APPROVAL SHEET

Title of Dissertation: Semi-supervised Expectation Maximization with Contrastive Outlier Removal.

Name of Candidate: Sumeet Menon
Doctor of Philosophy, 2022

Dissertation and Abstract Approved: David R. Chapman
(Dr. David Chapman)
(Assistant Professor)
(Computer Science)

Date Approved: 04/30/2022

ABSTRACT

Title of Document: SEMI-SUPERVISED EXPECTATION MAXIMIZATION WITH CONTRASTIVE OUTLIER REMOVAL.

Sumeet Menon, PhD 2022

Directed By: David Chapman, Assistant Professor, CSEE

Semi-supervised learning has proven to be one of the most widely used techniques to overcome the concern of limited labels. One of the concerns while using neural networks for semi-supervised learning in presence of an extremely small labeled dataset is the occurrence of confidently predicted incorrect labels. This phenomenon of confidently predicting incorrect labels for unsupervised data is called *confounding bias*. Even though pseudo-labeling and consistency regularization are among the state-of-the-art techniques for semi-supervised learning, these techniques are susceptible to the problem of confounding bias while using neural networks. We propose a methodology that could help neural networks overcome this problem by leveraging information from unlabeled images using cluster generating techniques and smoothness generating techniques in a tightly-coupled way to overcome the fundamental problem of outliers. These techniques could help the model to learn certain attributes from the image which could not be learned from the original resolution of the unlabeled images. We argue both theoretically and empirically that contrastive outlier suppression is a necessary yet overlooked criteria in the application of EM-derived latent bootstrapping, because discrimination models such as neural networks have the potential to make erronous predictions with high confidence if these datasets are far from the decision boundary, whereas generative methods for

which Expectation Maximization (EM) was originally designed have no such issue. Contrastive outlier suppression is derived under the assumption that the latent feature vector predictions should follow a multivariate gaussian mixture distribution. Our results show that contrastive latent bootstrapping greatly improves semi-supervised classification accuracy over a baseline, and furthermore when combined with a state-of-the-art consistency regularization method, our results achieve the highest reported semi-supervised accuracy for the CIFAR-10 classification using only 250 labeled sample images.

SEMI-SUPERVISED EXPECTATION MAXIMIZATION WITH CONTRASTIVE OUTLIER REMOVAL.

By

Sumeet Menon

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2022

Acknowledgements

I would like to thank and acknowledge each and every person who has helped me along this road of education. First and foremost I am forever grateful to my parents Mr. Suresh Menon and Mrs. Shainey Menon who have sacrificed a lot to help me reach where I am today. As a kid, I remember my family always being supportive to learn anything I ever needed to and I am forever in debt to everyone in my family for encouraging me to do my PhD and helping me throughout the journey especially Mrs. Sunita Menon and Dr. Achuth Padmanabhan.

Having an incredible mentor is what one needs to be able to discover their potential to work on something novel and for that I would like to thank Dr. David Chapman for being my advisor in the VIPAR lab and helping to bring out the best of me. I truly believe that apart from being a great mentor, he is a great saxophone player and I consider myself lucky to be able to perform as a duo with him on a few occasions.

I would like to thank Dr. Yelena Yesha for being a constant support and introducing me to the Center for Accelerated Real Time Analysis (CARTA) at UMBC. I would also like to thank Dr. Milton Halem and Dr. Phuong Nguyen for working with me on various CARTA projects. I would like to thank my committee members Dr.Tim Finin, Dr. Tim Oates and Dr. Alberto Santamaria-Pang for their time and efforts.

Lastly, I would also like to thank the computer science department for such a great program and my fellow-graduate students for working with me. I shall cherish all the moments spent in the lab and virtual meetings.

Table of Contents

[1]

[2]

This Table of Contents is automatically generated by MS Word, linked to the Heading formats used within the Chapter text.

List of Tables

List of Figures

Chapter 1: Introduction

Learning with limited labeled data samples is a problem that deep learning has not yet overcome. While humans can achieve this task seamlessly, contemporary Artificial Intelligence (AI) algorithms still struggle to do so. These AI algorithms need to be trained with large labeled data volumes in order to perform well. And while great strides have been made in transfer learning, borrowing labeled volumes from another domain, we have not yet seen a widespread application of semi-supervised deep learning and it is fair to say that semi-supervised deep learning has not advanced to the level necessary to be widely applied. The applications for semi-supervised learning are numerous as in many domains, unlabeled data is plentiful yet high quality labeled data is scarce. Data labeling remains a task that is time consuming, expensive and error prone. Medical imaging is a particularly promising area, as labeling for classification and segmentation tasks require board certified radiologists or other highly trained medical professionals in order to perform manual labeling. Semi-supervised learning has been one of the most used algorithms to learn with a limited amount of data but has still not achieved the best solution to the problem.

Semi-supervised learning becomes very challenging when the labeled data volumes are very small relative to the unlabeled volumes. In this case, most methods are susceptible to a kind of confounding bias, in which the model at first learns some error due to the small labeled sample, and then proceeds to pseudo label the data incorrectly thereby reinforcing its prior error. If the initial labeled sample is

sufficiently large this problem does not occur. But as we decrease the labeled sample size, and increase the unlabeled sample size, this confounding bias becomes more and more of a concern. There is a strong need for semi-supervised learning algorithms that can learn with small volumes of labeled data and large volumes of unlabeled data.

We believe that the key to optimally addressing the confounding bias problem is to combine the well known smoothness and manifold assumptions into a single deep learning algorithm framework. The state-of-the-art methods for semi-supervised deep learning assume either smoothness or manifold but not both simultaneously. At present, the methods based on the smoothness assumption outperform those that assume manifold, but to the best of our knowledge the combination of these assumptions has not been adequately explored in a deep learning context. We review a variety of methods and express our reasoning why this novel approach is likely to address the confounding bias problem and allow deep learning methods to learn from substantially smaller labeled data volumes. As there are many possible ways to develop an algorithm based on either assumption, we believe that the search for the best way of combining smoothness and manifold assumptions will be a topic that will lead to many publications as well as an advancement of the field toward industry applications for which large volumes of unlabeled data are obtainable, yet data labeling is prohibitively expensive.

Thesis Statement

The *confounding bias* problem with semi-supervised bootstrapping and pseudolabeling can be addressed and overcome through *contrastive outlier removal* of samples not well represented in the labeled set.

Contributions

- **Observed that outlier samples, unexplainable by labeled data, can cause confounding bias in pseudolabeling techniques.**
  Observed that pseudolabeling methods suffer from confounding bias caused by high confident but incorrect "outlier" predictions of samples not well represented by labeled data.
- **Combination of contrastive outlier removal and consistency regularization yields state-of-art results for semi-supervised classification.**
  A combination of contrastive outlier removal and consistency regularization yields the highest reported accuracy in semi-supervised classification as measured by the semi-supervised CIFAR10 task with only 250 or 4000 labeled samples when the labels follow a uniform distribution.
- **Contrastive k-nearest neighbors technique to remove outlier predictions.**
  This technique compares k low-dimensional feature maps of images and has been observed to remove predictions that cannot be explained by the model.
- **Improved technique to reduce confounding bias for self-training.**
  Contrastive k-nearest neighbor along with gaussian filtering reduces the confounding bias overall to as low as 0.96% per expectation maximization bootstrapping iteration.

- **Novel technique to determine the quantity of manually annotated labels during active semi-supervised learning.**

  ASEM achieves an accuracy of 92% for computer aided diagnosis based classification.

Chapter 2: Background of semi-supervised learning

Machine learning (ML) is the study of complex algorithms which improves itself automatically with the help of training data. Deep learning (DL) on the other hand is a subset of a broader family of machine-learning algorithms based on artificial neural networks representation learning. The learning primarily can be supervised, semi-supervised and unsupervised. Deep learning has seen tremendous success in the areas of image classification and speech recognition. In order to make a model learn these attributes, it is necessary to feed the model with a large supervised labeled dataset in order to avoid the model from over-fitting. Also, adding manually annotated high quality labels to the data is an expensive task and often it is not possible to receive labels of every data sample. Semi-Supervised Learning (SSL) overcomes this task with the help of using partially labeled and unlabeled data samples in the model while training. As mentioned, in semi-supervised learning you often have a dataset S that has a labeled set S_L and unlabeled set S_U. The labeled dataset is usually much smaller than the unlabeled dataset. The primary goal of semi-supervised learning is to leverage the unlabeled dataset to build a stronger performing model than the model trained on only the small amount of labeled dataset. Also, eventually getting a model that would converge closer to a model that was trained on the dataset S which was completely labeled. Using this data for training helps us to create a prediction function for the model to identify which class the data sample belongs to. For example, when we use only S_U (which is a relatively lower sample size than S_U) to classify data points, the prediction function identifies a decision boundary that separates one class from the other. The aim of semi-supervised

learning is to use the information from the unlabeled dataset S_U which could help to enhance the decision boundary of the model and make much more accurate predictions. Semi-supervised learning initially appeared in the form of self-training [4]. This paper focuses on training the model with the partially labeled dataset and predicting the labels for a subset of the unlabeled dataset. These predictions were then added to the training dataset iteratively to increase the accuracy of the model.

There have been several methods that have been tried over several years. These methods can be broadly classified into 4 categories [33]:

Consistency Regularization: These methods mainly focus on generating additional training samples without affecting the model and without changing the prediction of the unlabeled samples significantly. The model can be trained to have consistent predictions for unlabeled data points and its perturbed version [1,2,11,12,13,35,36].

Proxy-Label Methods: These methods emphasize on labeling a subset of an unlabeled dataset based on a particular condition and can be added to the training sample to be re-trained to increase the sample size of the dataset. Some of the examples of these methods are Self-training, Co-training and Multiview training [4,10,15, 25].

Generative models: When we have a dataset G which has similar characteristics but is not exactly the same class of the small sample size of the labeled dataset S, the dataset G can be generated using generative models. These generative models can be used to transfer the weights to generate the sample dataset S to increase the sample size of the labeled dataset. [21,24,38,40,41]

Graph-Based Methods: The data samples are considered as a nodes on the graph and the model's predictions are compared between the labeled and the unlabeled samples. The objective of these methods are to compare the similarity between two nodes based on how strong the edges are between the two nodes [33,43,44,8].

2.1 Main Assumption of Semi-Supervised Learning

Before diving deeper into some of these techniques, let's try to understand some of the general assumptions that need to be made during semi-supervised learning. Some of these assumptions about the data structure need to be true in order to perform sem-supervised learning on a finite dataset.

Some of these assumptions are as follows:

- The Smoothness Assumption

If there are two points x_1 and x_2 that lie close to each other in a high-density region then its corresponding output y_1 and y_2 should be close to each other [4]. In other words, if there are two data points that belong to the same class and belong to the same cluster in a high density region with all other data points with the same class, their corresponding output should also be close to each other. This is also true for the inverse scenario that if two points are separated from a low density region and do not belong to the same cluster, their corresponding outputs should also be away from each other. This assumption is mainly used during classification tasks which helps us to verify that the predicted sample can be added to the training sample or not.

- The Cluster Assumption

If there are two points that belong to the same cluster, they are likely to belong to the same class [4]. This assumption is made on the basis that the input samples that belong to the same class are considered as one cluster. This makes it easier to assign and verify if the model's prediction belongs to a particular class. It can also be called the low-density assumption. For example, if we make a decision boundary at a high density region of the input samples, we are likely to make clusters of samples that belong to different classes that violate the basic principles of semi-supervised learning.

- Manifold Assumption

The (high-dimensional) data lie (roughly) on a low-dimensional manifold [20]. In a high-dimensional dataset the requirement of volume of data-sets also go up during generative tasks which makes it quite challenging. This assumption is based on the fact that if the input data can be represented in a low dimensional space, the input unlabeled data can be used to make a lower dimensional representation, learn some special characteristics which can be used while learning the labeled samples which enhances the performance of the model while classification.

2.2 Consistency Regularization

The principle of consistency regularization is that the model has to be less sensitive even when there are different perturbations applied to the training data. One of the most common techniques used in consistency regularization is data augmentation. The idea behind data augmentation is to considerably increase the training data volume. In some methods, the input data is stretched and expanded which still keeps

the sample aligned to its original class. This helps the model to learn various aspects of the data that it was not able to learn with just one form of the input sample. In semi-supervised learning, consistency regularization is used with the idea that the model predicts the same class to the perturbed version of the unlabeled sample. In other words, the model should be able to predict the unlabeled sample x exactly the same way it predicts the class for the augmented version of x [2]. In the "Mean Teacher" [1] algorithm, the exponential moving averages of the model parameters are taken to get a much more stable target prediction and this method has significantly shown improvements in results. One of the drawbacks in these types of methods is that they use domain specific augmentations. These problems have been overcome by techniques like "Virtual Adversarial Training" [35]. These techniques believe in generating additive samples with similar characteristics to increase the data volume and thus avoiding random augmentations. There are also approaches such as Transformation Consistency where they propose that if the x and x' that are at a distance from each other are fed into the model their corresponding predictions y and y' need to be at the same distance from each other [36]. The most common distance measurement techniques are Mean Squared Error (MSE), Kullback-Leiber (KL) divergence and Jensen-Shannon (JS) divergence. In other words, consistency regularization obtains pseudo-labels from the model's predicted distribution based on the different augmentations applied to the input image [11, 12, 13, 37].

2.3 Proxy-Label Methods

Semi-supervised learning focuses on using the unlabeled data as a part of the training data along with the labeled samples. Proxy-label methods assign a soft label to these unlabeled samples. Self training is one of the oldest and simplest techniques that uses the model predictions for unlabeled data to be added on to the training sample followed by re-training the model with an increased sample size of training data. Formally, when a model m is training on a dataset x, the predictions of the unlabeled dataset m(x) is compared with a predetermined threshold T. If the prediction surpasses that threshold, it is then added to the dataset to re-train the model [4,10,15, 25]. Co-training is a part of multi-view training where a dataset S can be represented as 2 independent feature sets S1 and S2. After the model is m1 and m2 are trained on the respective datasets, at every iteration, the predictions that surpass the predetermined threshold from exactly one model are then passed to the training dataset of the final model [26, 23]. In recent times, co-training has been used in 3-D medical imaging where the coronal, sagittal and axial view of the data was trained on three different networks [9]. A consensus model between these three networks was used to predict the label for the unlabeled dataset. The major limitation with such types of models is that they are unable to correct their own mistakes and any bias or wrong prediction detected by the model results in confident but erroneous predictions. One of the papers proposed by Yalnizet et al. [39] uses self-training to improve ResNet-50 and work towards making a robust model even after being subjected to various perturbations. Initially, the model is trained on unlabeled images and their

proxy-labels which is then used to fine-tune the model with the help of the labeled images in the final stage.

2.4 Generative Methods

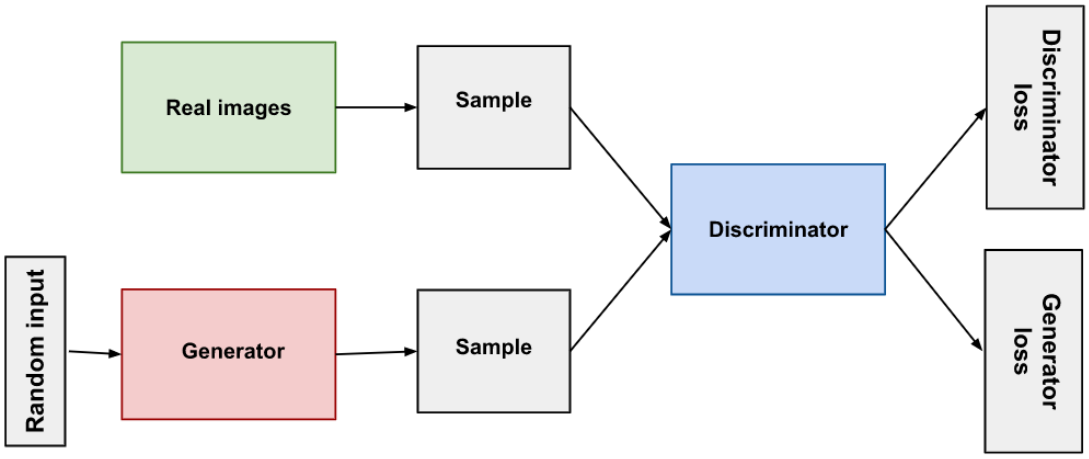

Figure 2.1: Generative Adversarial Network

Generative Adversarial Networks [42] consists of a generator G and a discriminator D (Figure 2.1). The generator receives a latent sample z~p(z) which is sampled from p(z) and tries to map it to the input sample. The discriminator on the other hand fetches a sample either from the generator or the real image and tries to classify it as "fake" or real. There have been many other proposed extensions using the hinge loss, wasserstein loss (WGAN) etc. The semi-supervised GAN (SGAN) [21] is an extension of traditional Generative Adversarial Networks. As such, the model works on the unsupervised mode and the supervised mode. In the unsupervised mode, the model is initially trained on unlabeled datasets for the generator to generate images and the discriminator extracts certain features to classify the generated images as real or fake. This discriminator is then used later as a starting point on the same dataset to

predict the class of the samples which is a part of the supervised mode. The features extracted by the discriminator during the unsupervised mode helps the model to learn certain features which could enhance the performance of the model during supervised classification. For example: Odena et al. show how GAN based semi-supervised learning on the MNIST dataset can perform as well or better than just a stand-alone CNN model [21]. Salimans et al. [38] present how Semi-supervised GANs perform equally well or better in many image classification techniques including the MNIST dataset. Fujino et al. [24] present how SGANs along with bias correction techniques like maximizing conditional entropy can be used with expectation maximization for semi-supervised learning. Kigma et al. [40] expanded the work on using generative techniques for generative sem-supervised techniques [41]. They use the generative distribution of the data to improve the classification performance of the model. Variational Autoencoders (VAE) are one of the approaches that use generative techniques for semi-supervised learning. Initially, the model is trained on both the labeled and the unlabeled data. Using the fully trained VAE, the observed labeled samples are modified into a low dimensional latent space sample. These samples can then be used with the associated label of the original data sample for classification. These low dimensional embeddings are much more informative since they are formed by independent Gaussian posterior parameterized by the encoder of the VAE.

2.5 Graph Based Methods

Graphs can be used to express the relations between any data points which could express the data in a general and global manner. In this technique, every data point x_i

is represented as a node on a graph and the edges e_i represent the relation between the two nodes. For eg. if we have a graph G = (V,E), where V = $\{v_1,v_2,v_3,...v_n\}$ are the nodes on the graph and E = $\{e_{ij=1}^{n}\}$ represent the relation between the vertices. This relation between nodes and edges on the graph are represented with the help of an adjacency matrix A_{ij}. In the adjacency matrix the relation between the nodes is represented by a non-negative number and if there is no relation, $A_{ij} = 0$. The adjacency matrix can either be derived using the similarity metric between two data-points [43] or could be derived using knowledge graphs. Graph based classification can be broadly classified into four categories: node classification, link prediction, clustering, and visualization [33]. Graph based techniques can be transductive and inductive in nature as well. Transductive methods are capable of only producing labels for the samples seen during training while inductive methods are capable of both getting information from the labeled samples and transferring the information to the unseen and unlabeled samples. The node classification approaches are one of the most used methods in semi-supervised learning [44]. This method works with a general idea that the nodes which are close to each other have the same label or the labels that are close to each other have the same embedding. Xu et al. [8] proposed using the technique of graph heat to capture information from the neighboring nodes. GraphHeat is a set of filters which takes the graphical representation as an advantage to get information from neighboring nodes for smooth labelling.Qimai et al. [14] present two methods in specific called Graph Labeled Propagation (GLP) and Improved Graph Convolutional Networks (IGCN). In Graph Labeled Propagation, a low pass filter is used on the data which is treated as a graph

to predict the vertices and then pass this new version of the data into a classifier. The advantage is that we are able to inject graphical relations into the data which shows a better performance. IGCN on the other hand replaces the weight matrix with a k-exponent weight matrix which makes the graph smoother and shows better performance than GLP. Papandreo et al. depicts a graphical model to evaluate the joint PDF of picture names, pixel esteems, and pixel marks utilizing pixel names as an idle variable [19]. Yang et al. [27] confirm that graphs can basically be constructed by us as a feature, which could be the distance between the instances or directly derived from external data such as knowledge graphs. In the paper, they have chosen a data-set in which a graph is already given which does not intersect with the features. They have concentrated on a transductive and inductive framework which gives out a graphical representation of the information gained from the networks.

2.6 Related Methods

Active Learning

If some of the most sophisticated supervised learning tasks had the capability to choose the data to be learned from, they could perform much better than usual. Active learning is an interactive platform where at any point, the true label of the desired sample can be fed into the model to enhance its performance [45]. The two most used measures of choosing samples during active learning are informativeness and representativeness [33]. Informativeness is the measure which tells us how the model is affected by the addition of a particular unlabeled sample to the model.

Representativeness on the other hand measures how well an instance can be used to describe the structure of input patterns. Phuong et al. present how active semi-supervised learning can be used to classify medical modalities [46]. This work presents how the intervention of radiologists in the active-learning section provides diagnostic labels to the unlabeled samples that could enhance the performance of the statistical model with expectation maximization. Active learning relates to the common problem of semi-supervised learning which is improving predictions with a very limited set of labeled data.

Transfer Learning

Transfer Learning is one approach that helps a statistical model to learn from a dataset with similar characteristics and structure like the limited, labeled training sample. Once the model is trained on the dataset from a similar domain, this model can then be used to transfer this knowledge on the training sample. This helps our model to achieve a better start point of convergence and extracts the important features from the training data efficiently. Razavian et al. proves that transfer learning permits CNNs to take in robust highlights from an enormous dataset which improves the classification performance [20]. Transfer learning can also be used in generative approaches to generate input samples for classification. We have also demonstrated that while generating COVID-19 x-rays [47]. Due to the limitations of the COVID-19 x-ray samples, we trained a Res-Net 50 model on similar pneumonia based x-rays and then this model was transferred to generate COVID-19 x-rays using a mean teacher approach. Shin et al. evaluate the use of state of the art (2016) convolutional neural networks for transfer learning to perform Computer Aided Diagnosis (CAD) and

compare the performance of these models versus CNN's being trained from scratch [30]. The author assesses their theory utilizing two CAD issues: thoraco-abdominal Lymph node (LN) identification and Interstitial Lung Diseases (ILD) characterization. Transfer learning can be broadly classified into 3 techniques:

1. Feature Extraction: This technique primarily focuses on using a pre-trained network's intermediate layers on another dataset for feature extraction. This technique works particularly well when you train a model on some dataset and reuse the intermediate layers and replace the fully connected layers while applying the model on a completely different dataset [52].
2. Fine Tuning: This technique in addition to the feature extraction method is focused on using the entire pre-trained network on a new dataset. While training it on the new dataset the entire model is trained from the first epoch as it is.
3. Two - Stage Transfer Learning: In contrast to the fine tuning technique, this method trains the final fully connected layers first for a few epochs and then the entire as it is believed that training the entire network from scratch might transfer harmful gradients to the fully connected layer which could degrade the models performance [52]. When we add new fully connected layers to a pre-trained intermediate layer network, the randomly initialized weights might cause instability in the model's learning.

Weakly - Supervised Learning

Since manually captioning of data samples can be very time consuming and expensive, weakly supervised learning is one of the approaches that have been used. The use of low quality images and larger training samples are constructed with the help of these annotators [48]. This method is very similar to supervised learning but the only difference is that real labels are replaced by weak labels. For example, In segmentation tasks, it is very difficult to obtain pixel level annotations so inexact locations like bounding boxes are taken into account. In such scenarios semi-supervised learning can be used with a limited quantity of labeled images and a larger quantity of weakly labeled images. Berthelot et al. explains the importance of augmentation anchoring with both weak and strong augmentations [5]. Papandreo et al. also present the combination of weakly labeled foreground and background pixel-level scores for the images with the image level labels. Rosenberg et al. also present the work of using weakly supervised labels to estimate the weak labels for the unlabelled dataset, compute expected statistics using fully labeled and weak label samples and update the parameters of the detection model [25].

2.7 Limitations of Semi-Supervised Learning

Unless the learner is absolutely certain about the assumptions between the labeled and the unlabeled data (if no assumptions are made between the labeled and the unlabeled data then it is called no-prior knowledge setting) then the one cannot hope a significant improvement over supervised learning with the use of semi-supervised learning [49]. Although it has been assumed that having extra knowledge about the

distribution of the unlabeled samples helps to enhance the performance of semi-supervised learning, it has been proved that having extra information about the unlabeled distribution also can lead to degrading the performance of the semi-supervised model. This generally happens when there is no relation assumed between the labeled and the unlabeled information which feeds the model with unwanted noise and takes the model away from the target.

2.8 Problems of Semi-Supervised Learning

Semi-supervised learning is mainly focused on leveraging the information from the unlabeled data to generalize a group of data to a particular class. Let's say we want to perform a binary classification task where we have a set of labeled and unlabeled data. In semi-supervised learning, we assume $n << N$ where n consists of labeled data and N is the set consisting of both labeled and unlabeled data. One of the simplest methods to tackle this problem would be self learning where we train our model on the labeled data n to predict the unlabeled data (N-n). These predictions are then used along with the labeled data to train the model again with the assumption that the model would learn additional features that were not present in the small subset of labeled data. These methods often fail to converge as we do not make an assumption for the relation between the labeled and the unlabeled data. In these methods, when the model is trained initially on only the labeled images to predict the labels for the unlabeled images, it tends to sometimes yield false predictions as well. When these uncertain labels for the unlabeled samples are added to the training dataset to re-train the model, there is a possibility to overfit the model to this dataset

which is a mix of true positives and false positives. This scenario where we overfit the model to these incorrect predictions is known as *confirmation bias*. Our main focus in this proposal is to build a framework which would consider all the assumptions of semi-supervised learning to reduce this condition of confirmation bias.

Recently, there have been approaches that include mixing up of the data and using various consistency regularization techniques that use data augmentation to overcome confirmation bias [2]. A network is said to be over-confident when it is trained on hard labels and it is this scenario which is taken care of by randomly mixing labels of different classes to reduce the prediction-confidence of these neural networks [7]. In other words, naive pseudo labeling results in overfitting models due to confirmation bias and hence, Arazo et al. had proposed a mix-up augmentation technique and a minimum number of labeled samples per mini batch are effective regularization techniques to overcome confirmation bias [7].

The first most addressed problem with semi-supervised learning is how to choose the subset of newly labeled data from the unlabeled data as an extension to the previously labeled data [50]. There have been techniques that use a confidence interval to determine the threshold for the newly labeled data. Sometimes, using a very high threshold does not always result in the improvement of the classification performance whereas using a very low threshold also results in misclassified examples which causes the model to diverge. Hence selecting an optimal threshold

value for a particular dataset needs to be addressed. Our model specifically focuses on overcoming this limitation.

The second most addressed problem is the use of the discriminator to finally classify the class of the input data. Some of the standard approaches that the previous work suggests is the use of the classifier which was used to train the labeled samples after the last iteration. This could lead to overfitting problems as well. There have also been approaches that suggest the use of an upper bound error rate to stop the iterations for the discriminator [51].

There are papers that take one or two out of the three assumptions of semi-supervised learning into consideration but do not consider all three assumptions while performing semi-supervised learning. We propose to consider all the three assumptions while constructing our model which we strongly believe can help to overcome the issue of confirmation bias. Graph-Based learning typically works on local similarity indexes to construct a graph over all data points. It is important to find a local similarity matrix to consider such assumptions for graphs. The high dimensionality datasets like images work specifically on the smoothness based assumptions where the model needs to be constant even for minor perturbations of the input images [28]. Semi-supervised learning too works on the similar assumptions as its supervised learning counterparts. The support vector machine instances often consider the cluster based assumptions which states that the decision boundary of the clusters should be in the low-density region. It has still not been discovered which

method and which particular assumption suits the best for which dataset and we try to focus on proposing a generalized technique that considers all the three assumptions which could be more robust and that could be widely applicable. There is no established method over the years to determine which assumption and method works the best for any particular dataset and that is what we try to achieve in this proposed technique.

2.9 Problems of Semi-Supervised Learning in Medical Imaging

Our proposed work has a lot of applications on medical imaging as well. Computer- Aided Diagnosis (CAD) has been explored a lot in the past and we know that labeled images are one of the most talked about hurdles that a researcher has to deal with. Deep learning requires a large amount of labeled data to establish a relation with the data distribution and the associated label. This could be a very challenging and expensive task as clinically labeling X-rays, (Computer Tomography) CT-scans or any other medical modality requires a radiologist. Medical imaging requires to surpass the metrics of sensitivity and specificity which is often signified by the Receiver-Operating-Characteristics (ROC) curve. Sensitivity calculates the amount of true positives in the set whereas specificity measures the amount of true negatives in the set. CAD focuses on increasing the amount of true positives that the model predicts. Another important problem is using hard labels with a large quantity of high resolution medical modalities does not allow the model to learn the detailed aspects of the data distribution. There are techniques that focus on CAD using the smoothness assumption to classify the ct-scans and also x-ray generation. There have been proposed techniques using only the cluster assumption during generation to determine

a threshold for the predicted images. Manifold assumptions have also been used to learn from low-dimensional representations of the input data but as to our knowledge all the three assumptions are not used in CAD which we have tried to achieve incrementally in our preliminary work. We have also demonstrated the use of a combination of active learning and semi-supervised learning with the help of human intervention for providing the model with labeled necessary samples. This work builds upon our recent work demonstrating that a Semi-supervised EM (SEM) algorithm was able to perform as well as a cross-validated Lung Cancer screening accuracy as compared to a fully-fully-supervised technique [10]. We have also demonstrated our algorithm that incorporates Active learning in combination with Expectation-Maximization (EM) in order to further improve cross-validated screening accuracy. The active learning component allows the algorithm to interactively suggest images to radiologists that need to be labeled, and the semi-supervised learning using EM allows the algorithm to incorporate a larger unlabeled training image dataset along with a smaller labeled dataset.

2.10 Dataset

- *MNIST Dataset*

The MNIST dataset consists of images of 10 classes which are digits from 0 to 9. These images consist of handwritten and is one of the datasets that have been used widely for deep learning and computer vision related problems. The MNIST database contains 60,000 training images and 10,000 testing images. Half of the training set and half of the test set were taken from NIST's training dataset, while the other half of

the training set and the other half of the test set were taken from NIST's testing dataset.

- *CIFAR 10 Dataset*

The CIFAR-10 dataset contains 60,000 32x32 color images in 10 different classes. The 10 different classes represent airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. There are 6,000 images of each class. It is also one of the most used datasets for validating the performance of semi-supervised learning approaches.

- *Medical Datasets*

We propose to use publicly available Covid-19 CT-scans and x-rays. The publicly available Covid-19 images consist of 1210 images of the Covid-19 class. We also propose to use a publicly available Kaggle dataset that consists of 5400 slices of malignant nodules. This helps us to verify our algorithm on a medical dataset problem as well.

Chapter 3: Deep Expectation-Maximization for Semi-Supervised Lung Cancer Screening

We begin by presenting a semi-supervised algorithm for lung cancer screening in which a 3D Convolutional Neural Network (CNN) is trained using the Expectation

Maximization (EM) meta-algorithm. Semi-supervised learning allows a smaller labeled data-set to be combined with an unlabeled data-set in order to provide a larger and more diverse training sample. EM allows the algorithm to simultaneously calculate a maximum likelihood estimate of the CNN training coefficients along with the labels for the unlabeled training set which are defined as a latent variable space. We evaluate the model performance of the Semi-Supervised EM algorithm for CNNs through cross-domain training of the Kaggle Data Science Bowl 2017 (Kaggle17) data-set with the National Lung Screening Trial (NLST) data-set. Our results show that the Semi-Supervised EM algorithm greatly improves the classification accuracy of the cross-domain lung cancer screening, although results are lower than a fully supervised approach with the advantage of additional labeled data from the unsupervised sample. As such, we demonstrate that Semi-Supervised EM is a valuable technique to improve the accuracy of lung cancer screening models using 3D CNNs.

3.1 Introduction

The accuracy of Computer Aided Diagnosis (CAD) for cancer screening has improved tremendously in recent years due to advances in Deep Learning[54, 32, 31, 55, 56, 29, 57]. The most successful deep learning algorithm to date for image classification being the Convolutional Neural Network (CNN). However, a major limitation of most deep learning algorithms including CNNs is that they are fully supervised. As such, low data volumes of labeled imagery are often a limiting factor, especially in the medical imaging domain in which accurate data labels require a great deal of clinical training to be able to construct. Semi-supervised learning is an

approach to expand the volume and diversity of the labeled training sample set by making use of an additional unlabeled training sample. This approach can increase data volumes thereby potentially improving screening accuracy. However, semi-supervised learning introduces additional complexity into the training process. Expectation maximization (EM) is a classical statistical meta-algorithm for estimating a model given some variables existing in a latent variable space [58]. Although in machine learning EM is often associated with relatively simple Gaussian Mixture Models (GMM), it can also be applied to more complicated deep learning models including CNNs thereby enabling CNNs to be trained in the presence of latent variables. In our approach, we perform semi-supervised learning by employing the EM meta-algorithm to train the maximum likelihood CNN model in the presence of both observed and unobserved (latent) image labels. For training and evaluation we combined two lung cancer screening data-sets: The Kaggle Data Science Bowl 2017 (Kaggle17) as well as the National Lung Screening Trial (NLST). Kaggle17 has a total of 1375 patients and the Computed Tomography (CT) scans include image volumes with associated binary clinical labels for 365 patients diagnosed with lung cancer within one year of the scan. The CT-scans for which the patient was diagnosed with lung cancer are labeled as 1 and the remaining are labeled as 0. NLST is another data-set we have used for training our model. The subset of the NLST made available to us for this model contained 4075 patients out of which 639 patients had been diagnosed with lung cancer. Combining two cancer screening data-sets in this way introduces potential challenges with cross-domain training. Even though both data-sets represent imagery from a similar lung cancer screening task, there are small

discrepancies that may impact the classification performance. Furthermore, the NLST data-set is 4x larger than the Kaggle17. As such, we want to evaluate the ability to train a model using supervised imagery from one data-set and to incorporate unsupervised imagery from another data-set to perform semi-supervised learning. In this study, we demonstrate that a semi-supervised training approach using EM is able to achieve greater accuracy than a fully supervised approach using either the NLST or Kaggle17 data-set on its own. We find that semi-supervised learning with EM is able to increase the available labeled data volume and thereby improve the accuracy of deep cancer screening tasks.

3.2 Related Work

Semi-supervised learning comprises a variety of methods to combine labeled and unlabeled training data to improve model performance [54; 59; 26; 60; 61]. Generative techniques attempt to model the probability distribution of the unlabeled imagery as a function of the model and labeled imagery [62; 63]. The most influential generative technique is Expectation Maximization introduced by Dempster et. al 1977 [64]. EM is originally intended to be applied to generative models although many important classifiers including CNNs are of the discrimination variety. Nevertheless, due to their ability to infer label probabilities from source imagery, it is possible to make the assumption that the discrimination model is approximately generative [65]. Papandreou et. al (2015) has combined EM with CNNs to infer pixel segmentation from weak image labels [19] demonstrating that EM is compatible with CNNs. There are also methods that use the labels predicted in the first iteration that combine with

the multiple machine learning models through ensemble methods [59] called Co-Training. Active learning has also been combined with Expectation Maximization [26] which is a related method that helps human annotators guide the SemiSupervised learning by the machine selecting data-points to fully label. In this scenario, the model constantly trains on new data and labels being fed in every iteration to classify the unlabelled data. Finally, extensions to EM have been incorporated into a variety of shallow classifiers including SVMs and HMMs. These alternative techniques include Co-Training and Co-EM which introduce additional non-linearity into Semi-Supervised learning by incorporating multiple views of the unlabeled imagery [61].

3.3 Data Preparation

Both the Kaggle17 and NLST datasets contain chest CT scans with slice thickness less than 3mm. These CT scans are a 3D volume with every voxel value having a single intensity value in accordance with the Hounsfield scale's standardized units. Data preprocessing was performed prior to model training. Each axial slice of the data-set is 512 × 512 and the number of axial slices per CT scan varied between 150 to 225 in each volume. We have created chunks of 20 slices for every patient and re-sized each image to 50 × 50. The suspicious nodules included in the data-set are on the order of 1 cm3 , slices from an example CT image are shown in figure 1

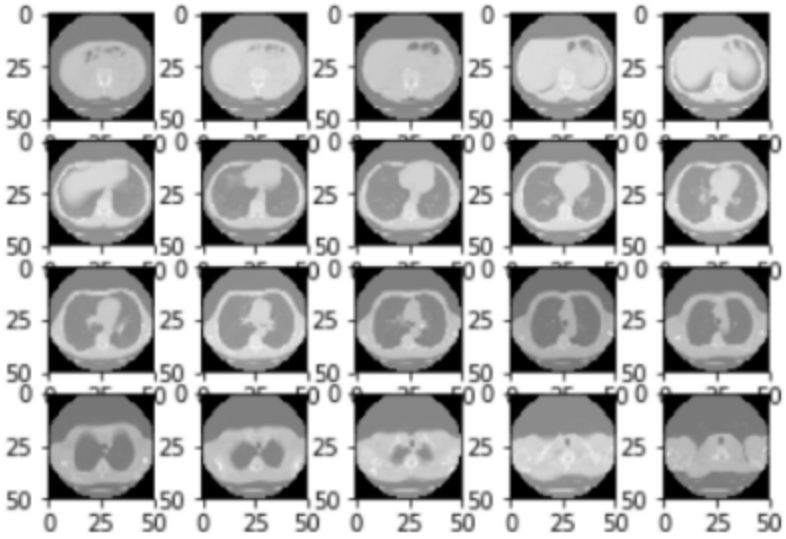

Figure 3.1: Axial slices of lung CT-scan subsequent to preprocessing.

3.4 CNN Architecture

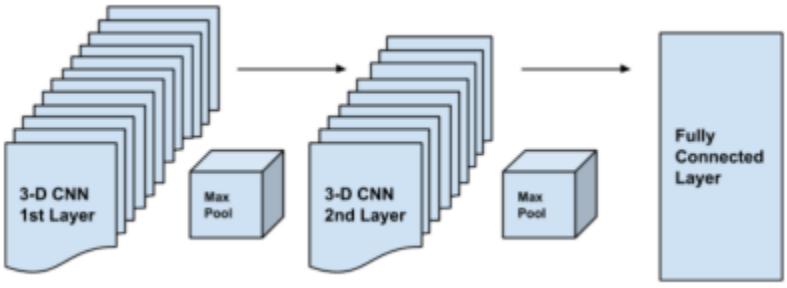

Figure 3.2: Architectural diagram of 2 layer 3-D CNN model.

Layer Number	*Name*	*Output Shape*
0	Input	50
1	3-D CNN	32
2	Max-Pooling	32
3	3D-CNN	64
4	Max-Pooling	64
5	Fully Connected Layer	1024

Table 3.1: 2 layer 3-D CNN model

We evaluate the Semi-Supervised EM method using two very different CNN architectures for lung cancer screening. The first architecture is a relatively simple 2-layer 3D CNN as seen in Figure 3.2, and the other is a deeper 3D AlexNet architecture as seen in Figure 3.3. The architecture of our 3D 2-layer CNN in Figure 3.2 consists of 2 layers of CNN with a fully connected dense layer in the end. We have passed a sliding 3 × 3 window over each 3-D image for feature extraction. In the first layer of the CNN, the sliding window generates features and passes it to the max pooling layer which reduces the size of the feature maps before passing it to the next convolution layer. AlexNet comprises 11 layers which are a combination of convolutional, max pooling, and fully connected layers. The detailed description is illustrated in figure 3.3.

Two layer 3-D CNN

In this model the 3D volume data is first passed through the 3D-CNN layer with 1 channel where it detects 32 features and is passed to the next max-pooling layer. It computes the first 32 features using a window of size 3 × 3 × 3. In the max-pooling layer, it computes the highest pixel-values and creates a new image. In the next 3D-CNN layer it computes 64 features using 32 channels and is then passed on to the next max-pooling layer. Finally, it is passed through the fully connected layer which

computes 1024 features. The architectural diagram for this model can be seen in Figure 3.2 and output shape in Table 3.1.

AlexNet

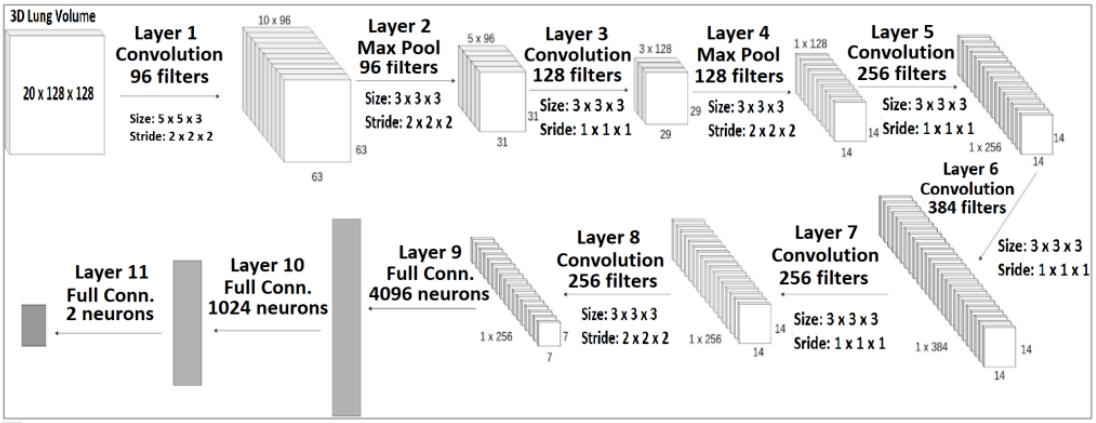

Figure 3.3: Architectural diagram of 3D AlexNet CNN.

In this model as shown in Figure 3.3, the 3-D Lung volume is passed through the first layer of CNN with 96 filters, size being 5 × 5 × 3 with a stride of 2 × 2 × 2 and setting and zero padding. It is then passed on to the next layer of maxpooling with the same 96 filters with the size as 3 × 3 × 3 and stride as 2 × 2 × 2. In the next layer of CNN the filters are increased to 128 keeping the same size but reducing the stride to 1 × 1 × 1. In the 4th layer during max-pooling the strides and the size is unchanged. The next layer of CNN has 256 filters with size 3×3×3 and stride 1×1×1 but with no padding. The 6th layer is a CNN with 384 filters with the same size and stride and zero padding is added. The 7th layer again is a CNN layer with 256 filters with size 3x3x3 and the stride as 2×2×2. The 9th,10th and 11th layers then consist of 4096, 1024 and 2 neurons respectively. These are fully connected layers.

3.5 Expectation Maximization

The EM algorithm is used to find the maximum-likelihood of a model in the presence of observed and latent variables. It is a technique that can be used in a semi-supervised approach to infer unknown labels while training. In our method, based on the labeled data-points, EM initially generates a classifier θ. The next step consists of performing an iterative procedure where EM uses θ to classify the data and then generate a new MAP hypothesis based on the labels inferred in the previous step. EM is an iterative method that attempts to determine the latent variable Z which in our case is a set of unknown image labels, such as to maximize the likelihood of observing the image X given a CNN model. The likelihood of a latent variable is given by the integral of the joint probability density over all possible values of the latent variable Z.

$$L(\theta; X) = p(X|\theta) = \int p(X, Z|\theta)\, dz \quad [3.1]$$

EM attempts to solve the above integral by alternating between Expectation and Maximization steps. Expectation is in which we calculate the expected value of the latent variables given the tth iteration of the model θ^t. In the context of a deep learning framework, the expected value of $E_{Z|X,\theta_t}$ can be computed by classifying label probabilities of the unlabeled imagery using the t[th] iteration of the model coefficients θ^t.

$$Q(\theta|\theta_t) = E_{Z|X,\theta_t}[logL(\theta;\ X, Z)] \quad [3.2]$$

The Maximization step is to compute the maximum likelihood model θ^{t+1} given our current expected value of the latent variables Z. This can be accomplished by retraining the deep learning model using the expected value of the image labels at the t[th] iteration.

$$\theta^{t+1} = argmax_\theta Q(\theta|\theta^t) \quad [3.3]$$

Our algorithm is similar to Co-EM [61] which is a semi-supervised algorithm which makes use of the hypothesis learned in one view to make use of probability to label the examples in the other data-set. It runs EM in every iteration and interchanges the labels that it has learned using probability. The major difference with this algorithm is that it does not set the labels in one iteration but based on the probability, it changes the labels and gets it close to the cluster it belongs to. There are several variants of EM for semi-supervised learning that evaluate this system in slightly different ways. In order to improve convergence, we decided to incorporate the unlabeled imagery gradually over multiple iterations of EM rather than all at once. When training on NLST and evaluating on Kaggle17, in the first iteration we predict the labels of the first 200 images in the Kaggle17 data-set. After prediction of the labels we concatenate these labels to the respective images and append it to the training dataset to pass it through the model again to predict the labels of the next 200 images in the next iteration. We repeat this process until the likelihood of the labels are maximized and the labels of all the images in the unknown data-set are predicted.

3.6 Deep Expectation Maximization Evaluation

We evaluate the accuracy of the Semi-Supervised EM methodology for cross-domain lung cancer screening. We compare our results to a fully supervised baseline as well as a fully supervised upper-bound. To perform this evaluation, we combined the NLST and Kaggle17 data-sets in several ways as described in Tables 3.2 and 3.3. In both Tables we wish to evaluate the ability to train the CNN model on one Lung

Cancer data-set (either Kaggle17 or NLST) and evaluate classification accuracy on the other data-set. This task is cross-domain in the weak sense that the Kaggle17 and NLST datasets are highly related but differences can affect classification accuracy. In Table 3.2 we train the CNN models using supervised data from the Kaggle17 data-set and evaluate the accuracy for cancer screening on the NLST data-set. Respectively, in Table 3.3 we train the CNN models using supervised data from NLST and evaluate using Kaggle17. The question we wish to answer is to what extent we are able to improve the accuracy of this cross-domain classification task by incorporating unlabeled data from the evaluation domain's training set using Semi-Supervised EM. As such, both Tables 3.2 and 3.3 have three columns. The first column shows a baseline fully supervised approach using only the labeled out-of-domain data-set for training. The second column shows a fully supervised upper-bound of using both in-domain and out-of-domain data for training. The third column "Semi-Supervised EM" shows the extent to which including unlabeled data from the evaluation domain using the proposed methodology is able to improve the classification performance. Table 3.2 shows the performance of supervised baseline, supervised upper-bound, and Semi-Supervised EM to classify NLST imagery using supervised imagery from Kaggle17. We divide the training, validation and test data into 80, 10 and 10 percent respectively. The 2 layer 3-D CNN model gives us baseline accuracy of 75.95% which improves to 77.5% with the addition of unlabeled imagery from NLST. Similarly the AlexNet gives us a baseline accuracy of 79.36% which improves to 81.1% using Semi-Supervised EM. We see that for both CNN architectures, the improvement of incorporating unlabeled imagery from NLST with Semi-Supervised

EM is roughly half that of an upper-bound using fully labeled imagery from both data-sets combined. Table 3.3 shows a similar cross-domain evaluation of supervised baseline, supervised upper-bound, and Semi-Supervised EM but this time to classify Kaggle17 imagery using labeled imagery from NLST. In this process, we have trained the data-set on 80% of the NLST data-set, validated on 10% of the NLST data-set and tested the accuracy of the model on 10% of the Kaggle17 data-set. The 2 layer 3-D CNN model gives us baseline accuracy of 76.75% which improves to 78.1% with the addition of unlabeled imagery from NLST. The AlexNet, however shows an improvement in this case with a labeled baseline accuracy of 72.9% which improves to 74.4% using Semi-Supervised EM. The NLST data-set is much larger than the Kaggle17 data-set with 4075 and 1375 CT scans respectively. The larger improvement of the 2-layer 3D CNN relative to the AlexNet can be explained because in Table 3.3 the unlabeled imagery is a smaller fraction of the overall data volume relative to the experiment in Table 3.2. We see in both of our cross-domain lung cancer screening experiments that the Semi-Supervised EM (column 3) is able to improve the classification accuracy over a baseline supervised algorithm using only the out-of-domain imagery and labels. Also, as expected we also see that the classification accuracy is less than the upper-bound of incorporated fully supervised labels from both data-sets. Furthermore, we see that this improvement is more pronounced in the experiment of Table 3.2, in which we use labeled imagery from the smaller Kaggle17 data-set and incorporate unlabeled imagery from the larger NLST data-set.

Train - Test	*(Supervised) Kaggle17 only*	*(Supervised) Kaggle17 + NLST*	**Semi-Supervised EM**
(2 layer 3D-CNN) NLST	75.95%	80.37%	**77.5%**
(AlexNet) NLST	79.36%	83%	**81.1%**

Table 3.2: Train:Kaggle17 Test:NLST

Train - Test	*(Supervised) NLST only*	*(Supervised) NLST + Kaggle17*	**Semi-Supervised EM**
(2 layer 3D-CNN) Kaggle17	76.75%	81%	**78.1%**
(AlexNet) Kaggle17	72.9%	77.7%	**74.4%**

Table 3.3: Train:NLST Test:Kaggle17

3.7 Findings

We have demonstrated that Semi-Supervised EM, when applied to computer-aided lung cancer screening with CNN models, is able to increase accuracy of cross-domain classification by incorporating unlabeled imagery. Semi-Supervised EM is a technique to infer the maximum likelihood CNN coefficients in the presence of labeled and unlabeled imagery. This technique can therefore improve and increase the availability of training data which is often a limiting factor for cancer screening applications. Our findings show that using labeled imagery with the smaller Kaggle17 data-set and incorporating unlabeled imagery from the larger NLST dataset provides

roughly half of the accuracy benefit as incorporating fully labeled imagery from NLST. We believe that semi-supervised learning could have a major impact on the performance of Deep CAD algorithms which are an area of active research. These results help us to indicate that Semi-Supervised EM is an appropriate methodology for these purposes, and is compatible with the genre of CNN architectures in active research and development for oncology detection and diagnosis applications.CNN architectures in active research and development for oncology detection and diagnosis applications.

Chapter 4: Active Semi-Supervised Expectation Maximization Learning for Lung cancer Detection from Computerized Tomography (CT) images with Minimally Labeled Training Data

Artificial intelligence (AI) has great potential in medical imaging to augment the clinician as a virtual Radiology Assistant (vRA) through enriching information and providing clinical decision support. Deep learning is a type of AI that has shown promise in performance for Computer Aided Diagnosis (CAD) tasks. A current barrier to implementing deep learning for clinical CAD tasks in radiology is that it requires a training set to be representative and as large as possible in order to generalize appropriately and achieve high accuracy predictions. There is a lack of available, reliable, discretized and annotated labels for computer vision research in radiology despite the abundance of diagnostic imaging examinations performed in routine clinical practice. Furthermore, the process to create reliable labels is tedious, time consuming and requires expertise in clinical radiology. We present an Active Semi-supervised Expectation Maximization (ASEM) learning model for training a Convolutional Neural Network (CNN) for lung cancer screening using Computed Tomography (CT) imaging examinations. Our learning model is novel since it combines Semi-supervised learning via the Expectation-Maximization (EM) algorithm with Active learning via Bayesian experimental design for use with 3D CNNs for lung cancer screening. ASEM simultaneously infers image labels as a latent variable, while predicting which images, if additionally labeled, are likely to improve classification accuracy. The performance of this model has been evaluated

using three publicly available chest CT datasets: Kaggle2017, NLST, and LIDC-IDRI. Our experiments showed that ASEM-CAD can identify suspicious lung nodules and detect lung cancer cases with an accuracy of 92% (Kaggle17), 93% (NLST), and 73% (LIDC) and Area Under Curve (AUC) of 0.94 (Kaggle), 0.88 (NLST), and 0.81 (LIDC). These performance numbers are comparable to fully supervised training, but use only slightly more than 50% of the training data labels.

Keywords: Lung Cancer screening, Active Learning, Semi-Supervised Learning, CT, Label Acquiring, Computer-Aided Diagnosis, Expectation Maximization, Artificial Intelligence, Deep Learning

4.1 Introduction

Deep learning using Convolutional Neural Networks (CNNs) has greatly improved the performance of Computer Aided Diagnosis (CAD) algorithms for cancer screening in recent years [54 , 32 , 31 , 67 , 57 , 70 , 71]. However, a disadvantage of many deep learning classification techniques including CNNs is that these algorithms are fully supervised and therefore require very large datasets with manual annotation by expert radiologists in order to achieve high accuracy. Typically these fully annotated datasets are on the order of thousands of images, whereas clinical Picture Archiving and Communication Systems (PACS) even at a community hospital contains millions of unlabeled or weakly labeled Radiology examinations. As such, a major challenge in applying deep learning based CAD clinically is to be able to make use of larger unlabeled radiology imaging datasets and to combine these datasets with smaller highly annotated datasets.

Methods to reduce the amount of manual annotation necessary while maintaining or improving accuracy are an important contribution because manual annotation for medical imagery is time consuming, costly, and requires expert labelers with a high level of expertise in radiology.

Accurate Chest CT annotation for lung cancer screening requires a board certified diagnostic radiologist (4 years of medical school, 1 year of internship, and 4 years of diagnostic radiology residency), ideally with additional experience or subspecialization in Thoracic Radiology or Oncologic Radiology (1-2 years additional fellowship or clinical experience). Furthermore, challenging tasks such as nodule segmentation and malignancy assessment require additional annotation that is beyond the routine clinical standard of care and therefore is not readily available. The misclassification rate for CNNs has been empirically estimated to decay exponentially as data volumes increase [72]. Therefore, the absence of large annotated datasets are currently a limiting factor in the clinical application of deep learning for radiology. Active learning and Semi-supervised learning algorithms have the potential to enable deep learning based CAD to further improve performance by making use of large clinical image volumes collected by institutions thereby greatly reducing the necessary labeling burden.

In this study we investigate a novel learning model that combines Active learning with Semi-supervised learning in order to reduce the amount of annotation necessary to create a CNN based CAD algorithm for Chest CT cancer screening examinations. Lung cancer screening was recently identified as contributing to the largest year-over-year decline in cancer deaths ever recorded. [73] We demonstrate that an

Active Semi-Supervised Expectation Maximization (ASEM) algorithm is a viable approach for training deep CNN based CAD algorithms using CT exams. This work builds upon our recent work demonstrating that a Semi-supervised EM (SEM) algorithm was able to improve cross-validated Lung Cancer screening accuracy as compared to a fully-supervised technique [74]. We expand on this algorithm by incorporating Active learning in combination with Expectation-Maximization (EM) in order to further improve cross-validated screening accuracy. The active learning component allows the algorithm to interactively suggest images to radiologists that need to be labeled, and the semi-supervised learning using EM allows the algorithm to incorporate a larger unlabeled training image dataset along with a smaller labeled dataset. The suggested images from the large unlabeled pool are selected by a validating classification uncertainty method.

4.2 Related Work

Recently, [71] presented an artificial intelligence (AI) system which can potentially outperform human experts in breast cancer prediction. To evaluate its effectiveness in the clinical setting, they have curated a large characteristic dataset from the UK and large enhanced dataset from the USA and have manifested complete reduction of 5.7% (USA) and 1.2% (UK) in false positive rate and 9.4% (USA) and 2.7% (UK) in false negative rate. Expectation Maximization (EM) is an influential generative meta-algorithm for latent variable training and has been employed for semi-supervised learning [58]. Generative algorithms model the probability distribution of unlabeled imagery as a function of model and labeled imagery [62 , 63]. Although EM assumes an underlying generative model, recent work in

combining EM with discriminative CNN architectures have been shown to be successful in practice, likely due to the non-linearity of CNNs. EM is applied to improve semantic segmentation of general imagery using CNNs [19]. The method achieves 73.9% accuracy with a small number of pixel-level annotated images which is almost competitive with the fully-supervised model's accuracy of 79%.

Active learning or "optimal experimental design" in statistics is part of the machine learning field, where the learner selectively asks (or queries) experts for more ground truth labels in order to achieve its desirable outcome (e.g model's accuracy or better learning with less samples). As such, Active learning methods choose the most informative unlabeled samples for annotation by a human radiologist. The selection process requires the learning algorithm to provide a query strategy to select unlabeled data points that are most likely to improve the model accuracy if labeled. By using uncertainty sampling [75], the way to select is by picking the least certain label and requiring experts to annotate. Recently, active learning has been to overcome data scarcity issues with current models by incrementally choosing the most revealing unlabeled samples, querying their labels and putting them to the labeled data set [76].

The Monte Carlo dropout method is used to estimate the level of uncertainty in the active learning process or look ahead technique to select samples [77]. Better uncertainty estimation is obtained using ensemble models [78]. Previous work combined the approach proposed by [77] and data augmentation (generate new training samples from a latent variable, discriminate between real and fake samples) for classification learning tasks [79].

Active learning alone has been applied for characterization of endothelial cells in human tumors [80] and predicting positive p53 cancer rescue regions by using the most informative information method [81]. There is a recent active learning framework that is presented by [82] for skin lesion analysis which is cost-effective by selecting and employing much fewer labeled samples while the network still attains state-of-the-art performance. Their active learning method tends to enhance the annotation coherence. The authors have selected their samples to be highly supportive and have used dataset of the ISIC 2017 Skin lesion Classification challenge and attained state-of-the-art performance by using 50% of the data for the first task and 40% of the data for the second task of skin lesion classification. Previous work in [83] developed a model that recognizes anomalies within plain-text-based reports which could then be utilized further as a method to create labels for models depending on CT scans thereby aiming to decrease human efforts in labeling CT scans. A systematic approach named NoduleX was proposed by [84] which uses a deep learning CNN as well as a radiomics approach for prediction of lung nodule malignancy using CT images of the LIDC dataset.

4.3 Methodology

The ASEM method combines both Semi-supervised and Active learning to improve the accuracy of the model prediction with as few known labels as possible. Active learning techniques require an Oracle step in which the algorithm asks for more

ground truth from the unlabeled data during the ASEM process. Figure 4.1 shows an overview of our proposed learning model. We first train an initial model using a subset of the training data which is fully labeled. Subsequently, the ASEM model alternates between Semi-supervised Expectation and Maximization steps as well as Active learning Oracle, and Active Retraining steps. Each ASEM iteration requires retraining the model in the Maximization and Active Retraining steps with improved estimates of the latent variables either due to Expectation, or due to the Oracle. The computational burden of retraining the model for each ASEM iteration however is greatly reduced by re-using the weights from the previous ASEM iteration rather than retraining from a random seed (see table 4.4 for the runtime performance of our ASEM-CAD).

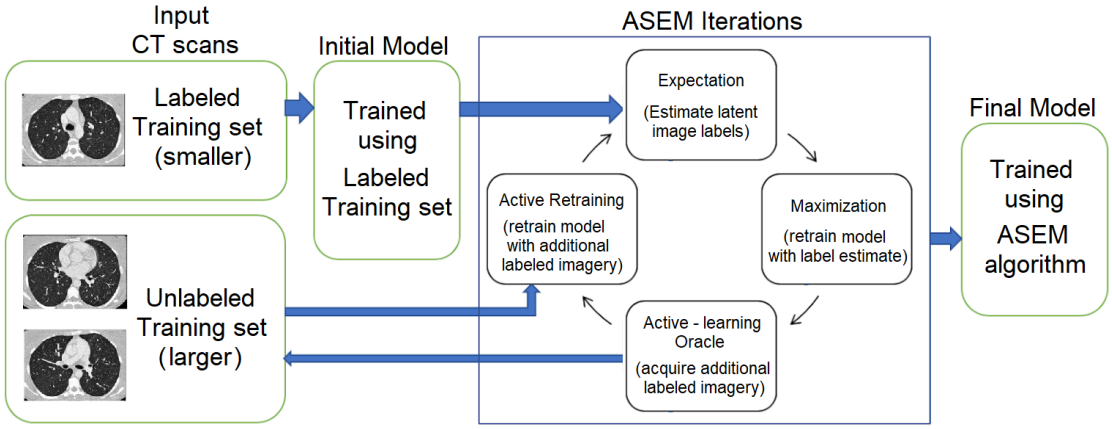

Figure 4.1 Overview of proposed model for lung cancer detection.

The ASEM algorithm is an alternating local maximization-maximization algorithm in which we attempt to perform a maximum likelihood estimate of our model parameters θ and expected experimental design ξ in the presence of latent variables Z and given the ability to actively label a finite number of observations y. Our goal is to show that the maximum likelihood of the model is improving after each Active Learning step and after each EM step.

A theoretical detail is that EM steps attempt to maximize likelihood whereas Active learning minimizes cross entropy. These have equivalent global optima under the assumption of statistical independence and have approximately equivalent optima in practical machine learning applications as follows,

$$-\sum_i p(X_i)\, log(\, p(X_i \mid \theta)) \;\approx\; -\, log(p(X|\theta)) \qquad [4.1]$$

The Expectation and Maximization steps maximize likelihood of the model under all possible values of the latent variable space [58]. The likelihood of a latent variable is given by the integral of the joint probability density over all possible values of the latent variable Z.

$$L(\theta; X) \;=\; p(X|\theta) \;= \int p(X,\, Z|\theta) dZ \qquad [4.2]$$

EM attempts to solve the above integral by alternating between Expectation and Maximization steps. Expectation is in which we calculate the expected value of the latent variables given the tth iteration of the model θ^{t}. In the context of a deep

learning framework, the expected value of $E_{Z|X,\Theta_t}$ can be computed by classifying label probabilities of the unlabeled imagery using the tth iteration of the model coefficients θ^t.

$$Q(\theta|\theta^t) = E_{Z|X,\theta^T}[log\ L(\theta; X, Z)] \quad [4.3]$$

The Maximization step is to compute the maximum likelihood model parameters θ^{t+1} given our current expected value of the latent variables Z. This can be accomplished by retraining the deep learning model using the expected value of the image labels at the tth iteration.

$$\theta^{t+1} = argmax_\theta Q(\theta|\theta^t) \quad [4.4]$$

Our active learning process is designed to select data points for an expert human to label during the EM iteration training. In this way we make selective incremental improvements to data label quality. The active learning steps optimize the expected posterior cross entropy of the model given an alternate experimental design ξ with the addition of a labeled sample y_i. We cannot measure the posterior cross entropy directly because we must select a sample before acquiring it's true label via Oracle. As such the expected posterior cross entropy is as follows:

$$U(\xi) = -\int log(p(X|\theta, y_i, \xi))\, dy_i \quad [4.5]$$

This quantity can be rewritten using Bayes rule for bayesian experimental design as follows,

$$U(\xi) = -\int log(p(y_i|\theta, X, \xi)\frac{p(y_i|\theta,\xi)}{p(X|\theta,\xi)} dy_i \quad [4.6]$$

This integral would be expensive to compute as it would require retraining the algorithm for every possible sample choice and every possible sample label prior to choosing the appropriate sample.

However, we can make an approximation that a single sample does not change the model prediction of most samples more than a small amount at a time, but rather the predicted sample inself y_i has the greatest local contribution to posterior cross entropy.

Under this assumption the change in posterior cross entropy is approximately equal the the normalized classification entropy over all possible K labels as follows,

$$\Delta U(\xi) \approx I_{norm}(y_i) = \frac{-1}{log(K)} \sum_{k=1}^{K} p(y_{ik})\ log(p(y_{ik})) \quad [4.7]$$

We can perform a small number of Active Learning steps in a batch rather than performing a single active learning step. In this case, our expected utility becomes

the normalized classification entropy of all of the selected samples in the batch as follows:

$$avg(I_{norm}(Y)) = \frac{1}{|Y|} \sum_{y \in Y} I_{norm}(y)) \quad [4.8]$$

As such the ASEM algorithm alternates between steps 3, 4, and 8 in order to improve the maximum likelihood estimate and reduce classification cross entropy in the presence of latent variables while optimizing Bayesian experimental design. An important point is that algorithm, as a maximization-maximization meta-algorithm does not guarantee convergence to a global optimum, but rather will achieve a local optimal experimental design as well as a local optimal estimate of latent variables for semi-supervised learning.

4.4 Data and Experimental Design

We analyze the performance of the active semi-supervised EM (ASEM) model using 3 lung cancer screening datasets: Kaggle, NLST, and LIDC-IDRI. The Kaggle Data Science bowl (2017) (Kaggle17) is a benchmark dataset for Computer Aided Diagnosis (CAD) algorithms for Non-Small Cell Lung Cancer (NSCLC) cancer screening using Low Dose Computed Tomography (LDCT) scans. Each volumetric scan contains a varying number of Chest CT image slices, and each slice is the standard resolution of 512x512 pixels. We experiment with the Kaggle17 dataset, which consists of a total of 1375 patients. These scans are labeled as 1 for cancer (i.e.

diagnosed with lung cancer within one year of the scan) and 0 for non-cancerous. NLST was a landmark 2011 study that proved that high risk individuals (60+ yrs old, and heavy smokers) who receive periodic LDCT lung cancer screening exams have greater life expectancy and lower mortality than if these individuals were to receive periodic chest x-ray screenings. We used 4075 LDCT scans from the NLST dataset, and each scan was labeled as 1 if the patient was diagnosed with cancer or 0 if the patient was not diagnosed with cancer. Of these 4075 scans, 639 patients were diagnosed with lung cancer.

The LIDC-IDRI Dataset [85, 86] is a publicly available dataset that consists of diagnostic and lung cancer screening thoracic computed tomography (CT) scans with marked-up annotated lesions. This dataset is a web-accessible international resource initiated by the National Cancer Institute (NCI), and then further developed by the Foundation for the National Institutes of Health (FNIH) and going along with the Food and Drug Administration (FDA). LIDC is used for the purpose of research towards development, training and assessing of computer-aided diagnostic (CAD) methods for detecting and diagnosing lung cancer in its early stages. This dataset is created in collaboration with seven academic centers and eight medical imaging companies that have 1018 CT cases where it has thoracic CT scans associated with an XML file. The LIDC study has annotations that are provided by four experienced

Thoracic Radiologists who reviewed each of the 1018 CT cases in the LIDC/IDRI cohort and marked lesions into 3 categories based on the nodule size. Nodules > or = 3mm have a greater probability of being malignant than nodules <3mm and

non-nodule > or = 3mm. The malignancy rating is given from 1-5 depending on the size and features of the nodule.

The following section presents how we build and evaluate our ASEM model for Computer Aided Diagnosis, called ASEM-CAD for lung cancer detection using the above 3 datasets.

Data Pre-processing

In Both Kaggle and NLST datasets, each patient CT scans have varied slice numbers. For each patient, we create standard 3D volume data as input for the model by resizing the 512x512 image pixels of multiple DICOM slices into a standard 50x50x20 voxel resolution. The third dimension is reduced to 20 by chunking slices into 20 chunks then average. Thus, the input 3D volume for each patient has 50x50x20 dimensions, each associated with a label: either 1 (cancer) or 0 (non-cancer). Kaggle has 1357 patients with 356 cancer cases. Cancer cases represent 26% of the Kaggle dataset and non-cancer cases represent 74%. For NLST, after preprocessing we have 2538 cases, 397 cancers and 2141 non-cancer.

The third dataset, LIDC, has 1010 CT scans, with each slice having 512x512 pixels. We crop 4253 nodules which cover the size of nodules according to annotations that are provided by the four experienced radiologists. Their annotations are given in the form of nodule Region Of Interest and their Z-positions. Thus we crop the the 32x32x16 dimension nodules using the spatial coordinate centered at the annotated location of the CT scans. For assigning the labels of the nodule, we use rating scores

provided by board certified radiologists with levels 1 and 2 as a non-cancer nodule (benign), level 4 and 5 as cancer nodule (malignant). The score level 1 meaning highly benign, 2 as moderately benign (non-cancer), 4 as moderately suspicious to be malignant and 5 as highly likely to be malignant (cancer). Nodules labeled by a radiologist as having an intermediate malignancy (rating 3) are not considered for classification in this paper. In summary, we have 4253 nodules, each has 32x32x16 dimensions and an associated label 1 (cancer) and 0 (non-cancer). There are 1653 cancerous nodules. 2600 nodules belong to benign cases.

Neural Architecture and Training Procedures

The ASEM-CAD neural architecture has six CNN layer blocks. Each has Convolutional 3D layers, LeakyRelu, BatchNormalization, MaxPooling3D, DropOut with 32, 32, 64, 64, 64, 64 feature maps. The Convolutional 3D uses a 3x3x3 filter. Then followed by Dense, BatchNormalization, DropOut layer with 256 features. The last layer has Dense 1024 and 2 classes. For the LIDC dataset, we use a simpler CNN architecture with layer block 1, 2, 3, 4, 7 and 8. The CNN feature maps are 8, 8, 16, 16 for LIDC experiments. For all experiments, data is split as 80% of the input dataset is used for ASEM to train and evaluate the model. The remaining 20% of the dataset is used for testing. The ASEM training procedure is as follows: the initial model is fully trained with 50% of all labels until convergence using category Cross-entropy loss. The ASEM-CAD is trained using RMSprop optimizer with a learningRate of 0.0001. The Initial model is fully trained using 500 epochs. Then

each ASEM meta-iteration (EM iteration) is trained with 10 epochs. The ASEM model is trained in batches of 32 samples. The Initial model is saved.

Then EM iterations can load the initial model's parameters and start the Active EM training. The Active component selects 10 samples and asks an Oracle for the label during an ASEM iteration. The number of ASEM' Active EM iterations is set to 5. In addition, we apply Label Smoothing, BatchNormalization, and Early Termination techniques for training our ASEM-CAD.

4.5 ASEM Results

In this section, we present our experimental findings of the performance of the ASEM algorithm in comparison to fully supervised learning as well as in comparison to the SEM algorithm. We calculate Receiver Operating Curves (ROC), and present accuracy, Area Under Curve (AUC), sensitivity, specificity, and precision as evaluation metrics. We evaluate the ASEM-CAD using the Kaggle17, NLST, and LIDC datasets and compare it with fully supervised training as well as Semi-Supervised EM. We compare the following methods,

Supervised 1: Using only 50% of these labeled datasets;

Supervised 2: Using 100% labeled dataset;

SEM-CAD: Start with 50% labeled data initially then using full dataset for EM iterations.

ASEM-CAD 1: Active semi-supervised with 50% labels and additional labels

with Max. Classification Entropy;

ASEM-CAD 2: Active semi-supervised with 50% labels, add additional labels with above Avg Classification Entropy.

Most lung cancer datasets including Kaggle17, NLST, and LIDC have an unbalanced number of cancer vs non cancer cases with a greater number of non-cancerous cases relative to cancerous cases. Yet in clinical practice, it is necessary to bias the final threshold of any cancer screening test such as to over-predict false positives in order to reduce the probability of predicting false negatives. In order to provide a more complete picture, ROCs are calculated by varying the prediction threshold between 0 to 1 and plotting sensitivity against 1 - specificity if all predictions above the varied threshold are classified as cancerous. AUC varies from 0 to 1 (higher is better), and is defined as the integral of sensitivity with respect to 1-specificity over the domain of the ROC curve. Tables 4.1, 4.2, and 4.3 calculate an inflection point along this ROC curve and present sensitivity, specificity, and precision. Table 4.1 shows the performance of the ASEM-CAD algorithm over the Kaggle17 dataset, Table 4.2 shows the performance over the NLST dataset, and Table 4.3 shows the performance over the LIDC-IDRI dataset.

Table 4.1. ASEM performance over the Kaggle17 dataset

Experiments	Number of Samples	Test_ACC	AUC	Sensitivity	Specificity	Precision
Supervised 1	50% labels only	0.87	0.85	0.69	0.94	0.79
Supervised 2	100% labels	**0.92**	**0.92**	**0.81**	**0.95**	**0.85**
SEM-CAD	50% label initially	0.91	0.92	0.88	0.92	0.76
ASEM-CAD1	50%, add labels with Max. Classification Entropy	**0.92**	**0.94**	**0.78**	**0.96**	**0.88**
ASEM-CAD2	50%, add labels with above Avg Classification Entropy	0.85	0.81	0.66	0.9	0.67

Over the Kaggle17 dataset, ASEM-CAD1 outperformed Supervised 2 algorithms in AUC (0.94 vs 0.92), this is notable because Supervised 2 has the benefit of using 100% of the training labels.

At the inflection point, Specificity and Precision were higher although Sensitivity was slightly lower as compared to fully-supervised learning model using 100% labels (table 4.1). It also showed in table 4.1 that ASEM-CAD1 outperforms the SEM-CAD in similar metrics.Noticeably, ASEM-CAD1 performed much better by 7.9%, 13%, 15%, 6%, and 23% in all metrics respectively in its order in table 4.1 over our ASEM-CAD2. ROC curves comparing Supervised 2, SEM-CAD, and ASEM-CAD1 are shown in Figure 4.2.

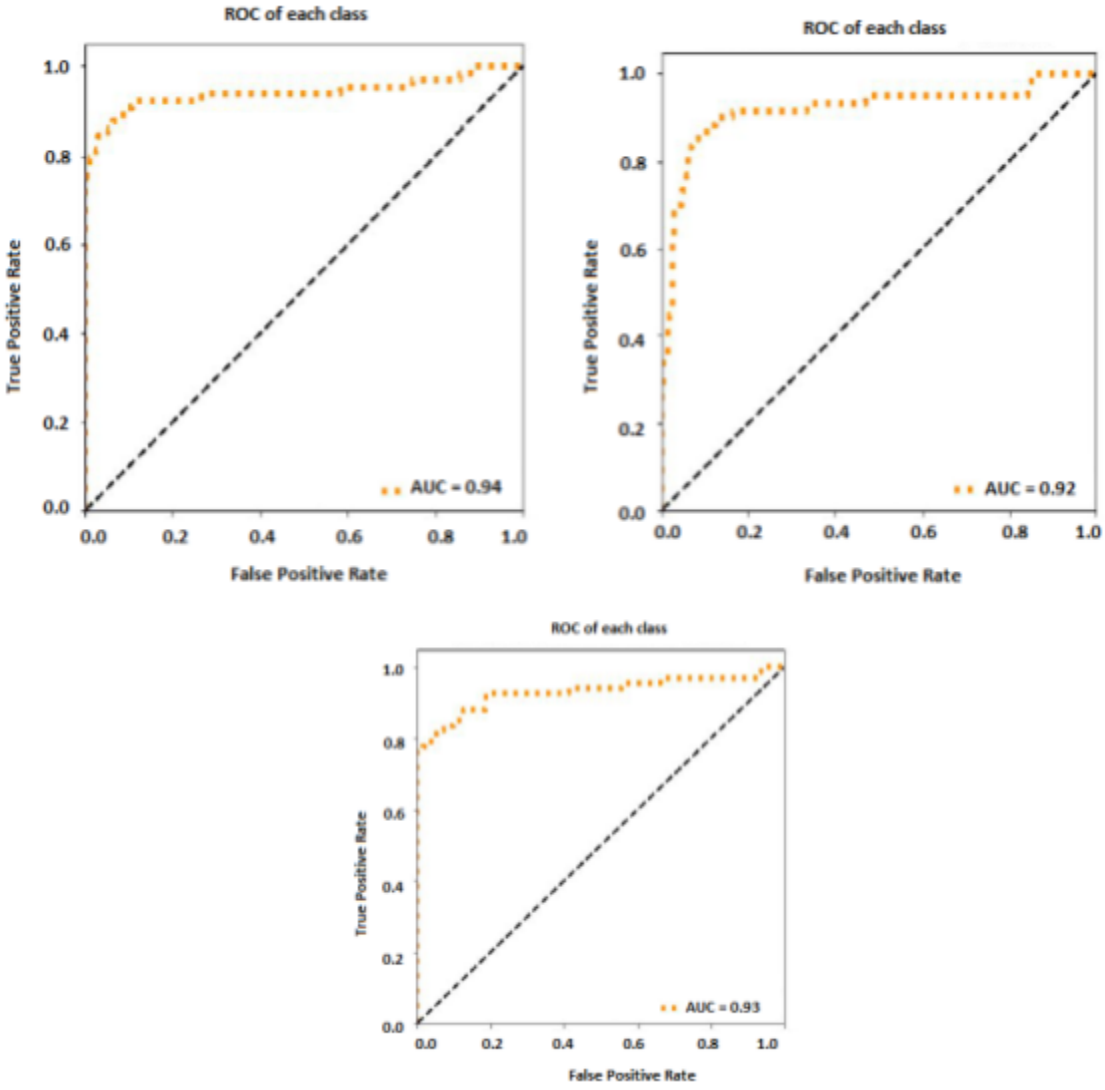

Figure 4.2 ROC analysis of the Kaggle dataset **a) left Supervised 2** (100% labels) **b) middle semi-supervised SEM-CAD** (50% labels). **c) right our ASEM-CAD1**, Active Semi-Supervised (50% labels, add labels with Max. Classification Entropy). Note: this ROC curve is reported per run not on average of multiple runs presented in table 4.1.

Table 4.2 shows a similar ROC analysis of ASEM-CAD1 against supervised and semi-supervised techniques. We see that the ASEM-CAD1 algorithm achieves AUC of 0.88 which is comparable and slightly greater than Supervised 2 which achieves AUC of 0.87. Supervised 2 has the benefit of using all of the labeled data, whereas ASEM-CAD1 uses only slightly more than half of the labeled data. Figure 4.3 shows the ROC curves comparing ASEM-CAD2 versus Supervised 2, we see that ASEM-CAD2 exhibits comparable performance characteristics to Supervised 2 in addition to achieving similar AUC.

Table 4.2. ASEM paerformance over the NLST dataset.

Experiments	Number of Samples	Test_ACC	AUC	Sensitivity	Specificity	Precision
Supervised 1	50% labels only	0.92	0.87	0.65	0.97	0.75
Supervised 2	100% labels	**0.94**	**0.87**	**0.72**	**0.97**	**0.90**
SEM-CAD	50% label initially	0.90	0.89	0.77	0.92	0.67
ASEM-CAD1	50%, add labels with Max. classification entropy	**0.93**	**0.88**	**0.56**	**0.99**	**0.94**
ASEM-CAD2	50%, add labels with above Avg classification entropy	0.92	0.86	0.63	0.99	0.91

Table 4.3 compares the performance of ASEM versus supervised and semi-supervised methods for nodule malignancy estimation using the LIDC-IDRI dataset.

We see that ASEM-CAD1 achieves AUC of 0.81 which is very comparable performance to Supervised 2 (AUC 0.82), by using only slightly more than 50% of the data labels, as opposed to 100% of the data labels. For this dataset ASEM-CAD1 and ASEM-CAD2 achieved comparable AUC performance and these algorithms

outperformed SEM-CAD. At the inflection point ASEM-CAD1 achieves slightly greater sensitivity but slightly lower specificity than Supervised 2. We compare the ROC curves for Supervised 2 vs ASEM-CAD1 in Figure 4.4, and we find that these curves have similar accuracy performance characteristics.

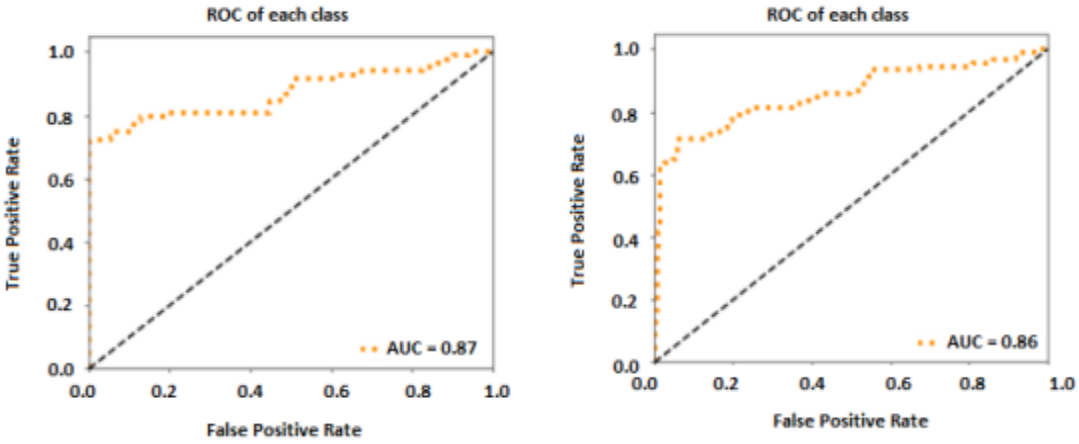

Figure 4.3 ROC analysis of the NLST dataset a) left fully-supervised(100% labels). b) Right ASEM-CAD2 Active Semi-supervised (50% labels), add labels with above Average Classification Entropy.

Table 4.3. ASEM performance over the LIDC-IDRI dataset.

Experiments	Number of Samples	Test_ACC	AUC	Sensitivity	Specificity	Precision
Supervised 1	50% labels only	0.67	0.82	0.91	0.52	0.54
Supervised 2	100% labels	**0.74**	**0.82**	**0.78**	**0.72**	**0.64**
SEM-CAD	50% label initially	0.71	0.81	0.82	0.64	0.59
ASEM-CAD1	50%, add labels with Max. classification entropy	**0.73**	**0.81**	**0.79**	**0.70**	**0.62**
ASEM-CAD2	50%, add labels with above avg classification entropy	0.73	0.80	0.82	0.67	0.60

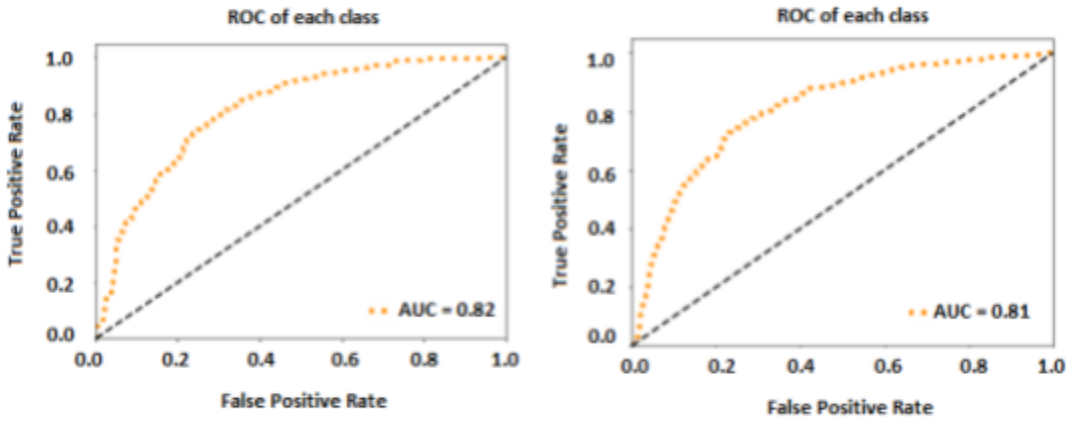

Figure 4.4 ROC analysis of the LIDC dataset a) left fully-supervised (100% labels)-Supervised2. b) Right Active Semi-supervised (50% labels, add label with Maximization Classification Entropy) ASEM-CAD1.

Also notable is that the ASEM algorithm, despite iteratively retraining of the CNN models more than 10 times, adds less than 50% additional overhead to the overall training time. Table 4.4 shows the wall-time runtimes of the ASEM-CAD algorithm for training using a customized computer with AMD 1885 MHz 32 cores, 658 GB, 3 NVIDIA GeForce RTX, each GPU has 11 GB memory. The reason that the runtime is manageable (30%-50% increase) as opposed to a factor 10x or more is because we save and reuse the CNN weights after each iteration as opposed to retraining the CNN from random weights. As a maximization-maximization procedure, each ASEM iteration can be thought of as a local hillclimb in order to further improve the maximum likelihood estimate of the model parameters. Thus, the weights from the previous ASEM step a good initial guess to the weights of the subsequent ASEM step, thereby reducing the number of epochs necessary (and thus total walltime) for the ASEM Iterations.

Table 4.4 Training wall time of the ASEM-CAD algorithm

Dataset	Number of Images	Total runtime in Minutes			
		Initial Model	ASEM Iterations	Total Time	Percent Increase
Kaggle	1357	26	8	34	31 %
NLST	2538	47	15	62	32 %
LIDC	4253	8	4	12	50 %

4.6 Learnings

ASEM-CAD is a new CNN based CAD model which combines both semi-supervised and active learning to detect lung cancerous nodules and lung cancer cases using CT scans, while reducing the number of labeled scans necessary to train the neural architecture. ASEM-CAD has been evaluated using three public chest CT datasets for lung-cancer screening: Kaggle17, NLST, LIDC. Our experiments showed ASEM-CAD can detect lung cancer with high AUC performance comparable to that of fully supervised learning, but with only slightly more than 50% of the training labels. The ASEM-CAD1 vs Supervised2 AUC performances were: NLST (0.88 vs 0.87), Kaggle17 (0.94 vs 0.92), and LIDC-IDRI: (0.81 vs 0.82). The Active learning component asks for additional ground truth of unlabeled data which has a high level of classification uncertainty (high entropy) during the EM training process. This

selection process results in better performance as compared to purely Semi-Supervised learning as well (SEM-CAD).

In conclusion, we have demonstrated that ASEM-CAD is able to detect suspicious lung nodules with comparable accuracy as using a fully supervised algorithm but with far fewer labeled images. ASEM-CAD may help to provide medical imaging researchers and commercial vendors with a more practical approach to train more powerful artificial intelligence based virtual radiology assistants (vRA) to augment radiologists interpreting oncologic imaging in the setting of lung cancer screening and perhaps other diagnostic radiology examinations more generally. In the future, we expect that Semi-Supervised and Active learning will play an increasingly larger role in the development of Deep CAD algorithms as these techniques will make it possible to learn from large clinical PACS datasets while reducing the need for manual annotation by radiologists.

Chapter 5: Semi-supervised Contrastive Outlier - removal for Pseudo - Expectation - Maximization (SCOPE)

Semi-supervised learning is the problem of training an accurate predictive model by combining a small labeled dataset with a presumably much larger unlabeled dataset. Many methods for semi-supervised deep learning have been developed, including pseudolabeling methods, consistency regularization, and contrastive learning methods. Pseudolabeling methods however are highly susceptible to *confounding*, in which pseudolabels are erroneously assumed to be true labels, thereby causing the model to reinforce its prior biases over successive iterations and thereby deviate substantially from truth. We present a new approach to suppress confounding errors through a method we describe as contrastive outlier removal. This method can be derived from Expectation Maximization (EM), a latent variable framework which can be extended toward understanding cluster-assumption deep semi-supervised algorithms.

5.1 Introduction

Learning high quality model representations from limited labeled data is a problem that deep learning has not yet overcome. While humans can achieve this task seamlessly, contemporary deep learning algorithms still struggle to do so. These deep learning algorithms need to be trained with large labeled data volumes in order to perform well. While great strides have been made using transfer learning, which borrows labeled samples from another domain, semi-supervised deep learning has not advanced to the level of maturity necessary to gain widespread adoption. The applications for semi-supervised learning are numerous because in many domains, unlabeled data is plentiful yet high quality labeled data is scarce. Data labeling

remains a task that is time consuming, expensive and error prone. Semi-supervised learning becomes very challenging when the labeled data volumes are very small relative to the unlabeled volumes. In this case, many methods, especially proxy-label bootstrapping techniques are susceptible to a kind of confounding bias, in which the model at first learns some error due to the small labeled sample, and then proceeds to pseudo label the data incorrectly thereby reinforcing it's prior error. If the initial labeled sample is sufficiently large the chances of this problem happening is much less. But as we decrease the labeled sample size, and increase the unlabeled sample size, this confounding bias becomes more and more of a concern. From a standpoint of practical use-cases, it is desirable to discover methods capable of learning with as little labeled data as possible, but this scenario is exactly the situation in which semi-supervised learning is most difficult to perform.

Problem Definition of Semi-Supervised Learning

Semi Supervised learning can be defined as the problem of learning an accurate predictive model using a training dataset with very few labeled samples but a much larger number of unlabeled samples. Let us say that we have a set of training samples X and training labels Y , the samples can be further defined as a set of mutually exclusive unlabeled samples X_U and labeled samples X_L along with supervised labels Y_L and unobservable (latent) unsupervised labels Y_L, where $X = X_L \cup X_U$ and $Y = Y_L \cup Y_U$. In practice, the number of unlabeled

samples is also typically much larger than the number of labeled samples $|X_L| << |X_U|$.

The accuracy of semi-supervised learning is typically evaluated using the multi-class classification task via cross validated benchmark on a withheld test set X_T, Y_T. It is further assumed that the training and test sets follow the same distribution of samples and labels (if different distributions are assumed the problem becomes unsupervised domain adaptation). The goal of semi-supervised learning is to minimize the expected testing loss as follows.

$$min_{\theta} E\ (\ L(Y_T - \widehat{Y}_T)\ ;\ X_L,\ X_U,\ Y_L) \qquad [5.1]$$

Although we do not have access to the unobserved labels Y_U one can attempt to predict these labels using a technique called pseudolabeling. We define the pseudolabels as $\widehat{Y}_U^{\ t}$ which are the predicted labels of the unlabeled set at the t^{th} Expectation Maximization (EM) iteration. If we define our predictive model as $F\ (X, \theta^t)$, where the algebraic form of expression F represents the network architecture, and parameters θ^t represents the model parameters at EM training iteration *t* then, the predicted pseudolabels can be defined as follows,

$$\widehat{Y}_U^{\ t} = F(X_U, \theta^t) \qquad [5.2]$$

Self-training had first attempted to solve this issue [25,26] using proxy-label techniques that trained a classifier with a small amount of labeled sample and labeled a subset of the unlabeled sample. These unlabeled images were added to the training dataset only if it passed a certain threshold. Since the classifier was trained initially only with a small subset of labeled samples, these models face a problem of confidently predicting wrong labels. The problem of confidently predicting wrong labels which might cause the model to diverge is called *confounding bias*.

5.2 Related Work

Contrastive Loss is a prominent distance criteria for smoothness based semi-supervised deep learning technique, and is the foundation for Momentum Contrast (MoCo), its successors and related approaches (He et. al 2020, Oord et. al 2018, Henaff 2020, Hjelm et. al 2018, Tian & Isola 2020, Misra et. al 2020, Li et. al 2020) [87,88,89,90,91,92,93]. There are several forms of contrastive loss (Hadsell et. al 2006, Wang et. al 2015, Wu et. al 2018 Hjelm et. al 2018) [94,95,96,90] but in its most general form one must define a similarity loss LS to penalize similar samples from having different labels, as well as a difference loss LD to penalize different samples from exhibiting the same label. Consistency Regularization Is another smoothness based strategy that has led to MixMatch and its derivatives for image classification (Berthelot et. al 2019, Sohn et. al 2020, Kurakin et. al 2020, Mustafa & Mantuik 2020) [2,3,4,36]. Consistency regularization assumes that if one augments an unlabeled sample, it's label should not change; i.e. smoothness between samples and simple augmentations thereof.

Semi-supervised learning initially appeared in the form of self-training [36]. This paper focuses on training the model with the partially labeled dataset and predicting the labels for a subset of the unlabeled dataset. These predictions were then added to the training dataset iteratively to increase the accuracy of the model. In semi-supervised learning, consistency regularization is used with the idea that the model predicts the same class to the perturbed version of the unlabeled sample. In other words, the model should be able to predict the unlabeled sample x exactly the same way it predicts the class for Augmented(x) [90]. In the "Mean Teacher" [1] algorithm, the exponential moving averages of the model parameters are taken to get a much more stable target prediction and this method has significantly shown improvements in results. One of the drawbacks in these types of methods is that they use domain specific augmentations. These problems have been overcome by techniques like "Virtual Adversarial Training" [35]. These techniques believe in generating additive samples with similar characteristics to increase the data volume and thus avoiding random augmentations. There are also approaches such as Transformation Consistency where they propose that if the x and x' that are at a distance from each other are fed into the model their corresponding predictions y and y' need to be at the same distance from each other [14]. The most common distance measurement techniques are Mean Squared Error (MSE), Kullback-Leiber (KL) divergence and Jensen-Shannon (JS) divergence. In other words, consistency regularization obtains pseudo-labels from the model's predicted distribution based on the different augmentations applied to the input image [19 , 20, 21,22].

Semi-supervised learning focuses on using the unlabeled data as a part of the training data along with the labeled samples. Proxy-label methods assign a soft label to these unlabeled samples. Self training is one of the oldest and simplest techniques that uses the model predictions for unlabeled data to be added on to the training sample followed by re-training the model with an increased sample size of training data. Formally, when a model m is training on a dataset x, the predictions of the unlabeled dataset m(x) is compared with a predetermined threshold T. If the prediction surpasses that threshold, it is then added to the dataset to re-train the model [16,27,28,29]. Co-training is a part of multi-view training where a dataset S can be represented as 2 independent feature sets S1 and S2. After the model is m1 and m2 are trained on the respective datasets, at every iteration, the predictions that surpass the predetermined threshold from exactly one model are then passed to the training dataset of the final model [30,31]. In recent times, co-training has been used in 3-D medical imaging where the coronal, sagittal and axial view of the data was trained on 3 different networks [32]. A consensus model between these 3 networks was used to predict the label for the unlabeled dataset. The major limitation with such types of models is that they are unable to correct their own mistakes and any bias or wrong prediction detected by the model results in confident but erroneous predictions. One of the papers proposed by Yalnizet et al. [13] uses self-training to improve ResNet-50 and work towards making a robust model even after being subjected to various perturbations. Initially, the model is trained on unlabeled images and their proxy-labels which is then used to fine-tune the model with the help of the labeled images in the final stage.

5.3 Theoretical Justification

Pseudolabeling as an EM approximation

Contrastive outlier suppression improves upon this semi-supervised EM framework by addressing several limitations including being designed to suppress the *confounding bias* issue. Intuitively, *confounding errors* occur when a neural network model predicts unlabeled samples incorrectly with high confidence and when these unlabeled samples are added to the labeled training set. For the purposes of measurement and evaluation, we define a *confounding error* is when an erroneous pseudolabel is added to the labeled set. We believe a fundamental source of incompatibility between the EM framework (for cluster-assumption semi-supervised learning), and deep neural networks is that EM is designed for generative models [58] , whereas most deep neural network are inherently discrimination models.

Why is the difference between generative and discrimination models important to address confounding? One major reason is because deep neural networks, as discrimination models, have the ability predict an incorrect label category with high confidence. This inconvenient property has led to a number of difficulties with DNNs including the existence of adversarial examples which intentionally trick DNNs into incorrect predictions . We hypothesize that this property also exacerbates confounding: if an erroneous prediction is made with high confidence in an early round of EM iteration, then this erroneous prediction may reinforce biases in later rounds of retraining.

Figure 4.1 illustrates why discrimination models are capable of making inaccurate predictions with high confidence in the event of encountering outlier samples that are not well represented by the labeled data used for prior iterations of training.

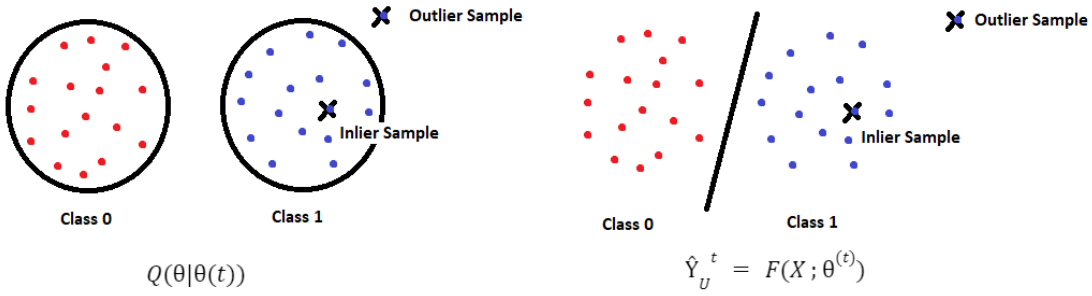

Figure 5.1. Illustration of the incompatibility of discrimination models with the cluster assumption. Left: for the generative mode, the outlier sample is (correctly) considered low probability of being within either cluster. Right: the outlier sample is (incorrectly) considered to be part of cluster 1 with high probability due to the large distance from the decision boundary.

Theoretically, pseudo-labeling as well as latent bootstrapping methods rely on the *cluster assumption* [16]. The cluster assumption can be intuitively paraphrased as follows: *If there are 2 points that belong to the same cluster, then they (very likely) belong to the same class* [16]. As such, clustering methods assume that the data is separable into K clusters $C_1 \ldots C_K$ in which the true decision boundary lies in-between the clusters, and does not pass through any individual cluster. The cluster assumption can be mathematically defined as follows,

$$\exists C_1 \ldots C_K \;\; s.t. \;\; if \;\; x_i, x_j \in C_k \;\; then \;\; y_i = y_j \qquad [5.3]$$

Pseudolabeling is a special case of the cluster assumption, in which one not only assumes the decision boundary lies in-between the clusters, but further assumes the stronger condition that there is only one cluster per label category. The Expectation Maximization (EM) algorithm [23] is the foundation for many clustering techniques. Maximum likelihood estimation of simultaneous cluster assignment Z and model parameters θ can be obtained iteratively as follows

Expectation : $$Q(\theta|\theta^{(t)}) = E_{Z|X,\theta}(t)\ log\ L\ (\theta\ ;\ X\ ,\ Z\ ) \quad [5.4]$$

Maximization : $$\theta^{(t+1)} = argmax_{\theta}\ Q(\theta \mid \theta^{t}) \quad [5.5]$$

Pseudolabeling can be theoretically justified as interpretation of EM, in that the latent variable Z is defined as the unobservable (unsupervised) training labels Y_U. Furthermore the observed variable X is defined as the intersection of all observable data measurements available for training, including the supervised training data X_S, the supervised training labels Y_S, as well as the unsupervised training data as follows X_U,

$$Z = Y_U \quad [5.6]$$

$$X = X_S \cap Y_S \cap X_U \quad [5.7]$$

Furthermore, given the basic statistical identity that $L(a|b) = p(b|a)$, the Expectation step under this interpretation is presented as follows.

$$Q(\theta|\theta^t) = E_{Y_U|X_S,Y_S,X_U,\theta^t} \log p(X_S, Y_S, X_U, Y_U|\theta) \quad [5.8]$$

One must further assume sample independence of the individual samples lying within the training dataset. Under this common assumption, the supervised and unsupervised contributions to the maximum likelihood expectation step can be split additively as follows,

$$Q(\theta|\theta^t) = E_{Y_U|X_S,Y_S,X_U,\theta^t} \log p(X_S, Y_S|\theta) + E_{Y_U|X_S,Y_S,X_U,\theta^t} \log p(X_U, Y_U|\theta) \quad [5.9]$$

Which can be simplified as

$$Q(\theta|\theta^t) = \log p(X_S, Y_S|\theta) + E_{Y_U|X_U,\theta^t} \log p(X_U, Y_U|\theta) \quad [5.10]$$

Supervised branch *Unsupervised branch*

One can also apply an additional bayesian identity in that $p(a, b \mid c) = p(a|b, c)\ p(b|\ c)$. As such, the expectation can be expanded as follows,

$$Q(\theta|\theta^t) = \log p(Y_S|X_S,\ \theta)\ p(X_S|\ \theta) + E_{Y_U|X_U,\theta^t} \log p(Y_U|X_{U,}\theta)\ p(X_U|\ \theta)$$

[5.11]

For Discrimination Models, the model parameters are not used to generate the training samples $p(X|\theta)$, but instead to calculate directly the posterior predicted probabilities $p(Y|X, \theta)$. As such, discrimination models can be viewed as making a simplifying approximation in which the component $p(X|\theta)$ is omitted from the maximum likelihood calculation by setting this value to 1 as follows,

$$p(X_S|\theta) = p(X_U|\theta) = 1 \quad [5.12]$$

Under this approximation, the expectation simplifies as follows,

$$Q(\theta|\theta^t) = log\, p(Y_S|X_S, \theta) + E_{Y_U|X_U,\theta^t} log\, p(Y_U|X_{U,}\theta) \quad [5.13]$$

One can make use of the sample independence assumption over *N* supervised samples, *M* unsupervised samples, and *C* categories to rearrange this expression in more explicit form as follows,

$$Q(\theta|\theta^t) = \sum_{i=1}^{N}\sum_{c=1}^{C} Y_{Sic}\, log\, p(Y_{Si=c}|X_S, \theta) + \sum_{i=1}^{M} E_{Y_U|X_U,\theta^t} \sum_{c=1}^{C} Y_{Uic}\, log\, p(Y_U|X_{U,}\theta) \quad [5.14]$$

Supervised Log Loss Unsupervised Expected Log Loss

When presented in this form, it becomes clear that such a cluster-based method should contain both supervised and unsupervised optimization terms, and that the supervised term resembles the negative of the well known multi-class log loss function. This term is negated only for the reason that EM attempts to *maximize* the expectation and is thus by definition a negated loss. The second term is the unsupervised contribution to the expectation. It can be seen that the unsupervised contribution is in some sense similar to the multi-class log-loss, but with an additional caveat, that this term should be averaged over all possible values of the unobserved training labels Y_U, given access to the unlabeled data X_U and the previous model parameters θ^t, from timestep *t*.

Although the EM iteration should yield a maximum likelihood semi-supervised training algorithm under appropriate conditions, traditional pseudolabeling makes use of one additional assumption in that the expected value is distributed inside the sum as well as inside of the log term as follows. In theory expected value operations follow the distributed property for summations, but do not strictly follow the distributive property for all convex functions such as log probabilities. Nevertheless, distributing this term through both summation and logarithm terms is widely employed as a baseline pseudolabeling strategy. We define this approximation as $\widehat{Q}(\theta|\theta^{t})$ as follows.

$$\widehat{Q}(\theta|\theta^{t}) = \sum_{i=1}^{N}\sum_{c=1}^{C} Y_{Sic}\, log\, p(Y_{Si=c}|X_{Si},\theta) + \sum_{i=1}^{M}\sum_{c=1}^{C} \widehat{Y}_{Uic}\, log\, p(\widehat{Y}_{Ui=c}|X_{Ui,}\theta) \quad [5.15]$$

Supervised Log Loss Unsupervised Expected Log Loss

where $\widehat{Y}_{Ui}$ is the pseudolabel as follows,

$$\widehat{Y}_{Ui} = E_{Y_{U}|X_{U},\theta^{t}}(Y_{Ui}) \quad [5.16]$$

SCOPE as an improved EM approximation

The purpose of the SCOPE methodology is to reduce confounding errors by improving upon an unsatisfactory assumption of baseline Pseudolabeling as an EM approximator. Notably, let us recall equation [5.11] as follows,

$$Q(\theta|\theta^t) = log\, p(Y_S|X_S, \theta)\ p(X_S|\theta) + E_{Y_U|X_U,\theta^t} log\, p(Y_U|X_{U,}\theta)\, p(X_U|\theta) \quad [5.17]$$

Baseline pseudolabeling makes the convenient yet unsatisfactory assumption that $p(X_S|\theta) = p(X_U|\theta) = 1$. This assumption is convenient, because discrimination models such as deep neural classifiers do not attempt to measure the probablity of the sample occurring given the model parameters $p(X|\theta)$, but instead attempt to directly infer the predicted probabilities $p(Y|X, \theta)$.

SCOPE is based on an improved assumption that again also does not require the descrimination model to be able to predict the sample probabilities as part of a differentiable loss, but does require some ability to perform non-differentiable outlier suppression based on a previous estimate of the model parameters $p(X|\theta^t)$ as follows,

$$Q(\theta|\theta^t) \approx log\, p(Y_S|X_S, \theta)\ p(X_S|\theta^t) + E_{Y_U|X_U,\theta^t} log\, p(Y_U|X_{U,}\theta)\, p(X_U|\theta^t) \quad [5.18]$$

Replacing $p(X|\,\theta)$ with $p(X_S|\,\theta^t)$ is asymptotically justifiable, because as the EM model converges, the differences between the model parameters θ and θ^t in successive EM iterations will become negligible. A practical advantage of approximating $p(X|\,\theta)$ using $p(X|\,\theta^t)$ for the purposes of semi-supervised deep learning is that $p(X|\,\theta^t)$ does not technical depend on the current estimate θ and therefore does not require a differentiable form of this expression to be integrated into the gradient descent.

As $p(X|\,\theta^t)$ does not need to take differentiable form, it is possible to approximate this quantity using a Bernoulli distribution, which we define as $\widehat{p}(X|\,\theta^t)$ being a binary approximation of $p(X|\,\theta^t)$ as follows,

$$\widehat{p}(X|\,\theta^t) \approx p(X|\,\theta^t) \quad \text{where}$$

$$\widehat{p}(X|\,\theta^t) = \begin{cases} 1/N & \text{if } p(X|\,\theta^t) > \tau \\ 0 & \text{otherwise} \end{cases} \qquad [5.19]$$

The use of a binary approximation implies an outlier removal strategy. As such, if one can identify samples for which $p(X|\,\theta^t)$ is unlikely, these samples can be removed as outliers from the Maximization step. Whereas, if one can identify samples for which $p(X|\,\theta^t)$ is likely, these samples can be included as inliers to the

maximization. If one repeats the derivation of equation [5.15] but instead using this assumption for unlabeled data points, one arrives at the following which describes the SCOPE methodology in its most general form.

$$\widehat{Q}(\theta|\theta^{t}) = \sum_{i=1}^{N}\sum_{c=1}^{C} Y_{Sic}\, log\, p(Y_{Si=c}|X_{Si},\theta) + \sum_{i=1}^{M}\sum_{c=1}^{C} \widehat{Y}_{Uic}\, log\, p(\widehat{Y}_{Ui=c}|X_{Ui,}\theta)\, \widehat{p}(X_{Ui}|\,\theta^{t}) \quad [5.20]$$

Supervised Log Loss Unsupervised Expected Log Loss

where $\widehat{Y}_{Ui}$ is the pseudolabel as follows,

$$\widehat{Y}_{Ui} = E_{Y_{U}|X_{U},\theta^{t}}(Y_{Ui}) \quad [5.21]$$

Where $\widehat{p}(X_{Ui}|\,\theta^{t})$ is a binary outlier removal term for unlikely unlabeled samples

Consistency Regularization

Consistency regularization is an approach in which one adds an additional constraint that the predicted label should not change through augmentation. Let us define A as the space of augmentation parameters, and α as the augmentation function. Consistency regularization introduces the following constraint.

$$p(Y|\alpha(X, A_{1}), \theta) = p(Y|\alpha(X, A_{2}), \theta) \quad for\ all\ \ A_{1}, A_{2} \in A \quad [5.22]$$

Consistency regularization is often defined with α as a random function, but it is equivalently presented here with α as a deterministic function but randomly chosen parameters A_1 , A_2 $\in$ A, where A is the space of augmentation parameters. This constraint states algebraically that augmentation should not change the predicted labels. Consistency regularization, like other forms of regularization, can be implemented by adding a penalty $L_{consist}$ to the overall loss function for optimization. Or conversely, as EM is a *maximization* procedure, by subtracting the following penalty $L_{consist}$ from the overall maximization step term $\widehat{Q}(\theta|\theta^t)$.

$$L_{consist} = E_{A_1, A_2 \in A} \; L[\, p(Y|\alpha(X, A_1), \theta) - p(Y|\alpha(X, A_2), \theta)\,] \quad [5.23]$$

Consistency regularization is highly dependent on the ability to obtain a viable augmentation function that is unlikely to alter the true label of the image. Recent work in the use of Consistency regularization for semi-supervised learning has yielded a number of augmentation functions that perform well for image and digit classification datasets [2,3,5,36]. SCOPE makes use of Control-Theory Augment (CT-Augment), Cut-out Augment, and Rand Augment.

5.4 SCOPE Meta-Algorithm and Architecture

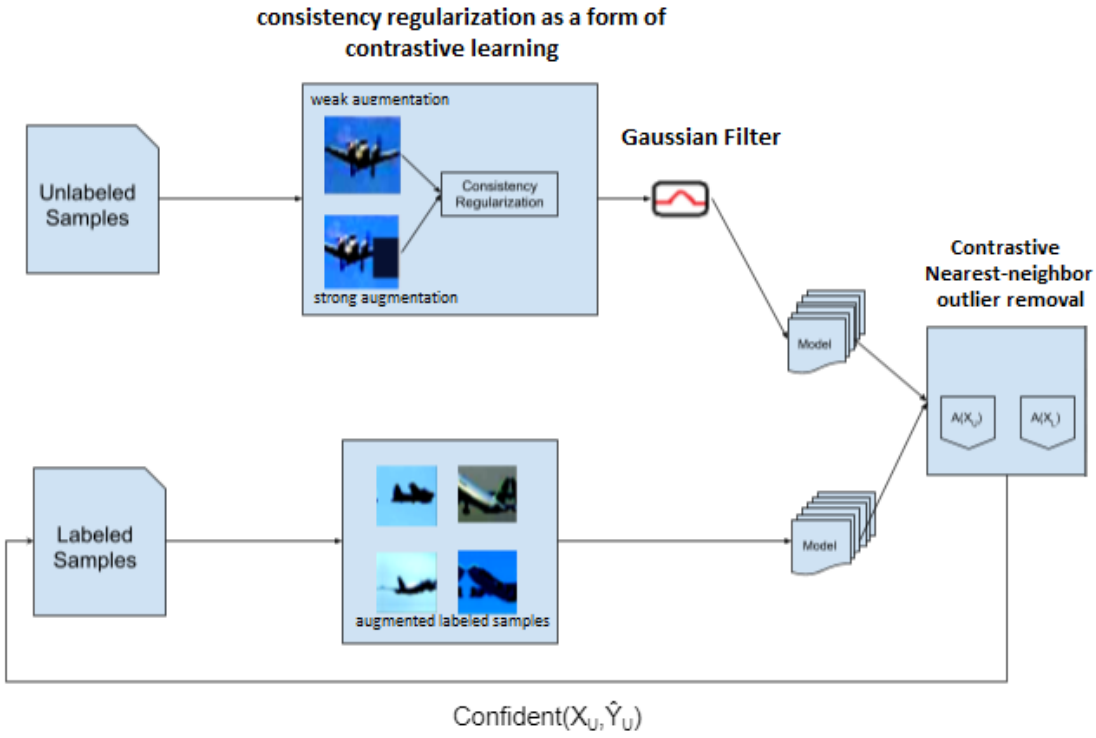

Figure 5.2. Description of the SCOPE Meta-learning algorithm

Figure 5.2. shows the scope Meta-learning algorithm which implements equation [5.20] as well as extends this EM approximator using additional information. Figure 5.2 describes the application of equation [5.20]. The meta-learning algorithm consists of a supervised branch and an unsupervised branch. The supervised branch consists of the supervised images and its corresponding labels which is the lower-half of the figure 5.2. The images in the supervised branch are passed through an augmentation function which consists of basic augmentations such as left-shift, right-shift, rotate etc. As shown in the equation [5.20], the supervised branch has a supervised loss

which in this architecture is the cross-entropy between the true labels and the predicted labels by the model.

The top half of Figure 5.2 describes the unsupervised branch. The unlabeled images are then passed through the consistency regularization layer. Consistency regularization as a form for contrastive learning is used as a part of the proposed approach [13]. The unlabeled samples are passed into the model as 2 branches. The first branch being the weak augmentation of the unlabeled sample and the second branch being the strong augmentation of the same unlabeled sample [3]. The model's predictions for the weak augmented images are threshold to a confidence score of 0.95 and every image that surpasses this threshold is considered as the true label for the unlabeled sample. The model's prediction for the strong augmentation image is considered as the prediction for the unlabeled sample and a cross-entropy loss is applied between the predicted class for the weak augmented sample and the predicted probability distribution of the strong augmented image. This loss term as a whole is considered as the unlabeled loss. The labeled loss on the other hand is the cross-entropy between the true label of the sample and the predicted probability distribution of the labeled sample. This technique has achieved very good results for semi-supervised learning [3]. Neural networks have the characteristics to predict incorrectly with high confidence. In the proposed approach the gaussian filter as shown in Figure 5.2 helps to not miss out on correctly predicted low confidence unlabeled samples which is described in detail below in section 4.2. The gaussian filter tries to fit the probabilities of the predicted class in a gaussian probability

density function and adds the correctly predicted low confidence samples as a part of the gradient descent. The proposed approach also leverages from increasing the labeled samples in every epoch by adding high confidence unlabeled samples which are described in detail in section 4.4. The contrastive nearest-neighbor outlier removal section in Figure 5.2 compares the cosine similarity between the features vectors of an unlabeled sample which is predicted to be of a certain class with the feature vectors of the labeled samples of the sample class. There are many potential ways to implement a binary outlier removal term $\widehat{p}(X_{Ui}|\,\theta^t)$. We analyze two simple heuristics that can be used for outlier removal. Notice that because we are working with a discrimination model, the model parameters θ^t do not easily allow one to estimate $p(X_{Ui}|\,\theta^t)$. However, it is possible to make use of similarity learning.

An iterative bootstrap

As explained in section 3.2, one approach toward implementing a binary outlier removal term $\widehat{p}(X_{Ui}|\,\theta^t)$ is to define a Labeled set $X_L{}^t$ at EM iteration *t*, which contains the true labels as well as the Pseudolabels at iteration *t*. If

One approach toward implementing a binary outlier removal term $\widehat{p}(X_{Ui}|\,\theta^t)$ is to make use of contrastive learning. If the sample X_{Ui} is within a threshold distance λ of *k* other inlier samples $x_1 \ldots x_k \in X_S$, then we assume that $p(X_{Ui}|\,\theta^t) > \tau$ and

therefore assign $\widehat{p}(X_{Ui}|\,\theta^t)$ to $1/N$. However, if fewer than k inliers to X_{Ui} are within the threshold distance λ, then we assume $p(X_{Ui}|\,\theta^t) < \tau$ and therefore assign $\widehat{p}(X_{Ui}|\,\theta^t)$ to 0, thereby removing the outlier from consideration in gradient descent via the maximization term of EM.

Gaussian Filtering and Outlier Removal

As previously described, we define $\widehat{p}(X_{Ui}|\,\theta^t)$ is a binary outlier removal term, in which we attempt to remove samples that may be unlikely to appear given the model parameters. The most straightforward way to implement such a term would be to measure $p(X\ |\,\theta^t)$ directly and then to determine if $p(X\ |\,\theta^t)$ exceeds a threshold τ. As the label categories are mutually exclusive, one may use the following identity.

$$p(X\,|\,\theta^t) \;=\; \sum_{c=1}^{C} p(X\,|Y=c,\ \theta^t)\,p(Y=c\,|\,\theta^t) \qquad [5.24]$$

And given that $p(Y=c)$ is mutually independent of θ^t (as we have no information of X), this simplifies to the following,

$$p(X\,|\,\theta^t) \;=\; \sum_{c=1}^{C} \frac{N_{Y=c}}{N}\,p(X\,|\,Y=c,\ \theta^t) \qquad [5.25]$$

The quantity $p(X\,|\,Y=c,\ \theta^t)$ can be straightforwardly estimated using the pseudo-labeled set if one can make use of θ^t to produce manifold that projects the data X into a space that is approximately normally distributed. Define $F(X,\ \theta^t)$ as the

output of a neural network taking X as input, and using trained weights θ^t. One can therefore estimate $p(X \mid \theta^t)$ as follows,

$$p(X \mid \theta^t) = \sum_{c=1}^{C} \frac{N_{Y=c}}{N} p(F(X, \theta^t) \mid Y = c) \quad [5.26]$$

For simplicity the Gaussian filtering technique for scope uses the output predicted probabilities as the manifold $F(X, \theta^t)$, and furthermore makes use of the diagonal

According to our second hypothesis, the labels of the labeled and the unlabeled samples of a particular class should follow a gaussian distribution. Let $F(X, \theta^t)$ be the manifold of samples X which is the probability distribution of the feature vectors produced by the model.

Our proposed method considers the probability distribution $F(X, \theta^t)$ of the samples who have received a same pseudo label value $Y = c$. Let $\alpha(X, A_1)$ and $\alpha(X, A_2)$ be the batch of the unlabeled strong and weak augmentations at time t that belong to class c, where c is the pseudo label predicted by the model.

We initialize the mean μ_c^t and the standard deviation σ_c^t for the probability distribution $F(X, \theta^t)$ where $Y = c$.

$$p(F(X,\ \theta^t)\ |\ Y\ =\ c)\ \ =\ \ \frac{1}{\sqrt{2\pi\sigma}}\ exp(\frac{-0.5\ *\ ((F(X,\ \theta^t))\ -\ \mu_c^t)^2}{(\sigma_c^t)^2}) \qquad [5.27]$$

Algorithm 5.1: Pseudo Code for Normal Distribution Filter

```
For c = 1 … C:
```

$$\mu_c^t\ \ =\ E_{X|Y=c}\ F(X,\ \theta^t)$$

$$\sigma_c^t\ \ =\ \sqrt{E_{X|Y=c}\ (F(X,\ \theta^t)\ -\ \mu_c^t\ )^2}$$

```
Pseudo_Labels_To_Include= []
```

$p(F(X,\ \theta^t))\ =\ 0$

```
For c = 1 … C:
```

$$p(F(X,\ \theta^t))\ \ +=\ \ \frac{1}{\sqrt{2\pi\sigma}}\ exp(\frac{-0.5\ *\ ((F(X,\ \theta^t))\ -\ \mu_c^t)^2}{(\sigma_c^t)^2})$$

```
Pseudo_Labels_To_Include.append(|p(F(X, θ^t)) >= τ)|)
```

We see that after the normal distribution filter we are able to segregate the labels which belong to a particular class with a little more confidence which also helps to solve the confounding bias issue.

Contrastive Learning

Contrastive Learning can be understood as learning by comparing multiple input samples that are in some sense similar to each other (Le-Khak 2020). Contrastive Learning can be used to estimate

Discriminative models use contratrastive learning as an approach to group similar samples closer and different samples away from each other [100]. The distance between these samples is often measured with a distance metric which helps to evaluate if the sample lies close or far to another sample. The discriminator is trained in a way that it learns if an original sample and the augmented version of the original sample lie close to each other in a feature space which is further used to fine tune the model. In the general form of contrastive loss, the similarity metric is usually cosine similarity where we want the similar vectors to have a cosine similarity distance close to 1.

The manifold assumption states that a high-dimensional data-point can be represented in a low-dimensional feature space which could be used to learn special feature-representations that could not be learned from a high-dimensional feature space alone. Now to pseudo-label the unlabeled samples, cosine similarity is used as a contrastive loss metric to evaluate the angular distance between the low-dimensional feature space of the unlabeled sample with a pseudo label of class c and the low-dimensional feature space of the labeled sample with a supervised label of class c.

Cosine similarity works best for this approach because it compares the low-dimensional representations of angular distance between 1 being perfectly similar and -1 being completely different.

$$d(F(X_{U'}, \theta^t), F(X_{S'}, \theta^t)) \;=\; ||F(X_{S'}, \theta^t)||\,||\,F(X_{U'}, \theta^t)\,||\,cos\theta \qquad [5.28]$$

Where $cos\theta$ is represented using the dot product and magnitude as

$$cos\theta \;=\; \frac{F(X_{U'}, \theta^t)\,F(X_{S'}, \theta^t)}{\sqrt{F(X_{U'}, \theta^t)^2}\,\sqrt{F(X_{S'}, \theta^t)^2}} \qquad [5.29]$$

Contrastive Nearest Neighbor Outlier Removal

Similar to the previous section where we describe $\widehat{p}(X_{Ui}|\,\theta^t)$ as the outlier removal term, we describe how we use a nearest neighbor comparison to add confidently predicted unlabeled samples to the labeled training set. As we have derived above from equations for maximization $Q(\theta|\theta^t)$ and the probability of a image given the model parameters $p(X\,|\,\theta^t)$,

Another way of estimating if $p(X|\,\theta^t) > \tau$ is to make use of a contrastive learning approach based on k-nearest neighbors. Intuitively, if a sample X is nearby other supervised samples in X_S used for fitting the model θ^t, then one might determine that $p(X\,|\,\theta^t) > \tau$. Conversely, if a sample X is far from all known supervised samples in X_S one may conclude that $p(X\,|\,\theta^t) < \tau$. This intuition has a theoretical basis, because the K-Nearest Neighbor algorithm is the optimal classifier assuming a Variable-width Balloon Kernel Density Estimator as follows,

$$p(X \mid \theta^t) = \frac{1}{n\,h^D} \sum_{i=1}^{n} K\left(\frac{F(X,\theta^t) - F(X_i,\theta^t)}{h}\right) \quad [5.30]$$

$$\text{where} \quad h = \frac{k}{(n\,p(X \mid \theta^t))^{1/D}}$$

$$\text{where} \quad K(z) = \frac{1}{\sqrt{2\pi}} exp\left(-\frac{1}{2}z^2\right)$$

It can be shown that under these assumptions, $p(X \mid \theta^t)$ increases monotonically as sum of euclidean distances to the k nearest samples $d(X,\ X_s,\ k)$ decreases as follows,

$$\forall\, \tau \qquad \exists\, \gamma \quad s.t.$$

$$p(X \mid \theta^t) > \tau \qquad iff \qquad d(X,\ X_s,\ k) < \gamma \quad [5.31]$$

The nearest neighbor contrastive outlier removal step for the SCOPE algorithm is based on this strategy with two minor practical tweeks. First, rather than euclidean distance between sample feature spaces $F(X_i, \theta^t)$ and $F(X_j, \theta^t)$, cosine similarity is used. This is because cosine similarity is a standard distance metric for use with contrastive learning techniques that compare individual sample feature vectors. Secondly, rather than comparing distance to all samples within the supervised set, this comparison is made only for the samples within the predicted pseudo-label category.

Our algorithm:

Let X_{Si} and $F(X_{Si},\ \theta^t)$ be the labeled image and the corresponding manifold.

Similarly let X and $F(X,\ \theta^t)$ be the unlabeled image of interest and the corresponding

manifold which belongs to class c where c = 1..10. If the unlabeled manifold $F(X, \theta^t)$ has a cosine similarity score of above γ with at least k labeled samples with the same pseudo label of the corresponding class we add X to the list of probable labeled candidates.

Algorithm 5.2: Pseudo Code for contrastive Nearest Neighbor Outlier Removal

```
Dict = {0:[], 1:[].... 9:[]}          # dictionary with keys

For c in range(10):
      L = { X_i : x_i ∈ X_s  and  Y_i = c }
      U = { X_i : x_i ∈ X_U  and  Y_i = c } #C is the class from 0 to
9
      Dict_2 = {U_1 :0 … U_n :0 }
      For i in range(len(L)):
            For j in  range(len( U)):
                  If (Cosine Similarity(L[i], U[j] )) > γ:
                        Dict_2[U[j]]+=1
      For k in Dict_2:
            If Dict_2[k]>=k:
                  Dict[l].append(k)
```

5.5 SCOPE Experimental Setup and Results

SCOPE was evaluated according to the test accuracy on the CIFAR-10 dataset to determine the extent to which our algorithm can enhance semi-supervised classifiers to correctly classify the test data. The algorithm was later compared to some of the benchmark semi-supervised learning algorithms. CIFAR-10 is a widely used dataset

which has been used by multiple state-of-the-art models to evaluate the performance of the algorithm. This dataset consists of 50000 images and their corresponding labels for training and 10000 images and labels for testing. The dataset consists of 10 classes and hence makes it challenging for classifiers. As other models the proposed algorithm uses Wide ResNet as the classifier. This paper evaluates the test results on 2 particular splits of labeled and unlabeled data. The first experiment consists of a training sample size of 250 labeled images and 49750 unlabeled images whereas the second experiment consists of a training sample size of 4000 labeled images and 46000 unlabeled images. In the first experiment, with 250 labels, the π-Model yields an accuracy of 45.74% and the pseudo-labeling paper reports an accuracy of 50.22%. Mean teacher which is one of the advanced techniques which uses the exponential moving averages of weights, yields an accuracy of 67.67%. MixMatch, which uses k number of augmentations, averages the model predictions over all the augmentations and uses temperature sharpening reports with an accuracy of 88.95%. The fix-match algorithm which had reported better results than all the above approaches used consistency regularization and exponential moving average of the weights reported an accuracy of 94.93%. The proposed algorithm in this paper uses consistency regularization along with contrastive latent bootstrapping using outlier removal yields 95.46% accuracy with 250 labeled samples.

In the second experiment, with 4000 labeled images and 46000 unlabeled images, the π-Model yields an accuracy of 85.99% and the pseudo-labeling paper reports an accuracy of 83.91%. The mean-teacher algorithm receives an accuracy of 90.81%

which performs better than the mean-teacher approach. The fix-match approach has reported an accuracy of 95.69%. The semi-supervised contrastive latent bootstrapping along with consistency regularization achieves better results with 4000 labeled samples with an accuracy of 95.82%.

Method	Accuracy with 250 Labels	Accuracy with 4000 Labels
π-Model	45.74% (44.76,46.72)	85.99% (85.29,86.66)
Pseudo-Labeling	50.22% (49.24,51.20)	83.91% (83.17,83.63)
Mean Teacher	67.68% (66.75,68.60)	90.81% (90.23,91.37)
Mix-Match	88.95% (88.32,89.56)	93.58% (93.08,94.05)
Fix-Match	94.93% (94.48,95.35)	95.69% (95.27,96.08)
SCOPE	**95.52% (95.10,95.92)**	**95.82% (95.41,96.20)**

Table 5.1: Accuracies of different architectures on CIFAR-10

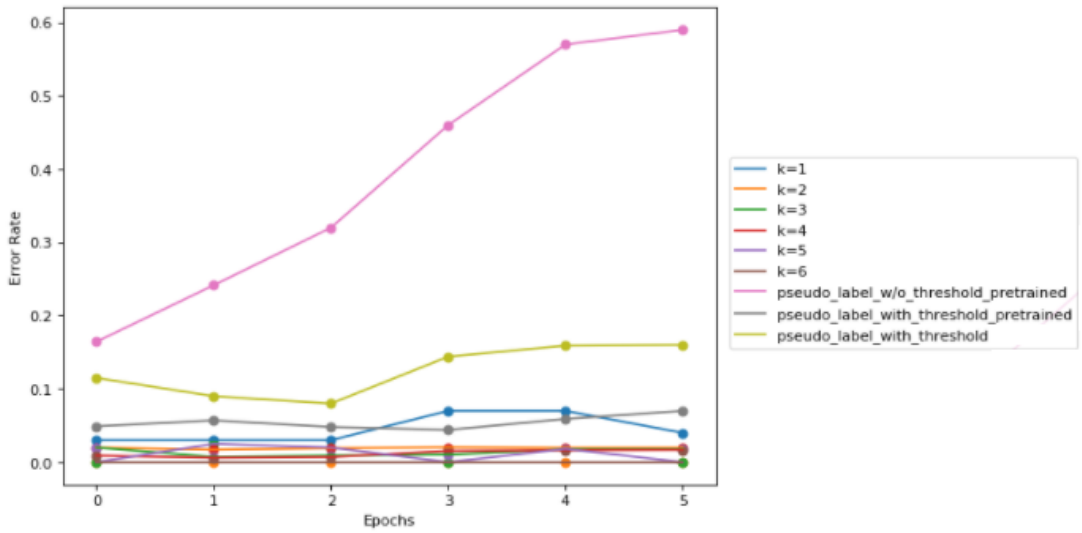

Figure 5.3 Confounding error rate per epoch.

The above figure 5.3 explains the confounding error rate of the proposed model while using various numbers of neighbors k. This figure helps to identify the amount of incorrectly predicted samples added to the labeled branch of the model which causes the model to diverge further. We see that as the number of neighbors used to evaluate the class of the model increases, the confounding error rate of the unlabeled samples being added to the labeled branch of the training data decreases. As shown in the above diagram, when k=1, the confounding error being added to the model is maximum. When k=2, the error rate drops as compared to when k=1. Similarly the error rate for k=3 is lesser than when k=2 and greater than when k=4. We use the value k=6 as the error rate is the least. Also, 3 other experiments which show the confounding error rate addition of unlabeled samples as labeled data in every epoch were performed. When pseudo labeling is performed with a pre-trained consistency regularization model like fixmatch without any threshold, the confounding error rate is maximum and it keeps getting worse after every epoch. The same model when pseudo labeled with a threshold of 0.95 i.e adding the pseudo labeled samples to the labeled training dataset only if it achieved a prediction score of ≥ 0.95, the confounding error rate is still higher than pseudo-labeling with scope. The same model without any pre-training when psudolabaled with a threshold of 0.95, the confounding error rate is still high and in every epoch keeps getting worse. This means that a lot of wrongly labeled samples are added to the labeled training set which causes the model to diverge. The confounding error rate per epoch plot helps to

understand that the importance of scope which introducing pseudolabeling with consistency regularization.

K Neighbors	Confounding Error Rate
k=1	2.3%
k=2	2.13%
k=3	1.28%
k=4	1.2%
k=5	0.98%
k=6	0.97%

Table 5.2 Total confounding error rates with different values of k

The confounding error rate of the pseudo labels with different values of k (hierarchical nearest neighbors) were calculated to evaluate the effectiveness of the algorithm. It is observed that when the feature spaces of the unlabeled sample for a predicted pseudo label are compared with the supervised images of the same class, the confounding error rate of the unlabeled samples being added to the labeled branch of the training data significantly decreases. A stress test is performed to find out at what value of k can we get the best confounding error rate. As shown in the table below when the value of k =1, a confounding error rate of 2.3% was achieved and as the value of k was increased to k=2, the error rate dropped to 2.13%. Further when the value of k was increased to 3,4 and 5 the confounding error rate drops to 1.28%, 1.2% and 0.98% respectively. The best error rate was achieved when the value of k was 6 which yielded an error rate of 0.97%. This experiment concludes that

measuring the angular distance of the feature spaces of similar classes helps to decrease the error rate and increase the accuracy of the model.

Method	Accuracy with 250 labeled samples	Accuracy with 4000 labeled samples
Without SCOPE (Fixmatch)	94.93% (94.48, 95.35)	95.69% (95.27, 96.08)
Scope with only gaussian filter	95.28% (94.85, 95.69)	95.69% (95.27, 96.08)
Scope with only contrastive nearest neighbor	95.39% (94.96, 95.79)	95.71% (95.29, 96.10)
SCOPE	**95.52% (95.10, 95.92)**	**95.82% (95.41,96.20)**

Table 5.3: Ablation Study for SCOPE

An ablation study was performed to study the performance of the model by using only the gaussian filter and by using only the pseudolabeling with outlier removal and bootstrapping. Scope with only the gaussian filter yields an accuracy of 95.28% with only 250 labeled samples and yields an accuracy of 95.698% with only 4000 labeled samples. Scope with only pseudo-labeling using outlier removal and bootstrapping yields an accuracy of 95.39% with only 250 labeled samples and 95.71% with only 4000 labeled samples. This experiment helps to demonstrate the effectiveness of using both the gaussian filter and the pseudo-labeling using outlier removal and bootstrapping as it performs better than using only one of the features by itself.

5.6 Learnings

Semi-supervised learning has made a lot of progress in the recent past but most of these algorithms focus on complicated loss functions. This paper focuses on

developing a simpler contrastive latent bootstrapping algorithm which follows the framework of expectation maximization. The use of hierarchical cluster assumption along with the smoothness assumption for consistency regularization yields better results in an expectation maximization framework. SCOPE shows the importance of gaining information by comparing the feature representations of images belonging to the same class with a similarity metric. These high confidence predictions for the unlabeled images are then bootstrapped and added to the training data which increases the amount of labeled samples. Thus when the model is re-trained in the next epoch, the parameters are tuned even further which is analogous to the maximization step. SCOPE achieves a better performance on the CIFAR-10 dataset as compared to the other methodologies. These results signify that such simple and elegant frameworks can help yield better test accuracies by labeling unlabeled images on the fly and can be very inexpensive and useful for highly unlabeled datasets.

Chapter 6: Future Work and Conclusion

6.1 Future Work

Immediate Improvements

SCOPE as a technique uses a novel gaussian filter and contrastive k-nearest neighbor as contrastive outlier removal techniques. As immediate improvements, we would like to evaluate our technique on a few more datasets which includes medical images as well and analyze the ROC's to evaluate the ability of the model to acknowledge

false positives. As obtaining manually annotated CT scans by board certified radiologists can be really expensive we would like to test our model on the Kaggle'17 Lung Cancer dataset and the National Lung Screening Trial dataset (NLST). The CIFAR-10 dataset consists of 10 classes and we would further like to evaluate our model on a dataset like CIFAR-100 which has 100 different classes. We would also try a combination of different outlier removal techniques to evaluate the performance of the model. Another experiment that we would like to perform is to compare the performance of the model on a dataset which has a class imbalance and how this problem of class imbalance could be addressed by using a different distribution filter.

Semi-Supervised Learning Extensions

SCOPE addresses the problem of confounding bias by using an ensemble of consistency regularization and the cluster assumption through expectation maximization. There is other relevant work on the use of either the smoothness or the manifold assumption and we would like to incorporate this related work into our method. We would like to evaluate our model by combining the smoothness and the manifold assumption based techniques as well. Semi-supervised Generative adversarial networks (SGANS) address the problem of limited labeled training dataset by generating synthetic samples which could be used to improve discriminative models. Working toward the discovery of a novel algorithm that combines generative methods would be an interesting study.

Other Problems

The work presented in this dissertation mainly addresses the problem for images and we would like to compare the performance of our model on a textural dataset to evaluate the robustness of our model. If we have two neural networks with the same architecture and we train one network on an image dataset and the other network on the textural dataset, even though the feature maps produced by the model would be in the same space for both the datasets, the distribution of these feature vectors are certain to be different [102]. Hence the study of evaluating semi-supervised neural networks on textural datasets could be interesting.

Significance

Semi-supervised self-training network models have focused on ways to infer information from unlabeled data with pseudolabeling techniques. The problem that needs to be addressed is that neural networks can often predict incorrect pseudolabels as the amount of labeled training data might not be enough to learn about the entire distribution of a particular class which results in outliers. The work previously done in the domain of semi-supervised learning techniques have not specifically targeted the problem of reducing confounding bias by identifying outliers in proxy-label methods. SCOPE addresses this problem of outliers and how they can be resolved using a contrastive outlier removal strategy which is the significance of this technique in the domain of semi-supervised learning. We have discovered that confounding errors can be caused by outlier predictions and that this appears to be a fundamental limitation of discrimination models that yield high confidence for samples far from

the decision boundary. This limitation appears to be an incompatibility between discrimation models and proxy-label methods. We believe that future research should attempt to target this limitation in ways that expand and build upon the outlier removal strategies analyzed as part of this dissertation.

6.2 Conclusion

Annotating data in general is a very expensive task and semi-supervised learning has proven to be a promising area of research to help overcome this problem. This dissertation summarizes the steps taken towards developing with a novel sophisticated solution for semi-supervised classification. Initially we describe the entire spectrum of research being done in the domain which is a part of the smoothness, cluster and the manifold assumption. The active semi-supervised expectation maximization technique uses a bayesian inference technique which helps to determine the moment for human intervention and provide annotated samples to the semi-supervised learning model. This section focuses on active-learning and helps to reduce the dependency of a large amount of labeled samples. ASEM was evaluated on 3 publicly available medical ct-scans and yields better AUC scores as compared to similar techniques. The deep expectation-maximization for semi-supervised lung cancer screening demonstrates good results as compared to a completely supervised model, but the selection criteria of adding confident unlabeled samples to the training dataset can cause *confounding bias* which could cause the model to diverge. The Semi-supervised Contrastive Outlier - removal for Pseudo - Expectation - Maximization (SCOPE) technique helps to reduce these outliers by increasing the amount of confidently predicted unlabeled samples by comparing their

low-dimensional feature vectors with k-neighbors of the labeled samples. This technique also consists of a gaussian filter which filters outliers in the unlabeled loss during the gradient descent. Thus, this technique yields better results and can be considered as an improvement over the consistency regularization techniques. Overall, this dissertation demonstrates techniques to address the confounding bias error with neural networks for semi-supervised learning and we hope this provides a new perspective towards using semi-supervised learning for classification with very limited labeled samples.

Bibliography

[1] Tarvainen, A., & Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. arXiv preprint arXiv:1703.01780.

[2] Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., & Raffel, C. (2019). Mixmatch: A holistic approach to semi-supervised learning. arXiv preprint arXiv:1905.02249.

[3] Sohn, K., Berthelot, D., Li, C. L., Zhang, Z., Carlini, N., Cubuk, E. D., ... & Raffel, C. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. arXiv preprint arXiv:2001.07685.

[4] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. IEEE Transactions on Neural Networks, 20(3):542–542, 2009

[5] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In Eighth International Conference on Learning Representations, 2020.

[6] Lan-Zhe Guo, Zhen-Yu Zhang, Yuan Jiang, Yu-Feng Li, and Zhi-Hua Zhou. Safe deep semisupervised learning for unseen-class unlabeled data. In International Conference on Machine Learning, 2020.

[7] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. arXiv preprint arXiv:1908.02983, 2019.

[8] Bingbing Xu, Huawei Shen, Qi Cao, Keting Cen, and Xueqi Cheng. Graph Convolutional Networks using Heat Kernel for Semi-supervised Learning. In IJCAI, 2019.

[9] Xia, Y., Liu, F., Yang, D., Cai, J., Yu, L., Zhu, Z., Xu, D., Yuille, A., Roth, H., 2020. 3d semi-supervised learning with uncertainty-aware multi-view co-training

[10] Zhai, X., Oliver, A., Kolesnikov, A., & Beyer, L. (2019). S4l: Self-supervised semi-supervised learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 1476-1485).

[11] Jeong, J.; Lee, S.; Kim, J.; and Kwak, N. 2019. Consistency based semi-supervised learning for object detection. In NeurIPS, 10759–10768.

[12] Verma, V., Kawaguchi, K., Lamb, A., Kannala, J., Bengio, Y., & Lopez-Paz, D. (2019). Interpolation consistency training for semi-supervised learning. arXiv preprint arXiv:1903.03825.

[13] Yalniz, I. Z., Jégou, H., Chen, K., Paluri, M., & Mahajan, D. (2019). Billion-scale semi-supervised learning for image classification. arXiv preprint arXiv:1905.00546.

[14] Qimai Li, Xiao-Ming Wu, Han Liu, Xiaotong Zhang, and Zhichao Guan. Label efficient semi-supervised learning via graph filtering. CoRR, abs/1901.09993, 2019.

[15] Livieris, I.; Drakopoulou, K.; Tampakas, V.; Mikropoulos, T.; Pintelas, P. Predicting secondary school students' performance utilizing a semi-supervised learning approach. J. Educ. Comput. Res. 2018.

[16] Park, S., Park, J., Shin, S.-J., and Moon, I.-C. Adversarial dropout for supervised and semi-supervised learning. AAAI, 2018.

[17] Luo, Y., Zhu, J., Li, M., Ren, Y., and Zhang, B. Smooth neighbors on teacher graphs for semisupervised learning. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pp. 8896–8905, 2018.

[18] G. Synnaeve, Q. Xu, J. Kahn, T. Likhomanenko, E. Grave, V. Pratap, A. Sriram, V. Liptchinsky, and R. Collobert, "End-toend asr: from supervised to semi-supervised learning with modern architectures," 2019.

[19] Papandreo, George, et al. "Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation." Proceedings of the IEEE international conference on computer vision. (2015) [partly graphical]

[20] Sharif Razavian, A., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 806-813). (Transfer Learning)

[21] Odena, A. (2016). Semi-supervised learning with generative adversarial networks. arXiv preprint arXiv:1606.01583 .

[22] A. Baraldi, L. Bruzzone and P. Blonda, "A multiscale expectation-maximization semisupervised classifier suitable for badly posed image classification," in IEEE Transactions on Image Processing, vol. 15, no. 8, pp. 2208-2225, Aug. 2006.

[23] V.J. Prakash, L.M. Nithya, A survey On semi-supervised learning techniques, Int. J. Comput. Trends Technol. 8 (2014) 25–29.

[24] Fujino, A., Ueda, N., & Saito, K. “A hybrid generative/discriminative approach to semi-supervised classifier design”. AAAI-05, the Twentieth National Conference on Artificial Intelligence Jan 2005

[25] C. Rosenberg, M. Hebert, and H. Schneiderman, “Semi-Supervised Self-Training of Object Detection Models,” Proc. Seventh Workshop Applications of Computer Vision, vol. 1, pp. 29-36, Jan.2005

[26] Blum, A., Mitchell, T. "Combining labeled and unlabeled data with co-training" COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann, 1998, p. 92-100.

[27] Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In International Conference on Machine Learning (ICML), 2016.

[28] Jesper E Van Engelen and Holger H Hoos. 2020. A survey on semi-supervised learning. Machine Learning 109, 2 (2020), 373–440

[29] Hua, Kai-Lung, et al. "Computer-aided classification of lung nodules on computed tomography images via deep learning technique." *OncoTargets and therapy* 8 (2015).

[30] Shin, H. C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., ... & Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, *35*(5), 1285-1298.

[31] Wang, Dayong, et al. "Deep learning for identifying metastatic breast cancer." *arXiv preprint arXiv:1606.05718* (2016).

[32] Bejnordi, Babak Ehteshami, et al. "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer." *Jama* 318.22 (2017): 2199-2210.

[33] Y. Ouali, C. Hudelot, and M. Tami, “An overview of deep semisupervised learning,” CoRR, vol. abs/2006.05278, 2020.

[34] S. Laine and T. Aila. Temporal ensembling for semisupervised learning. arXiv preprint arXiv:1610.02242, 2016.

[35] Takeru Miyato, Shin-ichi Maeda, Shin Ishii, and Masanori Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE transactions on pattern analysis and machine intelligence, 2018.

[36] Transformation Consistency Regularization– A Semi-Supervised Paradigm for Image-to-Image Translation [Regularization paper]

[37] M. Sajjadi, M. Javanmardi, and T. Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In Advances in Neural Information Processing Systems, 2016.

[38] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In NIPS, 2016.

[39] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. arXiv preprint arXiv:1905.00546, 2019.

[40] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In Advances in neural information processing systems, pages 3581–3589, 2014

[41]Kingma, Diederik P and Welling, Max. Auto-Encoding Variational Bayes. In The 2nd International Conference on Learning Representations (ICLR), 2013.

[42] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680,2014.

[43] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5070–5079, 2019.

[44] Arik Azran. The rendezvous algorithm: Multiclass semi-supervised learning with markov random walks. In Proceedings of the 24th international conference on Machine learning, pages 49–56, 2007.

[45] B. Settles. Active learning literature survey. Computer Science Technical Report 1648, University of Wisconsin-Madison, January 2010.

[46] P. Nguyen, D. Chapman, S. Menon, M. Morris and Y. Yesha, "Active semi-supervised expectation maximization learning for lung cancer detection from Computerized Tomography (CT) images with minimally label training data" in Medical Imaging 2020: Computer-Aided Diagnosis, International Society for Optics and Photonics, vol. 11314, pp. 113142E, 2020.

[47] S. Menon, J. Galita, D. Chapman, A. Gangopadhyay, J. Mangalagiri, P. Nguyen, and M. Morris, “Generating Realistic COVID-19 Xrays with a Mean Teacher+Transfer Learning GAN”, arXiv preprint arXiv:2009.12478, 2020.

[48] Alex Ratner, P Varma, B Hancock, and Chris Ré. Weak supervision: A new programming paradigm for machine learning (2019).

[49] Tyler Tian Lu. Fundamental limitations of semi-supervised learning. Master's thesis, University of Waterloo, 2009.

[50] Anh-Cuong Le, Akira Shimazu, and Le-Minh Nguyen. 2006. Investigating problems of semi-supervised learning for word sense disambiguation. In Proc. ICCPOL-06.

[51] Goldman, S., Zhou, Y.: Enhancing supervised learning with unlabeled data. In: Proc. ICML 2000, pp. 327–334 (2000)

[52] https://www.toptal.com/machine-learning/semi-supervised-image-classification

[53] H. Zhao, J. Zheng, W. Deng and Y. Song, "Semi-supervised broad learning system based on manifold regularization and broad network", IEEE Trans. Circuits Syst. I Reg. Papers, vol. 67, no. 3, pp. 983-994, Mar. 2020.

[54] Litjens, Geert, et al. "A survey on deep learning in medical image analysis." Medical image analysis 42 (2017): 60-885.

[55] Learning deep spatial lung features by 3D convolutional neural network for early cancer detection Taolin Jin1, Hui

[56] J. Xu, X. Luo, G. Wang, H. Gilmore, A. Madabhushi "A deep convolutional neural network for segmenting and classifying epithelial and stromal regions in histopathological images Neurocomputing", 191 (2016)

[57] Lakhani, Paras, and Baskaran Sundaram. "Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks." Radiology 284.2 (2017): 574-582.

[58] Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B (Methodological), 39(1), 1-22.

[59] Pan, Sinno Jialin, and Qiang Yang. ”A survey on transfer learning.” IEEE Transactions on knowledge and data engineering 22.10 (2010): 1345-1359.

[60] B. Settles. Active Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Clay-pool Publishers, 2012.

[61] Ulf Brefeld, Tobias Scheffer ”Co-EM Support Vector Learning”

[62] Zhu, X. J. (2005). Semi-supervised learning literature survey. University of Wisconsin-Madison Department of Computer Sciences.

[63] Castelli, V., & Cover, T. M. (1996). The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. IEEE Transactions on information theory, 42(6), 2102-2117.

[64] Trajanovski, S., Mavroeidis, D., Swisher, C. L., Gebre, B. G., Veeling, B., Wiemker, R., ... & McKee, B. J. (2018). Towards radiologist-level cancer risk assessment in CT lung screening using deep learning. arXiv preprint arXiv:1804.01901.

[65] Mann, G. S., & McCallum, A. (2010). Generalized expectation criteria for semi-supervised learning with weakly labeled data. Journal of machine learning research, 11(Feb), 955-984.

[66] Ziang Yan, Jian Liang, Weishen Pan, Jin Li, Changshui Zhang ”Weakly- and Semi-Supervised Object Detection with Expectation-Maximization Algorithm”

[67] Hua, Kai-Lung, et al. ”Computer-aided classification of lung nodules on computed tomography images via deep learning technique.” OncoTargets and therapy 8 (2015).

[68] Zhao, B., Feng, J., Wu, X., & Yan, S. (2017). A survey on deep learning-based fine-grained object classification and semantic segmentation. International Journal of Automation and Computing, 14(2), 119-135.

[69] Ion Muslea, Steven Minton, Craig A. Knoblock ”Active + Semi-Supervised Learning = Robust Multi-View Learning”

[70] Ozdemir, Onur, Rebecca L. Russell, and Andrew A. Berlin. "A 3D Probabilistic Deep Learning System for Detection and Diagnosis of Lung Cancer Using Low-Dose CT Scans." arXiv preprint arXiv:1902.03233 (2019).

[71] McKinney, Scott Mayer, Marcin Sieniek, Varun Godbole, Jonathan Godwin, Natasha Antropova, Hutan Ashrafian, Trevor Back et al. "International evaluation of an AI system for breast cancer screening." Nature 577, no. 7788 (2020): 89-94.

[72] Cho, Junghwan, Kyewook Lee, Ellie Shin, Garry Choy, and Synho Do. "How much data is needed to train a medical image deep learning system to achieve necessary high accuracy?." arXiv preprint arXiv:1511.06348 (2015).

[73] Siegel, R.L., Miller, K.D. and Jemal, A. (2020), Cancer statistics, 2020. CA A Cancer J Clin, 70: 7-30. doi:10.3322/caac.21590

[74] Menon, S. , Chapman, D. , Nguyen, P. , Yesha, Y. , Morris, M. , Saboury, B . (2019) “Deep Expectation-Maximization for Semi-Supervised Lung Cancer Screening”, Proceedings of ACM SIGKDD 2019, Anchorage, Alaska.

[75] Lindenbaum, Michael, Shaul Markovitch, and Dmitry Rusakov. "Selective sampling for nearest neighbor classifiers." Machine learning 54, no. 2 (2004): 125-152.

[76] Mahapatra, Dwarikanath, Behzad Bozorgtabar, Jean-Philippe Thiran, and Mauricio Reyes. "Efficient active learning for image classification and segmentation using a sample selection and conditional generative adversarial network." In International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 580-588. Springer, Cham, 2018.

[77] Gal, Yarin, Riashat Islam, and Zoubin Ghahramani. "Deep bayesian active learning with image data." In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 1183-1192. JMLR. org, 2017.

[78] Beluch, William H., Tim Genewein, Andreas Nürnberger, and Jan M. Köhler. "The power of ensembles for active learning in image classification." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9368-9377. 2018.

[79] Tran, Toan, Thanh-Toan Do, Ian Reid, and Gustavo Carneiro. "Bayesian generative active deep learning." arXiv preprint arXiv:1904.11643 (2019).

[80] Padmanabhan, Raghav K., Vinay H. Somasundar, Sandra D. Griffith, Jianliang Zhu, Drew Samoyedny, Kay See Tan, Jiahao Hu et al. "An active learning approach for rapid characterization of endothelial cells in human tumors." PloS one 9, no. 3 (2014).

[81] Danziger, Samuel A., Roberta Baronio, Lydia Ho, Linda Hall, Kirsty Salmon, G. Wesley Hatfield, Peter Kaiser, and Richard H. Lathrop. "Predicting positive p53

cancer rescue regions using Most Informative Positive (MIP) active learning." PLoS computational biology 5, no. 9 (2009).

[82] Shi, Xueying, Qi Dou, Cheng Xue, Jing Qin, Hao Chen, and Pheng-Ann Heng. "An Active Learning Approach for Reducing Annotation Cost in Skin Lesion Analysis." In International Workshop on Machine Learning in Medical Imaging, pp. 628-636. Springer, Cham, 2019.

[83] Benitez, Matias, James Tian, Mark Kelly, Vignesh Selvakumaran, Matthew Phelan, Maciej Mazurowski, Joseph Y. Lo, Geoffrey D. Rubin, and Ricardo Henao. "Combining deep learning methods and human knowledge to identify abnormalities in computed tomography (CT) reports." In Medical Imaging 2019: Computer-Aided Diagnosis, vol. 10950, p. 109500V. International Society for Optics and Photonics, 2019.

[84] Causey, Jason L., Junyu Zhang, Shiqian Ma, Bo Jiang, Jake A. Qualls, David G. Politte, Fred Prior, Shuzhong Zhang, and Xiuzhen Huang. "Highly accurate model for prediction of lung nodule malignancy with CT scans." Scientific reports 8, no. 1 (2018): 1-12.

[85] McNitt-Gray, M. F., Armato III, S. G., Meyer, C. R., Reeves, A. P., McLennan, G., Pais, R. C., ... & Laderach, G. E. (2007). The Lung Image Database Consortium (LIDC) data collection process for nodule detection and annotation. Academic radiology, 14(12), 1464-1474.

[86] Armato III, S. G., McLennan, G., Bidaut, L., McNitt‐Gray, M. F., Meyer, C. R., Reeves, A. P., ... & Kazerooni, E. A. (2011). The lung image database consortium

(LIDC) and image database resource initiative (IDRI): a completed reference database of lung nodules on CT scans. Medical physics, 38(2), 915-931.

[87] He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 9729-9738).

[88] Oord, A. V. D., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.

[89] Henaff, O. (2020, November). Data-efficient image recognition with contrastive predictive coding. In International Conference on Machine Learning (pp. 4182-4192). PMLR.

[90] Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., & Bengio, Y. (2018). Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:1808.06670.

[91] Tian, Y., Krishnan, D., & Isola, P. (2020). Contrastive multiview coding. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16 (pp. 776-794). Springer International Publishing.

[92] Misra, I., & Maaten, L. V. D. (2020). Self-supervised learning of pretext-invariant representations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 6707-6717).

[93] Li, J., Zhou, P., Xiong, C., & Hoi, S. C. (2020). Prototypical contrastive learning of unsupervised representations. arXiv preprint arXiv:2005.04966.

[94] Hadsell, R., Chopra, S., & LeCun, Y. (2006, June). Dimensionality reduction by learning an invariant mapping. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) (Vol. 2, pp. 1735-1742). IEEE.

[95] Wang, X., & Gupta, A. (2015). Unsupervised learning of visual representations using videos. In Proceedings of the IEEE international conference on computer vision (pp. 2794-2802).

[96] Wu, Z., Xiong, Y., Yu, S. X., & Lin, D. (2018). Unsupervised feature learning via non-parametric instance discrimination. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3733-3742).

[97] Ouali, Y., Hudelot, C., & Tami, M. (2020). An overview of deep semi-supervised learning. arXiv preprint arXiv:2006.05278.

[98] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In Proceedings of the .33rd Ann& Meetzng of the Assoczation for Computational Linguistics, pages 189-196, 1995.

[99] McClosky, D., Charniak, E., and Johnson, M. Reranking and selftraining for parser adaptation. In ACL, pp. 337–344, 2006.

[100] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, “A survey on contrastive self-supervised learning,” Technologies, vol. 9, no. 1, p. 2, 2021.

[101] Le-Khac, Phuc H., Graham Healy, and Alan F. Smeaton. "Contrastive representation learning: A framework and review." *IEEE Access* 8 (2020): 193907-193934.

[102] J. D. Prusa and T. M. Khoshgoftaar, ‘‘Improving deep neural network design with new text data representations,’’ J. Big Data, vol. 4, no. 1, p. 7, Mar. 2017.