.03	-1.90	98.44	-0.95	-8.34	93.40	-2.85	**+6.44**	-2.90
Ours	97.91	93.82	**-1.71**	97.84	**+0.06**	**-0.51**	93.86	**+0.73**	-3.49	-0.01

Table 2.2: Consolidated report of **Accuracies** (in %) and percentage change in the same during Continual Learning on A1, N1, A2 and N2 (in the same order) evaluated using Fine Tuning (FT), LWF, MER and ours

2.5.3 Brain Ventricle Segmentation

Ventricle segmentation from MRI is of interest in many applications. To demonstrate that our method can also be used on 3D data, experiments were done on this segmentation task also. Three public datasets were chosen: MICCAI [3], with 15, 1mm T1 weighted MRI scans, IBSR [51] with 18, 1.5mm T1 weighted volumes and CANDI [23] 30, 1.5mm T1 weighted scans. These datasets have extreme diversity in distributions: While MICCAI has higher resolution as compared with IBSR, the CANDI dataset contains the MRI scans of only children and adolescent subjects which vary reasonably when compared to that of adult scans in terms of size and anatomy.

For training, we used the M-NET model [37] and converted it into Bayesian M-NET by replacing the middle order of the network with BCNNs. Table 2.3 shows the experiments conducted on the three datasets $\{D_1, D_2, D_3\}$ which corresponds to: IBSR, MICCAI, and CANDI respectively for continual learning using Fine Tuning (FT), LWF, MER and our method. Each of these datasets are split into 80% of train and 20% of test sets.

Since all three datasets have different distributions, at the end of training, we have three models $\{m_1, m_2, m_3\}$ in the Ensemble. It is worth nothing that MER is not as effective. The performance is lower on subsequent datasets ($D_i, i > 1$) because the distributions between what is held in its memory and the training dataset are extremely different. Similarly, a classic case of stability-plasticity trade-off can be observed in LWF method, where the model is not plastic enough to learn new data, hence the low dice score on CANDI dataset for LWF. We skipped EWC for this task because EWC involves calculation of Fisher information, which is computationally very expensive. Our method has performed consistently well across all the datasets with a little gracious forgetting.

Methods	**IBSR**	**MICCAI**	**Change on IBSR**	**CANDI**	**Change on IBSR**	**Change on MICCAI**
FT	98.62	98.60	-19.44	98.43	-65.12	-56.95
LWF	97.90	72.37	-18.06	33.83	-64.25	-32.70
MER	97.90	92.26	**-1.95**	87.89	-26.33	-32.17
Ours	98.55	98.59	-2.34	98.37	**-3.50**	**-4.58**

Table 2.3: Consolidated report of **Dice scores** (in %) and percentage change in them during Continual Learning on IBSR, MICCAI, CANDI (in the same order) evaluated using Fine Tuning (FT), LWF, MER and our methods.

2.6 Conclusion

In this work, we proposed a novel approach to the problem of CL. Unlike existing approaches which largely rely on modifying the regularisation component, our approach takes a more fundamental route by addressing the problem at the network architecture level by employing Bayesian networks to compute uncertainty in the decisions arrived at by the system. It enables (i) learning efficiently using old knowledge as the knowledge is maintained by storing only non-redundant models and (ii) retrieval of knowledge using an uncertainty measure. The experimental results show that our method outperforms state-of-the-art methods like LWF, EWC, MER consistently over a variety of problems such as disease detection and anatomy segmentation which pose different kinds of challenges for CL. A key advantage of the proposed system is the utilization of existing models for transfer learning and wisely adding of models in an ensemble which calls for a minimal amount of explicit memory required to store models as compared to storing batches/episodes from previous datasets.

Chapter 3

Manifold Learning to address Catastrophic Forgetting

3.1 Introduction

Catastrophic Forgetting (CF) mostly occurs in cases where the datasets lie on different manifolds from each other as a result of which a single network will fail to converge on all of them when the network is trained sequentially. Existing regularisation techniques do not take the dataset shifts into account which is a core reason for CF. Dataset shift occurs when the joint distribution of any two datasets are different, i.e. $P_1(x, y) \neq P_2(x, y)$. Dataset shift may appear as one of the following manifestations [40]:

1. **Covariate Shift:** In this type of shift, the conditional probability $P(y|x)$ remains the same, but the input distribution $P(x)$ changes from the current dataset to the future data i.e. $P_1(y|x) = P2(y|x)$ and $P1(x) \neq P2(x)$

2. **Prior Probability Shift:** It is considered as the reverse case of the Covariate Shift. In this type of shift, the distribution of classes varies from one dataset to another. It is defined as the case where $P1(x|y) = P2(x|y)$ and $P1(y) \neq P2(y)$

3. **Concept Shift:** Also referred to as Concept Drift, it occurs when the relationship between the input and the class variables changes i.e. change in the concept to be learned. Mathematically, it can be defined as $P1(y|x) \neq P2(y|x)$ and $P1(x) = P2(x)$

In this paper, we focus on this core aspect, specifically on the Covariate shift, where we assume that datasets coming from different sources have different distributions because of acquisition, demographic changes, quality, etc., but the underlying relationship between the images and class variables remain the same i.e. there will be no concept drift and prior probability shifts.

We propose a simple yet effective solution in the form of a Reformer which brings or ”reforms” the data from all the datasets together towards a common manifold. These reformed samples are then passed to the network to learn the task. This should enable the deep learning system to be robust to the changes in dataset distributions and limit CF to a tolerable extent. The idea of a Reformer was originally introduced in [39] to defend against adversarial attacks in the image classification task. The proposed approach is memory efficient as it doesn't require storing of old samples. Unlike in other state-of-the-art regularization-based techniques, using the proposed approach, the network can learn seamlessly through the sequence of datasets without the necessity to undergo complex computations during the training phase.

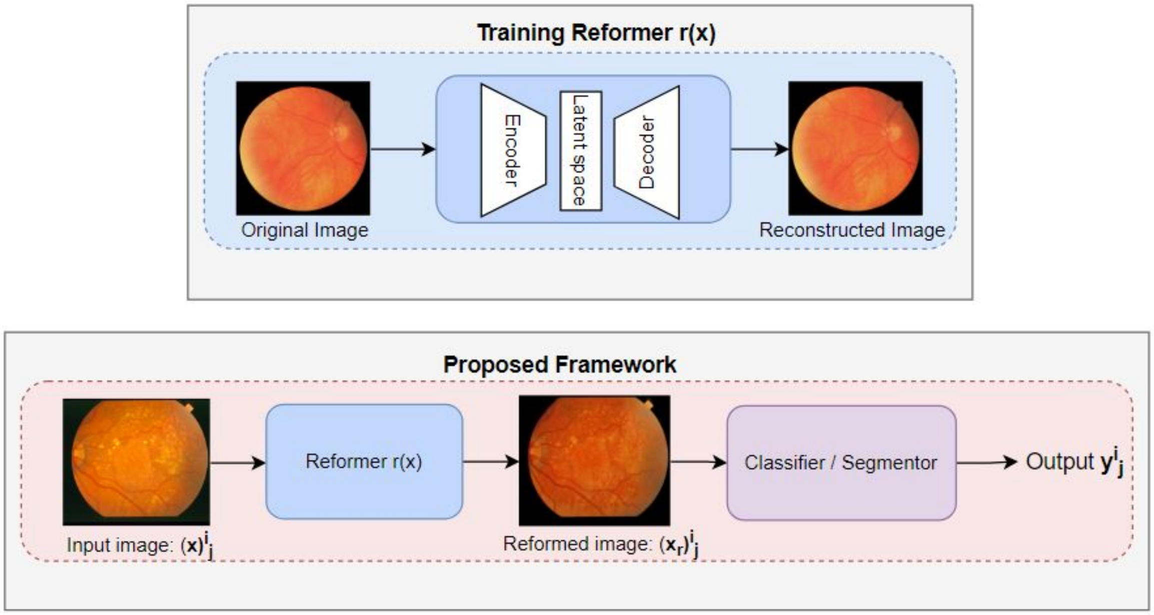


Figure 3.1: Proposed Framework in which a Refomer r(x) is trained on a dataset to learn the manifold $\mathcal{M}$. Given an input $(x)^i_j$, which represents j^{th} sample from the dataset $\mathcal{D}_i$, the Reformer outputs $(x_r)^i_j$ such that it is closer to the learned manifold $\mathcal{M}$. Finally, these reformed samples are fed to the target classifier/segmentor network

3.2 Methodology

3.2.1 Manifold Learning using Autoencoders

Our strategy of learning of a common manifold across datasets is driven by a working assumption that natural, high dimensional data concentrates close to a non-linear, low dimensional manifold. For instance, if an ambient space represented as $\mathcal{X}$ has a probability distribution μ, then the support of μ

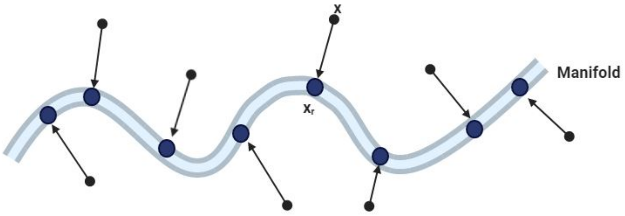


Figure 3.2: The effect of autoencoder that projects the incoming sample x on to the learned manifold

(i.e. a set of possible values of that distribution) is a low-dimensional manifold $\mathcal{M}$ given by:

$$\mathcal{M}(\mu) = \{x \in \mathcal{X} | \mu(x) > 0\} \tag{3.1}$$

Autoencoder networks have been successfully used to learn this manifold structure of data and to obtain a parametric representation of such a structure in various applications [29].

An autoencoder has two parts: Encoder and Decoder. The input to the encoder ψ_e is a sample $x \in \mathcal{X}$ which gets mapped to its latent space representation, $z \in \mathcal{Z}$ such that, $z = \psi_e(x)$. Therefore, given a manifold $\mathcal{M} \subset \mathcal{X}$, the encoder is a mapping $\psi_e : \mathcal{M} \longrightarrow \mathcal{Z}$. The decoder, $\psi_d : \mathcal{Z} \longrightarrow \mathcal{X}$ maps the latent representation z to x_r which is an approximation to x, such that $x_r = \psi_d(z) = \psi_d \circ \psi_e(x)$. Parametrized by θ, both encoder and decoder are trained together to minimize the reconstruction errors using a loss function $\mathcal{L}(x, x_r)$, given by:

$$\underset{\theta}{min} \mathcal{L}(\theta) = \underset{\theta}{min} \frac{1}{n} \sum_{i=1}^{n} \left\| x^i - \psi_d^\theta \circ \psi_e^\theta(x^i) \right\|^2 \tag{3.2}$$

The result is an approximate manifold $\hat{\mathcal{M}} = \psi_d^\theta \circ \psi_e^\theta(\mathcal{M})$ given an input manifold $\mathcal{M}$ as illustrated in Fig 3.2.

3.2.2 Proposed solution

In order to make the network learn seamlessly from all the datasets without undergoing CF, we propose a solution that moves or ”reforms” the data from all the datasets together towards a common manifold using an autoencoder called Reformer. These reformed samples are used by the classifier/segmentor network to learn the task, as illustrated in Fig 3.1.

The Reformer is a function $\mathbf{r} : \mathbf{x} \longrightarrow \mathbf{x_r}$ that maps the input x to the reformed output x_r. As elaborated in Algorithm 4, given a sequence of datasets $\{\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_{k-1}\}$, the Reformer r is trained on the first dataset $\mathcal{D}_0$ to learn the manifold $\mathcal{M}$. When a sample from subsequent $k - 1$ datasets arrives, it is passed as an input to this pre-trained reformer which moves/reforms the sample as shown in Fig 3.2.

A denoising autoencoder is chosen as the Reformer to learn meaningful representations of data instead of a plain autoencoder that may merely copy its input by learning an identity function. Training the Reformer is a critical task as the reformed outputs should not introduce additional artifacts and should

Algorithm 3 Proposed Solution

```
Input: Reformer $\mathbf{r} : \mathbf{x} \longrightarrow \mathbf{x_r}$, Classifier/Segmentor network $\mathbf{f}$ and a sequence of $\mathbf{k}$ datasets
        $\{\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_{k-1}\}$
for $x$ in $\mathcal{D}_0$ do
    $x_r \leftarrow r(x)$
    Optimize $r(x)$ using $\mathcal{L}(x, x_r)$ in equation 3.2
end
for each $D_i$ in $\{\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_{k-1}\}$ do
    for $(x)^i_j$ in $D_i$ do
        $(x_r)^i_j \leftarrow r((x)^i_j)$
        $(y)^i_j \leftarrow f((x_r)^i_j)$
        Optimize the network f using its corresponding loss function and optimizer
    end
end
```

preserve important details in the input image. When the model is trained on the reformed images from a new dataset, the weights/parameters of the model will not be changed drastically as these reformed images lie on a common manifold on which the model has been trained before. Thus this enables the model to learn seamlessly on the new datasets without CF.

Figure 3.3 depicts the effect of reformer in bringing samples belonging to two different datasets (DRIVE and STARE datasets) to a common manifold. The t-SNE representation of DRIVE and STARE images forms two separate clusters. However, when a reformer trained on the DRIVE dataset is used to reform STARE images, the t-SNE representation of the latter gets closer to the cluster belonging to the DRIVE dataset.

3.3 Experiments and Results

The Reformer was designed to be a 20-layer denoising auto-encoder with skip connections between corresponding encoder and decoder layers. The bottleneck (latent space) layer had 128 features. All convolutional layers had a kernel size of 3 with padding by 1. The Reformer was trained for 50 epochs approximately using the Adam optimizer and MSE (Mean Squared Error) loss, with a learning rate of 0.001. The input image was of size $224 \times 224 \times 3$ and each image was subjected to Gaussian noise with a 0-mean and standard deviation of 0.25.

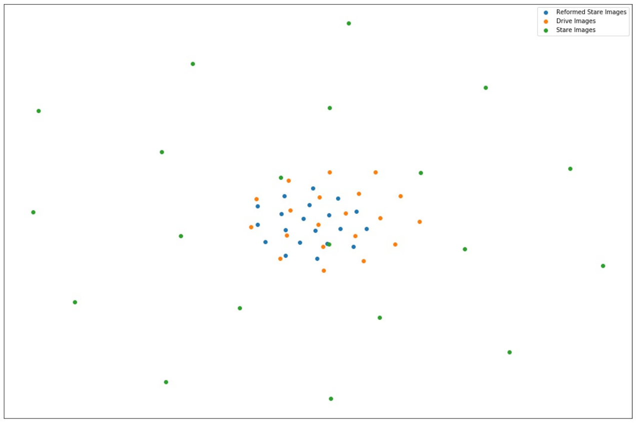


Figure 3.3: t-SNE representation of DRIVE, STARE images, and the reformed images of STARE dataset with the help of a reformer trained on DRIVE dataset.

The proposed solution for addressing CF was assessed using the above Reformer on two classical problems of interest in CAD design: Segmentation (Retinal Vessel) and Classification (Skin Melanoma). The classifier/segmentor was trained on a sequence of datasets with the assumption that there is *no access* to previously used datasets.

For each of the problems, comparison with the regularization methods EWC, LwF, and SI was also done by evaluating them with and without a reformer.

The CF in a network was quantified by computing the percentage change in the performance of the network on the old dataset when it is trained on a new dataset. Ideally, this percentage change should be greater than or equal to 0; increasing negative values indicate higher degree of CF in the model.

3.3.1 Retinal Vessel Segmentation

The task here is to segment the blood vessels in a given retinal image. We have used three publicly available datasets: DRIVE [49], STARE [18], and HRF [6] for this task.

3.3.1.1 Implementation

Datasets - The DRIVE (Digital Retinal Images for Vessel Extraction) database has 40 images (divided equally into train and test sets) obtained from a diabetic retinopathy screening program in The Netherlands. The images are captured in digital form from a Canon CR5 nonmydriatic 3CCD camera at a 45-degree field-of-view. The images are of size 768×584 resolution. The STARE (STructured Analysis of the Retina) dataset has 20 retinal images. All these images were acquired using a TopCon TRV-50 fundus camera at a 35-degree field-of-view and subsequently digitized at 605×700 pixels in resolution. Finally, the HRF (High-Resolution Fundus) Image database has a total of 45 retina images acquired using a Canon CR-1 fundus camera with a 45-degree field-of-view digitized at 3504×2336 resolution.

Segmentation network - Attention-based U-net architecture [41] was used for segmenting the blood vessels. The training was done using an Adam optimizer with an initial learning rate of 0.001 and with 5-fold cross-validation since the dataset was small. The model was trained sequentially on the datasets in the following order one-by-one: Dataset $\mathcal{D}_0$: DRIVE, Dataset $\mathcal{D}_1$: STARE, and Dataset $\mathcal{D}_2$: HRF. The Reformer was trained on Dataset $\mathcal{D}_0$ (DRIVE). The skip connections used in the Reformer helped to retain intrinsic details, which is essential for tasks like vessel segmentation. This pretrained Reformer was used to reform images from all the other subsequent datasets. The percentage change in Dice score of the model before and after training on a new dataset is taken as the quantitative estimate of CF.

3.3.1.2 Segmentation Results

The results of the performance in CF reduction are given in Table 1 covering all possible settings as per the training order. Here, $\mathcal{D}_i \longrightarrow \mathcal{D}_j$ denotes CF on the dataset $\mathcal{D}_j$ after the model is trained sequentially on datasets $\{\mathcal{D}_i, \mathcal{D}_{i-1}, ..., \mathcal{D}_{j+1}\}$ where $j < i$

In the experiment, the order of the training is first with low resolution images and then high resolution images. Hence, the segmentation network gets tuned to the dataset on which it is last trained on, and consequently state-of-the-art methods suffer from high CF as indicated by large negative values. The results for Finetuning with and without Reformer in Table 1 shows that there is a significant reduction in CF when a Reformer is introduced. There is further boost in performance improvement when the Reformer is used in conjunction with state-of-the-art methods.

Methods	$\mathcal{D}_1 \longrightarrow \mathcal{D}_0$	$\mathcal{D}_2 \longrightarrow \mathcal{D}_1$	$\mathcal{D}_2 \longrightarrow \mathcal{D}_0$
Finetuning (FT)	-51.60	-67.14	-69.99
FT + Reformer	**-0.84**	**-8.08**	**-3.04**
EWC	-49.30	-60.71	-67.04
EWC + Reformer	**+0.014**	**-5.56**	**-2.18**
LwF	-50.8	-58.28	-70.37
LwF + Reformer	**+0.81**	**-3.73**	**-1.4**
SI	-29.73	-29.53	-50.57
SI + Reformer	**-4.19**	**-0.77**	**-1.99**

Table 3.1: Quantitative analysis of Catastrophic Forgetting (CF) in segmentation. Here, $\mathcal{D}_i \longrightarrow \mathcal{D}_j$ denotes CF on the dataset $\mathcal{D}_j$ after the model is trained sequentially on datasets $\{\mathcal{D}_i, \mathcal{D}_{i-1}, ..., \mathcal{D}_{j+1}\}$ where $j < i$. $\mathcal{D}_0$: DRIVE, $\mathcal{D}_1$: STARE, and $\mathcal{D}_2$: HRF

3.3.2 Skin Melanoma Classification

For the melanoma classification task, we used three publicly available datasets: Derm7pt [20], ISIC20 [14] and ISIC19 [16, 13, 1].

3.3.2.1 Implementation

Datasets - The ISIC20 dataset has 33,126 dermoscopic images of 2056 patients from Europe, North America, and Australia with an average of 16 lesions per patient and 584 confirmed melanomas. The ISIC'19 archive contains over 13,000 dermoscopic images collected from leading clinical centers worldwide and acquired from a variety of devices within each center. The Seven point checklist dataset for skin images (Derm7pt) includes over 2000 clinical and dermoscopy color images, along with corresponding structured metadata tailored for training and evaluating computer-aided diagnosis (CAD) systems.

The melanoma classification system was trained in the following order: First on A, next on B and then finally on C, where Dataset $\mathcal{D}_0$: Derm7pt, Dataset $\mathcal{D}_1$: ISIC20, and Dataset $\mathcal{D}_2$: ISIC19. The reformer was trained on Dataset $\mathcal{D}_0$ (Derm7pt) and this pretrained reformer was used to reform images from the remaining datasets.

Classfication network - EfficientNet-b6 architecture [50] was used to classify an image as melanoma or benign; it was trained using Adam optimizer with an initial learning rate of 0.0001 and with Binary Cross Entropy loss function. The percentage change in AUC scores before and after training on a new dataset was taken as an estimate of CF.

3.3.2.2 Classification Results

In this experiment, along with the standard regularization methods like EWC, LwF and SI, we compared our method with a recent paper on domain adaptation (DA) [31]. This method too, like ours, assumes that old (source) data is not available when the new (target) data arrives. The method makes the model adapt to target domain by freezing the classifier module of source data and learning the target-specific feature extraction module.

The results given in Table 2, show that the Reformer is able to reduce CF even with Finetuning, with the degree of reduction dependent on the order of training. Since the order of training goes from small to the large size datasets, the model becomes generalized at the end, resulting in tolerable CF (moderately low negative values) in all the cases. However, it can be observed that when the Reformer is used in conjunction with regularization methods brings CF score close to zero, and in some cases, it serves to improve the performance of the network on old datasets as indicated by the positive values. These results are quite encouraging towards the development of effective classification systems across different scenarios (limited training dataset with changing dataset distributions due to acquisition systems, demography, resolution, etc.) Furthermore, this also opens up the possibility of using the latent features of the Reformer as inputs to the network for the classification task.

Methods	$\mathcal{D}_1 \longrightarrow \mathcal{D}_0$	$\mathcal{D}_2 \longrightarrow \mathcal{D}_1$	$\mathcal{D}_2 \longrightarrow \mathcal{D}_0$
Finetuning (FT)	-4.59	-6.97	-4.12
FT + Reformer	**-3.48**	**-0.45**	**-2.66**
EWC	-2.09	-2.17	-2.90
EWC + Reformer	**-0.93**	**+2.57**	**-1.03**
LwF	-1.04	+2.95	-3.61
LwF + Reformer	**-0.71**	**+5.30**	**-0.80**
SI	+2.07	+4.09	+0.57
SI + Reformer	**+4.91**	**+8.24**	**+8.07**
DA	-16.67	-2.87	-3.41

Table 3.2: Quantitative analysis of Catastrophic Forgetting (CF) in classification. Here, $\mathcal{D}_i \longrightarrow \mathcal{D}_j$ denotes CF on the dataset $\mathcal{D}_j$ after the model is trained sequentially on datasets $\{\mathcal{D}_i, \mathcal{D}_{i-1}, ..., \mathcal{D}_{j+1}\}$ where $j < i$; $\mathcal{D}_0$: Derm7pt, $\mathcal{D}_1$: ISIC20, and $\mathcal{D}_2$: ISIC19 datasets sequentially.

3.4 Conclusions

In this work, we presented a simple yet novel solution to alleviate CF when a model is trained on a sequence of datasets with different distributions. We focused on a setting where the task and anatomy are the same throughout the training process while the difference lies in the distributions of datasets. Our method outperformed existing state-of-the-art methods by a large margin for both classification and segmentation problems. This indicates that mapping the different datasets to a manifold is an effective and efficient (since there is no additional memory requirement) solution to addressing CF. We also examined the option of integrating the Reformer with state-of-the-art regularization methods. Experimental results showed further improvement compared to when getting trained on original samples alone which is quite attractive. It has to be noted that the proposed framework is only suitable for scenarios where the images represent the same anatomy (or scene). While we used a simple denoising autoencoder to implement the Reformer, other methods for learning a manifold can be explored in the future. Similarly the approach can also be extended to multi-task learning on different datasets.

Chapter 4

Incremental learning for a flexible CAD system design

4.1 Introduction

Deep Neural Networks (DNNs) have shown remarkable performance in a broad range of computer vision tasks, including in the medical domain. With FDA approval for the first standalone Computer-Aided Diagnostic (CAD) system in the recent past, incremental learning (IL) is of interest to increase the scope of existing CAD systems. This is also of practical interest because annotations are labour intensive to generate and hence annotations for different tasks may not be available at a time.

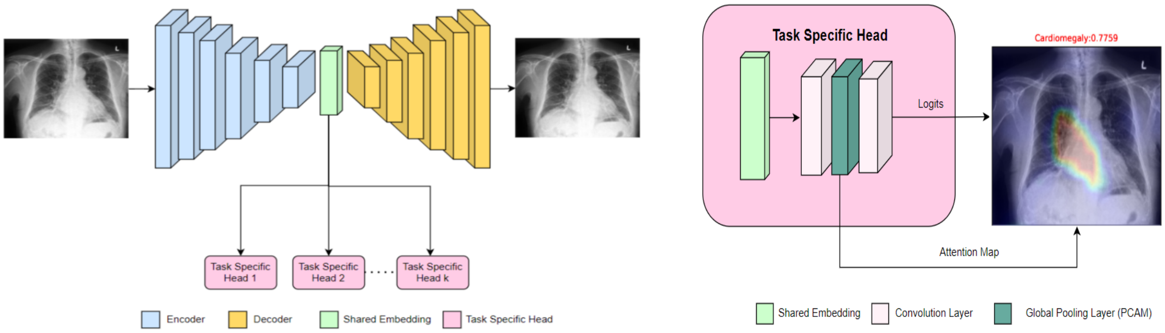
When a pre-trained model is adapted to learn on newly added classes, its performance on old classes generally drops drastically. This phenomenon is called Catastrophic Forgetting (CF) [35]. CF arises due to the fact that during incremental learning, the weights which were optimal for the old classes can undergo significant change in order to perform better on the new classes.

The solution we propose for IL to learn new tasks has two key parts a) introduction of an auxiliary task that is common to the course of IL and utilise the features learnt in this auxiliary task to the class-specific heads. The feature space of the auxiliary task is stable during IL and this ensures low/no CF even with the addition of many new classes. b) imposition of a knowledge distillation loss *and* attention loss on *all* the old classes; this ensures that the model remembers the causative factors behind all the decisions.

LwM[9] also uses a similar loss, but considers only the loss from attention maps of the class that has the highest probability, whereas we consider that of *all* classes. Intuitively, remembering the reason behind why an image does not belong to certain classes should be as important as remembering why it belongs to a specific class. This is critical in multi-label tasks, where each image can have multiple labels, and it is necessary to remember **all** the decisions on old classes.

4.2 Proposed solution

We propose a model with a multi-head setting, where a unique classification head is created for each task as depicted in Figure 4.1. When a model is trained sequentially on new tasks, the weights/parameters



(a) Overall architecture

(b) Task-specific head

Figure 4.1: The proposed architecture. (a) Overview of the multi-head model with an auxiliary reconstruction task. (b) Detail of a task-specific head with corresponding attention map generation.

of the model will change to adapt to the new task which leads to CF. This is prevented using a training regime with the following components: (1) Mean Squared Error (MSE) loss for auxiliary task; (2) Knowledge distillation loss; (3) Attention Loss. In addition to these losses, the network is made to learn the new classes with the help of an additional loss term: $\mathcal{L}_{class}$ which is a binary cross-entropy loss between the predicted and ground truth labels.

4.2.1 Auxiliary Task

A potential reason behind CF is that weights learnt for a new task may not be optimal for the old tasks. Hence, our strategy is to create an auxiliary task which is common through the IL process and feed the hidden representation learnt from this task to the task-specific heads. The auxiliary task we choose is the reconstruction of input images. Given that the anatomy is fixed across the tasks, the reconstruction of images should not result in any drastic change of weights in its feature space. This in turn should help reduce CF. The MSE loss between input image x and the reconstructed image x_{recons} is used as loss for the auxiliary task $\mathcal{L}_{aux}$:

$$\mathcal{L}_{aux}(x, x_{recons}) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \|x(i,j) - x_{recons}(i,j)\|^2 \tag{4.1}$$

4.2.2 Knowledge Distillation Loss

A distillation loss is imposed on the probabilities of all the old classes when the model is trained on a new class. This ensures that the model outputs the same decisions on old classes without forgetting [30]. Let $\mathbf{y_o}$ and $\hat{\mathbf{y}}_\mathbf{o}$ be the recorded and current probabilities, respectively, on the old classes. Then, the knowledge distillation Loss $\mathcal{L}_{kd}$ is defined as follows:

$$\mathcal{L}_{kd}(\mathbf{y_o}, \hat{\mathbf{y}}_\mathbf{o}, l) = \sum_{i=1}^{N} {y'}^{(i)}_{o} \log \hat{y}'^{(i)}_{o} \tag{4.2}$$

where, N is the number of old class labels; $y_o^{'(i)}, \hat{y}_o^{'(i)}$ are the modified versions of $y_o^{(i)}, \hat{y}_o^{(i)}$ respectively and are computed as

$$y_o^{'(i)} = \frac{(y_o^{(i)})^{1/T}}{\sum_j (y_o^{(j)})^{1/T}}, \hat{y}_o^{'(i)} = \frac{(\hat{y}_o^{(i)})^{1/T}}{\sum_j (\hat{y}_o^{(j)})^{1/T}} \tag{4.3}$$

where T is the temperature parameter. When $T > 1$, the probabilities become softer and thus give information about those classes whose predictions are stable across the old and new models.

4.2.3 Attention Loss

Our intuition is that rather than just remembering the old decisions, ensuring that the model also remembers the reason behind that decision would be a better strategy. We do this by imposing an additional loss called attention loss, $\mathcal{L}_{att}$. It is an L1-loss between attentions of old classes before and after training on the new class. This ensures that the reason behind the decisions, i.e. explanations/attentions are preserved during incremental learning of new tasks.

Algorithm 4 Proposed Solution

```
Input: Input: Set of Tasks $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_c\}$ where $\mathcal{T}_i$ represent the task of binary classification of
        the class $i$ that contains input image $x$ and its corresponding class ground truth $y_{gt}$ as $(x, y_{gt})$
for each $\mathcal{T}_i$ in $\mathcal{T}$ do
    $f_{base} = f.copy()$
    $x_{recons}, \hat{y}, \hat{\mathcal{A}} = f(x)$, where $x \in \mathcal{T}_i$
    $l_c = \mathcal{L}_{class}(y_{gt}, \hat{y})$
    $l_{aux} = \mathcal{L}_{aux}(x, x_{recons})$
    if $(i > 1)$ then
        $\_, y, \mathcal{A} = f_{base}(x)$
        $l_{kd} = \mathcal{L}_{kd}(y, \hat{y}, i-1)$
        $l_{att} = \mathcal{L}_{att}(\mathcal{A}, \hat{\mathcal{A}}, i-1)$
    end
    $\mathcal{L}_{total} = \alpha l_c + \beta l_{aux} + \gamma l_{kd} + \delta l_{att}$
end
```

Using any explainable method $\mathcal{XAI}$ like GradCAM [15], PCAM [52], the attention loss is computed as below:

$$\mathcal{L}_{att}(\mathcal{A}_{\mathbf{o}}, \hat{\mathcal{A}}_{\mathbf{o}}, N) = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathcal{A}_o^{(i)} - \hat{\mathcal{A}}_o^{(i)} \right\| \tag{4.4}$$

where N is the number of old classes; $\mathcal{A}_o^{(i)}$ and $\hat{\mathcal{A}}_o^{(i)}$ are the recorded and current attention maps of the class i from the model f which can be obtained as follows:

$$\mathcal{A}_o^{(i)} = \mathcal{XAI}(f(x^{(i)})), \hat{\mathcal{A}}_o^{(i)} = \mathcal{XAI}(\hat{f}(x^{(i)})) \tag{4.5}$$

here, x represents the input image and f and $\hat{f}$ represents the model before and after training on new classes.

4.2.4 Algorithm

During the course of IL a new task-specific head is added as shown in Figure 4.1(b) whenever the model is trained on a new task. Each head gives attention maps and auxiliary task output along with the class probabilities. A total loss function $\mathcal{L}_{total}$ is calculated as weighted average of binary cross entropy loss $\mathcal{L}_{class}$ and 3 other losses defined earlier:

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{class} + \beta\mathcal{L}_{aux} + \gamma\mathcal{L}_{kd} + \delta\mathcal{L}_{att} \tag{4.6}$$

where $\alpha + \beta + \gamma + \delta = 1$. The step by step details are provided in Algorithm 4.

4.3 Experiments and Results

4.3.1 Dataset details

Experiments were done on CheXpert dataset, a large public dataset of chest X-rays, consisting of 224,316 chest radiographs of 65,240 patients. The images were collected from Stanford Hospital between 2002 and 2017 [12]. Though the dataset has 14 different categories, we focus only on 5 for illustrative purposes: Cardiomegaly, Edema, Consolidation, Atelectasis, and Pleural Effusion.

4.3.2 Implementation details

The autoencoder network uses DenseNet121 [19] as the encoder and the corresponding decoder is constructed connected through skip connections, similar to UNet [45]. The output of the encoder (i.e. the latent space features) is given as input to the task-specific heads, each of which consists of a convolution layer and a global pooling layer called PCAM [52] which helps generate localized attention maps in chest x-rays. These maps are passed on to a final convolution layer with 1×1 kernel to generate the logits of the network as shown in Figure 4.1(b). The model was trained on the 5 following classes

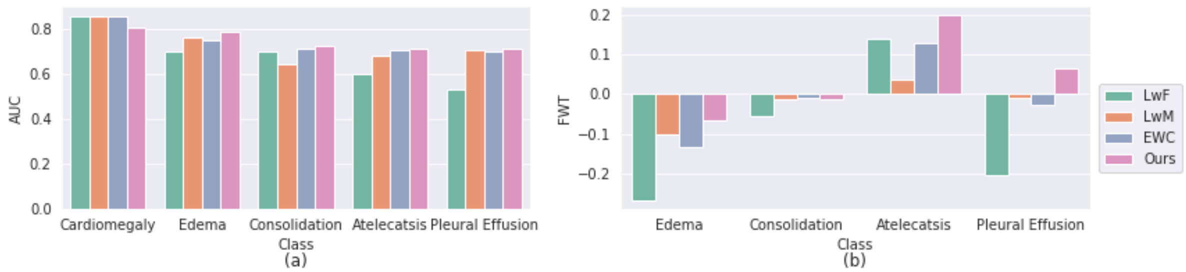


Figure 4.2: (a) AUC values of all classes during incremental learning (b) FWT on class i after training on class $i-1$

sequentially, in the same order: Cardiomegaly, Edema, Consolidation, Atelecatsis and Pleural Effusion. We used $\alpha = 0.4$, $\beta = 0.2$, $\gamma = 0.2$, and $\delta = 0.2$ as the weights for the total loss L_{total} in equation 4.6. We also report results of 3 state of the art methods in IL which are based on similar assumptions as ours: LwM [9], LwF [30], EWC [26])

4.3.3 Results

AUC was used to assess classification during the course of IL, while transfer metrics were used to assess the influence of a new task on the older tasks.

Backward Transfer (BWT) [33]: BWT is the change in a model's AUC value (${y_c}^{(i)}$) on a task T_i after learning sequence of new tasks $\forall_j T_j$ where $j > i$.

$$BWT(c) = {y_c}^{(j)} - {y_c}^{(i)} \tag{4.7}$$

Ideally, $BWT \geq 0$; high negative values indicate high degree of CF and a positive value indicates a boost in the AUC for the old task after learning a new task.

Forward Transder (FWT) [33]: FWT measures the influence of IL on the new task T_i. Hence, it is computed by assuming no IL or a randomly initialised network as a reference.

$$FWT(c) = {y_c}^{(i)} - r_c \tag{4.8}$$

where r_c is the AUC value of class c for a randomly initialized model before training and y_c is the AUC of the model before starting IL on task T_i. A positive FWT indicates that the model is able to exploit the knowledge learned from previous tasks.

Table 4.1 lists the BWT for each class after incrementally learning new classes. Our method is seen to outperform all the state-of-the-art methods by a good margin and for a few cases, IL seems to actually have helped increase the performance of the old class as BWT is above zero.

New class	LwF	LwM	EWC	Ours
On Cardiomegaly				
Edema	-0.047	-0.040	-0.040	**-0.002**
Consolidation	-0.151	-0.058	-0.080	**-0.005**
Atelecatsis	-0.123	-0.060	-0.136	**0.004**
Pleural Effusion	-0.107	-0.091	-0.146	**0.004**
On Edema				
Consolidation	-0.018	-0.013	-0.023	**0.001**
Atelecatsis	-0.078	-0.038	-0.025	**-0.016**
Pleural Effusion	-0.179	-0.036	-0.060	**0.003**
On Consolidation				
Atelecatsis	-0.073	-0.018	-0.016	**-0.01**
Pleural Effusion	-0.037	-0.029	-0.025	**-0.02**
On Atelecatsis				
Pleural Effusion	**0.057**	-0.039	-0.004	0.012

Table 4.1: BWT on each class after training on new classes incrementally

New class	A	X	D	AD	XD	AX	AXD
Edema	-0.021	-0.035	-0.047	-0.007	-0.004	-0.004	**-0.002**
Consolidation	-0.012	-0.013	-0.151	-0.009	-0.015	-0.009	**-0.005**
Atelecatsis	-0.017	-0.02	-0.123	-0.006	-0.007	-0.007	**0.004**
Pleural Effusion	-0.022	-0.019	-0.107	-0.002	-0.002	-0.001	**0.004**

Table 4.2: BWT values obtained in the ablation study. Column headings indicate the entities included in the study. A:Attention loss (Modified LwM), X: Auxiliary task, D: Distillation loss (LwF), AD: Attention and Distillation losses and so on.

Figure 4.2(a) shows a bar plot of AUC scores for all the classes which are learnt incrementally using different methods. If a model struggles to achieve reasonable AUC score during the course of IL, we can infer that it has poor plasticity i.e. it is unable to adapt to new tasks. In terms of plasticity, our method for IL is seen to be on par with the state-of-the-art methods; LwF is seen to cause less plasticity as new classes have relatively low AUC score. This can be also be explained by the sudden surge in BWT measure on Atelecatsis after training on Pleural Effusion class using the LwF method.

From Figure 4.2(b) it can be observed that our method gave highest FWT scores for all the classes and it is due to presence of the auxiliary reconstruction task that the model has exploited to transfer old knowledge to new classes.

4.3.4 Ablation Study

Our proposed has three important components: Auxiliary Task, Attention Loss and Knowledge Distillation loss. Each component plays a crucial role in IL. An ablation study was done to assess the importance of these components in IL. The BWT values were computed for the Cardiomegaly class after incrementally learning new classes. These are listed in Table 4.2. It is notable that the results of the using 'A' alone, which corresponds to our Modified LwM approach, is much better than the original LwM proposed in [9]. This attests to our intuition that it is important to also remember why an image does not belong to certain classes. From the ablation study results, it can be established that inclusion of the auxiliary module along with either attention or distillation loss serves to increase the BWT value. The reason for this is that the presence of a common auxiliary task acts as a regularizer preventing drastic change of weights during IL. The proposed method calls for all 3 components in the design and it is seen to yields the best BWT value.

4.4 Conclusion

In this work, we proposed a new method for IL to alleviate CF. It was based on a reasoning that the network should remember not only the decisions but also its causation. We also introduced an auxiliary task to provide task-invariant features to the task-specific heads to reduce the CF further. Our idea was validated on a fairly difficult problem of detecting multiple diseases from chest X-rays. The proposed method performed best in terms of BWT with higher values in all cases with even a boost in performance on an old task after learning a new task in a few cases. These results are encouraging for considering a potential extension of the scope of a existing CAD system.

Chapter 5

Conclusions

In this thesis, we explored two main areas of Lifelong learning: (i) Continual Learning where the goal is to adapt to a sequence of datasets with different distributions without forgetting the knowledge acquired through old datasets and (ii) Incremental Learning where the model learns new classes without while maintaining the performance on the old classes. The summary of our proposed solutions is enumerated below:

- Chapter 2: Our solution is based on the idea of learning in three stages: Encoding, Storing and Retrieval. We argue that the crucial aspect of intelligence is not to act/decide when uncertain which can be quantified using Bayesian Neural Networks (BNN) that learn probability distributions over parameter space. We encode the knowledge in the form BNN models and store them as an ensemble. During inference, to retrieve the relevant knowledge out of the ensemble, we use the uncertainty measure that can reject the predictions done by models with high uncertainty. We evaluated our proposed solution on three major problems: Malaria disease classification, Skin lesion segmentation, and Brain MRI ventricle segmentation and our method has outperformed other state-of-the-art techniques by a significant margin.
- Chapter 3: We proposed an approach to handle a sequence of datasets from multiple sources with different acquisition systems. As these datasets lie on a different manifold, a single neural network will fail to converge on all the datasets with a mere finetuning approach. With the help of a Reformer, which is a denoising auto-encoder, we propose to move the samples from different datasets are moved to a common manifold. This step helps the model to adapt to different datasets and limits CF to a tolerable extent. The approach is assessed on two classical problems: Retina Vessel Segmentation and Skin Melanoma Classification. We compared our method with other state-of-the-art regularization-based methods and the experiment results indicate that using a Reformer as a prior step to CL training effectively addresses CF.
- Chapter 4: We proposed a solution for IL that has three major components: (i) An auxiliary task common to all the tasks enforces the model to generate task-invariant features, (ii) Distillation Loss on the decisions of old classes and (iii) Attention Loss on the attention maps belonging to

old classes ensures that model remembers its previous decisions and causative factors behind all the decisions. We demonstrate our approach by incrementally learning 5 different tasks on Chest-Xrays and compare the results with the state-of-the-art regularization methods. Our approach performs consistently well in reducing CF in all the tasks with almost zero CF in most of the cases unlike standard regularisation-based approaches.

5.1 Future Directions

In this thesis, we assumed that data belonging to the old tasks will no longer be accessible while learning new tasks. With the same assumption, we identified following areas to extend our work in future:

1. **Incremental Learning with overlapping classes:** As explained in Chapter 4, we have a dedicated head for each new task in a typical IL framework. However, having a different head is redundant if there are common classes among these tasks. Our proposed method can be extended further to cover the above practical scenario efficiently.

2. **Few-shot Continual Learning:** In the Chapters 2, 3, 4 we subject the model to IL/CL by training on new ”datasets” without forgetting the knowledge from old ”datasets”. These are standalone datasets with enough samples to perform supervised learning. However, in the real world, the data is not available in large amounts at a time. Hence, a few-shot CL system that continuously learns from only a few samples at a time while maintaining overall performance is desirable.

Related Publications

- Prathyusha Akundi and Jayanthi Sivaswamy. 2021. Manifold learning to address catastrophic forgetting. Proceedings of the Twelfth Indian Conference on Computer Vision, Graphics and Image Processing. Association for Computing Machinery, New York, NY, USA, Article 28, 1–5. https://doi.org/10.1145/3490035.3490287

- P. Akundi and J. Sivaswamy, ”Incremental Learning for a Flexible CAD System Design,” 2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI), 2022, pp. 1-4, doi: 10.1109/ISBI52829.2022.9761688.

Bibliography

[1] Bcn20000: Dermoscopic lesions in the wild. In *International Skin Imaging Collaboration (ISIC) Challenge on Dermoscopic Skin Lesion Analysis 2019*, 10 2019.

[2] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154, 2018.

[3] A. Asman, A. Akhondi-Asl, H. Wang, N. Tustison, B. Avants, S. K. Warfield, and B. Landman. Miccai 2013 segmentation algorithms, theory and applications (sata) challenge results summary. In *MICCAI Challenge Workshop on Segmentation: Algorithms, Theory and Applications (SATA)*, 2013.

[4] M. Berseth. Isic 2017 - skin lesion analysis towards melanoma detection. *ArXiv*, abs/1703.00523, 2017.

[5] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

[6] A. Budai, R. Bock, A. Maier, J. Hornegger, and G. Michelson. Robust vessel segmentation in fundus images. *International journal of biomedical imaging*, 2013, 2013.

[7] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

[8] M. R. Cowie, J. I. Blomster, L. H. Curtis, S. Duclaux, I. Ford, F. Fritz, S. Goldman, S. Janmohamed, J. Kreuzer, M. Leenay, et al. Electronic health records to facilitate clinical research. *Clinical Research in Cardiology*, 106(1):1–9, 2017.

[9] P. Dhar, R. V. Singh, K.-C. Peng, Z. Wu, and R. Chellappa. Learning without memorizing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5138–5146, 2019.

[10] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. D. Hoffman, and R. A. Saurous. Tensorflow distributions. *ArXiv*, abs/1711.10604, 2017.

[11] Y. Dong, W. D. Pan, and D. Wu. Impact of misclassification rates on compression efficiency of red blood cell images of malaria infection using deep learning. *Entropy*, 21(11):1062, 2019.

[12] I. et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 590–597, 2019.

[13] N. C. F. C. et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 168–172, 2018.

[14] R. et al. A patient-centric dataset of images and metadata for identifying melanomas using clinical context. *arXiv preprint arXiv:2008.07360*, 2020.

[15] S. et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[16] T. et al. The ham10000 dataset: A large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5, 03 2018.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[18] A. D. Hoover, V. Kouznetsova, and M. Goldbaum. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Transactions on Medical Imaging*, 19(3):203–210, March 2000.

[19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[20] J. Kawahara, S. Daneshvar, G. Argenziano, and G. Hamarneh. Seven-point checklist and skin lesion classification using multitask multimodal neural nets. *IEEE Journal of Biomedical and Health Informatics*, 23(2):538–546, 2019.

[21] R. Kemker and C. Kanan. Fearnet: Brain-inspired model for incremental learning. *arXiv preprint arXiv:1711.10563*, 2017.

[22] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017.

[23] D. N. Kennedy, C. Haselgrove, S. M. Hodge, P. S. Rane, N. Makris, and J. A. Frazier. Candishare: a resource for pediatric neuroimaging data, 2012.

[24] M. E. Khan and D. Nielsen. Fast yet simple natural-gradient descent for variational inference in complex models. In *2018 International Symposium on Information Theory and Its Applications (ISITA)*, pages 31–35. IEEE, 2018.

[25] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[26] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[27] Y. Kwon, J.-H. Won, B. J. Kim, and M. C. Paik. Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation. *Computational Statistics & Data Analysis*, 142:106816, 2020.

[28] T. L. Lai. *Introduction to Hastings (1970) Monte Carlo Sampling Methods Using Markov Chains and Their Applications*, pages 235–256. Springer New York, New York, NY, 1997.

[29] N. Lei, D. An, Y. Guo, K. Su, S. Liu, Z. Luo, S.-T. Yau, and X. Gu. A geometric understanding of deep learning. *Engineering*, 2020.

[30] Z. Li and D. Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[31] J. Liang, D. Hu, and J. Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020.

[32] D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning. In *NIPS*, 2017.

[33] D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning. *arXiv preprint arXiv:1706.08840*, 2017.

[34] A. Mallya and S. Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

[35] M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[36] K. B. McDermott and H. L. Roediger. Memory (encoding, storage, retrieval). *General Psychology FA2018. Noba Project: Milwaukie, OR*, pages 117–153, 2018.

[37] R. Mehta and J. Sivaswamy. M-net: A convolutional neural network for deep brain structure segmentation. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 437–440. IEEE, 2017.

[38] T. Mendonça, P. M. Ferreira, J. S. Marques, A. R. S. Marçal, and J. Rozeira. Ph2 - a dermoscopic image database for research and benchmarking. *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5437–5440, 2013.

[39] D. Meng and H. Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 135–147, 2017.

[40] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera. A unifying view on dataset shift in classification. *Pattern recognition*, 45(1):521–530, 2012.

[41] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.

[42] S. Rajaraman, S. K. Antani, M. Poostchi, K. Silamut, M. A. Hossain, R. J. Maude, S. Jaeger, and G. R. Thoma. Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. *PeerJ*, 6:e4568, 2018.

[43] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[44] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

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

[46] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[47] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, pages 4528–4537. PMLR, 2018.

[48] K. Shridhar, F. Laumann, and M. Liwicki. A comprehensive guide to bayesian convolutional neural network with variational inference. *ArXiv*, abs/1901.02731, 2019.

[49] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, and B. van Ginneken. Ridge based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, 23(4):501–509, 2004.

[50] M. Tan and Q. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[51] A. Worth. The internet brain segmentation repository (ibsr), 1996.

[52] W. Ye, J. Yao, H. Xue, and Y. Li. Weakly supervised lesion localization with probabilistic-cam pooling. *arXiv preprint arXiv:2005.14480*, 2020.

[53] J. Yoon, E. Yang, J. Lee, and S. J. Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.

[54] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. *Proceedings of machine learning research*, 70:3987, 2017.