2019

Medical image classification under class imbalance

Chuanhai Zhang
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/etd
Part of the Artificial Intelligence and Robotics Commons

Recommended Citation
Zhang, Chuanhai, "Medical image classification under class imbalance" (2019). Graduate Theses and Dissertations. 17130.
https://lib.dr.iastate.edu/etd/17130

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Medical image classification under class imbalance

by

Chuanhai Zhang

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Computer Science

Program of Study Committee:
Carl K. Chang, Major Professor
Wallapak Tavanapong
Johnny S. Wong
Yan-Bin Jia
Ying Cai

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation. The Graduate College will ensure this dissertation is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University
Ames, Iowa
2019

Copyright © Chuanhai Zhang, 2019. All rights reserved.
# TABLE OF CONTENTS

| LIST OF FIGURES | .......................................................... | v  |
| LIST OF TABLES | .......................................................... | viii  |
| ACKNOWLEDGMENTS | .................................................................. | ix  |
| ABSTRACT | .................................................................. | x  |

## CHAPTER 1. INTRODUCTION .............................................................................. 1

1.1. Background and Motivation ...................................................................... 1  
1.2. Contributions of Dissertation .................................................................. 3  
1.3. Content Guide ......................................................................................... 7  

## CHAPTER 2. RELATED WORK ........................................................................... 9

2.1. Methods for Handling Small Training Datasets ......................................... 9  
2.2. Class Imbalance ....................................................................................... 12  
2.3. Instrument Scene Detection in Endoscopic Procedures ........................... 16  

## CHAPTER 3. PROPOSED REAL DATA AUGMENTATION METHODS .............. 19

3.1. Introduction ............................................................................................. 19  
3.2. Data Augmentation Based on Probability (CNN + Probability) ............... 19  
3.3. Data Augmentation Based on Distance Function Learning ....................... 20  
3.4. Data Augmentation Using Triplet-based Methods .................................. 21  

3.4.1. Data Augmentation Based on Feature Learning (Triplet+L2) .............. 21  
3.4.2. Unified Learning of Feature Representation and Similarity Matrix ....... 22  
3.5. Experimental Environment and Results .................................................. 23  

3.5.1. Experimental Datasets .......................................................................... 24  
3.5.2. Performance Metrics ............................................................................ 27  
3.5.3. Experimental Parameters ...................................................................... 28  
3.5.4. Experimental Results .......................................................................... 30  
3.6. Chapter Summary .................................................................................... 37
CHAPTER 4. INSTRUMENT SCENE DETECTION IN ENDOSCOPY VIDEOS ..... 38
4.1. Introduction .................................................................................................................................................. 38
4.2. Endoscopy Image Classification Using Cable Footprint History Technique ........................................... 41
  4.2.1. Image Preprocessing ............................................................................................................................... 41
  4.2.2. Spatial Feature Extraction and Classification ......................................................................................... 42
  4.2.3. Cable Footprint History ......................................................................................................................... 44
4.3. Endoscopy Image Classification Using EndoCNN Technique .............................................................. 45
  4.3.1. Manual Creation of Training Datasets .................................................................................................. 46
  4.3.2. Classification of Endoscopy Images Using CNN ............................................................................... 47
  4.3.3. Hierarchical and Unified Data Augmentation (HuDA) ...................................................................... 48
  4.3.4. Data Augmentation Methods .............................................................................................................. 50
4.4. Instrument Scene Generation .................................................................................................................... 52
  4.4.1. Eliminate falsely detected cable images ............................................................................................. 53
  4.4.2. Locate Cable Scene Boundaries ........................................................................................................ 53
4.5. Experimental Environment and Results .................................................................................................. 54
  4.5.1. Experimental Datasets ......................................................................................................................... 54
  4.5.2. Model Parameters ............................................................................................................................... 56
  4.5.3. Performance Metrics .......................................................................................................................... 57
  4.5.4. Experimental Results ........................................................................................................................ 57
4.6. Chapter Summary ....................................................................................................................................... 62

CHAPTER 5. ACTIVE DEEP LEARNING FOR IMAGE CLASSIFICATION UNDER CLASS IMBALANCE .................................................................................................................. 64
5.1. Introduction ................................................................................................................................................. 64
5.2. Feature Representation and Similarity Learning ....................................................................................... 65
  5.2.1. Triplet Loss .......................................................................................................................................... 65
  5.2.2. Triplet Selection ................................................................................................................................. 67
5.3. Unlabeled Sample Selection ...................................................................................................................... 67
  5.3.1. Rare Class Sample Selection and Labeling .......................................................................................... 67
  5.3.2. Dissimilar Sample Selection and Labeling .......................................................................................... 68
  5.3.3. Pseudo-labeling of High Confidence Common Class Samples ......................................................... 68
5.4. Experimental Environment and Results .................................................................................................. 70
  5.4.1. Experimental Datasets and Settings ................................................................................................... 70
5.4.2. Performance Metrics ................................................................................................. 74
5.4.3. Experimental Results .............................................................................................. 74
5.5. Chapter Summary ......................................................................................................... 78

CHAPTER 6. CONCLUSION .................................................................................................. 79

REFERENCES ....................................................................................................................... 81
# LIST OF FIGURES

| Figure 3. 1 | Algorithms of CNN+Probability and CNN+Bilinear. | 20 |
| Figure 3. 2 | Two triplet-based recommendation algorithms. | 22 |
| Figure 3. 3 | The model consists of a batch input layer to a CNN followed by L2 normalization, which results in the embedding using the triplet loss based on Bi-linear distance. | 23 |
| Figure 3. 4 | Sample images for the six classes. (a) - (f): left cable body, right cable body, forceps head, snare head, retroflexion, and no object. | 24 |
| Figure 3. 5 | Sensitivity study on classification effectiveness improvement over not using any data augmentation. | 33 |
| Figure 3. 6 | Recommendation effectiveness of different real data augmentation methods. | 36 |
| Figure 4. 1 | Four types of instrument frames. (a) left cable body, (b) right cable. | 39 |
| Figure 4. 2 | Eight non-instrument types: (a) thick fold, (b) bright boundary, (c) thin snare-like fold, (d) protruding polyp, (e) picture-in-picture, (f) retroflexion, (g) blurry frame, (h) informative frame. | 39 |
| Figure 4. 3 | The hierarchical class structure of endoscopy images. | 40 |
| Figure 4. 4 | (a) 12 general insertion areas; (b)-(d) new Cartesian coordinates; (e) insertion direction calculation. | 42 |
| Figure 4. 5 | (a)-(d) Cable regions in one colonoscopy video; (e) cable footprint history of this video; (f) detected cable region. | 44 |
Figure 4.6 The EndoCNN structure. 16@64x64 represents 16 feature maps with a size of 64x64 as the output of the first convolutional layer. .......................... 47

Figure 4.7 HuDA algorithm ...................................................................................................................... 50

Figure 4.8 Sensitivity analysis of the color contrast threshold values tested on cable images in the image set. .......................................................................................................................... 56

Figure 4.9 Classification results of the EndoCNN Manual model. The first row shows success cases. The second row shows instrument images missed by EndoCNN. .................................................................................................................... 62

Figure 5.1 Illustration of our proposed SAL framework. SAL progressively selects samples from the unlabeled dataset for either manual labeling or pseudo-labeling. Then the CNN classifier and the Unified LF&SM similarity model are updated using the newly expanded training dataset. .................................................................................................................. 64

Figure 5.2 SAL algorithm ........................................................................................................................... 69

Figure 5.3 The first and second line: sample images with/without forceps head from the Endoscopy dataset. The third and last line: samples images with/without air plane from the Caltech-256 Dataset. ........................................................................... 71

Figure 5.4 Classification performance under different iterations on (a-c) Endoscopy and (d-f) Caltech-256 datasets. Our proposed method SAL performs consistently better than CEAL and AIFT. ........................................... 75

Figure 5.5 The first line: dissimilar images selected by SAL from the Endoscopy dataset. The second line: dissimilar images selected by SAL from the Caltech-256 Dataset. ........................................................................................................ 77

Figure 5.6 Classification performances under different iterations for two variants of SAL on Endoscopy (a-c) and Caltech-256 (d-f) datasets. SAL_Conf
uses pseudo-labeling of high confidence common class images,
SAL_noConf does not use any high confidence common class image.
# LIST OF TABLES

| Table 3. 1 | Seed training sets with different sizes or varieties or similarities | 26 |
| Table 3. 2 | Our CNN structure. The input and output sizes are described by | 30 |
| Table 4. 1 | The training and test datasets for image classification | 54 |
| Table 4. 2 | Manual effort required to create a large training dataset beyond collecting the seed training dataset of 25,000 images | 55 |
| Table 4. 3 | Parameters and values used in experiments | 55 |
| Table 4. 4 | Effectiveness of different instrument image classification models on the balanced test set. I represents an instrument frame or class; NI represents a non-instrument frame or class. | 58 |
| Table 4. 5 | Effectiveness of different instrument image classification models on the imbalanced test set. I represents an instrument frame or class; NI represents a non-instrument frame or class. | 58 |
| Table 4.6 | Performance of different scene detection methods, # GT represents the number of operation scenes in the ground truth. | 61 |
| Table 5. 1 | Endoscopy dataset | 70 |
| Table 5. 2 | Caltech-256 dataset | 70 |
| Table 5. 3 | Our network architecture excluding soft-max layer | 72 |
| Table 5. 4 | Rare class recommendation accuracy for the Endoscopy dataset | 76 |
| Table 5. 5 | Rare class recommendation accuracy for the Caltech-256 dataset | 76 |
ACKNOWLEDGMENTS

I am heartily thankful to my research advisor Dr. Wallapak Tavanapong. This thesis would not have been finished without her excellent guidance and support. She taught me how to think deeper in a research topic, how to come up with novel research ideas, how to write a paper with good quality, and many aspects that would influence my future.

I would like to thank my committee chair, Dr. Carl K. Chang, and my committee members, Dr. Johnny S. Wong, Dr. Ying Cai, and Dr. Yan-Bin Jia, for their excellent guidance, patience and support throughout the course of this research.

I would like to thank Prof. Piet de Groen and his research team at Mayo Clinic Rochester for providing colonoscopy video data, for his help on conference and journal paper revision. I would like to thank Prof. JungHwan Oh and his research group at the University of North Texas for his help on paper revision.

I would also like to express many thanks to my fellow graduate students in our research lab: Lei Qi, and Gavin Kijkul for their great help on this research.

I would like to express my deepest gratitude to my wife, Yan Su, who always helps me, encourages me during my PhD study. I would like to express my special thanks to my parents. My parents have been supporting me for 29 years so that I can keep studying without distraction in an affluent environment.

This research is supported in part by a grant from the NIH (Grant #1R01DK106130-01A1) and EndoMetric Corporation. Opinions and findings in this paper are those of the authors and do not represent those of the funding agencies. This thesis is dedicated to my father Linwang Zhang, my mather Fuhua Li, and my wife Yan Su.
ABSTRACT

Many medical image classification tasks have a severe class imbalance problem. That is images of target classes of interest, e.g., certain types of diseases, only appear in a very small portion of the entire dataset. These medical image classification tasks share two common issues. First, only a small labeled training set is available due to the expensive manual labeling by highly skilled medical experts. Second, there exists a high imbalance ratio between rare class and common class. The common class occupies a high percentage of the entire dataset and usually has a large sample variety, which makes it difficult to collect a good representative training set for the common class. Convolutional Neural Network (CNN) is currently a state-of-the-art method for image classification. CNN relies on a large training dataset to achieve high classification performance. However, manual labeling is costly and may not even be feasible, which limits CNN from offering high classification performance in practice. This dissertation addresses these two challenging issues with the ultimate goal to improve classification effectiveness and minimize manual labeling effort by the domain experts.

The main contributions of dissertation are summarized as follows. 1) We propose a new real data augmentation method called Unified LF&SM that jointly learns feature representation and a similarity matrix for recommending unlabeled images for the domain experts to verify in order to quickly expand the small labeled training set. Real data augmentation utilizes realistic unlabeled samples rather than synthetic samples. The key of real data augmentation is how to design an effective strategy to select representative samples for certain classes quickly from a large realistic unlabeled dataset. 2) We investigate the effectiveness of six different data augmentation methods and perform a
sensitivity study using training sets of different sizes, varieties, and similarities when compared with the test set. 3) We propose a Hierarchical and Unified Data Augmentation (HuDA) method to collect a large representative training dataset for the common class. HuDA incorporates a class hierarchy: class differences on the high level (between the rare class and the common class) and class differences on the low level (between sub-classes of the rare class or the common class). HuDA is capable of significantly reducing time-consuming manual effort while achieving quite similar classification effectiveness as manual selection. 4) We propose a similarity-based active deep learning framework (SAL), which is the first approach to deal with both a significant class imbalance and a small seed training set as far as we know.

Broader Impact: Triplet-based real data augmentation methods utilize the similarity between samples to learn a better feature representation. These methods aim to guarantee that the computed similarity between two samples from the same class is always bigger than the computed similarity between two samples from two different classes. First, our sensitivity study on six different data augmentation methods shows that triplet-based real data augmentation methods always offer the largest improvement on both the recommendation accuracy and the classification performance. These real data augmentation methods are easily extendable to other medical image classification tasks. Our work provides useful insight into how to choose a good training image dataset for medical image classification tasks. Second, to the best of our knowledge, SAL is the first active deep learning framework that deals with a significant class imbalance. Our experiments show that SAL nearly obtains the upper bound classification performance by labeling only 5.6% and 7.5% of all images for the Endoscopy dataset and the Caltech-256
dataset, respectively. This finding confirms that SAL significantly reduces the experts’ manual labeling efforts while achieving near optimal classification performance. SAL works for multi-class image classification and is easily extendable to other medical image classification tasks as well.
CHAPTER 1. INTRODUCTION

In this chapter, first we will introduce the background and motivation of our research. Next, we will summarize the main contributions of this dissertation. Finally, we will give a content guide about the organization of this dissertation.

1.1. Background and Motivation

Many medical image classification tasks have a severe class imbalance problem. That is images of target classes of interest, e.g., certain types of diseases, only appear in a very small portion of the entire dataset. These medical image classification tasks share two common issues. First, only a small labeled training set is available due to the expensive manual labeling by highly skilled medical experts. Second, there exist a high imbalance ratio between rare class and common class. The rare class only occupies a small percentage of the entire dataset. The common class occupies a high percentage of the entire dataset. What’s worse, the common class usually has a large sample variety, which makes it more difficult and time consuming to collect a good representative training set for the common class.

Convolutional Neural Network (CNN) is a feed-forward net which consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of successive pairs of convolutional and pooling layers, followed by several fully connected layers. CNN was first introduced in 1980 [1] and around 1998 achieved great success on handwritten digit recognition [2]. Recently, CNN has been applied successfully in ImageNet classification [3] (14,197,122 general images, 27 high-level categories, and 21,338 sub-categories) and has shown promise in medical imaging applications [4-23]. Indeed, CNN is currently a state-of-the-art method for image
classification. However, CNN relies on a large training dataset to achieve high classification accuracy. Nowadays, large collections of medical images are readily available. However, it is costly and may not even be feasible for medical experts to manually inspect huge unlabeled datasets to obtain enough representative examples of the rare classes of abnormal images representing a specific disease or intervention, which prevents CNN from achieving high classification accuracy in practice.

In general, there are three categories of methods to handle class imbalance. They are sampling based methods [24-36], cost-sensitive training [37-44], and active learning [45-58]. Sampling based methods aim to build a balanced dataset by considering the representative proportions of class examples in the distribution. These methods do not contribute to increase the number of realistic training samples. Instead of creating balanced dataset through different sampling strategies, cost-sensitive learning solves the imbalanced learning problem by using different cost matrices that describe the costs for misclassifying examples of different classes. Recall that CNN needs a large training set to get high accuracy. However, neither sampling based methods nor cost-sensitive training methods utilize the large unlabeled image set and contribute to create a large representative training set. Therefore, they are not sufficient to train a CNN classifier with high accuracy for handling data variety in practice.

Active learning [45] is a category of methods [45-58] which deal with a small training dataset. Active learning aims to minimize efforts of the domain expert in labeling the data by using a query strategy to choose necessary samples typically from an unlabeled dataset or from a synthesis of labeled samples. Active learning is an iterative solution. In each iteration, new unlabeled samples are selected and labeled by experts, and then a better
classifier is learned from the new expanded training dataset. Active learning has huge potentials to improve the classification accuracy of CNN.

To the best of our knowledge, there is no existing work on active deep learning for medical image classification under class imbalance. We aim at designing active deep learning algorithms which minimize the expensive labeling time while achieving high image classification accuracy using CNN. To design an effective active deep learning framework, First we design and investigate four real data augmentation methods to quickly expand the small labeled training dataset, especially for the rare class. Then, we utilize the best real data augmentation method to design three different recommendation strategies, which select unlabeled images from a large unlabeled image set. Finally, we design the first active deep learning framework to deal with both a significant class imbalance and a small labeled seed training set for image classification.

1.2. Contributions of Dissertation

1. We propose a new a new real data augmentation method called Unified LF&SM [59] to quickly expand the small labeled training dataset. Unified LF&SM is a triplet-based real data augmentation method, which utilize the similarity between samples to learn a better feature representation. Triplet-based methods aim to guarantee that the computed similarity between two samples from the same class is always bigger than the computed similarity between two samples from two different classes. We explore six different data augmentation methods: four RDA (real data augmentation) methods [59] and two SDA (synthetic data augmentation) methods: traditional data augmentation [34-36] and TANDA [60]. SDA methods use different strategies to synthesize images from original labeled realistic images. We carefully design six different training
datasets that have different numbers of training images (sizes), different image appearances (variety) in each training set, and different similarity scores to the test set. We perform a sensitivity study to determine the impact on the classification effectiveness due to the sizes, the varieties within the training data, and the similarity of the training images with those in the test dataset. This study thus aims to identify and confirm the drawback of each augmentation method. To the best of our knowledge, no existing research team has done a similar study. The most important findings are as follows.

(1) When the training dataset has low sample variety or is not very similar to the test set, traditional augmentation contributes little to improve classification effectiveness.

(2) Our triplet-based methods offer the most improvement among the compared methods, but have the tradeoff of manual verification of 5,000 images in our experiments. The traditional augmentation and the triplet-based methods may be used in combination to further reduce manual labeling effort.

(3) Not all RDA methods are better than traditional data augmentation. The methods that rely on CNN predicted probability (described in Section 3.2) do not provide much improvement when the training set size is small. Under this condition, the triplet-based methods learn better feature representation than using a CNN classifier without the triplet-based methods.

Our work provides useful insight into how to choose a good training image dataset for medical image classification tasks.
2. We propose two different approaches for the instrument scene detection task in endoscopic procedures: Cable Footprint [61] and EndoCNN [62]. EndoCNN outperform the state-of-the-art method for instrument scene detection on both the detection accuracy and the processing time. Viewing instrument and NI classes and corresponding sub-classes as a class hierarchy, we also propose a novel Hierarchical Unified Data Augmentation (HuDA) method to quickly collect a large representative image set for the common class. The class hierarchy is useful to learn a good feature representation which differentiate not only high-level classes (the instrument class and the NI class), but also low-level classes (subclasses of the instrument class or the NI class). Hence, the class hierarchy is very helpful to collect a representative image set for each subclass of the NI class. HuDA is generalizable to any medical image classification problem. The class hierarchy can be defined based on the problem domain. We train several EndoCNN models with different training datasets created using different data collection methods. We evaluate these EndoCNN models on both balanced test image set and unbalanced test image set to assess which data collection method makes EndoCNN most effective for practical use. In our experiments, EndoCNN trained using the training dataset expanded by HuDA offers the best average F1-score of 97.24\% on the balanced test set and the third best average F1-score of 92.77\% on the imbalanced test set, which are only about 0.8\% below those of the best EndoCNN (trained using a large training dataset collected manually). However, HuDA only costs 0.31\% fraction of manual time required to label a large training dataset manually. Hence, HuDA
is capable of significantly reducing time-consuming manual effort with slight loss in classification effectiveness.

3. We propose a novel similarity-based active deep learning framework (SAL) [63] that deals with class imbalance. To the best of our knowledge, SAL is the first active deep learning method for image classification under class imbalance. SAL consists of four key components. First, SAL uses a similarity-based loss function in learning both feature representation and a similarity function jointly. Second, SAL uses the learned information to recommend more rare class samples effectively and find atypical, not previously encountered common class images. Since existing active learning methods do not consider class imbalance, more common class samples, rather than rare class samples, tend to be selected for manual labeling. Also, existing active techniques do not consider similarity as in their query strategy. Third, SAL recommends high-confidence common class samples for automatic pseudo-labeling without experts’ labeling efforts. These samples are only used for training the CNN classifier, but they are not included in the training dataset for learning the similarity model. This is to ensure that all labeled samples in the training dataset for learning the similarity model have the correct labels. Finally, SAL uses a new stopping criterion based on the rare class recommendation accuracy.

We use three metrics for evaluating classification performance (average Recall, Precision, and F1-score [97]) of SAL against those of two recent active deep learning methods [77][79] on two challenging image datasets: the Endoscopy dataset and the Caltech-256 dataset [64]. These methods are not
designed to handle a significant class imbalance. Our experiments show that
SAL consistently outperforms the two methods on both datasets. To obtain the
upper-bound of the classification performance, a CNN is trained on the largest
training dataset. Our experiments show that SAL nearly obtains the upper
bound classification performance by labeling only 5.6% and 7.5% of all images
for the Endoscopy dataset and the Caltech-256 dataset, respectively. This
finding confirms that SAL significantly reduces the experts’ manual labeling
efforts while still achieving near upper bound performance.

1.3. Content Guide

In Chapter 2, we summarize related works dealing with small labeled training
datasets and class imbalance. Also, we describe the related works on instrument scene
detection in endoscopic procedures, which is a specific unbalanced medical image
classification task. In Chapter 3, we present our investigation of the effectiveness of six
different types of data augmentation methods to quickly expand the small labeled training
dataset (especially for the rare class) and improve the image classification effectiveness of
CNN. Also, we present the result of a sensitivity study of different data augmentation
methods using training sets of different sizes, and differing in varieties, and similarities
when compared with the test set. In Chapter 4, we describe two different approaches to
solve the problem of instrument scene detection in endoscopic procedures. We also propose
different types of methods to collect a large representative training dataset for the common
class. Especially, we design a novel Hierarchical Unified Data Augmentation (HuDA)
method to expand the image set of the common class. HuDA incorporates a class hierarchy
to learn a better similarity model. In Chapter 5, we propose the first active deep learning
framework (SAL), which deals with small labeled training dataset and significant class
imbalance simultaneously. We provide conclusions and descriptions of future work in Chapter 6.
CHAPTER 2. RELATED WORK

In this chapter, we introduce all related works of our research. In Section 2.1, we introduce related works for handling small training datasets. Next, we introduce related works for dealing with class imbalance in Section 2.2. Then, we introduce related works on instrument scene detection in Endoscopic procedures in Section 2.3.

2.1. Methods for Handling Small Training Datasets

Synthetic Image Augmentation

Deep neural networks usually have millions or billions of free parameters and require massive labeled data sets for training. In most cases, labeled data are far away from enough to avoid overfitting the trained classifier to the training set. Data augmentation is a technique that artificially expands labeled training sets by leveraging task-specific data transformations that preserve class labels. Data augmentation has quickly become a critical and effective tool for combating this labeled data scarcity problem.

Krizhevsky et al. [34] apply various transformations to generate more data from existing data. These transformations include random translations, rotations, flips, and addition of Gaussian noise. Each synthesized image from these transformations preserves the same class label of the original image. Ratner et al. [60] propose a method to automatically compose user-defined transformation functions (TFs) for data augmentation. First, they train a generative adversarial model using a set of TFs and unlabeled data from users. The generative adversarial model consists of a null class discriminator, D, which classifies between real images and synthesized images and a generator, G, which produces TF sequences. Then, they apply the learned TF sequences to each labeled image to synthesize images. Each synthesized image shares the same label as their corresponding
original image. Tran et al. [65] propose a Bayesian data augmentation algorithm, called Generalized Monte Carlo Expectation Maximization (GMCEM). GMCEM iteratively learns the training data distribution. GMCEM samples new synthetic training points from a previously learned training data distribution at each iteration. Then, GMCEM uses Monte Carlo to estimate the expected value of the network parameters. Antoniou et al. [66] propose a Data Augmentation Generative Adversarial Networks (DAGAN). DAGAN consists of a generator neural network and an adversarial discriminator neural network. The generator neural network consists of an encoder and a decoder. The encoder takes an input image and projects it down to a lower dimensional manifold term “bottleneck” vector. Next, a random vector is transformed and concatenated with the bottleneck vector. Then, the decoder network accepts the concatenated vector as input to generate an augmented image. The adversarial discriminator neural network is trained to discriminate between the samples from the real distribution (real images from the same class) and the fake distribution (automatically generated images). Finally, DAGAN generates new synthesized images and assigns the label of the original image to these synthesized images. These synthesized images look different enough to be a different sample compared with the original image.

**Semi-supervised learning for image classification**

Semi-supervised learning (SSL) methods typically make use of a small amount of labeled data with a large amount of unlabeled data. Many research works [67-72] have shown that unlabeled data, when used in conjunction with a small amount of labeled data, can produce considerable improvement in learning accuracy. Guillaumin et al. [67] propose a multimodal SSL method for image classification. They assume that the training data have
both image content and tag information, but the test data has only image content information. First, the labeled images are used to learn a multiple kernel learning (MKL) classifier that uses both the image content and tags as features. Next, the MKL classifier is used to predict the labels of unlabeled training images with associated tags. Second, both the labeled data and the output of the classifier on unlabeled data are used to learn a second classifier that uses only visual features as input. Zhu et al. [68] propose a multi-view SSL framework for image classification. During training, labeled images are used to train view-specific classifiers independently, and each view-specific classifier is then iteratively retrained with respect to a measure of confidence using initial labeled samples and additional pseudo-labeled samples. When testing, the maximum entropy principle is utilized to assign appropriate category labels to unlabeled images via optimally trained view-specific classifiers.

Blum et al. [69] propose a Co-Training framework for SSL. Co-Training assumes that each data instance has two different views and each view is sufficient for learning an effective classifier. Co-Training also assumes that two classifiers trained on the two views respectively have similar predictions on most unlabeled images. Based on these assumptions, Co-Training proposes a double view self-training algorithm. First, it learns a separate classifier for each view on the labeled image set. Second, the predictions of the two classifiers on the unlabeled image set are gradually added to the labeled image set to continue the training. Qiao et al. [70] propose Deep Co-Training for semi-supervised image recognition. Deep Co-Training extends the Co-Training framework by training deep neural network classifiers and adding a new view difference constraint. To prevent two neural networks from collapsing into each other, they add the view difference constraint by
training each network to be resistant to the adversarial examples of the other. As a result, each neural network can keep its predictions unaffected on the examples that the other network fails. Therefore, the two networks provide complementary information about the data because they are trained not to make errors at the same time on the adversarial examples for them. Peikari et al. [71] propose a cluster-then-label SSL approach for Pathology image classification. First, a clustering method is applied on extracted features to find the underlying structure of Pathology images (clusters of points forming high density regions). Second, a standard supervised SVM is applied to find the decision boundary using knowledge about the underlying structure of the images.

Wu et al. [72] propose a semi-supervised image classification method with self-paced cross-task networks. First, they select training samples with weights and create pairwise constraints of unlabeled images to train a cross-task network. The cross-task network does image classification and clustering simultaneously. Next, they predict class labels of unlabeled images using the trained cross-task networks. Then, they update the weights of unlabeled images and retrain the cross-task network.

2.2. Class Imbalance

In general, there exist three categories of methods to handle class imbalance as shown in Table I. They are sampling based methods [24-35], cost-sensitive training [36-44], and active learning [45-58]. Sampling based methods consist of random over sampling of rare classes [24], under sampling of a common class [24-28], and over sampling of a rare class using synthetic data augmentation [29-36]. The synthetic data augmentation for rare classes creates variations of existed samples of rare classes. Existing synthetic data augmentation methods are either on the feature level [29-33] or on the image (raw data) level [34-36].
Feature-level data augmentation assumes the feature vector of a sample is given and creates artificial data based on the feature space similarities between existing rare class examples. The most representative research work on feature-level data augmentation is SMOTE [29]. SMOTE concentrates on the characteristics of the minority class to guide the oversampling process. In the case of extreme imbalance, the few samples in the minority class offer minimal distributional information, which may be misleading due to rarity and noise. As a result, this may generate misleading synthetic training instances which harm the classifier. Recently, Sharma et al. [32] propose a new method called SWIM (Sampling WIth the Majority), which uses the rich distribution information inherent in the majority class to synthesize minority class samples. SWIM generates synthetic samples that are at the same Mahalanbois distance from the majority class as the known minority instances. SWIM does not require any knowledge about the distribution of the minority class.

Recently, Liu et al. [33] propose a new method called DFBS (discriminative feature-based sampling). Different from SMOTE (which only focuses on the minority class) and SWIM (which only focuses on the majority class), DFBS considers both majority classes and minority classes to learn feature embedding. First, DFBS utilizes a triplet loss based deep feature embedding model to learn a discriminative feature representation. Second, DFBS extracts feature vectors of the samples in the minority class using the learned feature extractor at the first step. Then DFBS utilizes a random combination method to generate synthetic samples by sampling feature vectors of the minority class. Lastly, DFBS performs verification on all synthetic samples to guarantee that they are close to the minority class and not be confused with the majority class.
Image-level data augmentation applies image processing operators such as shearing, shifting, and rotation directly on images in the labeled dataset to oversample the classes with fewer labeled samples. Sampling based methods aim to build a balanced dataset by considering the representative proportions of class examples in the distribution. These methods do not increase the number of real training samples.

Instead of creating balanced dataset through different sampling strategies, cost-sensitive learning solves the imbalanced learning problem by using different cost matrices that describe the costs for misclassifying examples of different classes. Recall that CNN needs a large training set to get high accuracy. However, neither sampling based methods nor cost-sensitive training methods utilize the large unlabeled image set to create a large representative training set. Therefore, they are not sufficient to train a CNN classifier with high accuracy in practice.

Stefanowski [98] describes several important factors which may have an impact on the performance of classifiers under class imbalance. These factors are the number of training samples, the imbalance ratio, overlapping between classes, noisy examples located far away from the decision boundary (deeper inside the distribution of the opposite class), and small disjuncts (decomposition of the classes into smaller sub-parts including too few examples). Stefanowski [98] creates several artificial data sets to experimentally check which of these factors are more critical for the classification performance. Stefanowski [98] concludes that (1) the small number of samples in the minority class is not the main source of difficulty for classification; (2) the degradation of classification performance is more related to other critical factors such as the presence of small disjuncts, overlapping between classes, and noisy examples located far away from the decision boundary.
However, it is noticeable that all synthetic data sets used in [98] are very simple and the features are known in advance. These conclusions may not be applicable to image classification under class imbalance, for which good feature representation is important, but more likely unknown in advance. In Chapters 3-5, we will show that (1) more number of rare class samples significantly improve the image classification performance since better image features can be learned using CNN; (2) sub-classes of both the rare class and the common class are very critical for the image classification performance; (3) the imbalance ratio between the rare class and the common class has a significant impact on the classification performance.

Active learning [45] is a category of methods [45-58] which deal with a small training dataset. Active learning aims to minimize efforts of the domain expert in labeling the data by using a query strategy to choose necessary samples typically from an unlabeled dataset or from a synthesis of labeled samples. Active learning is an iterative solution. In each iteration, a classifier is learned from the current training dataset in that iteration and new unlabeled samples are selected. Several query strategies have been explored (e.g., selecting samples at the margin border separating different classes [58]).

Attenberg and Provost show that traditional active learning is ineffective for extreme class imbalance [73]. They then propose a hybrid of active learning and guide learning where the human is tasked with searching desirable unlabeled examples using any existing tools. Other forms of interactions with the human oracle are considered [74]. Class Conditional Query (CCQ) [74] algorithm proposes a label and a subset of the unlabeled examples, then asks the human expert to choose from these examples the ones that agree with the proposed label. SEARCH [75], another recent active learning method, tasks the
human expert to find from the entire unlabeled dataset, counterexamples to all the hypotheses considered by the algorithm, when human labeling of the selected examples does not improve the classifier further. SEARCH and CCQ although dealing with class imbalance do not learn feature representation of the data.

To the best of our knowledge, all these existing active learning methods that deals with class imbalance assume that the feature representation of a training sample is known in advance. However, it is nearly impossible to get good feature representation of a medical image from few labeled images. Therefore, these methods are not extendable to medical image classification.

The research problem we address is medical image classification with single label under significant class imbalance without known feature representation in advance. To the best of our knowledge, there is no similar work that addresses this problem. The existing closest works [76-78] use active deep learning to learn feature representation and selects unlabeled images for the human oracle to label. However, they are not designed to address class imbalance. Some handle class imbalance using traditional data augmentation to balance the seed training dataset prior to selection of unlabeled sample images for labeling [79].

2.3. Instrument Scene Detection in Endoscopic Procedures

Colonoscopy is currently the gold standard for colorectal cancer screening. Upper Endoscopy (EGD) is the procedure for inspection of the stomach. In the US, Colorectal cancer is the second leading cause of cancer-related deaths behind lung cancer [80], causing about 50,000 annual deaths. Colorectal cancer and stomach cancer are the third and the fifth most common cancer in the world [81]. During the insertion phase of an endoscopic procedure, a flexible endoscope with a video camera at the tip is advanced
under direct vision via the anus for colonoscopy and via the mouth for EGD. The video camera generates video of the internal mucosa of the organ. During the withdrawal phase, the endoscope is gradually withdrawn with careful examination of the mucosa and necessary diagnostic or therapeutic operations are performed.

An instrument scene or operation scene is defined as a video segment corresponding to a single purpose diagnostic or therapeutic action [61]. One scene may consist of one or more operation shots such as several biopsy shots taken in close proximity in the colon. Automatic operation scene detection is useful for 1) post-procedure documentation and review for causes of complications due to operations; 2) deriving real-time objective quality metrics such as withdrawal time without operations; 3) quality assessment; and 4) building a content-based retrieval system for endoscopic research and education. We map the problem of detecting operation scenes to the problem of identifying instruments used in biopsy or therapeutic operations since the operations cannot be performed without these instruments.

We limit the discussion to the related recent work in endoscopy video analysis. Bouget et al. [82] present a review on vision-based and marker-less surgical tool detection and tracking, which summarized all related works from the year of 2000 to 2015. Ye et al. [83] propose a method to detect endoscopic scenes based on tracking and detection of visual landmarks on the tissue surface. None of them are specifically designed to detect operation scenes in colonoscopy or EGD videos.

Cao et al. [84] introduce algorithms for detection of operation shots based on image segmentation. Moment invariants and Fourier shape descriptors are used in [84] and the earlier work for matching the detected regions with the cable body template regions. The
reported average true positive ratio and false positive ratio are 0.94 and 0.10. The image segmentation step is slow, making the method unsuitable for real-time application. The methods are not designed for grouping operation shots for the same purpose (e.g., biopsies) in close proximity into the same operation scene.
CHAPTER 3. PROPOSED REAL DATA AUGMENTATION METHODS

In this chapter, we introduce our proposed real data augmentation methods. First, we briefly introduce the research problem. Second, we describe our proposed real data augmentation methods in from Section 3.2 to Section 3.4. Third, we discuss our experimental results in Section 3.5. Finally, we give a summary of this chapter in Section 3.6.

3.1. Introduction

Let $T$ be our labeled seed training image set, $|C|$ be the number of classes for the classification problem, and $N_j$ be the number of images in $T$ belonging to a class $j$. Let $U$ be an unlabeled dataset with a cardinality of $|U|$. Our goal is to recommend the set $R_j$ of $k$ most relevant unlabeled images from $U$ for each class $j$. We use CNN as our supervised deep learning classification algorithm. We investigate the simplest recommendation algorithm, recommending the $k$ most similar images for each class to improve the robustness of CNN. The higher the value of $k$ is, the more likely is a larger variation in the recommended examples. Note that even when very similar images are recommended, they are still useful since the images are from different videos not seen in the training set.

3.2. Data Augmentation Based on Probability (CNN + Probability)

After training a CNN classifier on $T$, we apply the classifier on each image $I_i$ in $U$ and obtain $p(i, j)$ indicating the probability of the image $I_i$ belonging to a class $j$ using the soft-max function at the last layer of the CNN. Figure 3.1 shows the recommendation algorithm. For each class $j$, we sort all unlabeled images in descending order based on their $p(i, j)$ (Figure 3.1, Lines 14-15) and select the top $k$ most similar images to recommend.
3.3. Data Augmentation Based on Distance Function Learning (CNN + Bilinear)

We train a CNN classifier on the training dataset $T$. Then we extract feature representation $v_i$ for the image $I_i$ using the trained CNN. Next, we apply OASIS [85] to learn the squared matrix $W$ used in the bilinear similarity function $S_W(v_i, v_j)$ in Equation (3.1) that assigns higher similarity scores to images in the same class. Figure 3.1 shows our method. For each class $j$, we sort all unlabeled images in descending order based on their similarity scores (Figure 3.1, Lines 12-13) and recommend $k$ most similar images to the class representative.

Input: Seed training set $T$, recommendation number $k$, unlabeled set $U$, algorithm name

Algorithm:
1. Initialization: $R_j \leftarrow \emptyset$, $j = 1, 2, ..., |\mathcal{C}|$
2. Train a CNN classifier $M$ on $T$
3. If name == "CNN+Bilinear"
   4. Extract feature vector $v_i$ from $M$; learn bilinear function $W$
5. End
6. For $j = 1, 2, ..., |\mathcal{C}|$
7.   If name == "CNN+Bilinear"
8.      Compute the feature center of all images of the class $j$
9.      in $T$: $\tilde{v} = \frac{1}{N_j} \sum_{i=1}^{N_j} v_i$
10. End
11. For $i = 1, 2, ..., |U|$
12.   If name == "CNN+Bilinear"
13.     Compute $S_W(i, j) = v_i^T W \tilde{v}$ as similarity.
14. Else if name == "CNN+Probability"
15.     Use the output $p(i, j)$ of classifier $M$ as similarity.
16. End
17. End
18. Sort images in $U$ based on similarity in descending order
19. Assign top $k$ images to the set $R_j$ for the class $j$
20. End

Output: $R_j$ for each class $j$

Figure 3.1 Algorithms of CNN+Probability and CNN+Bilinear.
\[ S_W(v_i, v_j) = v_i^T W v_j \quad (3.1) \]

### 3.4. Data Augmentation Using Triplet-based Methods

We describe two triplet-based recommendation methods. One is a direct application of the Facenet [86] triplet model. The other is the new triplet model that does joint optimization for learning both the feature representation and the bilinear similarity matrix.

#### 3.4.1. Data Augmentation Based on Feature Learning (Triplet+L2)

We train a Facenet triplet model on the seed training dataset \( T \) in order to learn an embedding (feature representation) function \( F(I_i) \), from an image \( I_i \) into its corresponding feature vector by minimizing the overall loss \( L \) calculated using Equation (3.2). The goal of Facenet is that the squared distance between the image \( I_i \) and the image \( I_i^+ \) of the same class as \( I_i \) must be at least \( \alpha \) smaller than the squared distance between the image \( I_i \) and image \( I_i^- \) of a different class as \( I_i \) as shown in Equation (3.3). The second term \( \lambda \sum_{\theta \in \mathcal{P}} \theta^2 \) in Equation (3.2) is the regularization term [87] to prevent overfitting; \( \lambda \) is the weight decay.

\[
L = \sum_{i=1}^{\left| \Gamma \right|} \max(0, \|F(I_i) - F(I_i^+)\|_2^2 + \alpha - \|F(I_i) - F(I_i^-)\|_2^2) + \lambda \sum_{\theta \in \mathcal{P}} \theta^2 \quad (3.2)
\]

\[
\|F(I_i) - F(I_i^+)\|_2^2 + \alpha < \|F(I_i) - F(I_i^-)\|_2^2, \quad \forall (I_i, I_i^+, I_i^-) \in \Gamma, \quad (3.3)
\]

where \( \alpha \) is an enforced margin between positive and negative pairs; \( \mathcal{P} \) is the set of all parameters in \( F(I_i) \); \( I_i^+ \) (positive) is an image from the same class as \( I_i \). \( I_i^- \) (negative) is an image from a different class as \( I_i \). \( \Gamma \) is the set of all possible triplets in the training set and has cardinality \( |\Gamma| \). Figure 3.2 shows our method based on the embedding function learned in Line 3 using the negative of the squared distance function (L2) in Line 12 to find similar images.
Input: Seed training set $T$, recommendation number $k$, unlabeled set $U$, algorithm name

Algorithm:
1. Initialization: $R_j \leftarrow \emptyset$, $j = 1, 2, \ldots, |C|
2. If name == "Triplet+L2"
3. Train a Facenet Triplet model on $T$ to get $F(x)$.
4. Else if name == "Unified LF&SM"
5. Train a Unified LF&SM model on $T$ to get $F(x)$ and $W$.
6. End
7. For $j = 1, 2, \ldots, |C|
8. Compute the feature center of all images of the class $j$ in $T$: $\bar{v} = \sum_{i=1}^{N_j} F(I_i)/N_j$
9. For $i = 1, 2, \ldots, |U|$
10. If name == "Triplet+L2"
11. Compute $d(i, j) = -\|F(I_i) - \bar{v}\|_2^2$ as similarity.
12. Else if name == "Unified LF&SM"
13. Compute $S_{(F,W)}(i, j) = F(I_i)^TW\bar{v}$ as similarity.
14. End
15. End
16. Sort images in $U$ based on similarity in descending order
17. Assign top $k$ images to the set $R_j$ for the class $j$
18. End
Output: $R_j$ for each class $j$

Figure 3.2 Two triplet-based recommendation algorithms

3.4.2. Unified Learning of Feature Representation and Similarity Matrix

Here we describe our proposed Unified Learning of Feature Representation and Similarity Matrix (Unified LF&SM). Figure 3.3 shows the new model structure which is trained on the seed training dataset $T$. We aim at finding $S_{(F,W)}(I_i, I_j)$, a pair of an embedding function $F(I_i)$ mapping an image $I_i$ into a feature vector and a bilinear similarity matrix $W$ that assigns higher similarity scores to images in the same class as shown in Equations (3.4) and (3.5) by minimizing the loss function in Equations (3.6) and (3.7).
The model consists of a batch input layer to a CNN followed by L2 normalization, which results in the embedding using the triplet loss based on Bi-linear distance.

\[
S_{(\mathcal{F},W)}(I_i, I_i^+) > S_{(\mathcal{F},W)}(I_i, I_i^-) + \alpha, \ \forall (I_i, I_i^+, I_i^-) \in \Gamma
\]  \hspace{1cm} (3.4)

\[
S_{(\mathcal{F},W)}(I_i, I_j) = (\mathcal{F}(I_i))^T W \mathcal{F}(I_j)
\]  \hspace{1cm} (3.5)

\[
L = \sum_{i=1}^{\vert\Gamma\vert} l_{\mathcal{F},W}(I_i, I_i^+, I_i^-) + \lambda \sum_{\theta \in P} \theta^2
\]  \hspace{1cm} (3.6)

\[
= \sum_{i=1}^{\vert\Gamma\vert} \max\left(0, \alpha - S_{(\mathcal{F},W)}(I_i, I_i^+) + S_{(\mathcal{F},W)}(I_i, I_i^-)\right) + \lambda \sum_{\theta \in P} \theta^2
\]  \hspace{1cm} (3.7)

The definition of \(\alpha\), \(\vert\Gamma\vert\), \(I_i^-\) and \(I_i^+\) are the same as in the method of Triplet+L2; \(P\) is the set of all parameters in \(F(I_i)\) and \(W\). Unlike the Facenet model that uses L2 distance and optimizes for feature representation, the new model does joint optimization on both the feature representation and the similarity matrix used in the calculation of the similarity function \(S_{(\mathcal{F},W)}(I_i, I_j)\). The loss function in Equation (3.7) is positive when images of different classes are more similar than images of the same class by \(\alpha\), which is not desirable and we want to prevent this from happening. Figure 3.2 shows our algorithm using the learned similarity matrix and the learned feature representation obtained in Line 5 to find unlabeled images similar with the training images in each class based on the similarity scores in Line 14.

3.5. Experimental Environment and Results

In this Section, first we will describe our experimental datasets in Section 3.5.1. Second, we will describe our performance metrics for comparing the effectiveness of different augmentation methods in Section 3.5.2. Third, we will describe all parameters
used in our experiments in Section 3.5.3. Lastly, we will discuss our experimental results in Section 3.5.4.

3.5.1. Experimental Datasets

We select two image classification problems in endoscopy video analysis: instrument image detection [61][62] and retroflexion image detection [88]. The instrument image classification is the basis to find a segment of video in which a diagnosis or therapeutic operation occurs. Similarly, the retroflexion image classification is the basis to find a video segment in which retroflexion is performed. Both applications are useful for automatic documentation and objective quality control of colonoscopy [89] for better patient care. Figure 3.4 shows sample images for left cable body, right cable body, forceps head, snare head, retroflexion, and no object class for common endoscopy images without any of the aforementioned objects. We solve these two problems using one six-class CNN classifier for left cable, right cable, forceps head, snare head, retroflexion, and no object. The two problems have a severe class imbalance problem; instrument images and retroflexion images are rare.

Figure 3.4  Sample images for the six classes. (a) - (f): left cable body, right cable body, forceps head, snare head, retroflexion, and no object.
Seed training video set: We collect 25 de-identified full-length endoscopic videos of colonoscopy and upper endoscopy captured using Fujinon or Olympus endoscopes. The average percentages of images of each class calculated on the training video dataset are as follows: left cable (2.01%), right cable (3.82%), forceps head (2.04%), snare head (1.51%), retroflexion (0.8%), no object (89.8%).

Unlabeled dataset $U$ consists of 679,576 unlabeled images ($|U| = 679,576$) from 228 endoscopic videos, which are automatically created by extracting one frame for every ten frames. Each unlabeled video is neither in the seed training set nor in the test dataset.

Test dataset consists of 21,000 images (3,500 test images for each class) from 58 endoscopic videos, which are automatically created by extracting one frame for every five frames. Each test video is different from any training video and any unlabeled video. The test dataset contains many rare-class images with quite different appearances (e.g., different instrument colors or shapes) from the training images. We use the balanced test dataset in order to clearly see the effect on classification effectiveness for each class.

Training datasets: It is accepted knowledge that augmentation methods will not perform well when the size of the training set is small, the images in the training set are similar, and the training images are very different from the testing images. However, we do not know how these features impact different augmentation methods. To study the effect of different data augmentation methods, we collect six seed training datasets (Table 3.1). Each seed training dataset in Table 3.1 is class-balanced. If using an imbalanced seed training set with many more samples of the majority class, the resulting classifier will overfit the majority class. In other words, it would tend to mis-classify a rare class sample as the majority class, which is not desirable for finding more rare class samples.
Table 3.1  Seed training sets with different sizes or varieties or similarities

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{1400}$</td>
<td>Select 1 frame from every 5 frames from all 25 training videos; finally 1,400 images per class are manually labeled.</td>
</tr>
<tr>
<td>$S_{700}$ ($S_{700}^{highvar}$)</td>
<td>Select 1 frame from every 2 frames in $S_{1400}$. This set finally contains 700 labeled images per class.</td>
</tr>
<tr>
<td>$S_{350}$</td>
<td>Select 1 frame from every 4 frames in $S_{1400}$. Finally, this set has 350 labeled images per class.</td>
</tr>
<tr>
<td>$S_{700}^{lowvar}$</td>
<td>Select the first half number of frames in $S_{1400}$. Finally, this set has 700 labeled images per class and covers about half number of training videos.</td>
</tr>
<tr>
<td>$S_{700}^{highsim}$</td>
<td>More similar with the test set, 700 labeled images per class.</td>
</tr>
<tr>
<td>$S_{700}^{lowsim}$</td>
<td>Less similar with the test set, 700 labeled images per class.</td>
</tr>
</tbody>
</table>

**Varying sizes:** We collect three training sets (S350, S700, and S1400) from the aforementioned training video set. We collect 1400 images for each class from the videos in the training video set to form $S_{1400}$ first. Then, 700 frames are taken from $S_{1400}$ by keeping one every two images from $S_{1400}$. Similarly, S350 are selected from S700. This is to maintain similar variety of the images in these sets.

**Varying varieties:** We fix the training set size to 700 images, but collect two datasets with low and high variation in image appearances ($S_{700}^{lowvar}$, $S_{700}^{highvar}$); the $S_{700}^{lowvar}$ set has images that are more similar to each other than those in $S_{700}^{highvar}$ which is same as $S_{700}$. 

**Varying similarities with the test set:** We use two datasets ($S_{700}^{lowsim}$, $S_{700}^{highsim}$). It is generally understood that if the training dataset is quite similar to the test dataset, similar classification effectiveness is expected. However, creating an objective measurement of similarity to the test set is not trivial and the method to do so has not been defined in the literature. We develop our own method as follows. First, we train a Unified LF&SM model
using our test set as described in Section 3.4.2. Next, we compute the similarity $S_{(F,W)}(i, j)$ between each image $I_i$ of class $j$ in S1400 and the feature vector center of all images of class $j$ in the test set. Then, for each class $j$, we rank images of class $j$ in S1400 in descending order according to their similarities with the test set. Finally, for each class $j$, we use the top half number of images of class $j$ in S1400 as $S^\text{highsim}_{700}$, and the bottom half number of images of class $j$ in S1400 as $S^\text{lowsim}_{700}$; $S^\text{highsim}_{700}$ is more similar with our test set than $S^\text{lowsim}_{700}$.

### 3.5.2. Performance Metrics

**Classification effectiveness:** We report the average recall and average precision for the six classes of the CNN classifier trained on the augmented image set by each augmentation method. Recall and precision for each class $j$ are defined in Equations (3.8) and (3.9), respectively. Average recall and average precision are defined in Equations (3.10) and (3.11), respectively. They are measured on the test dataset as described in Section II. The test dataset is carefully designed to include different appearances of rare classes and the common class. Because training a deep learning model is very time consuming (about 2 days for each of our experiments) and we had several experiments varying different augmentation methods on different training sets of different sizes, varieties, and test data similarities, we did not do 10-fold cross validation.

\[
\text{Recall}_j = \frac{\text{number of images correctly classified as class } j}{\text{number of images in class } j}
\]

\[
\text{Precision}_j = \frac{\text{number of images correctly classified as class } j}{\text{number of images classified as class } j}
\]

\[
\text{Average recall} = \frac{\sum_{i=1}^{\mid C \mid} \text{Recall}_i}{\mid C \mid}
\]
Average precision = $\frac{\sum_{i=1}^{\left|\mathcal{C}\right|} Precision_i}{\left|\mathcal{C}\right|}$

**Recommendation accuracy:** Let $TA(j, k)$ be the number of true accepts (correct recommendations) in the top $k$ recommended images for the class $j$. We define $RTA_{\text{Mean}}(k)$ and $RTA_{\text{min}}(k)$ as the average true accept ratio and minimum true accept ratio considering all classes for each $k$ value as in Equations (3.12) and (3.13), respectively. We set $\left|\mathcal{C}\right| = 6$. Higher values of $RTA_{\text{Mean}}(k)$ and $RTA_{\text{min}}(k)$ are expected, and indicate better performances.

$$RTA_{\text{Mean}}(k) = \frac{1}{\left|\mathcal{C}\right|} \sum_{j=1}^{\left|\mathcal{C}\right|} \frac{TA(j, k)}{k}$$

$$RTA_{\text{min}}(k) = \min_{1 \leq j \leq \left|\mathcal{C}\right|} \frac{TA(j, k)}{k}$$

### 3.5.3. Experimental Parameters

**Compared methods:** We evaluate six data augmentation methods (two synthetic data augmentation methods and four real data augmentation methods) against Baseline which does not augment the training image set. For synthetic data augmentation, we select two methods: the traditional augmentation and TANDA [60]. We pick traditional augmentation because it is commonly used. We select TANDA because it utilizes an unlabeled dataset as well as experts’ input in learning realistic transformation and use the learned information to synthesize images in the training dataset.

For traditional augmentation, we use KERAS Image Data Generator [36] to generate synthesized images for each image in the seed training dataset. We need to guarantee that the synthesized image content is not heavily deformed, and the synthesized image has the same label as that of the original image by the expert. We experiment with many different parameter values and found that when setting rotation as 30 degrees,
shearing as 0.01, shifting as 0.01, zooming as 0.01, we obtain a good synthesized image set. Finally, we synthesize 5,600 images for each class to expand the seed training dataset.

TANDA consists of two components: transformation operator sequence learning and end classifier learning. To save the training time, we randomly select 60,000 unlabeled images from our large unlabeled image set. Then we ran the original code [60] of TANDA on this selected image set for 2 epochs to learn the transformation operator sequence. We try different combinations of basic transformation operators and observe their corresponding image plots generated by the code. Finally, we select three basic transformation operators. They are shearing, rotation, and zooming. After obtaining the transformation operator sequence, we train the end classifier. For the end classifier, we design a new CNN classification model and set up parameters in the original code to synthesize 5600 images for each class.

We use a CNN structure similar to the VGG Net [90], but with much fewer parameters, as shown in Table 3.2. Our CNN models accept RGB images with size of 64x64 pixels. These images are derived from resizing the raw endoscopic images of our datasets. We implement our CNN and triplet models using Python and Google’s TensorFlow libraries [91]. When training the CNN classifiers described in CNN+Probability and CNN+Bilinear, we add a softmax layer as the last layer and set the batch size as 256, the epoch number as 200, the weight decay as 0.001, and the learning rate as 0.001. When training the triplet models to learn feature representation in Triplet+L2 and Unified LF&SM, we set the enforced margin $\alpha$ as 0.2, the weight decay $\lambda$ as 0.001, the initial learning rate as 0.1, and the epoch number as 200 (400 batches per epoch, 6 classes per batch, and 256 images by random selection per class). We learn the bilinear
similarity function in CNN+Bilinear using the Matlab code provided by the authors of OASIS and set the iteration number as 108. The feature vector of each image is from the output of the “Conv5” layer in Table 3.2.

Considering that our unlabeled dataset $U$ consists of 679,576 unlabeled images and the retroflexion class has the lowest ratio as 0.8% calculated from the training videos, we estimate the number of images of the retroflexion class from $U$ to be 5,436 ($\approx 679,576 \times 0.8\%$). Here we set the recommendation number $k$ as 5000 for each augmentation method to find as many appearances of the rarest class. Finally, we train a new CNN classifier for our six-class classification problem by adding the new correctly recommended ones out of the 5,000 recommended images for each class to the seed dataset for each recommendation method. This final CNN model is used for final classification on the test dataset. For Baseline and the two synthetic data augmentation methods, we use the same CNN structure, weight decay, and learning rate used for CNN+Probability for training.

### 3.5.4. Experimental Results

We vary the sizes, the varieties, and the similarities of the seed training sets in Section 3.5.1. The same test set described in Section 3.5.1 is used in all the experiments.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Size-in</th>
<th>Size-out</th>
<th>kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv1</td>
<td>64x64x3</td>
<td>64x64x16</td>
<td>3x3x16.1</td>
</tr>
<tr>
<td>Pool1</td>
<td>64x64x16</td>
<td>32x32x16</td>
<td>2x2x16.2</td>
</tr>
<tr>
<td>Conv2</td>
<td>32x32x16</td>
<td>32x32x32</td>
<td>3x3x32.1</td>
</tr>
<tr>
<td>Pool2</td>
<td>32x32x32</td>
<td>16x16x32</td>
<td>2x2x32.2</td>
</tr>
<tr>
<td>Conv3</td>
<td>16x16x32</td>
<td>16x16x64</td>
<td>3x3x64.1</td>
</tr>
<tr>
<td>Pool3</td>
<td>16x16x64</td>
<td>8x8x64</td>
<td>2x2x64.2</td>
</tr>
<tr>
<td>Conv4</td>
<td>8x8x64</td>
<td>8x8x128</td>
<td>3x3x128.1</td>
</tr>
<tr>
<td>Pool4</td>
<td>8x8x128</td>
<td>4x4x128</td>
<td>2x2x128.2</td>
</tr>
<tr>
<td>Conv5</td>
<td>4x4x128</td>
<td>1x1x256</td>
<td>4x4x256.1</td>
</tr>
</tbody>
</table>
Figure 3.5 shows classification improvement in recall and precision. Recall improvement over Baseline is the recall using the CNN classifier trained on the augmented training dataset less the recall using the CNN classifier trained on the seed training set without any augmented images. Precision is computed similarly but using the precision performance metric instead.

**Performance of TANDA:** TANDA offers little recall improvement over Baseline (3.0% and 0.2% on S350 and $S_{700}^{\text{highvar}}$, respectively). TANDA also offers little precision improvement over Baseline (2.7% and 1.0% on S350 and $S_{700}^{\text{highvar}}$, respectively). We did not expect to find that TANDA yields even worse precision and recall on $S_{700}^{\text{lowvar}}$, $S_{700}^{\text{highsim}}$, $S_{700}^{\text{lowsim}}$, and S1400 compared with Baseline. Therefore, we do not include the performance data of TANDA in Figure 3.5 and in the following discussion. Since the authors' code [60] does not store the synthesized images during testing, we could not analyze the cause for TANDA's unexpected poor performance.

**Impact of seed training set sizes:** Figures 3.5(a) and 3.5(b) show that when the training set size is small, CNN+Probability offers the least improvement even worse than the traditional augmentation (Tradition). This is likely because CNN is not able to learn an effective classification model with this small training set for this classification problem. Relying on the CNN predicted probability to recommend unlabeled images is not reliable with a very limited training set. The rest of the real data augmentation methods offer significant improvement. With same feature representation as CNN+Probability, CNN+Bilinear using the learned similarity matrix for recommendation offers 11.5% recall improvement and 9.8% precision improvement over CNN+Probability when the training set is the smallest. The two triplet-based methods win over Tradition by offering between
6.1% and 8.2% recall improvement and 5.3% and 8.4% precision improvement across all the training set sizes. The triplet-based methods utilize the limited training dataset significantly better. The Triplet +L2 method offers a slightly better classification effectiveness on average than Unified LF&SM. In one of the next sections we will see the benefit of Unified LF&SM over Triplet+L2 when it comes to providing more correct recommendations for manual verification.

**Impact of variety of training images:** We fix the number of images at 700. The training datasets used for this study are $S_{700}^{\text{lowvar}}$ and $S_{700}^{\text{highvar}}$ where the latter set has images that are different from each other than the former set. Figures 3.5(c) and 3.5(d) show the recall and precision improvement over Baseline when the training set variety is varied. The results show that increase in variety of the seed training set increases the average recall and precision for each augmentation method. Interestingly, the training set variety impacts Tradition the most and the triplet-based methods the least. With limited variety in the training set, traditional augmentation contributes very little improvement over no augmentation baseline: 0.9% for average recall and 2.1% for average precision on $S_{700}^{\text{lowvar}}$. Therefore, to use traditional augmentation, manual effort is required to ensure that the training dataset has sufficient variety. The two triplet-based methods achieve much higher average recall and average precision than the two other methods based on CNN. Triplet+L2 outperforms Baseline by at least 14% in average recall and precision. Compared to traditional augmentation, it offers between 8.2% and 12.1% improvement on average precision and 8.2% and 13.9% on average recall. Hence, Triplet+L2 is much more effective in improving the classification effectiveness than traditional augmentation.
Instead of spending manual effort to gather the training dataset with sufficient variety, we recommend using one of the triplet-based methods to automatically recommend the candidate images.

Impact of training set sizes on recall (a) and precision (b) improvement

Impact of training set variety on recall (c) and precision (d) improvement

Impact of test set similarity on recall (e) and precision (f) improvement

Figure 3.5 Sensitivity study on classification effectiveness improvement over not using any data augmentation.
**Impact of test set similarity:** Figures 3.5(e) and 3.5(f) show that traditional augmentation provides little improvement over Baseline: 0.5% average recall improvement and 1.2% on average precision improvement on $S_{700}^{\text{lowsim}}$. In contrast, the two triplet-based methods improve the average recall and average precision over Baseline the best by at least 16.9% and 16.2% on $S_{700}^{\text{lowsim}}$. Traditional augmentation is most sensitive, and the triplet-based methods are least sensitive to the similarity between the seed training set and the test set. This finding has an important clinical implication. *Regardless of how similar the initial training set is to the test set, the triplet-based methods yield the final training dataset that includes the image appearances found in practice.* Using traditional data augmentation, careful selection of the initial training set is very important to the effectiveness of the trained CNN in day-to-day practice. Careful selection is time consuming.

**Impact of seed training sets on recommendation accuracy**

If the recommended images do not belong to the class that they are recommended for, the recommendations are not accurate. As a result, medical experts will need to spend extra time to assign images to the correct class labels. Furthermore, this could create class imbalance in the expanded training dataset if no or few images are present or rare instrument classes.

**Impact of the training set sizes on recommendation accuracy:** Figures 3.6(a) and 3.6(b) show that increase in the size of the seed training set improves the recommendation accuracy for each method. Unified LF&SM offers the highest average true accept ratio and highest minimum true accept ratio. Triplet+L2 is the second best followed by the other two methods. Unified LF&SM requires the least verification time for the medical experts.
compared to the other three methods while it offers very good classification effectiveness. Note that methods with a similar recommendation accuracy may provide noticeable difference in classification effectiveness. For instance, CNN+Probability and CNN+Bilinear have close recommendation accuracy under 50% when the training set has only 350 images for each class. However, CNN+Bilinear provides much better recall and precision improvement compared to CNN+Probability. The images that are selected by these two methods are different. Those selected to augment the training dataset by CNN+Bilinear yield better classification results.

**Impact of variety of training sets on recommendation accuracy:** We fix the training set size at 700 images and vary the variety as discussed previously. Figures 3.6(c) and 3.6(d) show that Unified LF&SM gives the highest recommendation accuracies among all the four methods. The difference between the min. true accept ratio offered by Unified LF&SM and the other methods are significant. *Unified LF&SM recommends images in a difficult class much better than the other techniques.* Triplet+L2 has an increase of 9.0% on average true accept and an increase of 18.2% on the minimum true accept when the variety increases. Triplet+L2 gains the most from the increase in variety of the training dataset. The two triplet-based methods take advantage of the variety in the training set to give a better recommendation than CNN+Probability and CNN+Bilinear, respectively. Figures 3.6(c) and 3.6(d) show that increase in the variety of the seed training set results in increases in recommendation accuracy of nearly all methods.

**Impact of test set similarity on recommendation accuracy:** Figures 3.6(e) and 3.6(f) show that Triplet+L2 gains the most from the increase in the test set similarity (11.3% of average true accept and 22.6% of minimum true accept, respectively). If the seed training
Impact of training set sizes on (a) average true accept and on (b) minimum true accept

Impact of varieties of training samples on (c) average true accept and (d) on minimum true accept

Impact of test set similarity on (e) average true accept and on (f) minimum true accept

Figure 3.6  Recommendation effectiveness of different real data augmentation methods
set is not a good representative of the test set, more manual work by domain experts is required using Triplet+L2 compared to Unified LF&SM. Nevertheless, this manual effort pays off much better in terms of classification effectiveness when compared to using CNN+Probability and CNN+Bilinear for recommendation.

3.6. Chapter Summary

We investigate the impact of size, variety, and similarity of the seed training set with the test set on the effectiveness of different data augmentation methods, which give us insight into the most effective image selection process. As expected, the effectiveness of each of the data augmentation methods is impacted by the size and the variety of the seed training set as well as the similarity between the seed training set and the test set. Each real data augmentation method offers noticeably higher effectiveness when using a seed training set with a bigger size, or larger variety, or higher similarity with the test set. Among them, the triplet-based methods (Triplet+L2 and Unified LF&SM) always gave the best improvement in recall and precision.

The traditional image augmentation method (e.g., shearing, rotation, shifting) contributes little to improve the classification effectiveness of a CNN classifier when the seed training set has low sample variety or is less similar with the test set. This means that careful manual selection of the training dataset is required in order to obtain a CNN classifier that is effective in practice. This process is time consuming. In contrast, the two triplet-based methods can semi-automatically expand the seed datasets using unlabeled data. Unified LF&SM perform best to get high recommendation accuracy, but triplet+L2 perform best to improve classification effectiveness when using a CNN classifier trained on the augmented dataset.
CHAPTER 4. INSTRUMENT SCENE DETECTION IN ENDOSCOPY VIDEOS

In this chapter, we introduce our proposed methods for instrument scene detection in endoscopic procedures. First, we briefly introduce the research problem and our main contributions in Section 4.1. Second, we describe our proposed cable footprint history technique for endoscopy image classification in Section 4.2. Third, we describe our proposed EndoCNN technique for endoscopy image classification in Section 4.3. Fourth, we describe our proposed instrument scene generation technique in Section 4.4. Fifth, we discuss our experimental results in Section 4.5. Finally, we give a chapter summary in Section 4.6.

4.1. Introduction

In this chapter, we solve a specific unbalanced medical image classification task: instrument scene detection in endoscopic procedures [61][62]. In endoscopic procedures, the most commonly found instrument appearances are in three categories: snare, forceps, and cable body or catheter sheath. They have quite different appearances and features. Moreover, in EGD, the cable is nearly always located in the left half part of an image and is usually long and thin. See Figure 4.1(a); but in colonoscopy videos, the cable is located in the right half part of an image and is usually short and wide. See Figure 4.1(b). This is related to the endoscopes typically used for these procedures and a difference in working distance away from the mucosa. For EGD, an upper endoscope is used which has the working channel at the 9 o’clock position; for colonoscopies, a colonoscope is used which has the working channel at the 5 o’clock position. Sometimes an upper endoscope is used for a colonoscope and vice versa; however, these are rare events. Finally, we get four different types of instrument frames: left cable body, right cable body, forceps head and snare head as shown in Figure 4.1.
There are many more non-instrument (NI) images than instrument images per video. Furthermore, the within-class distance of the NI class is very large. Based on careful observation of a large dataset, we categorize NI images into eight types (Figure 4.2): (a) the wide thick fold with its shape like that of the cable body or the forceps head, (b) the bright boundary with its brightness close to that of the cable body or the forceps, (c) the thin fold with its shape like that of the thin wire of the snare, (d) the polyp with the shape look like that of a forceps head, (e) the picture-in-picture which happens when the endoscopist takes a picture to document the case, (f) the retroflexion with the tubular shape like the cable body shape, (g) the uninformative frame without information useful for diagnostic and (h) the informative frame showing the in-focus colon mucosa. This last type has a larger variation in the appearance as the camera moves through a different segment.
of the organ that has different appearances. For instance, the transverse colon is more triangular than the other segments. Therefore, endoscopy images have complex hierarchical class structure as shown in Figure 4.3. Our contributions are as follows:

- We propose a two-step visual analysis approach to detect operation scenes. In the first step, we train a classifier to classify each endoscopy image. One image is considered as an instrument image if the classifier outputs a 1. Otherwise, the image is considered as a non-instrument image. In the second step, we utilize our pre-defined rules based on temporal information to determine the boundaries of operation scenes given a series of binary numbers from the classification step.

- We design two different types of endoscopy image classification models to detect instrument images. One is the Cable Footprint History technique (Cable Footprint) [61] which uses the hand-crafted features. The other one is EndoCNN [62] which learns features automatically using deep CNN.
Tran et al. [100] propose a 3D-CNN for video classification tasks. Unlike traditional CNN which accepts single image as the input for image classification, 3D-CNN accepts key frames of each video as the input for video classification directly. However, it is quite hard to apply 3D-CNN to accurately detect instrument scene boundaries. As a result, 3D-CNN is not applicable to our instrument scene detection task. We sample \( t \) images per second from the input video, forming what we call reduced colonoscopy video. The smaller the value of \( t \), the larger the reduction in the processing time, but the larger the difference between the actual and detected scene boundaries. The details of each step of our instrument scene detection technique are discussed below.

### 4.2. Endoscopy Image Classification Using Cable Footprint History Technique

#### 4.2.1. Image Preprocessing

To reduce processing time, we sub-sample and classify pixels in each input image \( I_i \) of the reduced video into two sets: \( I_{ND} \)--- non-dark pixel set using Equation (4.1) where \( T_c \) is a constant in the range of 0 and 1 and \( I_D \)---dark pixel set for all the pixels excluded from \( I_{ND} \). Next, for each image, we compute the median values of the non-dark pixel values in CIE LUV color space denoted as \( M_{LUV} \). For each pixel \( p \) in the image, we compute the Euclidean distance \( d(p, M_{LUV}) \) between the pixel values in CIE LUV color space of \( p \) and \( M_{LUV} \). LUV is one of the uniform color spaces for which the distance function maps to perceptual distance well. We separate the foreground and the background using Equation (4.2) where \( T_F \) is a contrast threshold.

\[
I_{ND} = \{ p | p \text{ is a pixel of } I_i \text{ with all its normalized } R, G, B \text{ values } > T_c \} \quad \text{(4.1)}
\]

\[
d(p, M_{LUV}) > T_F, \quad d(p, M_{LUV}) = \begin{cases} 
\sqrt{\frac{(p-M_{LUV})^2}{3}} & \text{if } p \in I_{ND} \\
0 & \text{if } p \in I_D
\end{cases} \quad \text{(4.2)}
\]
We perform an erosion with a disk structuring element and remove regions that are too large or too small. Let $R_i$ represent the $i$-th remaining connected component. This step replaces the time-consuming image segmentation method used in the previous algorithms [84]. We did not use the well-known Otsu method [92] to obtain a dynamic threshold value for each image because it wrongly considered cable regions as background for images with strong light reflection with higher contrast than cable regions in our training sets.

4.2.2. Spatial Feature Extraction and Classification

A number of shape features have been proposed [93] with varying degrees of computational complexity. Instead of using invariant moments [94] to represent region shape, we introduce a new method based on the domain knowledge to calculate a new Cartesian coordinate system to derive region shape features. By examining the cable regions from 17 endoscopic videos in our training video set, we further refine possible insertion directions into twelve general triangular areas as shown in Figure 4.4(a). For each area, two of the borders are denoted by the two-headed arrow in the figure. The third border is a portion of the image border intersecting the first two borders. We define a new Cartesian coordinate system for each corresponding Area $k$ as shown in Figures 4.4(b)-4.4(d). For instance, for $Area_{(2,5,8,11)}$, we choose $X'$ perpendicular to the diagonal line and $Y'$

![Figure 4.4](image)

(a) 12 general insertion areas; (b)-(d) new Cartesian coordinates; (e) insertion direction calculation
perpendicular to $X'$ as shown in Figure 4.4(d). We derive four spatial features to represent
the cable shape as follows.

First, we assign a cable region $R_i$ to Area $k$ denoted by the triangle $Area_k$ if more
than half of the pixels of the region are in $Area_k$ as shown in Equation (4.3). The $|x|$ denotes
the number of pixels in region $x$. Next, we calculate the features using the area-based
coordinate system. Equation (4.4) shows the eccentricity defined as the ratio of the cable
region height in the longest extension of $R_i$ along the $Y'$ axis to the width (the longest
extension of $R_i$ along $X'$) of a region $R_i$.

\[
\text{Area number} = \{k \mid \frac{|R_i \cap Area_k|}{|R_i|} > \frac{1}{2}\} \quad (4.3)
\]

\[
eccentricity = \frac{\text{height}}{\text{width}} = \frac{\frac{\max_x \sum_{y'} R_i(x',y')}^2}{\frac{\max_y \sum_{x'} R_i(x',y')}^2} \quad (4.4)
\]

Next, we calculate the orientation of the region. To get a reliable orientation, we only use
the middle part of the region. We define two lines: $P_2P_4$ at the one-fourth height of the
region and $P_1P_2$ at the third-fourth height of the region as shown in Figure 4.4(e). The
orientation vector of the region is calculated using Equation (4.5). Now, we compute the
angle difference $\theta_d$ between the region orientation vector and the angle bisector of $Area_k$.
See an example $\theta_d$ in Figure 4.4(e). We calculate the distance $d$ from a region $R_i$ to the
border of its $Area_k$ defined in Equation (4.6) and the normalized area $s$ of the region in
Equation (4.7).

\[
\text{orientation vector} = \frac{P_3P_1 + P_4P_2}{2} \quad (4.5)
\]

\[
d = \min_{(x',y') \in R_i} \{d(R_i(x',y'), \text{border of } Area_k)\} \quad (4.6)
\]

\[
s = \frac{\sum_x \sum_y R_i(x,y)}{\text{Number of all pixels in the image}} \quad (4.7)
\]
4.2.3. Cable Footprint History

In Equation (4.8), we compute the weight $w_i$ of the pixel at the coordinate $(x, y)$ on image $i$ using the corresponding binary image $B_i$ where only pixels of the detected regions by the classifier have the values of one and the rest have zeros. In other words, the weight of a pixel depends on whether it is part of the detected cable region in the current frame multiplied by one plus the weight of the corresponding pixel in the previous frame. The implication of this recursive equation is that the weight of this $x$-$y$ location increases when it is part of the detected cable regions in consecutive frames. The weight is reset to zero whenever this location is not part of any detected cable regions. Other positive constant positive values instead of 1 can be used. We chose 1 for simplicity.

$$w_1(x, y) = B_1(x, y)$$

$$w_i(x, y) = B_i(x, y) * (w_{i-1}(x, y) + 1)$$  \hspace{1cm} (4.8)

We compute the cable footprint history for each frame $i$ from the first frame to the last frame of the video using Equation (4.9). Figures 4.5(a) - 4.5(d) shows cable regions of the same video. Figure 4.5(e) shows the cable footprint history of the last frame of the entire video. The brightest region marks the most common pixel locations of detected cable regions in the video.

$$H_1(x, y) = w_1(x, y)$$

---

**Figure 4.5**  (a)-(d) Cable regions in one colonoscopy video; (e) cable footprint history of this video; (f) detected cable region
\[ H_i(x, y) = H_{i-1}(x, y) + w_i(x, y) \] (4.9)

We use a binary threshold \( T_H \) to segment the cable footprint history of the last frame \( t \) of the video into a set of connected components. Let \( HR_j \) represent the \( j \)-th connected component in the set. We choose the brightest connected component \( R_\# \) as the insertion area of this video as illustrated in Equation (4.10).

\[ R_\# = \left\{ HR_j \mid \max_{(x,y) \in HR_j} H_t(x, y) = \max_{(x,y) \in I_t} H_t(x, y) \right\} \] (4.10)

After locating the insertion direction for the video, we discard all candidate regions that do not intersect with \( R_\# \) since they are not likely a true cable region. Finally, we assign each frame either 0 or 1. A frame is assigned a 0 if it does not have any remaining cable candidate region. Otherwise, we label it as 1.

### 4.3. Endoscopy Image Classification Using EndoCNN Technique

In this section, we explore a CNN based solution in solving the operation scene detection problem in endoscopy videos. An effective CNN classifier must offer high precision and recall for both instrument and NI images. To achieve that, it must be trained with a good training dataset that is sufficiently large and has representative images of different patterns of appearances of both rare classes and common class. However, this manual data collection and labeling process is very time consuming, which inspires us to explore the following questions.

**Question 1:** Given a small seed training image dataset, can we design a tool that automatically finds good candidate images of the common class from a large unlabeled medical image dataset to be verified for inclusion in the seed training set?

**Question 2:** How effective is a CNN classifier trained using the dataset collected by this tool? We envision that (1) the tool should work for any type of medical images and for
any medical image classification problem; (2) the tool must be effective in its recommendation to reduce the time for verification whether to include the recommended images into the seed training dataset.

First, we discuss the manual process for creating the training image sets. Next, we discuss the CNN architecture for instrument frame classification. Last, we discuss several data augmentation methods which expand the small seed training dataset.

4.3.1. Manual Creation of Training Datasets

Large manually created dataset: Recall that there exist four different types of instrument images and eight different types of non-instrument images as shown in Figures 4.1 and 4.2. An ideal training image set is expected to cover enough representative samples for each of these 12 types. Therefore, we manually create a large training dataset as shown in Table 4.1 that has representative images covering as much as possible common variations of instrument and non-instrument images from 228 endoscopy videos. For our large training dataset, we manually select about 4,000 frames for each of the 8 NI types, resulting in 30,000 NI images. We select 5,000 frames for each of the 4 instrument image classes, resulting in 20,000 instrument images. In this training dataset, we maintain a slight class imbalance at the imbalance ratio of 1.5 (30,000/20,000), but not as high as observed in the 228 full-length endoscopic videos. This is because we do not want the trained classifier to be biased toward the NI class, and incapable of recalling difficult instrument images.

Small seed dataset: We manually select 625 frames for each of the 8 NI types from the large manually selected dataset, resulting in 5,000 NI images. We use all the instrument images in the large dataset. Our reason to use fewer NI images in this study because we want to apply different data augmentation methods on the seed dataset to recommend more
NI images as we found via trials and errors that having different types of NI images is important to improve CNN effectiveness.

### 4.3.2. Classification of Endoscopy Images Using CNN

The architecture of the CNN impacts significantly to the classification performance and influences the size of the training data. If the architecture has many layers, we typically need a large training dataset. A too simple architecture, however, may not be able to characterize difference in image appearances [99]. Via experiments, we settle on the architecture shown in Figure 4.6. We call it EndoCNN [62] for ease of future reference. We configure EndoCNN for a two-class classification problem where an image is classified into one of the two classes: the instrument classes (Figure 4.1) and the NI class (Figure 4.2). We use the MatConvNet library [95] to implement EndoCNN with a loss function in Equation (4.13).

\[
L = -\frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{K} 1\{y^{(i)} = k\} \ln \left( p(k|x^{(i)}; \Omega) \right) + \frac{\lambda}{2} \sum_{w_j \in \Omega_w} w_j^2 \tag{4.13}
\]

where \( M \) represents the number of training images; \( K \) is the number of classes; \( x^{(i)} \) and \( y^{(i)} \) represent the \( i \)th image and its label, respectively; \( 1\{\text{statement} \} \) is an impulse function which returns 1 when the statement is true and returns 0 otherwise; \( \Omega \) represents all the weight parameters \( \Omega_w \) and the bias parameters. After the two-class classification by EndoCNN, we assign the \( i \)th frame a value \( L_i \) which is either 0 or 1. The frame is assigned

| Parameters         | Random [0, 0.01] | Learning rate | 0.001 | Momentum | 0.9 | Weight decay \( \lambda \) | 0.001 | Batch size | 256 | Network depth | 11 | Stride size | 1 | Padding size | 1 | Activation function | ReLu | Pooling method | Max-pool |
|--------------------|------------------|----------------|-------|-----------|-----|--------------------------|-------|-------------|-----|--------------|----|--------------|---|----------------|-----|----------------|---------|

Figure 4.6 The EndoCNN structure. 16@64x64 represents 16 feature maps with a size of 64x64 as the output of the first convolutional layer.
a label 0 if it is detected as any of the NI image types and a label 1 if detected as any one of the instrument types.

4.3.3. Hierarchical and Unified Data Augmentation (HuDA)

We hypothesize that learning of the feature representation and the similarity matrix together is more likely to counter the impact of a small seed training dataset. Therefore, we propose and evaluate a new method called HuDA that does a joint optimization of both and utilizes the class hierarchy in Figure 4.3 to find $S_{F,W}(I_i, I_j)$, a pair of an embedding function mapping an image $I_i$ into a feature vector $F(I_i)$ and a bilinear similarity matrix $W$ such that higher similarity scores are assigned to images in the same class than in different classes. The class hierarchy is useful to learn a good feature representation which differentiate not only high-level classes (the instrument class and the NI class), but also low-level classes (subclasses of the instrument class or the NI class). Hence, the class hierarchy is very helpful to collect a representative image set for each subclass of the NI class. HuDA is generalizable to any medical image classification problem. The class hierarchy can be defined based on the problem domain.

HuDA accepts image quadruplets $\langle I_i, I_i^+, I_i^\pm, I_i^- \rangle$ as the input. $I_i$ is an anchor image. Unlike any existing triplet model, HuDA incorporates a class hierarchy (e.g., as in Figure 4.3). $I_i^+$ (a positive image) is an image from the same sub-class as $I_i$, $I_i^\pm$ (a semi-positive image) is an image from the same high-level class but different sub-class as $I_i$, and $I_i^-$ (a negative image) is an image from a different high-level class as $I_i$. As an example for the instrument detection problem, if $I_i$ is an image of the forceps head class, then $I_i^+$ must be a different image of the forceps head sub-class; $I_i^\pm$ can be any instrument image in the left cable sub-class or the right cable sub-class or the snare head sub-class; and $I_i^-$ must be an
image of the non-instrument class. We propose a new quadruplet loss function (See Equation (4.12)) that incorporates the following two design criteria:

(1) the similarity score between $I_i$ and $I_i^+$ to be at least $\alpha$ bigger than the similarity score between $I_i$ and $I_i^{\pm}$ as shown in Equation (4.11).

(2) the similarity score between $I_i$ and $I_i^{\pm}$ at least $\alpha$ bigger than the similarity score between $I_i$ and $I_i^-$ as shown in Equation (4.11).

The design criteria are to enforce the learning process to be sensitive to different levels of mis-classification according to a given class hierarchy. Mis-classification of sub-classes within the same high-level class is preferable over mis-classification across the high-level class. For example, a mis-prediction of a snare head image as a forceps head class is not desirable, but the forceps head class is in the correct high-level instrument class (Figure 4.3). This mis-classification is more preferable over prediction of the image as one of the NI sub-classes. The model is trained to minimize the overall loss function defined by Equation (4.12). The overall loss is the sum of all quadruplet loss $l_{f,w}(I_i, I_i^+, I_i^\pm, I_i^-)$.

$$S_{(f,w)}(I_i, I_i^+) > S_{(f,w)}(I_i, I_i^{\pm}) + \alpha \quad \text{and} \quad S_{(f,w)}(I_i, I_i^{\pm}) > S_{(f,w)}(I_i, I_i^-) + \alpha,$$

$$\forall (I_i, I_i^+, I_i^{\pm}, I_i^-) \in Q$$  \hspace{1cm} (4.11)

$$L = \sum_{i=1}^{\left|Q\right|} l_{f,w}(I_i, I_i^+, I_i^{\pm}, I_i^-) + \lambda \sum_{\theta \in P} \theta^2$$

$$= \sum_{i=1}^{\left|Q\right|} \max \left(0, \alpha - S_{(f,w)}(I_i, I_i^+) + S_{(f,w)}(I_i, I_i^{\pm})\right)$$

$$+ \sum_{i=1}^{\left|Q\right|} \max \left(0, \alpha - S_{(f,w)}(I_i, I_i^{\pm}) + S_{(f,w)}(I_i, I_i^-)\right) + \lambda \sum_{\theta \in P} \theta^2$$  \hspace{1cm} (4.12)

where $\lambda$ is the weight decay; $\alpha$ is a positive constant that is an enforced margin between positive and negative pairs. $P$ is the set of all parameters in $F(I_i)$ and $W$; $Q$ is the set of all possible quadruplets in the training set and has cardinality $|Q|$. 

**Input:** Seed training set $T$, recommendation number $k$, unlabeled set $U$

**Algorithm:**
1. Initialization: $R_j \leftarrow \emptyset$, $j = 1, 2, ..., |C|$
2. Train a HuDA model on $T$ to get $\mathcal{F}(x)$ and $W$.
3. For $j = 1, 2, ..., |C|$
4. Compute the feature center $\overline{v}$ of all images in the class $j$ in $T$:
   $$\overline{v} = \sum_{i=1}^{N_j} \mathcal{F}(I_i) / N_j \quad \text{// } N_j \text{ is the number of images of class } j$$
5. For $i = 1, 2, ..., |U|$
6. Compute $S_{(F,W)}(i,j) = \mathcal{F}(I_i)^T W \overline{v}$ as similarity.
7. End
8. Sort images in $U$ based on similarity in descending order
9. Assign top $k$ images to the set $S_j$ for the class $j$
10. End
**Output:** $S_j$ for each class $j$

Figure 4.7  HuDA algorithm

Figure 4.7 shows the HuDA algorithm. HuDA uses the learned similarity matrix and feature representation obtained in Line 2 of Figure 4.7. For each class, Line 7 calculates a similarity score for each unlabeled image and the center feature vector of the class. Lines 9-10 find $k$ images most similar to the center feature vector of that class. Layers 2-10 of the CNN structure used in our HuDA model are same as those of the EndoCNN structure in Figure 4.6. The input layer is different from Figure 4.6 in order to accommodate quadruplet instead of a single image. Furthermore, we do not need the soft-max classification layer since HuDA does not give a classification label as output. We implement the HuDA model by modifying the FaceNet source code [86] written in Python and Google’s TensorFlow library [91].

**4.3.4. Data Augmentation Methods**

We have two major goals for the experiments. First, we want to find the best CNN model for classification of images into instrument and NI classes. Second, we want to
compare effectiveness of EndoCNNs trained with different data augmentation methods. Recall that we select 228 de-identified endoscopy videos to be used for collecting the training image datasets. The process for creating the large and small manually collected datasets is described in Section 4.3.1.

**Traditional augmentation:** We use KERAS [36] package to apply rotation (0°~180°), shearing (0~0.01), translation (0~0.01), zooming (0~0.01), whitening, and mirroring on the 5,000 NI images in the small seed dataset to increase the number to 32,000 images.

**Data augmentation based on shot-based selection:** This method uses Edge Change Ratio method [96] for shot segmentation on each of the 200 training videos and creates a shot boundary when the ratio of the number of edge changes between the two neighboring frames is greater than a threshold 0.5. The program extract the middle frame of each shot as the key frame, resulting in a total of 42,000 key frames. After manually deleting all instrument images from the key frames, we are left with 39,000 NI images. In other words, about 7.1% of the instrument frames are deleted. From the remaining images, our program randomly select 32,000 images as the training set for NI images. This method still requires manual effort to verify 42,000 key-frames.

**CNN+Bilinear:** Using the small seed training dataset described in Section 4.3.2, we ran CNN+Bilinear in Section 3.3 to recommend 4,000 images for each of the 7 NI image types (a)-(g) from unlabeled images extracted from an image set that are not used for training, validation, or testing. We exclude recommending images in type h (informative frame) since this type has a large variation in image appearances, which is also the reason we configure EndoCNN for 11 class classification: 4 instrument sub-classes
and 7 NI sub-classes. We manually check 28,000 recommended images and discard all mis-recommended instrument images (about 1.39% of all recommended images). Of the remaining NI images, 27,000 images are randomly selected from the remaining NI images and added to the seed training dataset.

**Unified LF&SM:** Using the small seed training dataset described in Section 4.3.2, we ran our program described in Section 3.4.2 to recommend 4,000 images for each of the 7 NI image types (a)-(g) from unlabeled images extracted from an image set that is not used for training, validation, or testing. We manually check 28,000 recommended images and discard all mis-recommended instrument images (about 0.41% of all recommended images). Of the remaining NI images, 27,000 images are randomly selected and added to the seed training dataset.

**HuDA:** Using the small seed training dataset described in Section 4.3.2, we ran our program described in Section 4.3.1 to recommend 4,000 images for each of the 7 NI image types (a)-(g) from unlabeled images extracted from an image set that is not used for training, validation, or testing. We manually check 28,000 recommended images and discard all mis-recommended instrument images (about 0.31% of all recommended images). Of the remaining NI images, 27,000 images are randomly selected and added to the seed training dataset.

### 4.4. Instrument Scene Generation

This step utilizes temporal information and domain knowledge to identify operation scenes. This step accepts $L$, a sequence of 0 and 1 from the previous step, as input and outputs the frame numbers indicating the boundaries of the detected operation scenes.
4.4.1. Eliminate falsely detected cable images

This step corrects the misclassification results. We initialize the output sequence \( L^* \) with zeros. We slide a window of \( W \) frames on \( L \) from the beginning to the end of \( L \) one digit at a time. Each time, we compute the sum of all the numbers under the sliding window. When the sum is equal to \( W \) (i.e., all the frames under the window are cable images), we copy all the numbers under the sliding window in \( L \) to \( L^* \). We set the window size \( W \) to \( t / 2 \) where \( t \) is the temporal sampling rate in frames per second used in the pre-processing step. This window size covers frames within half a second since we observe that true cable frames typically appear consecutively more than half a second.

4.4.2. Locate Cable Scene Boundaries

Like in our previous techniques [84], we scan \( L^* \) from the beginning to the end. We first determine a sequence \( S \) of consecutive frames from \( L^* \) with all the following properties.

(1) The sequence \( S \) starts and ends with a 1, followed by at least \( K \times t \) consecutive 0s. In other words, the first and the last frames in \( S \) are cable images. The value of \( K \) should be the maximum temporal distance between consecutive cable shots of the same scene learned from training.

(2) The sequence \( S \) must have the ratio between the total number of 1s (instrument images) and the length of \( S \) greater than a threshold \( r_1 \).

(3) The sequence \( S \) lasts at least 2 seconds based on a consultation with our endoscopist and our observation. A biopsy is typically short about 2-4 seconds. A scene can have multiple sequences.
If the temporal distance between two consecutive sequence $S$ is less than $T_t$ seconds, we group them in the same operation scene. We select the values of $r_1$ and $T_t$ based on experiments discussed in Section 4.5.2.

### 4.5. Experimental Environment and Results

**4.5.1. Experimental Datasets**

Table 4.1 describes the summary of the training and test datasets. In each of our training datasets except the small seed dataset, we have about 60% more NI images than instrument images to slightly reflect the bias seen in practice in endoscopic procedures with many more NI frames. Validation images used for validating all the EndoCNNs are taken from the same training video set. We ensure that none of the validation images are same as those in the training image datasets. No test video is included in the training and validation. As shown in Table 4.1, we collect both balanced test image set and imbalanced test image set. The imbalanced test set is collected by extracting 1 frame for every 20 consecutive frames of test videos, which keeps the imbalance ratio of the instrument class to the NI class closer to the ratio in practice.
Table 4.2  Manual effort required to create a large training dataset beyond collecting the seed training dataset of 25,000 images

<table>
<thead>
<tr>
<th>Method</th>
<th>Manual effort type</th>
<th>#images required manual effort</th>
<th>Mis-recommendation ratio (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Large manual training set</td>
<td>Labeling</td>
<td>2,027,000</td>
<td>0</td>
</tr>
<tr>
<td>Traditional augmentation</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Shot-based selection</td>
<td>Verification</td>
<td>42,000</td>
<td>7.1</td>
</tr>
<tr>
<td>CNN+Bilinear</td>
<td>Verification</td>
<td>28,000</td>
<td>1.39</td>
</tr>
<tr>
<td>Unified LF&amp;SM</td>
<td>Verification</td>
<td>28,000</td>
<td>0.41</td>
</tr>
<tr>
<td>HuDA</td>
<td>Verification</td>
<td>28,000</td>
<td>0.31</td>
</tr>
</tbody>
</table>

Table 4.3  Parameters and values used in experiments

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling rate $t$ fps (frames per second)</td>
<td>6</td>
</tr>
<tr>
<td>Image resolution after spatial subsampling (pixels x pixels)</td>
<td>112x112</td>
</tr>
<tr>
<td>Dark pixel threshold $T_c$ for preprocessing</td>
<td>0.3</td>
</tr>
<tr>
<td>Color contrast threshold $T_F$ for preprocessing</td>
<td>0.05</td>
</tr>
<tr>
<td>Disk structuring element for erosion</td>
<td>5</td>
</tr>
<tr>
<td>Range of acceptable region size $R_S$ in pixels</td>
<td>$50 &lt; R_S &lt; \text{image area /14}$</td>
</tr>
<tr>
<td>Threshold $T_H$ for determining true cable area</td>
<td>0.5</td>
</tr>
<tr>
<td>Ratio of cable frames in an operation shot ($r_1$)</td>
<td>0.1</td>
</tr>
<tr>
<td>Duration without cable images between consecutive operation shots within one scene ($K$) in seconds</td>
<td>20</td>
</tr>
<tr>
<td>Duration in seconds to group two consecutive shots in one scene ($T_1$)</td>
<td>50</td>
</tr>
</tbody>
</table>

Table 4.2 provides a summary of the manual effort required to obtain a large training image dataset beyond the small seed training dataset, which is great, but with a significant reduction in the classification effectiveness as will be seen in Section 4.5.4. Traditional augmentation does not require any manual effort. CNN+Bilinear, Unified LF&SM, and HuDA require only a tiny fraction, 0.013 ($28,000/2,027,000$) of the manual effort compared to that required to creating a large training dataset manually and only about 66% of the manual effort required using the shot-based selection.
4.5.2. Model Parameters

**Cable Footprint History Technique:** The parameter values used by Cable Footprint are summarized in Table 4.3. We chose the temporal sampling rate of 6 fps to minimize the minimum distance between the true scene boundary and the detected scene boundary to 16 ms. For spatial subsampling rate, any higher rate, resulting in a smaller image resolution does not provide good classification result though it reduces the processing time. For the optimal color contrast threshold value, we plot the percentage of correct foreground (cable region) detection with different threshold values using the cable images in the image set. The plot in Figure 4.8 shows that the color contrast threshold of 0.05 gives the highest correct foreground detection result. We observe that many cable images in an operation scene are difficult to detect for several reasons such as blurry images, strong light reflected regions with tubular shape, use of dye color, and too small cable regions. Therefore, we set the ratio of cable frames in an operation shot, $r_1$ to a small value of $\frac{1}{10}$.

**HuDA:** When training the HuDA model to learn the feature representation and the similarity matrix in Section 4.3.3, via empirical studies, we settle with the enforced margin...
α as 0.2, the weight decay λ as 0.001, the initial learning rate as 0.001, and the epoch number as 200 (200 batches per epoch, 11 classes per batch, and 125 images by random selection per class).

4.5.3. Performance Metrics

For image classification, we use the precision, recall and the average F1-score [97] to measure the effectiveness of classification of images into instrument or NI classes. For scene segmentation, we use the performance metrics defined as follows. A false scene is a detected scene not overlapped with any operation scene in the ground truth. A missed scene is an operation scene in the ground truth that is not detected. For each video, we use Equation (4.13) to compute the true positive ratio ($TPR_i$) for the ith operation scene in the ground truth and the weighted true positive ratio ($WTPR$). False positive ratio ($FPR$) for each video is computed using Equation (4.14) where $N_d$ is the number of detected scenes; $N_g$ is the number of operation scenes in the ground truth video; $SC_i$ is the ith operation scene in the ground truth video; $SD_j$ is the jth detected scene for the video; $|SC_i|$ is the time duration of scene $i$ in seconds in the video; $T$ is the time duration of the video in seconds.

We use four metrics: the total number of false scenes ($#F$), the total number of missed scenes ($#M$), $WTPR$, and $FPR$. High $WTPR$ and low $FPR$ are desirable.

$$WTPR = \frac{\sum_{i=1}^{N_g} |SC_i| \cdot TPR_i}{\sum_{i=1}^{N_g} |SC_i|} \text{ where } TPR_i = \frac{\sum_{j=1}^{N_d} |SD_j \cap SC_i|}{|SC_i|}$$

(4.13)

$$FPR = \frac{\sum_{j=1}^{N_d} (|SD_j| - \sum_{i=1}^{N_g} |SD_j \cap SC_i|)}{T - \sum_{i=1}^{N_g} |SC_i|}$$

(4.14)

4.5.4. Experimental Results

A. Comparison of Effectiveness of Image Classification

A.1 Effect of Different Manually Collected Training Sets
Recall that we have two manually collected training datasets: large and small sets. The small set has only 5000 NI images. Both sets have the same instrument images. Table 4.4 shows that when tested on the balanced test dataset, EndoCNN trained using the small training set offers 91.90% average F₁-score, 4.82% lower than that of EndoCNN trained on the large training set. When tested on the imbalanced test dataset, EndoCNN trained on the small training dataset performs 19.05% worse in average F₁-score than that of EndoCNN trained on the large training set as shown in Table 4.5. The most performance drop of 44.86% is in precision of instrument images. Many predicted instrument images are in fact false. This mis-prediction is a result of a small number of NI images and significant class imbalance in the small training dataset.

Table 4.4 Effectiveness of different instrument image classification models on the balanced test set. I represents an instrument frame or class; NI represents a non-instrument frame or class.

<table>
<thead>
<tr>
<th>Method</th>
<th>Training datasets</th>
<th># Training images</th>
<th># Test images</th>
<th>Recall (%)</th>
<th>Precision (%)</th>
<th>Ave F₁-score (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>I</td>
<td>NI</td>
<td>I</td>
<td>NI</td>
<td>I</td>
</tr>
<tr>
<td>EndoCNN</td>
<td>Manual</td>
<td>20000</td>
<td>30000</td>
<td>18105</td>
<td>18105</td>
<td>94.89</td>
</tr>
<tr>
<td>EndoCNN</td>
<td>Manual</td>
<td>20000</td>
<td>5000</td>
<td>18105</td>
<td>18105</td>
<td>95.19</td>
</tr>
<tr>
<td>EndoCNN</td>
<td>Tradition</td>
<td>20000</td>
<td>32000</td>
<td>18105</td>
<td>18105</td>
<td>93.43</td>
</tr>
<tr>
<td>EndoCNN</td>
<td>Shot-based selection</td>
<td>20000</td>
<td>32000</td>
<td>18105</td>
<td>18105</td>
<td>89.27</td>
</tr>
<tr>
<td>EndoCNN</td>
<td>CNN+Bilinear</td>
<td>20000</td>
<td>32000</td>
<td>18105</td>
<td>18105</td>
<td>93.13</td>
</tr>
<tr>
<td>EndoCNN</td>
<td>Unified LF&amp;SM</td>
<td>20000</td>
<td>32000</td>
<td>18105</td>
<td>18105</td>
<td>95.72</td>
</tr>
<tr>
<td>Cable Footprint</td>
<td>HuDA</td>
<td>20000</td>
<td>32000</td>
<td>18105</td>
<td>18105</td>
<td>96.33</td>
</tr>
</tbody>
</table>

Tradition represents the traditional augmentation method.

Table 4.5 Effectiveness of different instrument image classification models on the imbalanced test set. I represents an instrument frame or class; NI represents a non-instrument frame or class.

<table>
<thead>
<tr>
<th>Method</th>
<th>Training datasets</th>
<th># Training images</th>
<th># Test images</th>
<th>Recall (%)</th>
<th>Precision (%)</th>
<th>Ave F₁-score (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>I</td>
<td>NI</td>
<td>I</td>
<td>NI</td>
<td>I</td>
</tr>
<tr>
<td>EndoCNN</td>
<td>Manual</td>
<td>20000</td>
<td>30000</td>
<td>9695</td>
<td>128256</td>
<td>92.83</td>
</tr>
<tr>
<td>EndoCNN</td>
<td>Manual</td>
<td>20000</td>
<td>5000</td>
<td>9695</td>
<td>128256</td>
<td>94.16</td>
</tr>
<tr>
<td>EndoCNN</td>
<td>Tradition</td>
<td>20000</td>
<td>32000</td>
<td>9695</td>
<td>128256</td>
<td>93.4</td>
</tr>
<tr>
<td>EndoCNN</td>
<td>Shot-based selection</td>
<td>20000</td>
<td>32000</td>
<td>9695</td>
<td>128256</td>
<td>89.2</td>
</tr>
<tr>
<td>EndoCNN</td>
<td>CNN+Bilinear</td>
<td>20000</td>
<td>32000</td>
<td>9695</td>
<td>128256</td>
<td>91.9</td>
</tr>
<tr>
<td>EndoCNN</td>
<td>Unified LF&amp;SM</td>
<td>20000</td>
<td>32000</td>
<td>9695</td>
<td>128256</td>
<td>93.71</td>
</tr>
<tr>
<td>EndoCNN</td>
<td>HuDA</td>
<td>20000</td>
<td>32000</td>
<td>9695</td>
<td>128256</td>
<td>94.35</td>
</tr>
</tbody>
</table>

Recall that we have two manually collected training datasets: large and small sets. The small set has only 5000 NI images. Both sets have the same instrument images. Table 4.4 shows that when tested on the balanced test dataset, EndoCNN trained using the small training set offers 91.90% average F₁-score, 4.82% lower than that of EndoCNN trained on the large training set. When tested on the imbalanced test dataset, EndoCNN trained on the small training dataset performs 19.05% worse in average F₁-score than that of EndoCNN trained on the large training set as shown in Table 4.5. The most performance drop of 44.86% is in precision of instrument images. Many predicted instrument images are in fact false. This mis-prediction is a result of a small number of NI images and significant class imbalance in the small training dataset.
A.2 Effectiveness of Traditional Data Augmentation

EndoCNN trained using the augmented training dataset created with traditional augmentation has a drop in the average F_1-score of 20.39% (from 89.13% on the balanced test dataset as shown in Table 4.4 to 68.74% on the imbalanced test dataset as shown in Table 4.5). The most significant drop of 55.25% comes from precision of instrument images. Many predicted instrument images are false. EndoCNN with Tradition performs 5.8% worse in the average F_1-score on the imbalanced test dataset than that of EndoCNN trained using the small training dataset. It shows that traditional data augmentation is not helpful to improve classification effectiveness under class imbalance since it makes the classifier more biased.

A.3 Effectiveness of Real Data Augmentation Methods

We investigate four real data augmentation methods (described in Section 4.3.4): shot-based selection method, CNN+Bilinear, Unified LF&SM, and HuDA. The shot-based method enables EndoCNN to obtain high average F_1-scores of 93.44% and 88.37% on the balanced and imbalanced test datasets, respectively. See Tables 4.4 and 4.5. It gives about 20% improvement in the average F1-score over traditional augmentation on the imbalanced test dataset. However, it offers 5.22% lower average F_1-score compared to EndoCNN trained on the large manually collected dataset because some types of NI images are rarely selected. The eight types of NI images do not appear in roughly equal probabilities in endoscopy videos. Blurry images and images with the bright cable body like or forceps like shape appear nearly on each video, but retroflexion images and picture-in-picture images do not appear often.
EndoCNN trained using CNN+Bilinear to recommend candidate NI images given the seed dataset of 5,000 NI images gives the fourth best average F1-score of 94.85%, which is 2.39% below that of the best EndoCNN (97.24%) on the balanced test set. See Table 4.4. However, on the imbalanced test dataset, as shown in Table 4.5, EndoCNN using CNN+Bilinear does not perform as well. It gives an average F1-score of 87.69%, which is 5.9% below that of the best EndoCNN (93.59%) on the same test dataset. Table 4.5 shows that EndoCNN trained on the augmented training set using CNN+Bilinear gives a low precision of 66.9% for the instrument class on the imbalanced test set. The features learned from the seed training set with a small number of NI images are biased toward instrument images, which affects the similarity learning.

EndoCNN trained on the training dataset expanded from the seed training dataset using Unified LF&SM to recommend candidate NI images gives the second best average F1-score of 97.04% and 93.26% on the balanced and imbalanced test sets, respectively, which are only about 0.3% below those of the best EndoCNN. See Tables 4.4 and 4.5. Furthermore, Unified LF&SM gives high recommendation accuracy compared to the shot-based selection method since only 0.41% of the recommended images are manually removed, which is 6.69% lower than that using the shot-based selection method.

EndoCNN trained on the training dataset expanded from the seed training dataset using HuDA to recommend candidate NI images gives the best average F1-score of 97.24% on the balanced test set and the third best average F1-score of 92.77% on the imbalanced test set, which are only about 0.8% below those of the best EndoCNN. As described in Section 4.3.4 and in Table 4.2, HuDA gives higher recommendation accuracy compared to the Unified LF&SM method since only 0.31% of the recommended images are manually
removed, which is 0.1% lower than that using the Unified LF&SM method. HuDA provides the benefits of both producing good expanded training datasets while shortening the time for manual labeling of the training data compared with CNN+Bilinear.

B. Comparison of Effectiveness of Instrument Scene Detection

Table 4.6 shows that one of our best classifiers, EndoCNN Manual (trained using the large manually selected dataset) achieves higher average WTPR by 23.69% compared to Cable Footprint, but 0.33% higher average FPR on the dataset I, but lower for the dataset II. EndoCNN reduces the false scenes in the two datasets by 65% from 52 to 18 false scenes, especially on the dataset II. Moreover, EndoCNN only miss 4 operation scenes instead of 48 by Cable Footprint. EndoCNN recalls more true operation scenes than Cable Footprint does by 17% ((48-4)/258). In some videos, Cable Footprint gives WTPR of zero because either the video only contains cable body which has weak color contrast with the background or only forceps head is seen during the operation for most of the time. Therefore, Cable Footprint miss these operation scenes since it is not designed to detect the appearance of forceps and snare head. Figure 4.9 shows that the EndoCNN model succeeds to detect instruments which have different colors with their backgrounds, but may fail to detect instruments which have quite similar colors with their background.

Table 4.6 Performance of different scene detection methods, # GT represents the number of operation scenes in the ground truth.

<table>
<thead>
<tr>
<th>Data &amp; Methods</th>
<th>Cable Footprint</th>
<th>EndoCNN Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dataset I:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58 videos with instruments;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>average duration: 28 mins</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average WTPR (%)</td>
<td>72.69</td>
<td>96.38</td>
</tr>
<tr>
<td>Average FPR (%)</td>
<td>2.7</td>
<td>3.03</td>
</tr>
<tr>
<td># F</td>
<td>24</td>
<td>14</td>
</tr>
<tr>
<td># M</td>
<td>48</td>
<td>4</td>
</tr>
<tr>
<td># GT</td>
<td></td>
<td>258</td>
</tr>
<tr>
<td><strong>Dataset II:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35 videos without instrument;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>average duration: 18 mins</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average FPR (%)</td>
<td>0.79</td>
<td>0.14</td>
</tr>
<tr>
<td># F</td>
<td>28</td>
<td>4</td>
</tr>
</tbody>
</table>
In terms of processing time during testing, EndoCNN took 15 milliseconds per frame, which outperforms 37 milliseconds of Cable Footprint [61] on the same PC with 3.5 GHz Intel Core i7-3770 and 32GB RAM. The time taken for scene detection is less than the time interval (33 milliseconds) between two consecutive frames.

4.6. Chapter Summary

In this chapter, we present two contributions. First, we propose EndoCNN, a new real-time and effective solution for detecting operation scenes in screening endoscopy procedures. Our experimental results on 93 full-length EGD and colonoscopy videos show that EndoCNN outperforms the state-of-the-art method on both classification effectiveness improving the recall by 17% and reducing the false scenes by 65%. To the best of our knowledge, the test dataset we use is the largest test dataset on this problem. Second, we investigate four data augmentation methods that recommend unlabeled images for verification by the domain experts to expand the seed training dataset. Among them, HuDA that utilizes a class hierarchy to jointly learn feature representation and a similarity matrix, offers the best classification effectiveness. HuDA improves the average F1-score by 24.03% over traditional data augmentation on the imbalanced test image set. EndoCNN trained using the training dataset expanded by HuDA offers the best average F1-score of
97.24% on the balanced test set and the third best average F$_1$-score of 92.77% on the imbalanced test set, which are only about 0.8% below those of the best EndoCNN, but with only 0.31% fraction of manual time required to label a large training dataset manually. Because feature representation and similarity learning are jointly learned from the training dataset, HuDA is generalizable to other types of medical image classification, which should help save domain experts’ time for manual labeling. Our future work is to investigate active deep learning methods that further minimize the manual efforts of checking recommended images while giving the best classification effectiveness in practice.
In this chapter, we introduce our active deep learning framework for image classification under class imbalance. Section 5.1 presents the overview of our proposed active deep learning framework. The key part of this active deep learning framework, feature representation and similarity learning, is described in Section 5.2. We describe our proposed unlabeled sample selection strategies based on the learned similarity model in Section 5.3. Finally, we present our experimental results in Section 5.4 and the summary of this chapter in Section 5.5.

5.1. Introduction

We propose an effective similarity active learning framework (SAL) as shown in Figure 5.1 for an unbalanced two-class image classification task. Assume a dataset of two categories (the rare class and the common class) and n samples denoted as $D = \{x_i\}_{i=1}^n$. The common class has many more samples than the rare class. Let $D^R$ and $D^C$ represent the...
labeled samples of the rare class and the common class, respectively. Let $D^U$ represents the unlabeled samples of $D$. The label of $x_i$ is denoted as $y_i$, $y_i \in \{1,2\}$. In our investigated image classification problems, most samples are unlabeled and may be selected and labeled in the learning process.

In each iteration, SAL uses Unified LF&SM model (learned from Step 1, Figure 5.1) to obtain feature representation and a similarity matrix to rank unlabeled images in Step 2. SAL then selects some of them for the domain experts to manually verify or correct the predicted label in “Step 3: Manual Label”. In “Step 3: Pseudo-Label”, SAL assigns the “common class” pseudo label to images when there is high confidence that these can be assigned to the common class for training without manual labeling. The goal is to get the highest number of labeled rare class images and common class images while minimizing the total number of images the domain experts need to verify or relabel.

5.2. Feature Representation and Similarity Learning

5.2.1. Triplet Loss

Traditional CNN classification models accept a single image as an input and usually require many training images. Given a small initial training dataset, the traditional CNN classification models cannot learn a good feature representation. Different from traditional CNN, Unified LF&SM is a triplet-based model. Let $I$ be a labeled training image set with a cardinality of $N$, $I_i$ be the $i$th training image, $I_i^+$ (positive) be an image from the same class as $I_i$, $I_i^-$ (negative) be an image from a different class as $I_i$. Unified LF&SM accepts image triplet $\langle I_i, I_i^+, I_i^- \rangle$ as the input. In theory, the number of triplets is $O(N^3)$. Given the same number of training images, Unified LF&SM utilizes different triplet combinations of these images to learn a better feature representation than traditional CNN does.
Unified LF&SM aims at finding $S_{(\mathcal{F}, W^*)}(I_i, I_j)$, a pair of an embedding function $\mathcal{F}(I_i)$ mapping an image $I_i$ into a feature vector and a bilinear similarity matrix $W^*$ such that the computed similarity score between the image $I_i$ and the image $I_i^+$ of the same class as $I_i$ is at least $\alpha$ greater than the similarity score between the image $I_i$ and the image $I_i^-$ of a different class as shown in Equations (5.1) and (5.2). Since the similarity matrix $W^*$ used in [59] is not symmetric, we may get two different similarity scores $(S_{(\mathcal{F}, W^*)}(I_i, I_j)$ and $S_{(\mathcal{F}, W^*)}(I_j, I_i))$ between the same pair of images $I_i$ and $I_j$. To get a single similarity score between a pair of images, we construct a symmetric similarity matrix $W^*$ as shown in Equation (5.3); $W$ can be initialized as any square matrix of order $m$ (the dimension of the feature vector).

$$S_{(\mathcal{F}, W^*)}(I_i, I_i^+) > S_{(\mathcal{F}, W^*)}(I_i, I_i^-) + \alpha, \forall (I_i, I_i^+, I_i^-) \in \Gamma$$

(5.1)

$$S_{(\mathcal{F}, W^*)}(I_i, I_j) = (\mathcal{F}(I_i))^T W^* \mathcal{F}(I_j)$$

(5.2)

$$W^* = \frac{1}{2} * (W + W^T)$$

(5.3)

$$L = \sum_{i=1}^{\lvert \Gamma \rvert} l_{\mathcal{F}, W^*}(I_i, I_i^+, I_i^-) + \lambda \sum_{\theta \in \mathcal{P}} \theta^2$$

(5.4)

$$= \sum_{i=1}^{\lvert \Gamma \rvert} \max \left(0, \alpha - S_{(\mathcal{F}, W^*)}(I_i, I_i^+) + S_{(\mathcal{F}, W^*)}(I_i, I_i^-) \right) + \lambda \sum_{\theta \in \mathcal{P}} \theta^2$$

(5.5)

The similarity score model is learned by minimizing the triplet loss function as shown in Equations (5.4) and (5.5). The second term in Equation (5.5) is the regularization term to prevent overfitting and obtain a smooth model; $\lambda$ is the weight decay; $\alpha$ is an enforced margin between positive and negative pairs. $\mathcal{P}$ is the set of all parameters in $\mathcal{F}(I_i)$ and $W$; $\Gamma$ is the set of all image triplets in the training set and has cardinality $\lvert \Gamma \rvert$. 

5.2.2. Triplet Selection

To ensure fast convergence, it is crucial to select triplets that violate the triplet constraint in Equation (5.1). Schroff et al. [86] propose a method to select image triplets for FaceNet. They use all positive images and semi-hard negative images in a mini-batch. For each image triplet, their semi-hard negative image is further away than the positive image from the anchor. But we found that for Unified LF&SM model, this negative sample selection strategy can in practice lead to bad local minima early on in training. Therefore, our selection strategy selects a negative image that is closer to the anchor than the positive image to the anchor as in Equation (5.6).

\[ S_{(f,W^*)}(I_i, I_i^+) < S_{(f,W^*)}(I_i, I_i^-) \quad (5.6) \]

5.3. Unlabeled Sample Selection

5.3.1. Rare Class Sample Selection and Labeling

The manual labeling process involves examining a very large number of images to find sufficient representative images of the rare class, which is usually time-consuming and not practical. A small number of labeled rare class images do not result in CNN with high recall for the rare class. We need an efficient way to find more rare class images. Therefore, we use the similarity model learned in Section 5.2 to recommend rare class images by finding unlabeled images similar with the labeled rare class images. First, we compute the feature vector center \( \overline{v}_R \) of all the labeled rare class images as shown in Equation (5.7). Next, we compute the similarity score between each unlabeled image \( I_i \) and \( \overline{v}_R \) using Equation (5.8). Then, we sort all unlabeled images in \( D^U \) in descending order of the computed similarity scores and select \( k_1 \) unlabeled images with the highest similarity scores for the experts’ manual labeling.
\[
\vec{v}_R = \frac{1}{|D_R^k|} \sum_{i=1}^{|D_R^k|} \mathcal{F}(I_i) 
\]

\[
S_{(\mathcal{F},W^*)}(I_i, \vec{v}_R) = \mathcal{F}(I_i)^T W^* \vec{v}_R 
\]

### 5.3.2. Dissimilar Sample Selection and Labeling

The labeled seed training image set is usually small and usually does not contain all appearance variations (sub-classes) of a class. Therefore, we propose a dissimilar sample selection method to select unlabeled images with appearances not present in the original training dataset. We rank all unlabeled images in descending order of the similarity scores with the rare class center as described in Section 5.3.1. For each unlabeled image \( I_i \), we get a rare class ranking score \( \text{Rank}_i^R \). Higher \( \text{Rank}_i^R \) score indicates that \( I_i \) is less similar with the center of the rare class. Similarly, we compute a common class ranking score \( \text{Rank}_i^C \) for each \( I_i \). Higher \( \text{Rank}_i^C \) score indicates that \( I_i \) is less similar with the center of the common class. A dissimilar image is much likely to have both high \( \text{Rank}_i^C \) and \( \text{Rank}_i^R \) scores. Therefore, we compute a novel score, \( \text{Novel}_i \), for each \( I_i \) using Equation (5.9). Then, we sort all unlabeled images in descending order of their computed novel scores and select \( k_2 \) unlabeled images with the highest novel scores as recommended dissimilar samples for the experts’ manual labeling. In Equation (5.9), we use the minimum of the two ranks. If the two ranks of an image are very different, the image is much closer to one class than the other. The novel score is the smaller of the two. After sorting in descending order of the novel scores, this image is unlikely to be selected as an dissimilar image.

\[
\text{Novel}_i = \min(\text{Rank}_i^R, \text{Rank}_i^C) 
\]

### 5.3.3. Pseudo-labeling of High Confidence Common Class Samples

We select from \( D^U \) images whose common class ranking score \( \text{Rank}_i^C \) is smaller than \( \delta_t \times |D^U| \) as high confidence common class images. These images are most likely true
common class images due to their high similarity with the common class. Therefore, we assign the common class as the pseudo-label for these images. In subsequent iterations, as more labeled images are increasingly available to train the Unified LF&SM, a more accurate similarity model is obtained. As a result, lower $Rank^c_i$ scores are associated more

---

**Algorithm:** SAL

**Input:**

- $D^U$: Set of unlabeled images
- $D^L$: Initial training set of images with labels
- $k_1$: Rare class sample selection size
- $\delta_1$: Initial high confidence sample selection threshold
- $d_r$: Threshold incremental rate
- $\beta$: Rare class recommendation accuracy threshold

**Variables:**

- $Accuracy$: Rare class selection accuracy
- $D^R_1$: Set of recommended rare class images
- $D^S_1$: Set of recommended dissimilar images
- $N_R$: Number of true rare class images in $D^R_1$
- $N_C$: Number of true common class images in $D^R_1$

**Output:**

- CNN classifier

1: Initialize the CNN classifier and the Unified LF&SM model with $D^L$;
2: $Accuracy \leftarrow 1$; $\delta_t \leftarrow \delta_1$
3: While $Accuracy \geq \beta$ do
4: Move $k_1$ images selected using 5.3.1 from $D^U$ to $D^R_1$
5: Experts manually label all images in $D^R_1$; Move all images in $D^R_1$ to $D^L$
6: $Accuracy \leftarrow \frac{N_R}{k_1}$
7: If $Accuracy > \frac{1}{2}$ then
8: Move $(N_R - N_C)$ images selected using 5.3.2 from $D^U$ to $D^S_1$ //dissimilar image
9: Experts manually label $D^S_1$; Move all images in $D^S_1$ to $D^L$
10: End
11: Select high confidence common class images based on $\delta_t$ from $D^U$ to $D^H$
12: as in 5.3.3; Update $\delta_t$ as in Equation (5.10)
13: Update the Unified LF&SM model using the images with labels in $D^L$
14: Update the CNN classifier using $D^H \cup D^L$
15: End while
16: Return CNN classifier

---

Figure 5.2  SAL algorithm
with common class images while higher $R_{i}^{C}$ scores are associated more with rare class images. Therefore, at the end of each iteration $t$, we increase the high confidence sample selection threshold as shown in Equation (5.10).

$$
\delta_{t} = \begin{cases} 
\delta_{1}, & t = 1 \\
\delta_{1} + d_{r} \ast (t - 1), & t > 1,
\end{cases} \quad (5.10)
$$

where $\delta_{1} > 0$ is the initial threshold; $d_{r} > 0$ is a threshold incremental rate. Figure 5.2 presents the algorithm. The stopping criteria is reached when the rare class recommendation accuracy is not greater than the given threshold. Sample selection strategies of SAL at each iteration cost $\mathcal{O}(c \ast N \ast \log_{2}k)$ time, where $c$ is the number of classes, $N$ is the number of unlabeled images, and $k$ is the number of selected unlabeled samples.

5.4. Experimental Environment and Results

5.4.1. Experimental Datasets and Settings

Experiment Settings: Tables 5.1 and 5.2 summarize the Endoscopy dataset and the Caltech-256 dataset [64] we used. Figure 5.3 shows some samples. For both datasets, 

<table>
<thead>
<tr>
<th>Table 5.1</th>
<th>Endoscopy dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td># Videos</td>
<td>228 Full-length colonoscopy and EGD videos, covering 95 hours, 29.97 fps</td>
</tr>
<tr>
<td>Sampling Rate</td>
<td>Sample 1 frame for every 20 consecutive frames</td>
</tr>
<tr>
<td>Resolution</td>
<td>720 x 480 RGB image</td>
</tr>
<tr>
<td>Images Imbalance ratio (1:44)</td>
<td># Rare Class Images (Images with forceps head)</td>
</tr>
<tr>
<td></td>
<td># Common Class Images (images without forceps head)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 5.2</th>
<th>Caltech-256 dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Images Imbalance ratio (1:37)</td>
<td># Rare Class Images (air plane class)</td>
</tr>
<tr>
<td></td>
<td># Common Class Images (256 other object classes)</td>
</tr>
</tbody>
</table>
we randomly select 1/6 of the images of the rare class and the common class as the test set. 

Next, we randomly select 1/6 of the images of the rare class and the common class in the remaining image set as the validation set. For the Endoscopy dataset, we randomly select 1/5 images (111 images) of the rare class and the same number of common class images in the remaining image set as the initial seed training set. For the Caltech-256 dataset, we randomly select 1/10 images (476 images) of the rare class and the same number of common class images in the remaining image set as the initial seed training set. Finally, we use all remaining images as the unlabeled image set for incremental learning process.

For each iteration, we set $k_1$ (the number of selected rare class images) as the number of initial seed training images. We set the rare class recommendation accuracy threshold $\beta$ as 0.1. For the Endoscopy dataset, we set $\delta_1$ as 0.05 and $d_r$ as 0.05. For the Caltech-256 dataset, we set $\delta_1$ as 0.2 and $d_r$ as 0.06. We use the same network architecture for Endoscopy and Caltech256 datasets as shown in Table 5.3. Our CNN models accept RGB
images with the size of 64x64 pixels. These images are created by resizing the original images using the Bicubic interpolation method. We implement the CNN and triplet models using Python and Google’s TensorFlow library. When training the CNN for classification, we add a softmax layer after “conv5” and set the batch size as 256, the epoch number as 200, the weight decay as 0.001, and the learning rate as 0.001. When training the Unified LF&SM models in Section 5.2, we set the enforced margin $\alpha$ as 0.2, the weight decay $\lambda$ as 0.001, the initial learning rate as 0.1, and the epoch number as 200 (200 batches per epoch, 2 classes per batch, and half number of labeled images by random selection per class).

**Comparison Methods:** We compare our proposed SAL framework with two recent state-of-the-art active deep learning methods (AIFT and CEAL) and one baseline method (ALL):

- **ALL:** All the training samples are manually labeled and used to train the CNN classifier. This method is expected to obtain the best classification performance and regarded as the upper bound performance.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Size-in</th>
<th>Size-out</th>
<th>Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv1</td>
<td>64x64x3</td>
<td>64x64x16</td>
<td>3x3x16,1</td>
</tr>
<tr>
<td>pool1</td>
<td>64x64x16</td>
<td>32x32x16</td>
<td>2x2x16,2</td>
</tr>
<tr>
<td>conv2</td>
<td>32x32x16</td>
<td>32x32x32</td>
<td>3x3x32,1</td>
</tr>
<tr>
<td>pool2</td>
<td>32x32x32</td>
<td>16x16x32</td>
<td>2x2x32,2</td>
</tr>
<tr>
<td>conv3</td>
<td>16x16x32</td>
<td>16x16x64</td>
<td>3x3x64,1</td>
</tr>
<tr>
<td>pool3</td>
<td>16x16x64</td>
<td>8x8x64</td>
<td>2x2x64,2</td>
</tr>
<tr>
<td>conv4</td>
<td>8x8x64</td>
<td>8x8x128</td>
<td>3x3x128,1</td>
</tr>
<tr>
<td>pool4</td>
<td>8x8x128</td>
<td>4x4x128</td>
<td>2x2x128,2</td>
</tr>
<tr>
<td>conv5</td>
<td>4x4x128</td>
<td>1x1x128</td>
<td>4x4x128,1</td>
</tr>
</tbody>
</table>

The input and output sizes are described by rows $\times$ cols $\times$ #nodes. The kernel is specified as rows $\times$ cols $\times$ #filters, stride.
• **Active Incremental Fine-Tuning (AIFT)** [79]: AIFT incorporates data augmentation techniques into active deep learning in a novel manner. AIFT is based on an observation that all images synthesized from the same image share the same label and are naturally expected to have similar predictions by the current CNN. As a result, their entropy and diversity are a powerful indicator to the “worthy” of an unlabeled image in improving the current CNN’s performance. To enhance the robustness of AIFT, a filter selects some “hard” synthesized images and entropy and diversity are computed by selecting only a portion of synthesized images of each unlabeled image according to the probabilities predicted by the current CNN.

• **Cost-Effective Active Learning (CEAL)** [77]: CEAL incorporates deep convolutional neural networks into active learning. Unlike traditional active learning methods focusing on only uncertain samples, CEAL also discovers a large amount of high confidence samples from the unlabeled set for automatic pseudo-labeling.

All comparison methods select the same number of unlabeled images for experts’ manual labeling in each iteration and share the same CNN architecture with our SAL on both datasets. They also use the same parameter settings when training the CNN classifiers with our SAL as described in Section 5.4.1. They only differ in the strategy to select unlabeled images.

When implementing AIFT, we synthesize 15 images for each unlabeled image from the Caltech-256 dataset. We synthesize 5 images for each unlabeled image from the Endoscopy dataset since this dataset is large (more than 300,000 images). We use AIFT (Entropy + Diversity)$^\alpha$ and compute the entropy-diversity matrix using the code provided by the author of AIFT. We set $\alpha$ as $1/4$, which control the percentage of synthesized images selected for
each unlabeled image. We set the trade-offs $\lambda_1$ and $\lambda_2$ between entropy and diversity as the same value 0.5. When implementing CEAL, we found that if the initial high confidence sample threshold $\delta_1$ is not small enough, predicted pseudo-labels of many high confidence samples are wrong. Thus, we set $\delta_1$ as $10^{-7}$ to get a trade-off between a large number of selected samples and high prediction accuracy of high confidence samples for both datasets. We set the high confidence threshold $d_r$ as $0.05 \times 10^{-7}$ and $0.18 \times 10^{-7}$ for the Endoscopy dataset and the Caltech-256 dataset, respectively.

5.4.2. Performance Metrics

Let $TP_j$ be the number of all images correctly classified as class $j$, $GP_j$ be the number of all images belonging to class $j$ in the ground truth, $DP_j$ be the number of all images detected as class $j$. The traditional recall, precision, and $F_1$-score values for each class $j$ are used as shown in Equations (5.11) and (5.12).

$$Recall_j = \frac{TP_j}{GP_j}, \quad Precision_j = \frac{TP_j}{DP_j}, \quad j = 1, 2$$ (5.11)

$$F_1-Score_j = \frac{2 \cdot Recall_j \cdot Precision_j}{Recall_j + Precision_j}, \quad j = 1, 2$$ (5.12)

5.4.3. Experimental Results

Figure 5.4 illustrates the classification performance at a different iteration number of ALL, CEAL, AIFT and the proposed SAL on both the Endoscopy and Caltech-256 datasets. Classification performance at iteration 0 is the performance when the initial training dataset is used. As illustrated in Figures 5.4 (a) - 5.4 (f), SAL obtains the best average recall, precision, and $F_1$-score for nearly every iteration compared with CEAL and AIFT on both datasets. Furthermore, SAL increases classification performance faster than CEAL and AIFT do in early iterations. CEAL gives higher classification performance than AIFT. Figures 5.4 (a) and 5.4 (d) show that SAL gives a higher average recall value of 94%
and 92% than 91.7% and 90.9% of the second best method CEAL at the final iteration on the Endoscopy and Caltech-256 datasets, respectively. SAL offers nearly the same average recall, precision, and F1-score values as ALL, the best method, at the final iteration for both datasets. More importantly, only 5.6% and 7.5% of all unlabeled images for the Endoscopy and Caltech-256 dataset, respectively, need manual labelling by experts in order for SAL to achieve these excellent results (see the second row of Table 5.4 and Table 5.5). This is strong evidence that SAL effectively reduce the experts’ manual labeling efforts while offering high classification performance. Therefore, SAL has a competitive advantage in unbalanced deep image classification tasks.

Component Analysis: We analyze the effectiveness of our proposed SAL’s components (rare class selection and high confidence common class pseudo-labeling). Tables 5.4 and 5.5 show that 95.1% of rare class images from the unlabeled image set are selected by SAL even if SAL only selects 5.6% of unlabeled images for experts’ manual
Table 5.4 Rare class recommendation accuracy for the Endoscopy dataset

<table>
<thead>
<tr>
<th>Training iteration t</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of samples labeled (%) up to iteration t</td>
<td>0.4</td>
<td>0.9</td>
<td>1.3</td>
<td>1.9</td>
<td>2.5</td>
<td>2.9</td>
<td>3.4</td>
<td>3.8</td>
<td>4.3</td>
<td>4.7</td>
<td>5.1</td>
<td>5.6</td>
</tr>
<tr>
<td>Percentage of rare class samples labeled (%) up to iteration t</td>
<td>SAL 10.0</td>
<td>18.9</td>
<td>28.5</td>
<td>41.7</td>
<td>55.1</td>
<td>64.3</td>
<td>72.0</td>
<td>78.3</td>
<td>86.6</td>
<td>90.8</td>
<td>93.3</td>
<td>95.1</td>
</tr>
<tr>
<td>Percentage of samples labeled (%) up to iteration t</td>
<td>CEAL 10.0</td>
<td>10.4</td>
<td>11.6</td>
<td>15.7</td>
<td>20.3</td>
<td>24.9</td>
<td>30.6</td>
<td>35.6</td>
<td>41.0</td>
<td>45.7</td>
<td>49.0</td>
<td>50.9</td>
</tr>
<tr>
<td>Percentage of rare class samples labeled (%) up to iteration t</td>
<td>AIFT 10.0</td>
<td>10.4</td>
<td>12.0</td>
<td>14.8</td>
<td>19.5</td>
<td>25.9</td>
<td>30.4</td>
<td>34.4</td>
<td>38.1</td>
<td>40.8</td>
<td>42.9</td>
<td>44.1</td>
</tr>
</tbody>
</table>

Table 5.5 Rare class recommendation accuracy for the Caltech-256 dataset

<table>
<thead>
<tr>
<th>Training iteration t</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of samples labeled (%) up to iteration t</td>
<td>1.1</td>
<td>2.1</td>
<td>3.2</td>
<td>4.3</td>
<td>5.4</td>
<td>6.4</td>
<td>7.5</td>
</tr>
<tr>
<td>Percentage of rare class samples labeled (%) up to iteration t</td>
<td>SAL 20.1</td>
<td>40.5</td>
<td>61.3</td>
<td>76.9</td>
<td>83.4</td>
<td>88.4</td>
<td>91.5</td>
</tr>
<tr>
<td>Percentage of samples labeled (%) up to iteration t</td>
<td>CEAL 20.1</td>
<td>20.1</td>
<td>20.6</td>
<td>26.1</td>
<td>38.5</td>
<td>47.0</td>
<td>51.4</td>
</tr>
<tr>
<td>Percentage of rare class samples labeled (%) up to iteration t</td>
<td>AIFT 20.1</td>
<td>21.3</td>
<td>21.7</td>
<td>22.8</td>
<td>25.3</td>
<td>27.7</td>
<td>34.4</td>
</tr>
</tbody>
</table>

We also find that only a tiny percentage of dissimilar images belong to the rare class from the unlabeled image set are selected by SAL even if SAL only selects 7.5% of unlabeled images for experts’ manual labeling at the final iteration on the Caltech-256 dataset. Tables 5.4 and 5.5 also show that many more rare class images from the unlabeled image set are selected for manual labeling by SAL compared with CEAL and AIFT. As a result, SAL include many more labeled rare class images in the training set. That may be in part explain that SAL outperforms CEAL and AIFT on both datasets.

Figure 5.5 shows some images from the dissimilar sample selection as described in Section 5.3.2. These images are quite different from our initial training images in appearances. The five images in the second line of Figure 5.5 represent five different categories of objects missed in the initial seed training set for the Caltech-256 dataset. All these five missed object categories are considered as the common class in our experiments. We also find that only a tiny percentage of dissimilar images belong to the rare class. This may be because of the high imbalance ratio from the common class to the rare class in the sample selection space.
To understand the contribution of pseudo-labeling of high confidence common class images, we compare the CNN classification performance of two variants of SAL (SAL_Conf and SAL_noConf). Both methods use manually labeled images. The only difference between the two methods is that SAL_Conf uses pseudo-labeling of high-confidence common class images while SAL_noConf does not do this. Figure 5.6 shows

Figure 5.5  The first line: dissimilar images selected by SAL from the Endoscopy dataset. The second line: dissimilar images selected by SAL from the Caltech -256 Dataset.

Figure 5.6  Classification performances under different iterations for two variants of SAL on Endoscopy (a-c) and Caltech-256 (d-f) datasets. SAL_Conf uses pseudo-labeling of high confidence common class images, SAL_noConf does not use any high confidence common class image.
that SAL_Conf consistently outperforms SAL_noConf with a clear margin on both the Endoscopy dataset (the first row) and the Caltech-256 dataset (the second row). These results confirm that pseudo-labeling of high confidence common class images is useful and can significantly improve the classification performance.

5.5. Chapter Summary

In this paper, we propose the first active deep learning framework (SAL) for image classification under class imbalance. Unlike traditional active learning methods which either focus on finding uncertain samples or never consider class imbalance, SAL actively learns a triplet-based similarity model to recommend rare class samples with a high accuracy for experts’ manual labeling as well as recommend high confidence common class samples for automatic pseudo-labeling without any input from experts. We compare classification performance (average recall, precision, and F1-score) of different active learning methods on two challenging image classification datasets: an Endoscopy dataset and the Caltech-256 dataset. Our experiments show that SAL consistently outperforms other state-of-the-art active learning methods on both datasets. Our experiments also show that SAL obtains nearly the upper bound of classification performance by only labeling 5.6% and 7.5% of all images for the Endoscopy dataset and the Caltech-256 dataset, respectively. This finding shows that SAL significantly reduces the experts’ manual labeling efforts while achieving excellent classification performances. For the future work, we plan to improve the methods for recommending rare class and common class samples more accurately as well as investigate active deep learning methods for object recognition tasks.
CHAPTER 6. CONCLUSION

We have described our works on medical image classification under class imbalance in details. We have overcome several main challenges. We have proposed a new real data augmentation method called Unified LF&SM to quickly expand the small labeled training dataset. We have explored six different data augmentation methods: four RDA (real data augmentation) methods and two SDA (synthetic data augmentation) methods. We have carefully designed six different training datasets that have different numbers of training images (sizes), different image appearances (variety) in each training set, and different similarity scores to the test set. We have performed a sensitivity study to determine the impact on the classification effectiveness due to the sizes, the varieties within the training data, and the similarity of the training images with those in the test dataset. This study thus aims to identify and confirm the drawback of each augmentation method. To the best of our knowledge, no existing research team has done a similar study. Our work provides useful insight into how to choose a good training image dataset for medical image classification tasks.

We also have proposed two different approaches for the instrument scene detection task in endoscopic procedures: Cable Footprint and EndoCNN. EndoCNN outperform the state-of-the-art method for instrument scene detection on both the detection accuracy and the processing time. Viewing instrument / NI classes and corresponding sub-classes as a class hierarchy, we also have proposed a novel Hierarchical Unified Data Augmentation (HuDA) method to quickly collect a large representative image set for the common class. HuDA is capable of significantly reducing manual efforts with slight loss in classification effectiveness. HuDA is applicable to any medical image classification problem.
We also have proposed a novel similarity-based active deep learning framework (SAL). To the best of our knowledge, SAL is the first active deep learning method that deals with a significant class imbalance and small labeled training image set. Our experiments show that SAL consistently outperforms two state of the art methods on both datasets: the Endoscopy dataset and the Caltech-256 dataset. Our experiments also have show that SAL nearly obtains the upper bound classification performance by labeling only 5.6% and 7.5% of all images for the Endoscopy dataset and the Caltech-256 dataset, respectively. This finding confirms that SAL significantly reduces the experts’ manual labeling efforts while still achieving near upper bound performance.

Our future work are as follows: (1) explore more effective methods, which not only expand the number of training images, but also improve the sample variety of training images; (2) investigate the effectiveness of our proposed SAL method on multi-class image classification tasks; (3) explore methods to measure the sample variety of an image set and use them in the unlabeled sample selection method; (4) explore the recommendation method used in HuDA (e.g., recommend top-k skip n rather than top-k).
REFERENCES


[27] Y. Zhao and A. K. Shrivastava, "Combating Sub-Clusters Effect in Imbalanced Classification," in Proc. of IEEE 13th Int’l Conf. on Data Mining, Dallas, TX, 2013, pp. 1295-1300.


[56] Z. Ferdowsi, R. Ghani, and R. Settimi, "Online Active Learning with Imbalanced Classes," in Proc. of IEEE 13th Int’l Conf. on Data Mining, Dallas, TX, pp. 1043-1048, 2013.


