SEMI-SUPERVISED LEARNING WITH LABEL CONFIDENCE FOR AUTOMATIC KNEE OSTEOARTHRITIS SEVERITY ASSESSMENT

by

Yifan Wang

APPROVED BY SUPERVISORY COMMITTEE:

Dian Zhou, Chair

Tamil Lakshman

Jin Liu

Mehrdad Nourani

Copyright © 2022

Yifan Wang

All rights reserved

This dissertation is dedicated to my wife

for her accompanying

throughout the years.

SEMI-SUPERVISED LEARNING WITH LABEL CONFIDENCE FOR AUTOMATIC KNEE OSTEOARTHRITIS SEVERITY ASSESSMENT

by

YIFAN WANG, BS, MS

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN

COMPUTER ENGINEERING

THE UNIVERSITY OF TEXAS AT DALLAS

May 2022

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Dian Zhou, who has advised me throughout my research. He is always sharing insightful opinions. I have been influenced and motivated by his academic enthusiasm. I would like to thank my senior, Dr. Zhaori Bi. He helped me settle down five years ago when I traveled to the US alone. He also provided me with precious advice on my research. I would like to thank my wife, Mrs. Xuelu Wang. Her continuous encouragement and accompany give me the strength to overcome any difficulty. The days we spent together enriched my life. I also want to thank my group members, Mr. Xiannan Wang, and Mr. Tianning Gao. I would like to thank my friend, Mr. Qilin Si. I would like to thank the people I worked with, Mr. Ziyang Luo, Mr. Genchan Peng, etc. Finally, I would like to thank my parents. They supported me without asking anything from me. With these people, I become the person I am.

March 2022

SEMI-SUPERVISED LEARNING WITH LABEL CONFIDENCE FOR AUTOMATIC KNEE OSTEOARTHRITIS SEVERITY ASSESSMENT

Yifan Wang, PhD
The University of Texas at Dallas, 2022

Supervising Professor: Dian Zhou, Chair

Knee osteoarthritis (OA) is a chronic disease that considerably reduces patients' quality of life. Preventive therapies require early detection and lifetime monitoring of OA progression. In the clinical environment, the severity of OA is classified by the Kellgren and Lawrence (KL) grading system, ranging from KL-0 to KL-4. Recently, deep learning methods were applied to OA severity assessment, to improve accuracy and efficiency. Researchers fine-tuned convolutional neural networks (CNN) on the OA dataset and built end-to-end approaches. However, this task is still challenging due to the ambiguity between adjacent grades, especially in early-stage OA. Low confident samples, which are less representative than the typical ones, undermine the training process. Targeting the uncertainty in the OA dataset, we propose a novel learning scheme that dynamically separates the data into two sets according to their reliability. Besides, we design a hybrid loss function to help CNN learn from the two sets accordingly. With the proposed approach, we emphasize the typical samples and control the impacts of low confident cases. Experiments are conducted in a five-fold manner on five-class task and early-stage OA task. Our method achieves a mean accuracy of 70.13% on the five-class OA assessment task, which outperforms all other state-of-art methods. Despite early-stage OA detection still benefiting from the human intervention of lesion region selection, our approach achieves superior performance on the KL-0 vs. KL-2

task. Moreover, we design an experiment to validate large-scale automatic data refining during training. The result verifies the ability to characterize low confidence samples by our approach. The dataset used in this paper was obtained from the Osteoarthritis Initiative.

TABLE OF CONTENTS

ACKNOWLEDGMENTS v

ABSTRACT vi

LIST OF FIGURES x

LIST OF TABLES xi

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 DEEP LEARNING METHODS ON MEDICAL IMAGE ANALYSIS TASKS 7

2.1 Introduction 7

2.2 Deep Learning Methods for Image Classification 9

2.3 Deep Learning Methods for Object Detection 15

CHAPTER 3 SEMI-SUPERVISED LEARNING 18

3.1 Introduction 18

3.2 Proxy-label Training 21

CHAPTER 4 CONFIDENCE LEARNING THEORY 22

4.1 Introduction 22

4.2 Theorems and for Confident Learning 23

CHAPTER 5 LABEL CONFIDENCE ASSISTED MODEL TRAINING FOR OA ASSESSMENT 26

5.1 Estimating Label Confidence 26

5.2 Interactive Training with Label Confidence Information 30

5.3 Hybrid Loss Function 32

CHAPTER 6 DATA PREPROCESSING AND EXPERIMENT SETUP 35

6.1 The OAI Public Dataset 35

6.2 Knee Joint Area Segmentation 35

6.3 Training Scheme Implementation 43

6.4 Model Performance Evaluation 43

CHAPTER 7 RESULTS AND DISCUSSIONS 45

7.1 OA Severity Assessment 45

7.1.1 The Five-stage Task 45
7.1.2 The early-stage tasks 49
7.2 Label Confidence Estimation 52
7.2.1 Low Confidence Sample Characterization 52
7.2.2 Label Confidence Estimation Process 55
7.3 Effects of Hyper-parameters 57
7.3.1 Effects of warm-up epoch 57
7.3.2 Effects of λ in hybrid loss 58
CHAPTER 8 CONCLUSION 61
APPENDIX DATASET ACKNOWLEDGMENT 62
REFERENCES 63
BIOGRAPHICAL SKETCH 69
CURRICULUM VITAE

LIST OF FIGURES

1.1 Computer-aided OA diagnosis pipeline 3
2.1 Architecture of AlexNet 10
2.2 2-D Convolution and Pooling Operation 12
2.3 Resnet Block Architecture 13
2.4 Densenet Block Architecture 14
3.1 Self-Training Procedure 20
5.1 The overview of the training stage in the proposed scheme 27
5.2 The interaction between the training and validation stages 30
5.3 Comparisons of targeting distributions in classification tasks 33
6.1 The IOU score for each batch (iteration) during training 36
6.2 Detection count per image 38
6.3 Size of the original images and detections 39
6.4 Locations of knee detections 40
6.5 Object confidence score distribution (by YOLO model) 41
6.6 Knee joint areas located by the fine tuned YOLOv2 model 42
7.1 Metrics of 5-class task on each fold 47
7.2 Comparison of activated region via GradCAM 48
7.3 Metrics of different methods on each fold 51
7.4 Low confidence samples verified by individual annotator 54
7.5 Estimated confidence level after each epoch 56
7.6 Average training loss after each epoch 59
7.7 Averaged label confidence of KL-1 regarding different λ 60

LIST OF TABLES

1.1 Definition of Kellgren-Lawrence Grading System 2
5.1 Symbols Commonly Used in This Paper 28
6.1 Label Distribution of the Cropped ROI 43
7.1 Accuracy of the five-class task 46
7.2 Comparison of the accuracy on early-stage tasks 50
7.3 Comparisons of the Performance under Random Label-Shifting (LS) Conditions (five-class task, the densenet121 model) 55
7.4 Comparisons of Different Warm-up Epoch 57
7.5 Comparisons of Different λ 58

CHAPTER 1

INTRODUCTION

Knee osteoarthritis (OA) is a global chronic disease characterized by an irreversible degenerating process of the knee cartilage. According to the World Health Organization (WHO), 9.6% of men and 18% of women over 60 years can have symptomatic osteoarthritis (Wittenauer et al., 2013). As a leading cause of adult disability (Allen and Golightly, 2015), OA will affect at least 130 million people due to the aging population (Maiese, 2016). In clinical scenarios, risk factors such as body mass index, age, and sex (Kellgren and Lawrence, 1957) can be used to assess OA. However, as symptoms may not appear in the early stages of OA (Palazzo et al., 2016), doctors depend on medical imaging modalities for diagnosis. In particular, X-ray imaging is the most common technique due to its affordability and accessibility.

Based on the radio-graphical evidence such as osteophyte and narrow joint space, Kellgren and Lawrence proposed a grading system in 1957 (Kellgren and Lawrence, 1957) as indicated in Kellgren-Lawrence (KL) grading is the most commonly used classification system, which categorizes OA severity into five levels. Early-stage (KL-1 or KL-2) patients can take preventive measures, including exercises and weight control, to manage the degeneration process (Ryan, 2020). In late stages (KL-4), the only treatment is artificial joint replacement (van der Woude et al., 2016). Thus, it is critical to diagnose OA in early stages and to monitor the severity through the patients' life. Considering the potential demand for OA assessment, past studies developed automatic OA assessment methods to promote the efficiency of OA diagnosis and reduction of labor cost.

Part of this chapter is reprinted from © 2021 IEEE. Reprinted, with permission, from Y. Wang et al., "Learning from Highly Confident Samples for Automatic Knee Osteoarthritis Severity Assessment: Data from the Osteoarthritis Initiative," in IEEE Journal of Biomedical and Health Informatics, doi: 10.1109/JBHI.2021.3102090.

Table 1.1: Definition of Kellgren-Lawrence Grading System

Grade	Remarks
KL-0	No evidence of osteophyte
KL-1	Doubtful osteophyte
KL-2	Definite osteophyte; possible Joint Space Narrow (JSN)
KL-3	Moderate osteophytes, definite JSN, some sclerosis and possible deformity of bone ends
KL-4	Large osteophytes, definite JSN, sclerosis, and deformity of bone ends

The automatic OA severity assessment is challenging for two reasons: 1) The lesion area occupies a small portion of the X-ray image. The irrelevant parts like clothes, tissues, or muscles overwhelm the cartilage status and mislead final decisions. 2) As bones differ in shape and density from one to another, it is challenging to establish standard diagnostic criteria. A well-trained radiologist assesses the severity of OA based on personal experiences, which are difficult to be incorporated into the computer-aided system.

Recent machine learning-based research introduced two stages shown in Figure. 1.1 for automatic OA diagnosis: 1) The region of interest (ROI) segmentation, which suppresses noises by removing the background and irrelevant parts. 2) The machine learning-based OA severity classification, which standardizes and simplifies complicated diagnostic criteria. Knee segmentation is a challenging task, as OA lesion areas usually occupy a small portion in the original X-ray image. Shamir et al. (2008) successfully built a two-stage framework, including template matching for knee joints detection and a nearest neighbor classifier for severity estimation. To elaborate, the authors slid a window over the X-ray image and calculated the Euclidean distance between the pixels within the window and 20 pre-determined templates. The smallest distance determined the region of interest (ROI). Based on the pixel statics and digital signal transformations of the ROI, the nearest neighbor classifier distinguished samples of different KL levels. Later studies followed the same paradigm. Antony et al. (2016) introduced a support vector machine (SVM) to improve

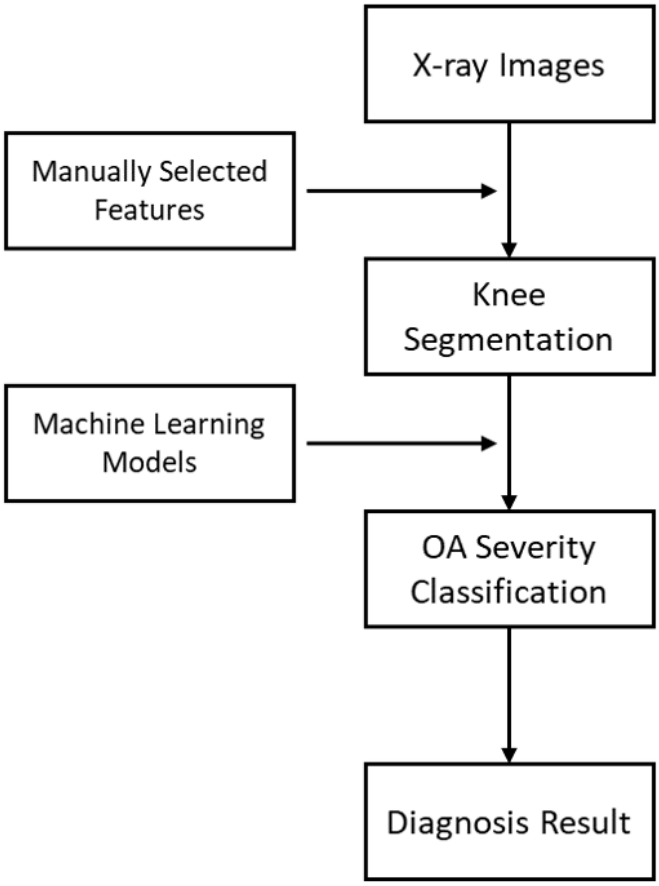


Figure 1.1: Computer-aided OA diagnosis pipeline. The entire process consists of two stages, including knee segmentation and OA severity classification. Typically, the knee segmentation depends on human efforts or manually selected features. Then, a machine learning model, like CNN, is trained to classify the knees based on the OA severity grades.

the accuracy of ROI detection. Other researchers used SVM (Sharma et al., 2016), neural network (Christodoulou et al., 2019), and random forest (Aprilliani and Rustam, 2018) to enhance the severity classification performance.

In the light of deep learning, the convolutional neural network (CNN) has been successfully applied to the medical imaging field for segmentation and classification (Yamashita et al., 2018; Lundervold and Lundervold, 2019). Recent studies of OA severity assessment proposed end-to-end approaches based on deep CNNs that improved both feature extraction efficiency and classification accuracy. Antony et al. (2017) proposed a method involving

two CNNs. The first CNN detected the knees' contour, and the second CNN used contents therein as inputs for classification. To segment knee joint areas, later studies(Chen et al., 2019) and (Suresha et al., 2018) employed the object detection CNNs like YOLO (Redmon and Farhadi, 2017) and RCNN(Ren et al., 2015). To extract better features, researchers proposed different learning tasks. For example, Tiulpin et al. (2018) proposed a Deep Siamese Network that learned the features from lateral and medial sides separately and fused them together for classification. The authors also extracted features from non-image data, including the health records of patients (Tiulpin et al., 2019). Nasser et al. (2020) used a deep auto-encoder with a discriminative regularization term in loss function, which helped the encoder maximize the distances between early-stage OA samples in the feature space.

However, OA severity assessment is still challenging for deep learning models. As shown in Table 1.1, the KL grading system is semi-quantitative. Suppose an image has significant evidence as listed all annotators will give a consistent KL grade, which indicates a high confidence in such a sample. Otherwise, when two or more different KL grades are assigned to the same image by each annotator, these samples and their labels are less confident. Culvenor et al. (2015) used two statistical measurements to describe the uncertainty in the KL grading system, including inter-observer and intra-observer reliability. Inter-observer reliability is to measure the agreement of ratings given by different annotators, while the intra-observer reliability measures the agreement of ratings given by the same person. For the KL grading, the inter-observer reliability is low (0.67), which confirms the existence of low confidence samples, whereas the intra-observer reliability is high (0.97), indicating the annotators' reliance personal experiences to make decisions. The Osteoarthritis Initiative (OAI) resolved this issue by introducing a third independent annotator. However, deep learning models treat all images and labels as equally confident in plain training. To some extent, deep CNNs are robust against data uncertainty (Drory et al., 2018). However, the uncertainty in the dataset can affect the later training epochs(Ma et al., 2018). Further,

when training on the typical data, CNNs will not memorize the training samples(Arpit et al., 2017). The aforesaid empirical studies indicate that if focusing on the OA samples with high confidence labels, CNNs can grain better generalization capability on unseen data. Noticing the label uncertainty, the authors used the Mean Squared Error (MSE) as the loss function to simulate the transition of KL levels (Antony et al., 2016). Chen et al. (2019) and Nasser et al. (2020) focused on discriminating the ambiguous samples by re-weighting the loss function. These studies do not refine the dataset regrading the confidence level of samples.

Uncertainty of labels and samples is one of the significant difficulties in the medical imaging field (Karimi et al., 2020). Intuitively, highly confident data can improve deep learning model performance. This strategy has been employed by recent studies when learning from uncertain annotations. Xue et al. (2019) used a label suppression approach for skin lesion classification. Samples with high loss values in each mini batch were considered uncertain and they were discarded during the back-propagation. Mirikharaji et al. (2019) prepared a small clean dataset for pre-training when handling the skin lesion segmentation task, so that the pre-trained model can generate an optimal pixel-level weight map, which helps with the training on the large-scale uncertain dataset. Their approach enhances the robustness of segmentation.

In this paper, we follow the two-step scheme for OA severity assessment by employing an object detection CNN to segment knee joint areas. Notably, we focus on the label uncertainty and propose a novel approach that helps the model to learn from the highly confident samples. Our contributions can be summarized as follows:

- We propose an integrated learning scheme which fuses label confidence estimation to characterize highly confident samples. The whole training process is self-boosting and entirely data-driven.

- We propose a hybrid loss function that emphasizes the importance of highly confident samples. To reduce the impact of empirical errors, we do not discard the low confident samples but control their impacts with a weight parameter.

- The experiment results show that we achieve a state-of-art performance on the five-class OA assessment task. Without human intervention, our method is competitive with the semi-automatic approach to early-stage OA detection task. We also verify the low confidence sample characterization by the case study and manual noise interference experiment.

This paper is organized as follows. Firstly, we introduce the related fields of our work, which consists three aspects. Chapter 2 summarizes the recent development of deep learning models the image classification and the object detection, which are the techniques used in the automatic knee OA assessment. Chapter 3 introduces the semi-supervised learning, especially the proxy-label training. Semi-supervised learning is the foundation of the proposed training scheme. Chapter 4 describes the label confidence theory, which provides the theoretical support to the proposed method. Then, Chapter 5 gives the details of the proposed method, including Chapter 5.1 for estimating label confidence, Chapter 5.2 for interactive training, and Chapter 5.3 for the proposed hybrid loss function. Next, Chapter 6 introduces the dataset preprocessing and experiment setup, including the data collection (Chapter 6.1), knee joint segmentation (Chapter 6.2), training scheme implementation (Chapter 6.3), and performance evaluation metrics (Chapter 6.4). Especially, as we used object detection in the preprocessing, the verification of the preprocessing is also included in Chapter 6. Finally, Chapter 7 shows the comprehensive results of the proposed training scheme in three aspects. Chapter 7.1 demonstrates the classification performance of the proposed method. Chapter 7.2 examines the label confidence characterization process. Chapter 7.3 is the ablation study of the hyper-parameters in the proposed scheme. To the end of this paper, we draw the conclusion in Chapter 8.

CHAPTER 2

DEEP LEARNING METHODS ON MEDICAL IMAGE ANALYSIS TASKS

2.1 Introduction

Imaging analysis is a vital technique used in the clinical environment. Common medical imaging modalities includes computed tomography (CT), magnetic resonance imaging (MRI), positron-emission tomography (PET), X-ray, and ultrasound imaging (UI). Medical imaging analysis tasks varies due to the diversity of medical imaging modalities and the desired output. However, fundamental medical image analysis tasks can be summarized from the following aspects (Zhou et al., 2021).

- Medical image reconstruction produces a visual representation for analysis from the electric or radio signals (Ahishakiye et al., 2021). For example, computerized tomography (CT) employs X-ray to reveal the details of vessels, bones, and soft tissues inside the body. Scanning results are converted into a plain image for analysis.

- Medical Image enhancement is to adjust images for specific visualization purposes. By improving image quality, clinicians can make better diagnosis and treatment plan decisions. For example, researchers developed histogram algorithms (Salem et al., 2019) to improve the contrast of X-ray images. Thus, doctors can have a clear view of the bone shapes.

- Medical image segmentation (Wang et al., 2020) is to highlight the target object or lesion areas. There are two segmentation categories, including the object level and pixel level. Object-level segmentation gives the bounding box of a target, while pixel-level segmentation aims to find all target pixels.

Part of this section is reprinted from © 2021 Y Wang et al. "An Automatic Knee Osteoarthritis Diagnosis Method Based on Deep Learning: Data from the Osteoarthritis Initiative", Journal of Healthcare Engineering, vol. 2021. https://doi.org/10.1155/2021/5586529.

- Medical image registration (Oliveira and Tavares, 2014) aligns the images obtained from various sources with the same coordinate system. In clinical, image deformation comes from different sources. On the one hand, medical image data taken for the same person involves different sensors, depths, and viewpoints. On the other hand, body movement like breath results in the deformation of exam outcomes. Registration helps integrate data from different sources and suppress the deformation.

Manually analyzing medical images requires intense human efforts and a large amount of time. To improve the efficiency and reduce the human intervention, researchers developed an automatic method to assist the medical image processing. However, developing a universal method for automatic medical image processing remains challenging for the following reasons.

- Modality diversity. Medical images are generated by different mechanisms such as CT and MRI, which causes the modality diversity. Handling different modalities requires human intervention and application-specific algorithms.

- Resolution density. Medical images have a high density in resolutions. For example, CT has the spatial resolution of millimeter level, resulting in an image of tens of million pixels. In addition, MRI images usually store the grayscale data in 12-bit to preserve a high dynamic range. On the other hand, typical computer images use 8-bit pixels. High-resolution density leads to extra computing burdens to traditional image processing methods.

- Standard variance. Due to the lack of a universal standard, existing medical images from different sources or equipment follow different protocols. Integrating the medical data of multiple standards requests numerous efforts to develop a preprocessing pipeline to bring all images to the same condition.

- Disease patterns. Evidence shown in medial images follows a long-tail distribution (Zou et al., 2004). While doctors can depend on personal experiences, it is hard to establish a golden standard for computer-aided analysis algorithms.

- Label absence and noise. Due to the high cost of manually labeling, large potions of medical images remain unlabeled. Alternatively, the labels given by crowd workers or the coarse machine learning models contain mistakes and errors, which increases the noise level. Thus, efficient noise suppression is essential before developing an automatic medical image method.

Recently, high level applications like computer aided diagnosis takes advantage of deep learning to overcome the above problems. This chapter gives a review of two related aspects where the deep learning methods significantly influence the computer aided diagnosis. Section 2.2 reviews the automatic diagnosis based on images, which is usually considered as a classification tasks. Section 2.3 reviews the lesion segmentation tasks, which is an essential preprocessing step before diagnosing.

2.2 Deep Learning Methods for Image Classification

Typically, researchers consider the diagnosis problem as a classification task. In the light of deep learning, CNNs are becoming a prevalent image classification technique. While there are several early implementations of convolution neural networks (CNN) (LeCun et al., 1998), the AlexNet (Krizhevsky et al., 2012) demonstrates the common computational components and hardware training schedule of modern deep learning models. On the one hand, AlexNet is trained on GPUs which showed the computational power impact on modern machine learning field. On the other hand, it builds up the common architecture, including a stack of convolutional blocks for feature extraction and several dense layers for classification tasks as shown in Figure. 2.1. Each convolution block has three categories of layers including

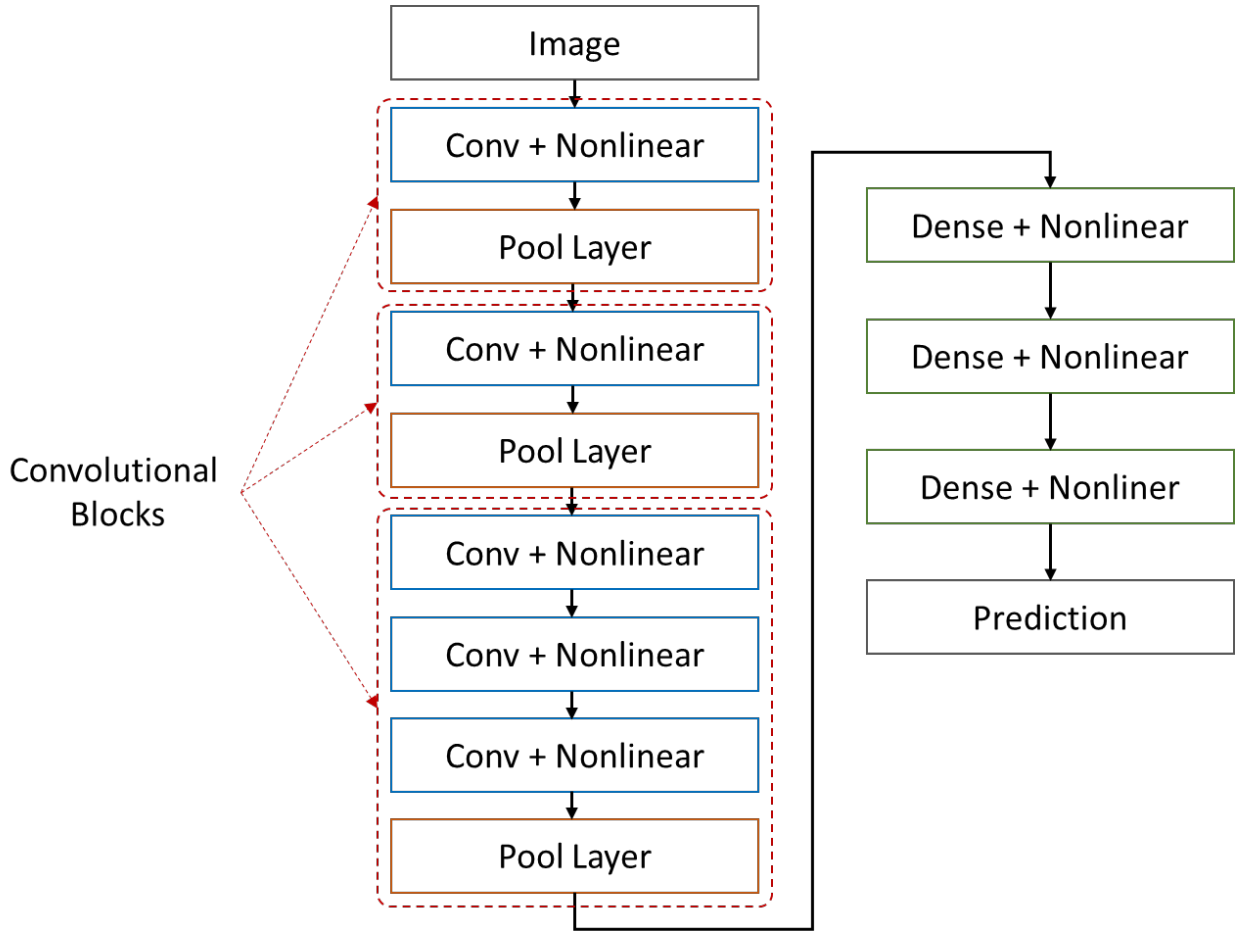


Figure 2.1: Architecture of Alex Net. There are three major components in this architecture, convolution layer, pooling layer, and the dense layer. In this figure, the convolution and its following nonlinear(activation) are shown in the blue box. One or more convolution layers and a pooling layer can be grouped together as the a convolutional block, which is the basic unit in later CNN architectures.

convolution layers, activation layers, and a pooling layer. A convolution layer performs 2-D convolution as shown in Equation 2.1

$$y[m,n] = \sum_{h=0}^{H} \sum_{w=0}^{W} k[h,w]x[m-h,n-w] \tag{2.1}$$

, where m, n are the location of inputs and outputs, and H, W are the height and width of convolution kernel. Weights of the convolution kernels are learned during the training.

Followed each convolution layer, a activation layer applies an nonlinear function to the convolution outputs. The idea of activation function roots in the neural network where it provides the nonlinear mapping between inputs and outputs. Typical activation functions include the sigmoid function (Equation 2.2) and tanh function.

$$\sigma(x) = \frac{1}{1 + \exp -x} \tag{2.2}$$

Recent researches propose several new activation functions to improve the training efficiency of large-scale deep network. For example, Agarap (2018) proposed ReLU function as shown in Equation 2.3.

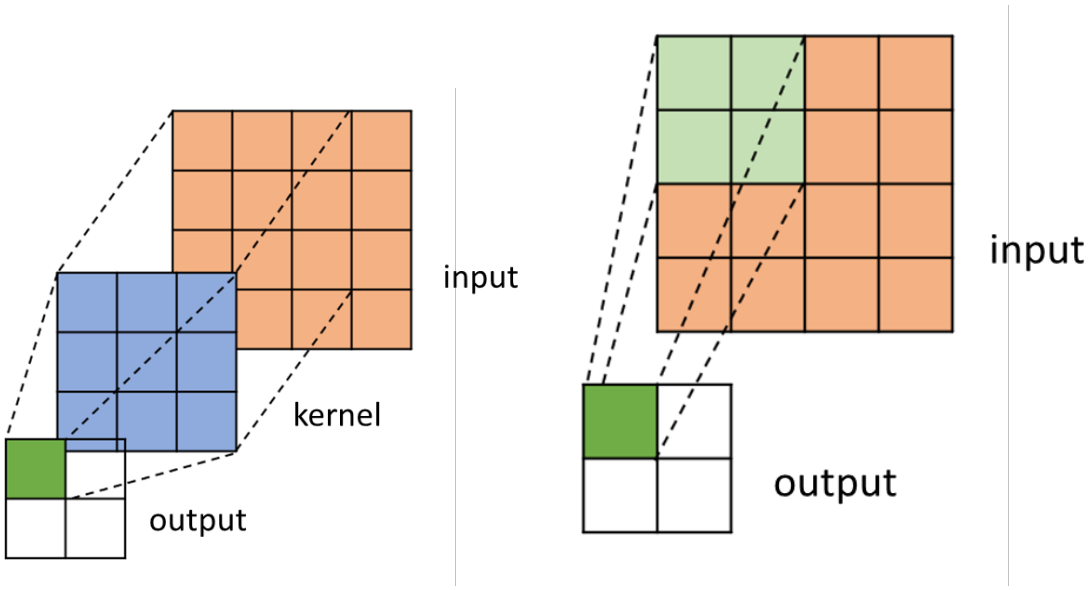
$$f(x) = \begin{cases} x & \text{for } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

Similarly, other researches proposed ELU(Clevert et al., 2015), CELU (Barron, 2017), and SELU(Klambauer et al., 2017).

The pooling layer performs a 2-D reduction operation as in Equation 2.4.

$$y[m,n] = \text{Pool}(x[m-h,n-w]), h \in [0,H_p], w \in [0,W_p] \tag{2.4}$$

Usually, the Pool function is either maximizing or averaging. Pooling layer aggregates the information of each window as shown in Figure. 2.2b. The convolutional blocks transforms and extracts features from the original image. Those features are embedded in a multi-dimensional tensor.



(a) 2D convolution with a 3×3 kernel

(b) 2D Pooling with the pool size as 2 and stride of 2

Figure 2.2: 2-D Convolution and Pooling Operation. As shown in Figure. 2.2a, a convolution layer uses a window to slide over the entire image and computes the outputs based on inputs and a kernel in each position. The kernel is a learnable weight matrix that is determined through training. Figure. 2.2b shows the pooling operation. Similar to the convolution, a pooling layer also uses a sliding window. However, there are no learnable parameters; and the output is computed from inputs only using maximizing or averaging.

Dense layers are conventional multi-layer neuron networks which take the feature embeddings as inputs and gives the classification results. A dense layer performs the dot product between inputs and learnable weights. Similar to the neuron networks, each dense layer also has a corresponding nonlinear activation layer.

The deep convolution neuron network show superior performance on image classification tasks. While the increasing depth improves the modeling capability, it also brings difficulty to the training, such as gradients vanishing. To exploits the power of network depth, following researches focused on enhance the computation stability and network architecture. Regarding the computational stability, Ioffe and Szegedy (2015) proposed the batch normalization technique, which is widely used in the CNN to accelerate the training. In term of

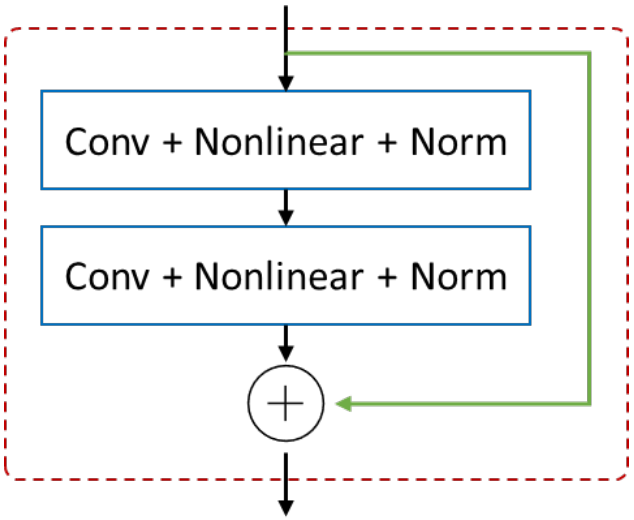


Figure 2.3: Resnet block architecture. In each resnet block, inputs has two paths. One goes through a stack of convolution, activation and normalization layers. The other one bypasses the above layers. At the end of each resnet block, two paths are merged by addition. Following the two-path fashion, Resnet block has many variants (He et al., 2016b). In this figure, the skip connection is marked as ghe green line.

network architecture, Resnet (He et al., 2016a) introduces the skip connection to the convolution blocks as shown in Figure. 2.3. The skip connection adds a second path of the information. At the end of each convolutional block, two branches are merged by addition operation. Follow the similarly idea, DenseNet (Huang et al., 2017) exploits the power of skip connection by merging the information from different convolutional layers as in Figure. 2.4.

The core learnable components for feature extraction component in CNN are the convolution layers. However, spatially distant feature extraction is challenging for a traditional CNN(Wu et al., 2020), of which convolutional filters only receive the information from a local region. For example, according to clinical experiences, primary evidences of OA, like the joint space narrowing, appear on both sides of a knee. As a result, the extracted feature maps do not address the relationship between different local regions. Researchers designed new convolution operators in the computer vision domain; for example, Yu and Koltun (2015) proposed the dilated convolution. For OA diagnosis, Tiulpin et al. (2018) divided the knee

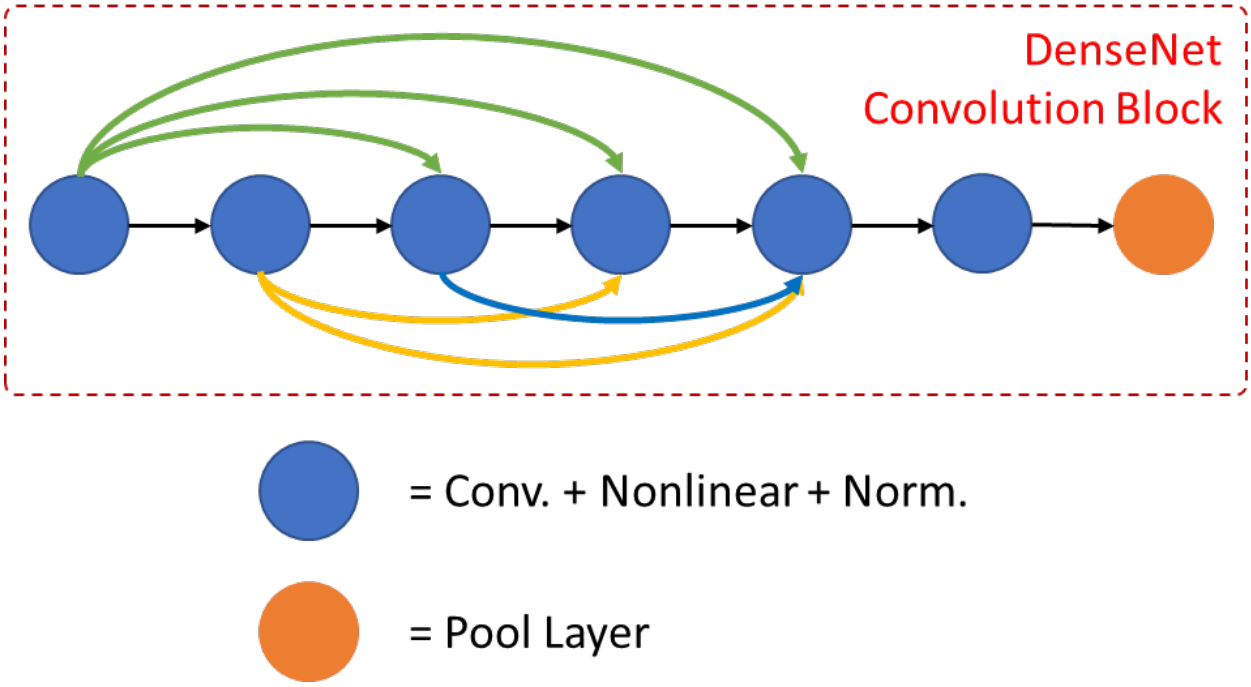


Figure 2.4: Densenet block architecture. The densenet block exploits the shortcut connection. The outputs of one convolutional layer are sent to all following convolutional layers within the same block as inputs. When joining multiple paths, densenet block uses concatenation instead of addition. In Figure. 2.4, skip connections from different convolutional layers are marked as green, yellow and blue lines, correspondingly.

joint areas into the left and right sides. Then, the authors used two CNNs to extract the features separately and fused them for classification.

Recently, Dosovitskiy et al. (2020) proposed the visual transformer, which takes advantage of relations between different local regions to boost the performance on multiple visual tasks. Transformer Vaswani et al. (2017) was first applied to the natural language procession (NLP) field based on the self-attention mechanism. In the implementation, inputs are first encoded into three components: the query, the key, and the value. Then the value is weighted by a mask calculated from the query and the key as in Eq. 2.5

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{Q \cdot K^T}{\sqrt{d_k}}) \cdot V \tag{2.5}$$

, where the Q, K, and V denotes the query, key, and value components. And d_k is the dimension of key components. For an NLP task, Q, K, and V are the sequences of feature

vectors extracted from each word in a sentence. The dot product of Q and K calculates the correlations between each pair of words in the sentence. Then, the softmax function normalizes the correlation and applies it to V as attention weights. In this case, only the features of the highly related words are emphasized. For a visual task, Dosovitskiy et al. (2020) took advantage of the self-attention mechanism and proposed the visual transformer. Images were divided into patches and reorganized into a sequence. Eq. 2.5 uncovers the relationship between each pair of patches by calculating the correlation, even if the two patches are distant in the original image. Additionally, Vaswani et al. (2017) and Dosovitskiy et al. (2020) applied the "multi-head" to the self-attention mechanism. The multi-head technique contains multiple parallel self-attention layers, enhancing the capability to exploit more specific locations simultaneously.

In medical field, enormous research applied the image classification by deep CNN into disease diagnosis especially when a large-scale dataset is available. Moreover,

2.3 Deep Learning Methods for Object Detection

In recent years, object detection CNNs help with locating and recognizing ROIs in plain images. Well-known object detection networks include faster R-CNN (Girshick, 2015), mask R-CNN(He et al., 2017), and YOLO(Redmon et al., 2016). Generally, object detection CNN includes a CNN backbone and two separate branches to determine the object class and location. CNN backbone can be implemented as the architectures mentioned in Chapter 2.2 without the last dense layers. Both the classification and the location branches are the small sub-network which utilize the features extracted by the CNN backbone to predict the class label and object coordinates. A common technique shared by all three approaches is bounding box regression (Girshick, 2015)(Redmon et al., 2016), which handles the object's location and size separately. The bounding box is a rectangle which tightly covers one object. Each object's location and size are determined by the center coordinates of its

bounding box, denoting as $x_{\text{obj}}, y_{\text{obj}}$, and the bounding box height and width, denoting as $h_{\text{obj}}, w_{\text{obj}}$. In general, CNNs do not directly predict location and size but retrieve them through a decoding process from the outputs. For example, the YOLO model relies on two additional information to decode its outputs for locations and sizes, respectively. For object locations, Redmon et al. (2016) introduced a reference grid system, which evenly divide entire image into small squared regions. Each grid has corresponding outputs x_{out} and y_{out} from the YOLO model. The predicted object center coordinates x_{pred} and y_{pred} are retrieved from Equation 2.6

$$\begin{cases} x_{\text{pred}} & = \sigma(x_{\text{out}}) \times l_{\text{grid}} + x_{\text{grid}} \\ y_{\text{pred}} & = \sigma(y_{\text{out}}) \times l_{\text{grid}} + y_{\text{grid}} \end{cases} \tag{2.6}$$

, where σ is the sigmoid function, l_{grid} is the grid edge length, and x_{grid}, y_{grid} are the top left corner coordinates of one grid. As the sigmoid function in Equation 2.6 restricts outputs to the range of $(0, 1)$, a grid only predicts the objects within the region. However, YOLO model detects an object at arbitrary location by applying Equation 2.6 to all grids. Regarding object sizes, the YOLO outputs are decoded referring to the "anchors". Anchors are the prior knowledge of the object size determined by clustering the sizes of all training objects. In practice, the YOLO model used cluster centroids as the expected shapes of an object. The outputs of YOLO, denoting as w_{out} and h_{out}, calibrate the anchors to retrieve the predicted size w_{pred} and h_{pred} using Equation 2.7

$$\begin{cases} w_{\text{pred}} & = w_{\text{anchor}} \times \exp(w_{\text{out}}) \\ h_{\text{pred}} & = h_{\text{anchor}} \times \exp(h_{\text{out}}) \end{cases} \tag{2.7}$$

, where the w_{anchor} and h_{anchor} are the width and height of the anchor. With Equation 2.6 and Equation 2.7, the YOLO model solves the detection problem as a regression task. Equa-

tion 2.8 computes the regression errors as follows.

$$\begin{cases} x_{\text{err}} & = (x_{\text{obj}} - x_{\text{pred}})^2 \\ y_{\text{err}} & = (y_{\text{obj}} - y_{\text{pred}})^2 \\ w_{\text{err}} & = (\sqrt{w_{\text{obj}}} - \sqrt{w_{\text{pred}}})^2 \\ h_{\text{err}} & = (\sqrt{h_{\text{obj}}} - \sqrt{h_{\text{pred}}})^2 \end{cases} \tag{2.8}$$

Notably, the errors in Equation 2.8 only account for the grids which contains ground truth object. To distinguish the objects from background, the YOLO model is also trained by a confidence error as in Equation 2.9

$$\text{conf}_{\text{err}} = \mathcal{L}_{\text{BCE}}(\sigma(p_{\text{out}}), p_{\text{obj}}) \tag{2.9}$$

, where $\mathcal{L}_{\text{BCE}}$ is the binary cross entropy loss function, p_{out} is the confidence output of each grid, and p_{obj} is an indicator of object existence. Particularly, if a grid covers at least one object center, the corresponding p_{obj} is set to 1. Otherwise, p_{obj} is set to be 0.

CHAPTER 3

SEMI-SUPERVISED LEARNING

3.1 Introduction

While the deep learning architectures have evolved in recent years, training a model with supervised learning methods is still challenging given limited labeled data and high label noise level. Semi-supervised learning refers to a training strategy for deep learning models. Unlike supervised learning, dataset for semi-supervised learning (SSL) contains both labeled data and unlabeled data for training. SSL leverage the difficulty caused by limited labeled samples. The motivation of SSL is to take advantage of information in unlabeled samples and improve the model performance. Typically, the semi-supervised learning contains several stages. Firstly, the model is trained on the labeled samples. Then, the trained model generates the labels for the unlabeled data. Next, a new model is trained on both given labels and generated label. The entire training procedure can be iterative until the above second and third stages converge.

The success of semi-supervised learning methods is based on several important assumptions (Ouali et al., 2020) that are summarized as follows.

- The smooth assumption, which means if two samples are close to each other in the input space of a model, the corresponding outputs should belong to the same cluster.

- The cluster assumption, which means if two samples (or their corresponding representations in feature space) are in the same cluster, they are likely to have the same label.

- The manifold assumption, which means the samples of high-dimensional representations can be roughly mapped to a low-dimensional manifold.

The above assumptions given the confidence of the confidence of the generated labels. Per application, the above assumptions are not always held for the dataset. Therefore it is necessary to analyze the application and the dataset before performing SSL.

Taking advantage of the unlabeled data, the SSL has been successfully applied in a wide range of practical applications. Especially, when labels are costly to obtain, the SSL provides a self-boost way to improve the model performance. The majority aspects of the applications are summarized as follows.

- Active Learning (Settles, 2009). The motivation of active is the let the model select the training sample for learning. In this paradigm, model training launches on a small potion of label data. Then the unlabeled sample are automatically evaluated by the information. In this case, the demand of new labels can be fulfilled by only annotating the most informative samples.

- Domain Adaption (Weiss et al., 2016). For a new application, the dataset may contains an extreme small number of labeled samples, which cannot support training complex models. If a related problem already has a large-scale labeled dataset, SSL helps with transfer the model from one to another efficiently.

- Weakly-Supervised Learning (Ratner et al., 2019). In the weakly-supervised scenario, labels are of large-scale but low-quality, such as the medical dataset annotated by crowed workers. The target of weakly-supervised learning is the same as supervised learning. However, in this case, the confidence of labels is unknown. The SSL helps with examining, refining the labels as well as training a high performance models.

As summarized above, weakly-supervised learning can be easily extended to noisy label learning. As mentioned in the last chapter, the sources of the label noise include the ambiguity of samples and the mistakes made by annotators. The label noise can be modeled

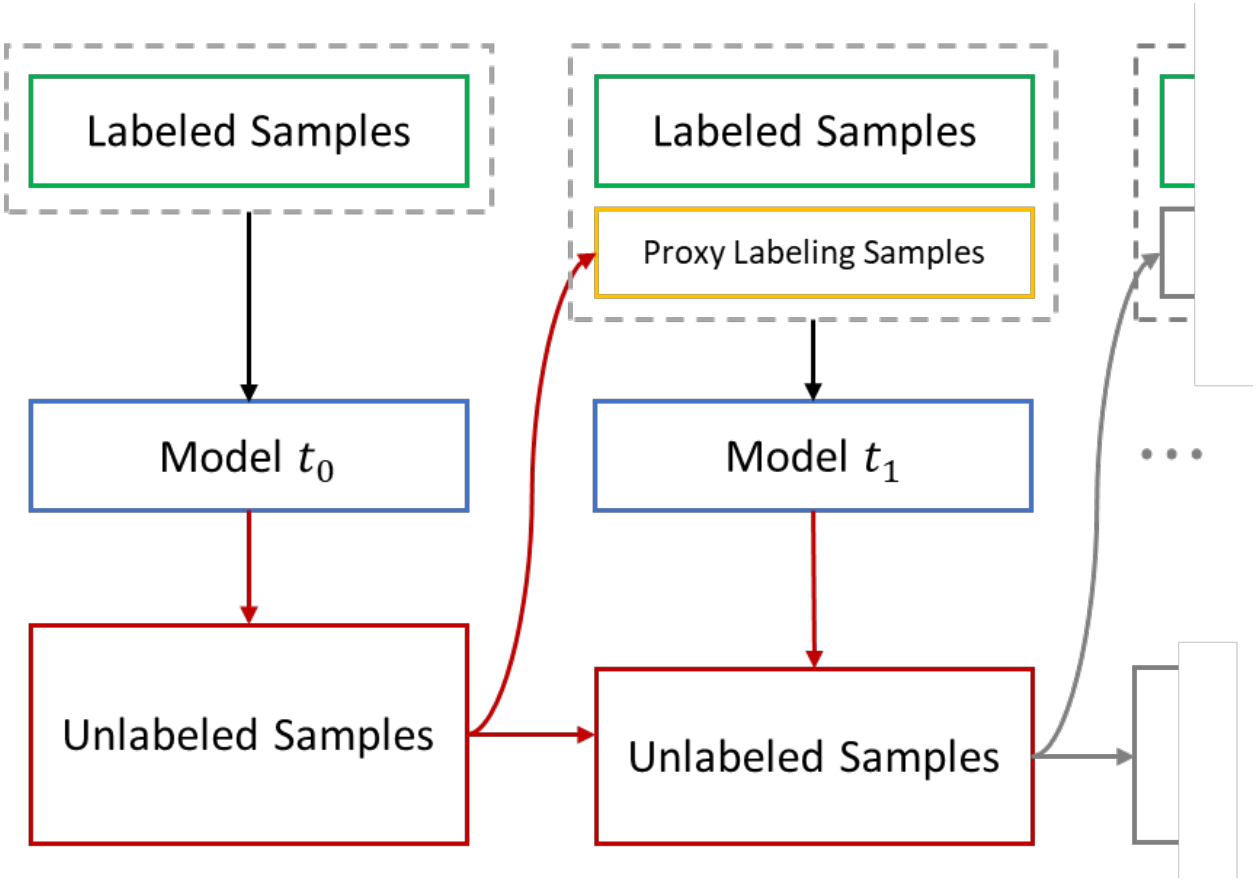


Figure 3.1: Self-training through proxy-labeling. Self-training is self-boost training scheme. In the first epoch, a model t_0 is trained on the labeled samples. Then t_0 predicts the labels on the unlabeled samples. The predictions are carefully selected in which the reliable ones are used for training in the following epoch. In each epoch, a model learns from all available labels including the given labels and the proxy-labels.

as the confidence level from the statistical view. The SSL can help characterize the high confidence samples and regularize the deep learning models.

Several methods to implement SSL include regularizing consistency, generative models, and graph-based models (Ouali et al., 2020). In the following section, we reviewed a method called proxy-labeling, closely related to handling the label noise problem.

3.2 Proxy-label Training

The proxy-label method takes advantage of the unlabeled data by two iterative stages in one training epoch. In the first stage, a deep learning model, the teacher model, is trained on the labeled data as normal. Then the trained model annotates the unlabeled data, which is called a proxy label. In the second stage, a student model learns both the given and the proxy label. Even though the proxy labels are weak and noisy, they provide extra information beyond the original labeled data. Based on behaviors of the above stage, the proxy-label method has two major training paradigms.

- Self-training. In a self-training scheme, the dataset is shared between training stages. As shown in Figure. 3.1, the proxy labels are the target for both teacher and student models. A self-training scheme is a self-boosting procedure. The proxy label's errors and bias in any stage affect the following training. To prevent the bias brought by the proxy labels, the self-training scheme only adds the most confident labels to the training set of the next epoch. Self-training scheme is straightforward to implement and has been successfully applied to the real problems.

- Multi-view training. In a multi-view training scheme, labeled samples are split into subsets, or "views", where any view contains sufficient information for model training. Each model in the multi-view training learns from one view of the dataset. In this case, the training of the models runs in parallel. Then each model annotates the unlabeled samples independently. In this case, each unlabeled sample has multiple proxy labels, which helps elect and refine.

The proposed method in this paper estimates the proxy labels but does not directly apply the proxy labels into training. Specifically, the proposed method exploits the proxy label probabilities to estimate the uncertainty in the dataset, which is introduced in the next chapter.

CHAPTER 4

CONFIDENCE LEARNING THEORY

4.1 Introduction

Nowadays, large-scale datasets with label noise are becoming increasingly common. For medical datasets, labeling issues are even worse when the golden assessment standards are missing. While doctors diagnose disease according to personal experiences, the ambiguity of labels causes difficulty in training a deep learning model. Confidence learning (Northcutt et al., 2021) is proposed to solve the issues of label uncertainty issue. The goal of confidence learning is to build a data-driven approach to learning theoretically and experimentally from uncertain data.

Early studies (Elkan, 2001; Forman, 2005) of label noise started with the binary classification. To analyze the noise ratios, the authors estimated the false positive and false negative rates. Forman (2008) introduced the threshold technique against the epistemic error of predictions. Regarding the learning process, the re-weighting loss function (Natarajan et al., 2013) addressed the issues of random classification noise. In general, the studies in the confidence learning field proposed three effective principles to tackle the label uncertainty as follows.

- Finding label errors with pruning or threshold (Chen et al., 2019).
- Re-weighting the loss function during training to prevent the error-propagation (Natarajan et al., 2017).
- Ranking the sample by order of confidence level to establish a robust learning process (Jiang et al., 2018).

The latest confidence learning studies integrate all the above principles to characterize the label uncertainty in the dataset, which are shown in the next section.

4.2 Theorems and for Confident Learning

The theoretical support of our work is based on the analysis of Northcutt et al. (2021). Past observations showed that the label uncertainty follows the class-conditional process assumption (Angluin and Laird, 1988). The class-conditional assumption states that certain class $i \in \mathcal{L}$ may be independently labeled as class $j \in \mathcal{L}$, where $\mathcal{L}$ is the class label set. Considering the labels as random variables, we can build a joint distribution to model the mislabeling from the possibility theory. Confident learning aims to estimate the joint distribution $Q_{\bar{y},y^*}$ then filter out the uncertain labels with $Q_{\bar{y},y^*}$, where $\bar{y}$ denotes the noisy label and y^* denotes the true but unknown label. There are three steps in the confident learning procedure.

- Step 1: Estimate the $\hat{Q}_{\bar{y},y^*}$ to characterize the label confidence level, where $\hat{Q}_{\bar{y},y^*}$ is a constant estimator of $Q_{\bar{y},y^*}$.
- Step 2: Characterize the low confident samples with the information in $\hat{Q}_{\bar{y},y^*}$.
- Step 3: Re-weight each class in the learning process after the noisy labels are removed.

Notably, in the first step, $\hat{Q}_{\bar{y},y^*}$ replaces $Q_{\bar{y},y^*}$ because y^* is unknown such that we can only find an estimator during the learning process.

Estimating $\hat{Q}_{\bar{y},y^*}$ requires the prediction probability matrix $\hat{P}_{k,i}$ of each sample, where the k-th sample is predicted as class i with the probability of $\hat{p}_{k,i}$. Usually, $\hat{P}_{k,i}$ is given by a CNN model $\mathcal{M}$ after the warm-up training. To estimate $\hat{Q}_{\bar{y},y^*}$, we count the samples according to the probability of predicted label it belongs to. In this case, noisy samples are grouped by their latent label y^*. A 2-D confident joint matrix $\hat{C}_{\bar{y},y^*}$ sums the counting results according to $\bar{y}$ and y^*. For example, $C_{\bar{y}=0,y^*=1} = 5$ means 5 samples are given label 0 but should be labeled as 1. Establishing the confident joint is based on Equation 4.2

$$C_{\bar{y},y^*} := |\hat{X}_{\bar{y}=i,y^*=j}| \text{ where}$$

$$\hat{X}_{\bar{y}=i,y^*=j} := \left\{x \in X_{\bar{y}=i} : \hat{p}(\tilde{y}=j;x,\mathcal{M}) \geq t_j, j = \underset{l \in \mathcal{L}:\hat{p}(\tilde{y}=l,x,\mathcal{M}) \geq t_l}{\operatorname{argmax}} \hat{p}(\tilde{y}=l,x,\mathcal{M})\right\}$$

, where $X_{\bar{y}=j}$ is the samples in the dataset X with given label j. A vital threshold in Equation 4.2 is t_j defined in Equation 4.1.

$$t_j = \frac{1}{|X_{\bar{y}=j|}} \sum_{x \in X_{\bar{y}=j}} \hat{p}(\tilde{y} = j; x, \mathcal{M}) \tag{4.1}$$

t_j is called "self-confidence", which is closely related to the model $\mathcal{M}$. Instead of the maximal predicted probability, "self-confidence" in Equation 4.2 improves the label confidence estimation robustness in a imbalanced class environment. With $C_{\bar{y},y^*}$, the estimated joint distribution $\hat{Q}_{\bar{y},y^*}$ is calculated from Equation 4.2

$$\hat{Q}_{\bar{y}=i,y^*=j} = \frac{\frac{C_{\bar{y}=i,y^*=j}}{\sum_j C_{\bar{y}=i,y^*=j}} \cdot |X_{\bar{y}=i}|}{\sum_{i,j} (\frac{C_{\bar{y}=i,y^*=j}}{\sum_j C_{\bar{y}=i,y^*=j}} \cdot |X_{\bar{y}=i}|)} \tag{4.2}$$

, where $i, j \in \mathcal{L}$ are the class labels. $\hat{Q}_{\bar{y}=i,y^*=j}$ shows the tendency of label shifting within the dataset.

There are two approaches to characterize the low confidence samples according to $\hat{Q}_{\bar{y},y^*}$. The first approach considers N_i samples in class i with the lowest self-confidence as the uncertain samples, where N_i is defined in Equation 4.3.

$$N_i = N \cdot \sum_{j \in \mathcal{L}, j \neq i} (\hat{Q}_{\bar{y}=i,y^*=j}[i]) \tag{4.3}$$

In Equation 4.3, N denotes the size of the entire dataset. The second approach ranks the samples with the confidence margin defined in Equation 4.4.

$$M = \hat{p}_{x,\bar{y}=j} - \hat{p}_{x,\bar{y}=i} \tag{4.4}$$

For each off-diagonal entry of $\hat{Q}_{\bar{y},y^*}$, we select $N_{i,j}$ samples with maximal M as the uncertain samples, where $N_{i,j}$ is defined as follows.

$$N_{i,j} = N \cdot \hat{Q}_{\bar{y}=i,y^*=j} \tag{4.5}$$

Beside, it is also possible to jointly apply the above two approaches to characterizing the low confident samples.

After low confident samples are removed, learning process re-weights each class for training. The class i is comprised by w_i defined in Equation 4.6.

$$w_i = \frac{1}{\hat{p}_{\bar{y}=i|y^*=i}} = \frac{\hat{Q}_{y^*}[i]}{\hat{Q}\bar{y}, y^*[i][i]} \tag{4.6}$$

Equation 4.6 indicates that the samples from class with lower label confidence are assigned to a higher weight. Reweighting remedies the dataset size shrinking due to removing the noisy samples.

Our work is based on the label confidence estimation theory. In addition, the proposed method establishes a dynamically way to estimate the label confidence during training. For medical dataset, directly removing samples may cause information loss. Therefore, we also re-design the loss function to incorporated the low confidence samples.

CHAPTER 5

LABEL CONFIDENCE ASSISTED MODEL TRAINING FOR OA ASSESSMENT

The proposed approach uses the label confidence information to enhance CNN's performance in OA assessment tasks, which includes two interactive stages. At the training stage, our approach characterizes the low and high confidence samples from each mini-batch as shown in Figure. 5.1. The hybrid loss function calculates the errors accordingly based on the samples' confidence information. In the validation stage, we estimate the label confidence which provides the references for the training stage to separate low and high samples. We introduce the label confidence estimation in Chapter 5.1, which lays the foundation for our approach. The details of the training stage and the hybrid loss function are discussed in Chapter 5.2 and Chapter 5.3, respectively. Symbols commonly used in this paper are defined in Table 5.1.

5.1 Estimating Label Confidence

Modeling the label uncertainty has been studied for decades (Angluin and Laird, 1988; Forman, 2005; Natarajan et al., 2013; Van Rooyen et al., 2015). In a recent research, Northcutt et al. (2021) use the probability theory to model the relationship between multiple labels in a dataset, called label confidence. Mainly, labels assigned to individual samples are not considered deterministic but generated by a distribution. For example, given a "doubtful" OA sample x belonging to the KL-1 class, multiple annotators can also assign KL-0 or KL-2 to it. In this case, the given label Y is defined as a random variable, which follows a conditional distribution $p_{Y|x}$. To estimate $p_{Y|x}$, we can count the assessments from different

Part of this chapter is reprinted from © 2021 IEEE. Reprinted, with permission, from Y. Wang et al., "Learning from Highly Confident Samples for Automatic Knee Osteoarthritis Severity Assessment: Data from the Osteoarthritis Initiative," in IEEE Journal of Biomedical and Health Informatics, doi: 10.1109/JBHI.2021.3102090.

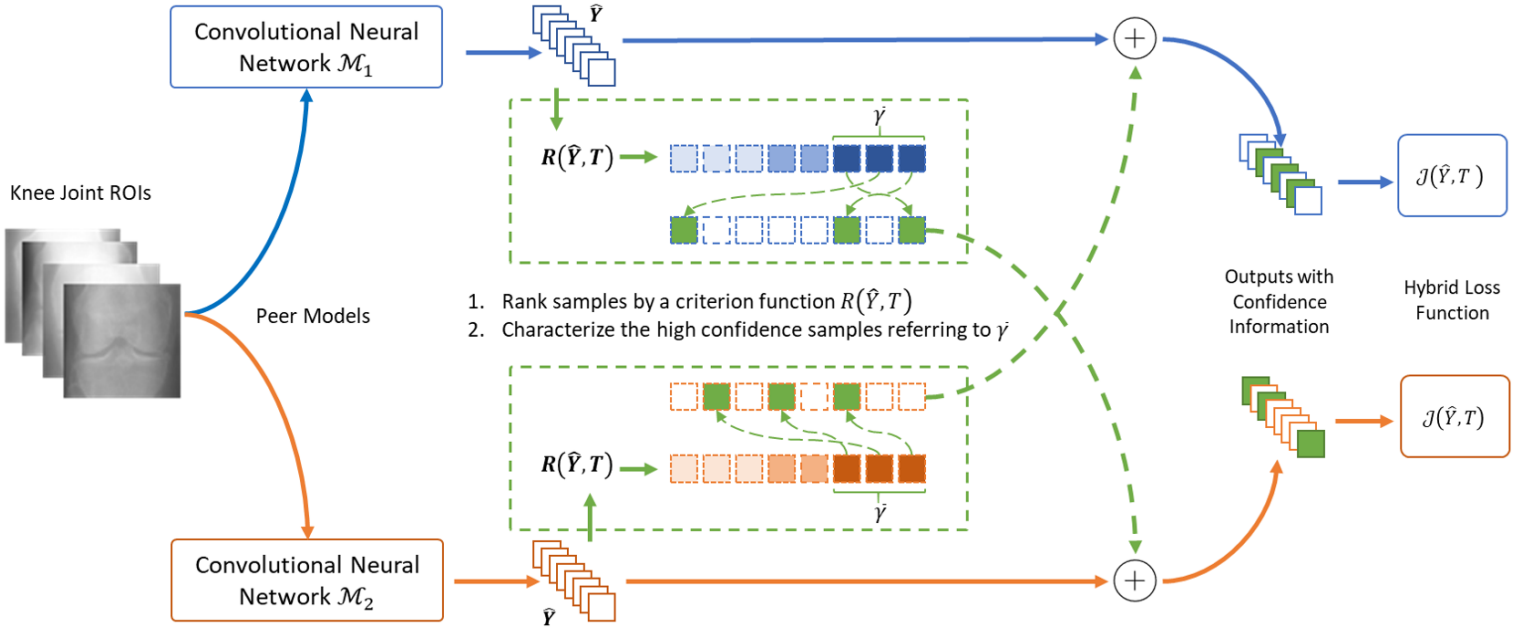


Figure 5.1: The overview of the training stage in the proposed scheme. We maintain two CNN models in the scheme, which take knee joint samples as inputs. Per mini-batch, the proposed method characterizes the high confidence samples via two steps. 1) Ranking the samples based on the confidence level estimated by $R(\hat{Y}, T)$, where $\hat{Y}$ is the CNN's outputs and T is the ground truth labels. 2) Separating the batch in the ratio of high confidence samples, denoted as $\bar{\gamma}$. Then, the peer models mutually exchange samples' confidence, which is the additional information for the hybrid loss function. In the figure, the blue and orange colors represent the training processes of two peer models, respectively. The green dashed boxes illustrate the process of characterizing high confidence samples. The solid lines represent the data flow, and the dashed lines are for confidence information flow.

Table 5.1: Symbols Commonly Used in This Paper

Symbol	Remarks
$\mathcal{K}$	the set of all KL grades
T	labels obtained from the OAI
Y	labels assessed by individual annotator
$\hat{Y}$	labels predicted by CNN
$\mathcal{D}$	the dataset obtained from the OAI
$\mathcal{T}$	the training set, $\mathcal{T} \subset \mathcal{D}$
$\mathcal{V}$	the validation set, $\mathcal{V} \subset \mathcal{D}$

annotators. There are two properties of this distribution. 1) $p_{Y|x}$ is non-categorical. As a comparison, we usually construct a categorical distribution $p_{T|x}$ from the true label T and use it as the target in classification tasks. 2) $p_{Y|x}$ is not uniform but skews to T. This property can also be illustrated in the above example. Annotators are not likely to assign KL-4 to a KL-1 sample because radio-graphic evidence of late-stage OA is absent. The second property indicates that samples of the same class share a similar distribution. Thus, $\gamma_{m,n} = p_{Y|T}(Y = m|T = n), \forall m, n \in \mathcal{K}$ describes the label uncertainty from the view of the entire dataset, which represents the probability that samples of class n are labeled as class m.

In our scheme, we incorporate the label confidence as a dynamic part in our scheme by performing the estimation on the validation set. Further, we focus on the $\gamma_k = p_{Y|T}(Y = k|T = k), \forall k \in \mathcal{K}$ which is the probability of samples of class k being correctly labeled. Given a CNN, the estimation process follows Northcutt et al. (2021). Firstly, we preserve the predicted label distribution $p_{\hat{Y}|x}$ of all the samples in $\mathcal{V}$. Secondly, we calculate the self-confidence of each class as defined in Equation 5.1

$$\epsilon_k = \frac{1}{|\mathcal{V}_k|} \sum_{x \in \mathcal{V}_k} p_{\hat{Y}|x}(\hat{Y} = k|x), \ \forall k \in \mathcal{K} \tag{5.1}$$

, where $\mathcal{V}_k$ is the set of samples with label k in $\mathcal{V}$. Thirdly, ϵ_k is used as the threshold to separate the samples in $\mathcal{V}$. The set of highly confident samples is defined as Equation 5.2.

$$C_k = \left\{x \in \mathcal{V}_k : p_{\hat{Y}|x}(\hat{Y} = k|x) > \epsilon_k\right\}, \forall k \in \mathcal{K} \tag{5.2}$$

At the same time, the low confidence set is defined as Equation 5.3.

$$\bar{C}_k = \left\{x \in \mathcal{V} \setminus \mathcal{V}_k : p_{\hat{Y}|x}(\hat{Y} = k|x) > \epsilon_k\right\}, \forall k \in \mathcal{K} \tag{5.3}$$

Finally, the estimated label confidence is defined in Equation 5.4.

$$\hat{\gamma}_k = \frac{|C_k|}{|C_k| + |\bar{C}_k|} \tag{5.4}$$

The effectiveness of Equation 5.1-Equation 5.4 requires the $p_{\hat{Y}|x}$ to be predicted by a model with a strong learning capability, which is stated as "error-free" condition (Northcutt et al., 2021). The "error-free" model can fit the $p_{Y|x}$ remarkably such that it behaves like a human annotator. Mistakes made by the "error-free" model are due to the divergence between $p_{Y|x}$ and $p_{T|x}$. Theoretical analysis by Northcutt et al. (2021) shows that $\hat{\gamma}_k$ obtained by the "error-free" model is a consistent estimator of γ_k. In practice, directly pursuing an "error-free" model is intractable, because information of Y is missing when we get the dataset from the OAI. However, CNN can approximate the "error-free" condition after a warm-up training, as it can learn the dominant pattern from initial epochs (Ma et al., 2018; Arpit et al., 2017). Through an average of all $\hat{\gamma}_k$, $\bar{\gamma}$ represents the ratio of high confidence samples whose labels are correctly assigned.

Unlike [34], which estimates the label confidence on the entire dataset, the proposed method only depends on the validation set. Statistically, $\mathcal{V}$ and $\mathcal{T}$ share the same label confidence, because they are independently sampled from one dataset $\mathcal{D}$. A benefit flowing from our adaption is that it does not affect the model learning by preventing data leakage. Such that we can embed the label confidence estimation into the standard training cycles. Further, we introduce an interactive training scheme and propose a hybrid loss function to enhance the label confidence estimation in the following sections.

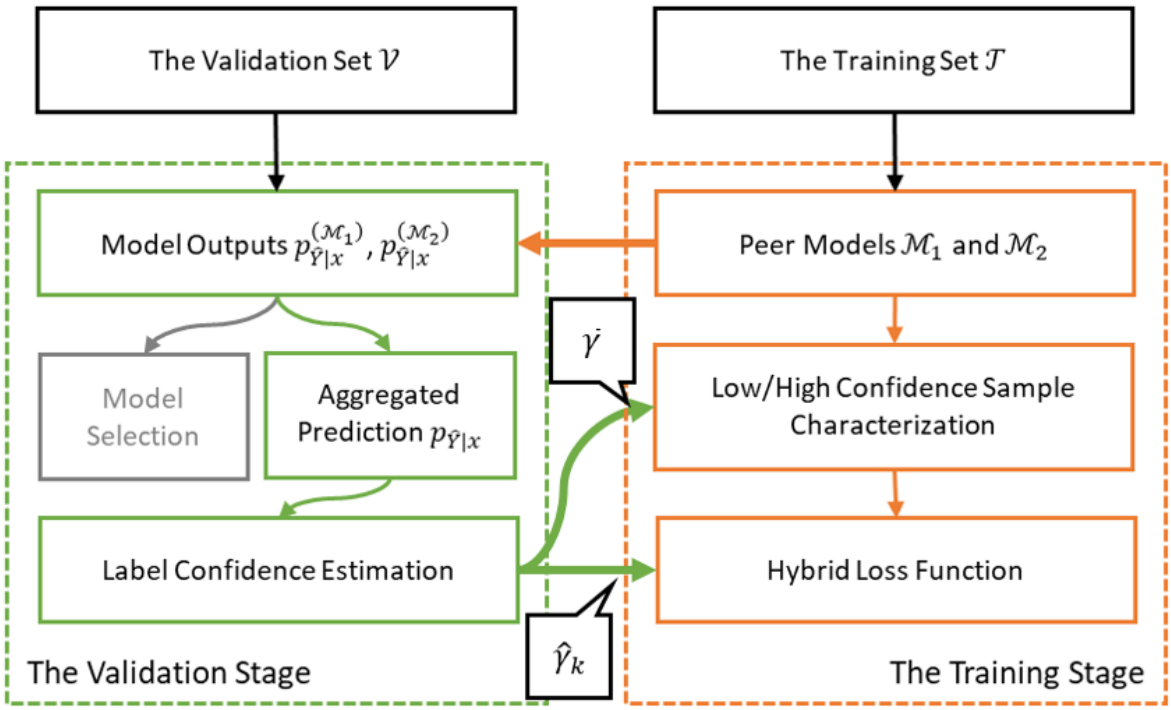


Figure 5.2: The interaction between the training and validation stages. Besides the model selection, the validation stage in our scheme estimates label confidence. The validation stage employs the peer models to obtain the aggregated predictions $p_{\hat{Y}|x}$. In turn, the training stage depends on the estimated $\hat{\gamma}_k$ and $\bar{\gamma}$ to characterize low and high confidence and calculating the errors. In the figure, the green boxes indicate the process of estimating label confidence.

5.2 Interactive Training with Label Confidence Information

The proposed scheme contains a training stage and a validation stage. During training, we maintain two peer models, which behave differently at each stage.

In the training stage, as shown in Figure. 5.1, peer models characterize the highly confident samples independently. For example, $\mathcal{M}_1$ characterizes the high confidence samples from each mini-batch as defined in Equation 5.5

$$H^{(\mathcal{M}_1)} = \left\{ x \in \mathcal{D}^{(batch)} : \mathrm{Ord}(R(x)) \leq \lfloor |\mathcal{D}^{(batch)}| \times \bar{\gamma} \rfloor \right\} \tag{5.5}$$

, where $\mathcal{D}^{(batch)}$ denotes a mini batch, R is a criterion function applied to each sample, and $\mathrm{Ord}(R(x))$ indicates the ordinal number of x's criterion in the mini batch. $R(x)$ reflects

the confidence level of the sample x. In the proposed method, $R(x)$ is implemented as the cross-entropy function as we define the OA severity assessment as a classification problem. For each sample, $R(x)$ is calculated from the CNN's outputs $\hat{Y}$ and ground truth label T. Then the whole batch is ranked in ascending order. According to $\bar{\gamma}$, samples with smaller $R(x)$ are characterized as high confidence samples. At the same time, the remaining samples compose the low confidence set as Equation 5.6.

$$L^{(\mathcal{M}_1)} = \mathcal{D}^{(batch)} \setminus H^{(\mathcal{M}_1)} \tag{5.6}$$

$\mathcal{M}_2$, $H^{(\mathcal{M}_2)}$ and $L^{(\mathcal{M}_2)}$ are defined in the same way. The training stage is similar to "co-teaching" (Han et al., 2018), which is featured by exchanging the loss values of samples between the peer models to learn from a noisy dataset. However, our implementation does not depend on the pre-determined threshold to filter the low confidence samples as we plug in the estimated $\bar{\gamma}$. Further, the criterion function in our scheme is different from the loss function.

At the validation stage, peer models are ensembled through the "bagging" method (Breiman, 1996). "Bagging" is an ensemble method which aggregates the results from multiple models to reduce the prediction variance. Usually, people need to sample independent sets from the original dataset and train multiple models before the model ensemble. As discussed earlier, two peer models separate the high and low confidence sets independently. After exchanging the characterization results, two models are trained on different subsets. Thus, the training stage assumes the role of independent sampling. When estimating label confidence, $p_{\hat{Y}|T}$ used in Chapter 5.1 is obtained by Equation 5.7. Bagging the results reduces the variance of predictions, which stabilizes label confidence estimation.

$$p_{\hat{Y}|x} = \frac{1}{2}\left(p_{\hat{Y}|x}^{(\mathcal{M}_1)} + p_{\hat{Y}|x}^{(\mathcal{M}_2)}\right) \tag{5.7}$$

Through the peer models and label confidence, two stages interact with each other. After the previous training epoch, the updated models estimate the label confidence at the

validation stage. At the end of the validation stage, $\hat{\gamma}_k$ and $\bar{\gamma}$ are fed back to the next training epoch. To this end, the proposed method is fully automatic and data-driven.

5.3 Hybrid Loss Function

As discussed in previous sections, our approach models the label uncertainty by label confidence and separates the high confidence samples during training. Further, we proposed a hybrid loss function targeting the empirical errors during the separation, which provides a second dimension for learning from high confidence samples. Empirical errors refer to the mistakes made by a machine learning model when generalizing on the unseen data. In the proposed approach, $\hat{\gamma}_k$ is estimated on the validation set. Such empirical errors are inevitable when applying the $\hat{\gamma}_k$ to the training set. As $\hat{\gamma}_k$ is approaching γ_k, the empirical errors become a minor factor for the characterization of low and high samples, such that we can directly prune the low confidence set. However, the proposed hybrid loss function provides a flexible way to handle the low confidence sets.

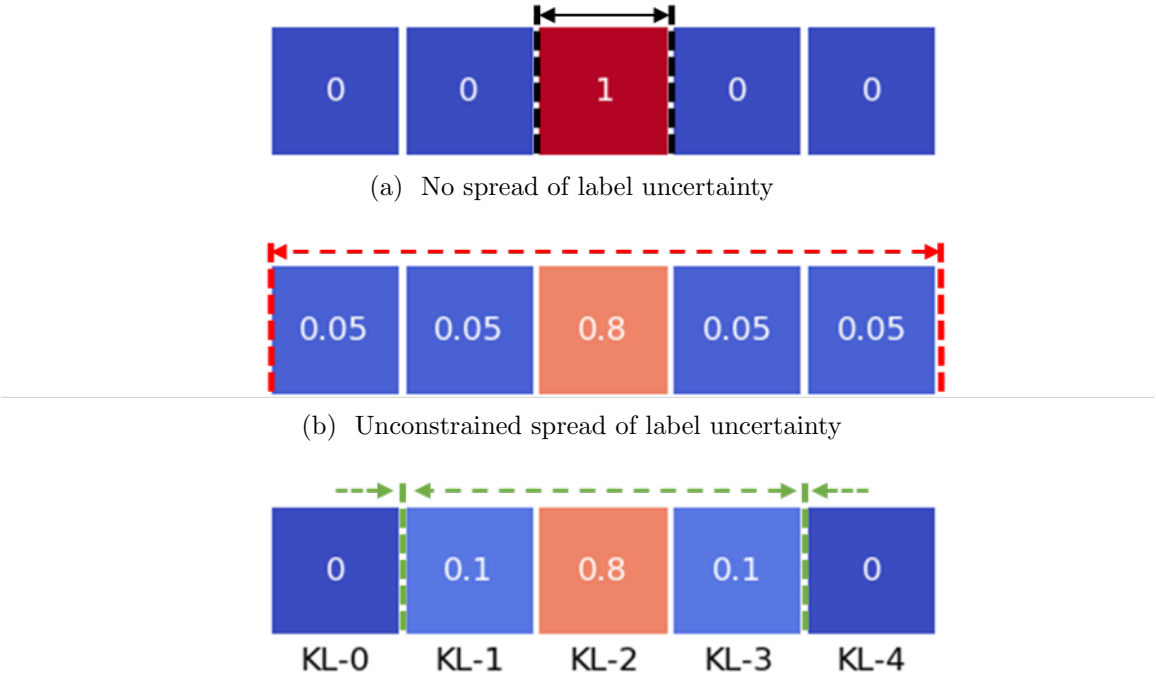
Taking $\mathcal{M}_1$ for example, the proposed loss function consists of two terms. The first term is the weighted cross-entropy loss function as Equation 5.8, which is applied to $H^{(\mathcal{M}_2)}$.

$$J_{wCE}(p_{T|x}, p^{(\mathcal{M}_1)}_{\hat{Y}|x}) = \sum_{k}^{\mathcal{K}} \frac{1}{\hat{\gamma}_k} p_{T|x}(T=k|x) \log(\hat{p}^{(\mathcal{M}_1)}_{\hat{Y}|x}(\hat{Y}=k|x)), x \in H^{(\mathcal{M}_2)} \tag{5.8}$$

For $L^{(\mathcal{M}_2)}$, the categorical target distribution $p_{T|x}$ is converted to a "smoothed" $\tilde{p}_{T|x}$ as Equation 5.9

$$\tilde{p}_{T|x} = \begin{cases} \hat{\gamma}_T & k = T \\ 0.5(1-\hat{\gamma}_T) & k = \text{adjacent classes of } T \\ 0 & \text{otherwise} \end{cases} \tag{5.9}$$

, where $\hat{\gamma}_T$ is the estimated label confidence of T. Design of $\tilde{p}_{T|x}$ is based on the prior knowledge that the distribution of Y skews on T. When T has only one adjacent class, we



(a) No spread of label uncertainty

(b) Unconstrained spread of label uncertainty

(c) Constrained spread of label uncertainty

Figure 5.3: Comparisons of categorical distribution (Figure. 5.3a), "smooth loss" distribution (Figure. 5.3b), and the proposed target distribution (Figure. 5.3c). Given T =KL-2, we smooth the categorical distribution but limit the spreading within the adjacent level of the ground-truth.

set the probability of T as $0.5(1+\hat{\gamma}_T)$. As shown in , $\tilde{p}_{T|x}$ is similar to "smooth loss"(Berrada et al., 2018) but it restricts the distribution within the adjacent classes of k. To measure the difference between $p_{\hat{Y}|x}^{(\mathcal{M}_1)}$ and $\tilde{p}_{T|x}$, we use the Kullback-Leibler divergence as the loss function for $L^{(\mathcal{M}_2)}$ as in Equation 5.10.

$$J_{KL}(\tilde{p}_{T|x}, p_{\hat{Y}|x}^{(\mathcal{M}_1)}) = \sum_{k}^{\mathcal{K}} \tilde{p}_{T|x}(T=k|x) \log\big(\frac{\tilde{p}_{T|x}(T=k|x)}{p_{\hat{Y}|x}^{(\mathcal{M}_1)}(\hat{Y}=k|x)}\big),\ x \in L^{(\mathcal{M}_2)} \tag{5.10}$$

If provided with more knowledge about the tendency of labeling, $\tilde{p}_{T|x}$ can be designed to be asymmetric. However, this is not the principal topic of this paper. Combining the above two items as well as a hyper-parameter λ to control the impact of low confidence samples,

the proposed loss function is as in Equation 5.11

$$J_{hybrid}^{(\mathcal{M}_1)} = \frac{1}{|H^{(\mathcal{M}_2)}|} \sum_{x \in H^{(\mathcal{M}_2)}} J_{wCE} \; + \frac{\lambda}{|L^{(\mathcal{M}_2)}|} \sum_{x \in L^{(\mathcal{M}_2)}} J_{KL} \tag{5.11}$$

, where the targets and model outputs are eliminated for clearance. The hybrid loss function helps with learning the high confidence samples by controlling the impact of empirical errors. Hyperparameter λ copes with the samples of different confidence levels. Loss function for $\mathcal{M}_2$ shares the same form as Equation 5.11, but uses the sample confidence information provided by $\mathcal{M}_1$.

CHAPTER 6

DATA PREPROCESSING AND EXPERIMENT SETUP

6.1 The OAI Public Dataset

The dataset used in this work is obtained from the OAI database. The OAI is a multi-center, longitudinal, prospective observational study of knee OA. It has established and maintained a comprehensive database including clinical evaluation data, radiological image, and a biospecimen repository. There are 4796 participants aged between 45 and 79 in the study of OAI. We used the X-ray screen data collected from the first visit of participants in our research. Specifically, we retrieved 4472 samples from 0.C.2 and 0.E.1 versions of the dataset.

6.2 Knee Joint Area Segmentation

The dataset obtained from the OAI contains the X-ray screening data and KL assessments. To prepare for classification tasks, we convert the screening data from DICOM[1] format to plain images and then segment the knee ROI. Plain images are extracted using Pydicom(Mason, 2011), during which we scale the 12-bit pixels to 8-bit. We use the YOLOv2(Redmon and Farhadi, 2017) to segment knee ROI. Firstly, we randomly select 200 images from the OAI dataset as the training set. Radiologists from Huashan Hospital, Fundan University annotate bounding boxes of knees in these images. Then we follow the

Part of this chapter is reprinted from © 2021 IEEE. Reprinted, with permission, from Y. Wang et al., "Learning from Highly Confident Samples for Automatic Knee Osteoarthritis Severity Assessment: Data from the Osteoarthritis Initiative," in IEEE Journal of Biomedical and Health Informatics, doi: 10.1109/JBHI.2021.3102090.

Part of this section is reprinted from ©2021 Y Wang et al. "An Automatic Knee Osteoarthritis Diagnosis Method Based on Deep Learning: Data from the Osteoarthritis Initiative", Journal of Healthcare Engineering, vol. 2021. https://doi.org/10.1155/2021/5586529.

[1]Digital Imaging and Communications in Medicine

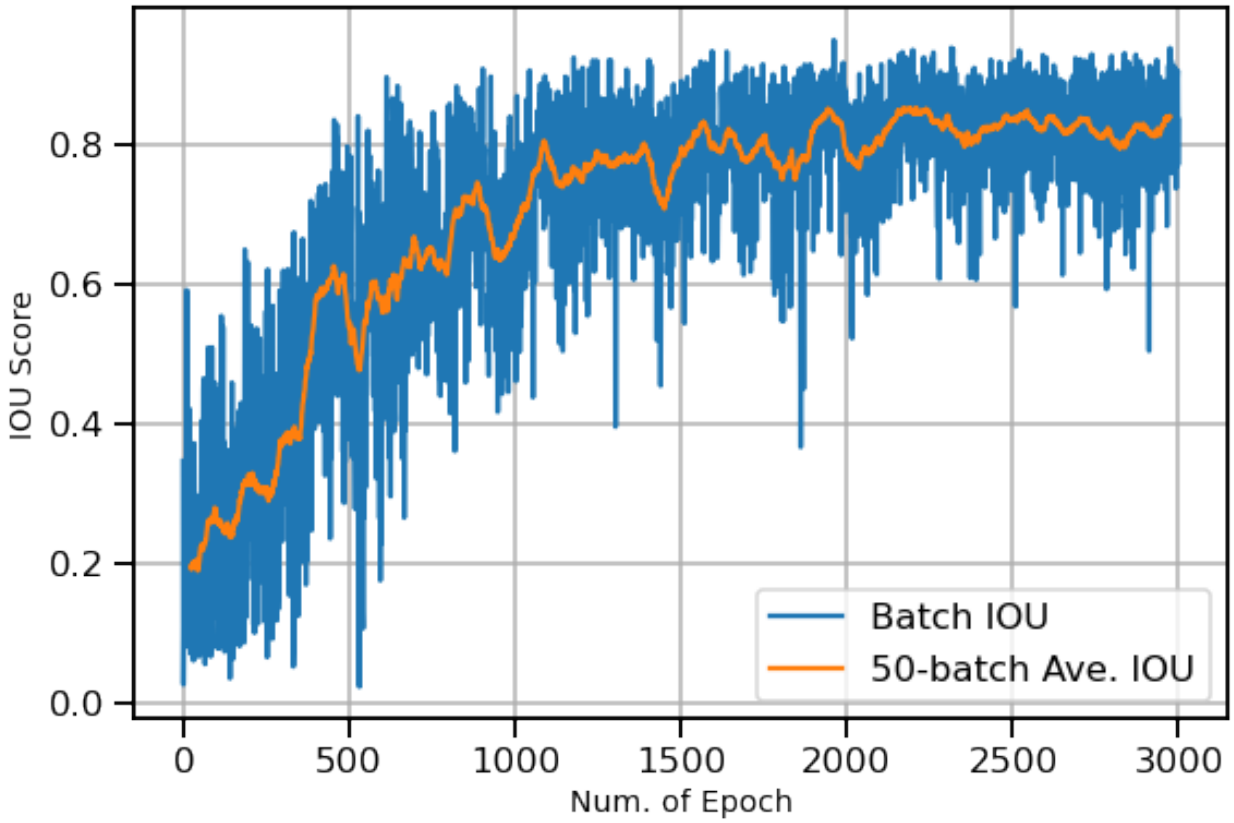


Figure 6.1: The IOU score for each batch (iteration) during training. The moving average value of 50 batches is marked as "50-Batch Ave. IOU" to show the general trend.

same settings as Redmon and Farhadi (2017) to finetune the YOLOv2 except that we set the number of class as 1 for our task. Meanwhile, we monitor the intersection over the union (IOU) score defined in Eq. 6.1

$$IOU = \frac{A \cap B}{A \cup B} \tag{6.1}$$

, where A denotes the predicted bounding box and B denotes the ground truth. Once the IOU scores converge, the training is terminated to avoid over-fitting. Figure. 6.1 shows the IOU score of each training batch. The moving average IOU scores over 50 batches were used as the indicator of convergences. After training, the IOU score on the annotated 200 samples reached 0.82.

After the finetuning, the YOLOv2 is further used to segment the remaining 4272 images. However, the IOU score is unsuitable for verifying knee detection because the remaining samples do not have annotations. To validate the segmentation results, we proposed four statistical forms of measurements.

- Detection count per image. Thanks to the consistency of the OAI dataset, all collected screen data consists of two knees. Therefore, two detections per image are expected.
- Detection size. The sizes of the knee are similar for humans. However, knee detection varies due to the scale of the X-ray image. From the statistical view, the detections are expected to tend to cluster.
- Detection location. A proper pair of knee joint detections should be located on the same height vertically and on both sides of the image horizontally.
- Object Confidence. As all images contain the knee joints, a reliable model should give a high confidence score on its detection.

From the four aspects mentioned above, we evaluate the detection results from the testing images. Firstly, Figure. 6.2 shows the distribution of detection per image. 98.22% of images in the dataset have two detections corresponding to the left and right legs. The YOLO model successfully detects two knees in most images.

Secondly, Figure. 6.3a and Figure. 6.3b show the distributions of height and width for both original X-ray images and cropped ROIs. As shown in Figure. 6.3a, the sizes of the original X-ray images are separated into two clusters. The centroids of the two clusters are located near $(600, 500)$ and $(1100, 850)$. Correspondingly, Figure. 6.3b shows two clusters regarding the sizes of the cropped ROIs, whose centroids are near $(150, 125)$ and $(250, 200)$. The clustering of cropped ROIs can be explained by the scale of original X-ray images, as the knees area is in proportion to the X-ray image size. Therefore, the sizes of detected ROIs

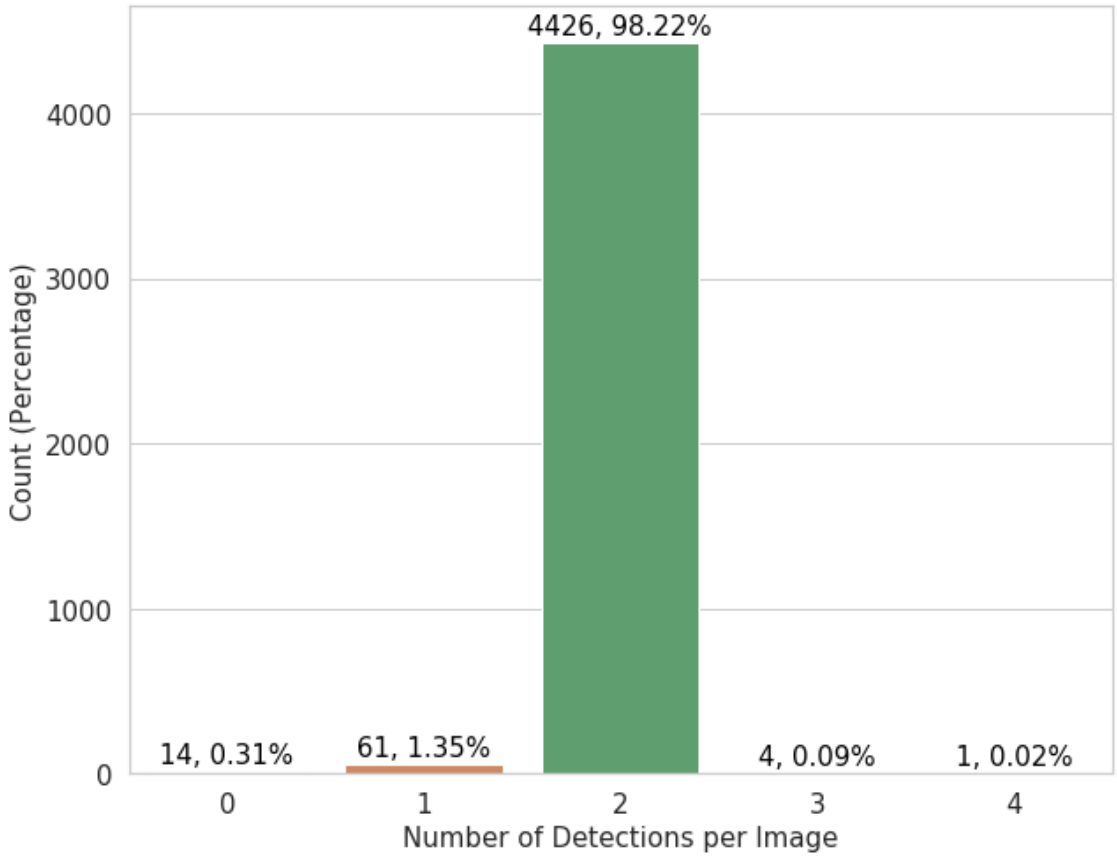
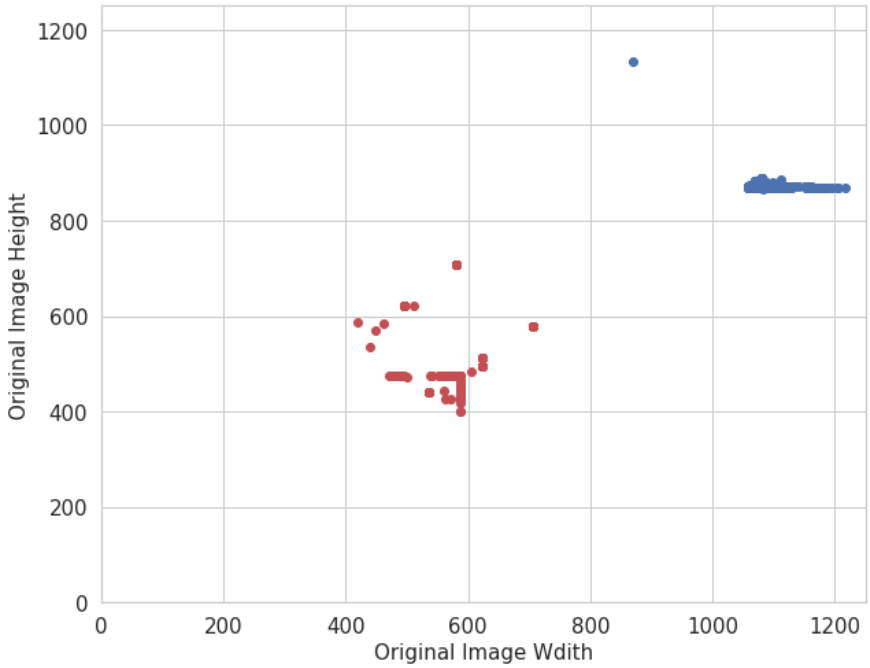


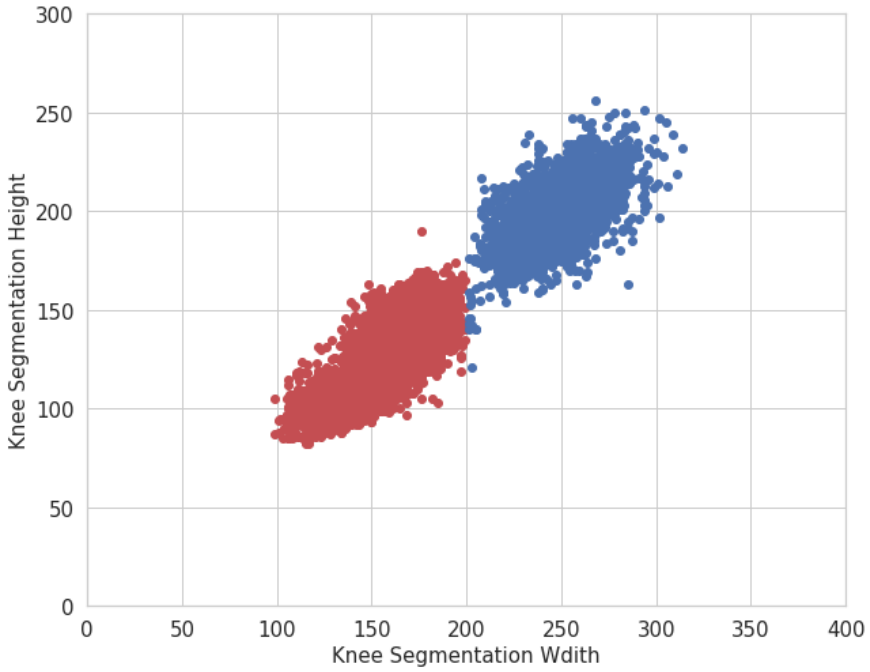
Figure 6.2: Detection count per image. In total, 4426 images have 2 detections, which account for 98.22% of the whole dataset.

are comparable. Moreover, the results also demonstrate that the trained YOLO model is robust to different X-ray image sizes.

Thirdly, Figure. 6.4 shows the locations of all detected knees, which are represented by the top-left corner's coordinates. We observe two groups of detections marked as "X" and "Y". The centroids of group "X" lie near $(150, 200)$ and $(400, 200)$. Regarding the ROIs in group "Y", their y-coordinates range between 280 and 400, and their x-coordinates are near 200 and 600. Vertically, all detections appear in the middle region of the X-ray image. Horizontally, the clusters distribute on the left and right sides.



(a) X-ray images size distribution



(b) Knee segmentation size distribution

Figure 6.3: Size of the original images and detections. (a) shows the size of original image. The OAI dataset has two clusters of shapes, as indicated in the red and blue colors. (b) shows the knee segmentation widths and heights. Two clusters regarding the size of the bounding box are marked by red and blue, respectively.

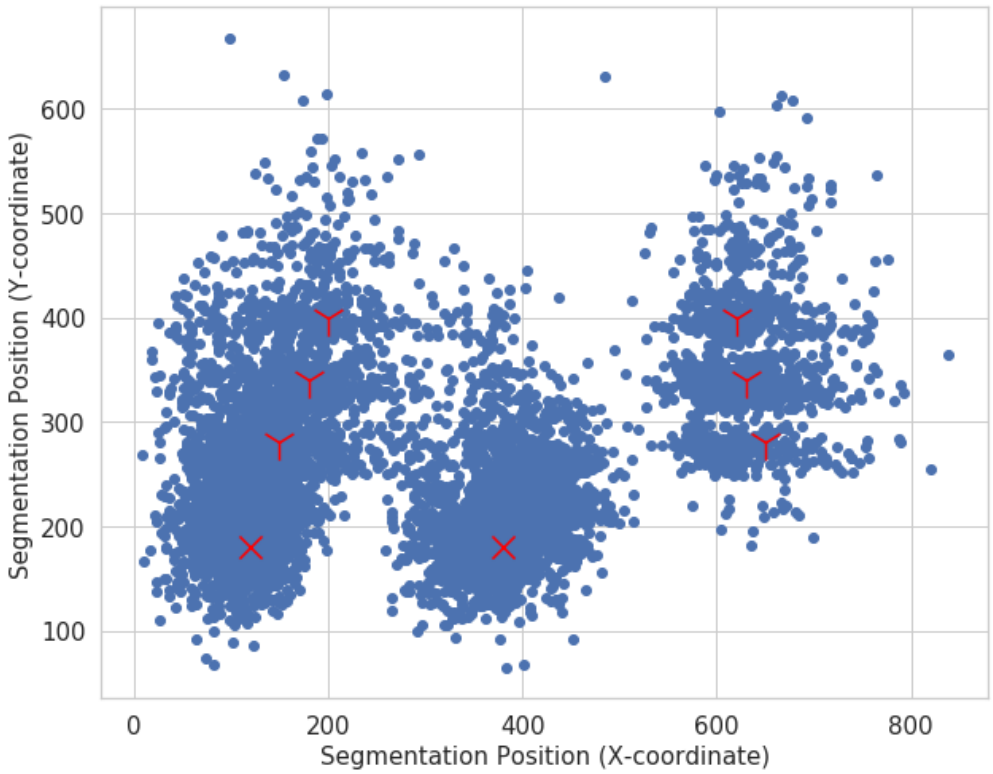


Figure 6.4: Locations of knee detections. Vertically, all detections appear in the middle region of the image except for outliers. Horizontally, two pairs of clusters are labeled as "X" and "Y" according to the positions. The clusters are caused by different X-ray image sizes and the knee alignment. Cluster "X" has only one pair of centroids. While cluster "Y" is split into 3 subgroups. Each subgroup on the right has its counterpart on the left.

Finally, Figure. 6.5 shows the confidence score distribution and the Kernel density estimation line based on the scores given by the YOLO. The center of the score distribution is roughly 85%. It indicates that the trained model has high confidence in its prediction.

Based on the above four measurements, we can conclude that segmentations on the whole dataset are accurate and valid. Furthermore, to mitigate the influence of invalid detections, we use the confidence score of 0.75 as a threshold to filter the ROIs referring to the predicted confidence score. In this way, we preserve 95% of ROIs, and their label distribution is shown in Table 6.1. Examples of detected knee join as well as YOLO's object scores are shown in

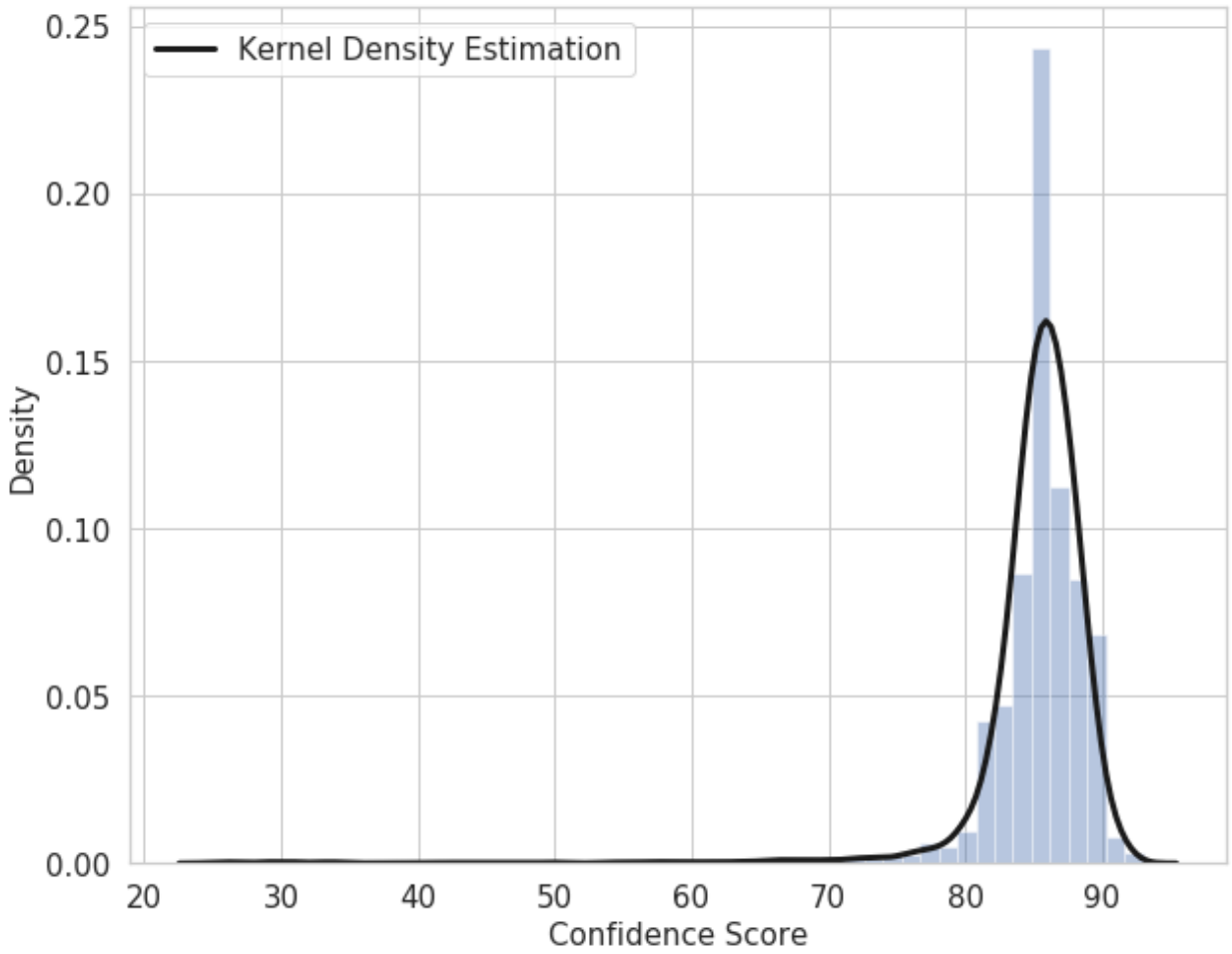
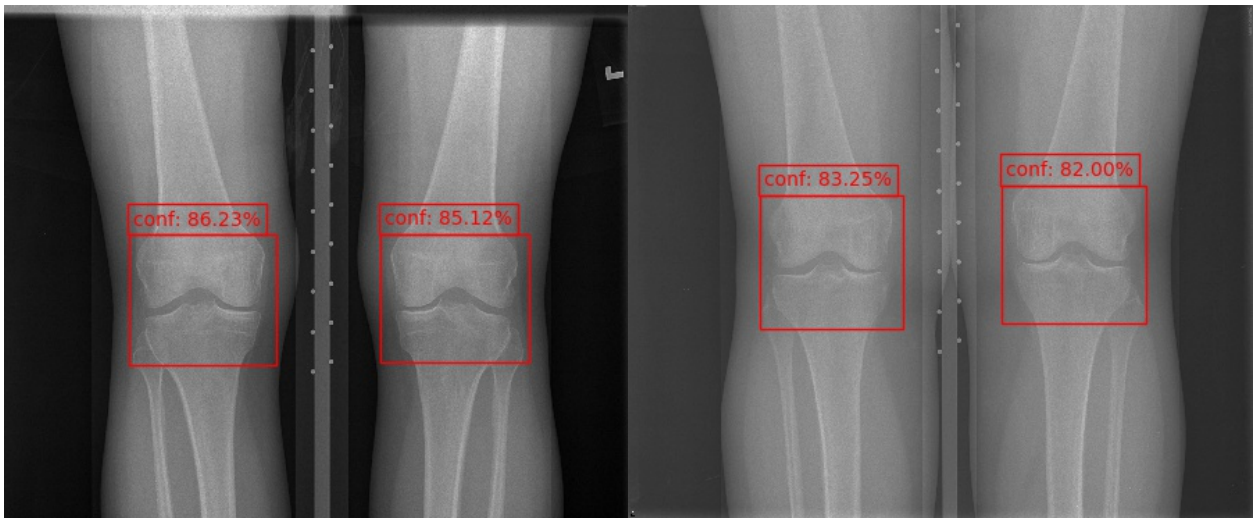
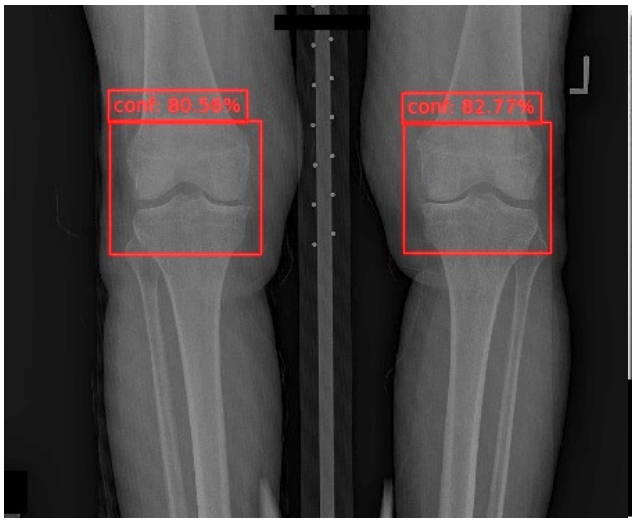


Figure 6.5: Object confidence score distribution (by YOLO model). The distribution of confidence scores is centered at 85. The density of low confidence detection (below 75) is nearly zero. The confidence score distribution indicates that the majority of detections are reliable.

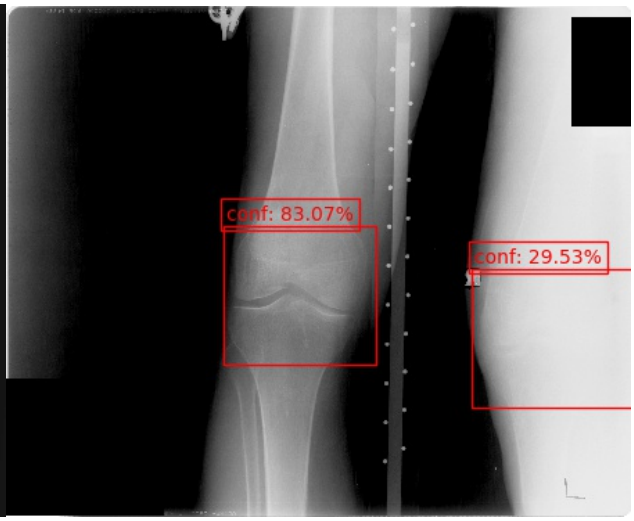


(a)

(b)



(c)



(d)

Figure 6.6: Knee joint areas located by the fine tuned YOLOv2 model. Figure. 6.6a, Figure. 6.6b, and Figure. 6.6c shows the proper bounding boxes, which give the knee joint areas a high object confidence score. Figure. 6.6d is a failed example due to the misalignment and low contrast. YOLOv2 scores 29.53% for this detection. To obtain segmentation, we need to crop the images referring to these bounding boxes and then resize the cropped ROIs. The segmentation details are described in Chapter 6.2.

Figure. 6.6. For the following classification process, we resize all cropped ROI into 224x224.

KL assessments are assigned to each ROI according to the patient ID and the side of the leg.

Table 6.1: Label Distribution of the Cropped ROI

	KL-0	KL-1	KL-2	KL-3	KL-4
Total ROI	3234	1475	2186	1141	266
Training Set	2264	1033	1531	799	187
Validation Set	323	147	218	114	26
Test Set	647	295	437	228	53

6.3 Training Scheme Implementation

We apply the proposed method to two CNN architectures in the experiments, viz., resnet34 (He et al., 2016a) and densenet121 (Huang et al., 2017). Deep learning models are implemented with Pytorch(Paszke et al., 2019). We integrate the CleanLab (Northcutt et al., 2021) into our training framework to estimate the label confidence at the training stage. As the baseline, "network-based"(Tan et al., 2018) transfer learning (denoted as "trans.") is compared for both tasks. Motivated by Chen et al. (2019) and Tiulpin et al. (2018), initial weights of CNN are obtained from the pre-training on the ImageNet(Deng et al., 2009) dataset to alleviate the difficulty of insufficient training data. We replaced CNN's last layer to adapt to the 5-class OA assessment. And we did not freeze any layers during training. We use an augmentation method similar to Chen et al. (2019) by randomly adjusting the image's brightness and contrast. The CNNs are finetuned for 12 epochs to ensure the optimization is converged. We use Adam optimizer with learning rate of 0.0001 and with weight decay of 1e-8. Besides the baseline, we compare our method with the published research, which shall be discussed in the next section. All experiments are run on Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz CPU. We use Nvidia Tesla V100 GPU to speed up training.

6.4 Model Performance Evaluation

We evaluate the performance on two tasks related to OA assessment, five-stage assessment task and early-stage assessment task. The five-stage assessment performance is evaluated

on all KL grades. We use accuracy score and Matthews Correlation Coefficients (MCC) as metrics. The MCC is widely used in the field of bioinformatics as a metric of imbalanced dataset. While MCC was firstly proposed for binary classification tasks, Jurman et al. (2012) extended it to multi-class scenarios. Let P denote the prediction indicator matrix where $P_{ik} = 1$ if i-th sample is predicted as k and let G denote the ground truth indicator matrix where $G_{ik} = 1$ if i-th label is k. The MCC is defined as Equation 6.2

$$MCC = \frac{\text{cov}(P, G)}{\sqrt{\text{cov}(P, P)\text{cov}(G, G)}}\text{cov}(P, G) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k\in\mathcal{K}}(P_{ik} - \bar{P}_k)(G_{ik} - \bar{G}_k) \tag{6.2}$$

, where N is the size of dataset and $\bar{P}_k, \bar{G}_k$ are the column-wise mean of P, G. The MCC ranges from -1 to 1 where 1 is for perfect classifier, and 0 is for random guess. The early-stage assessment includes KL-0 vs. KL-1, KL-1 vs. KL-2, and KL-0 vs. KL-2 classifications. We use accuracy score and F1-score as metrics for early-stage assessment task.

To simulate the varying label confidence levels in the dataset, our experiment is conducted in a 5-fold manner. The ROIs obtained are split into five folds using the stratified sampling by the KL-grade. In each step, we hold out one fold for testing. The other four folds are further split into training set and validation set in the ratio of 7:1. The validation set is used for model selection and label confidence estimation. The metric used for model selection is the accuracy score for all tasks. Notably, the KL label distributions are the same for all five folds as shown in Table 6.1, but experiments are independent of each other. We evaluate the performance separately and report the average results. Such a setup is similar to Nasser et al. (2020), which uses ten folds, we apply five folds to leave more testing data.

CHAPTER 7

RESULTS AND DISCUSSIONS

7.1 OA Severity Assessment

7.1.1 The Five-stage Task

Accuracy scores on the five-stage task are compared with recently published researches in Table 7.1. Results of different studies are grouped by the classification CNN architecture. For the reported accuracy, we carefully examine these studies from two aspects for a fair comparison, including the data source and preprocessing method. First, results reported in Table 7.1 are evaluated on the same OAI dataset as ours. Particularly, Tiulpin et al. (2018) use an additional dataset (Multicenter Osteoarthritis Study, MOST (Multicenter Osteoarthritis Study: `https://most.ucsf.edu/`) for training. Second, these researches employ a semi-automatic or fully automatic preprocessing method. Besides the baseline, we compare our work to "ordinal loss"(Chen et al., 2019) and "label smooth"(Berrada et al., 2018), which handle the label uncertainty through loss functions. Their results are obtained from our preprocessed data. We use the same weights for "ordinal loss" provided by Chen et al. (2019). For "label smooth," we set the smooth parameter as 0.1, which results in the best performance according to Berrada et al. (2018). For our method, we use $\lambda = 0.01$ in the proposed hybrid loss function. Warm-up epoch is 2 for resnet34 and 3 for densenet121. Hyper-parameters' effects are analyzed in the last section. As there are two models in our scheme, we also ensemble their results by Equation 5.7, which are marked as "bagging".

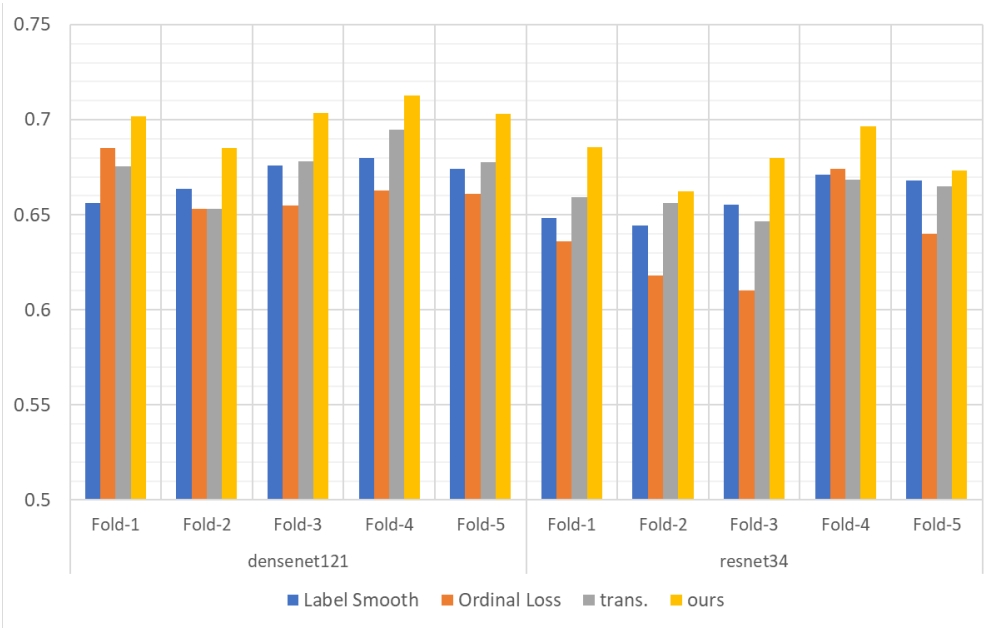
As shown in Table 7.1, our approach outperforms the previous methods and the baseline on the five-class tasks. The proposed method achieves an improvement of 4.76% (resnet34)

Part of this chapter is reprinted from © 2021 IEEE. Reprinted, with permission, from Y. Wang et al., "Learning from Highly Confident Samples for Automatic Knee Osteoarthritis Severity Assessment: Data from the Osteoarthritis Initiative," in IEEE Journal of Biomedical and Health Informatics, doi: 10.1109/JBHI.2021.3102090.

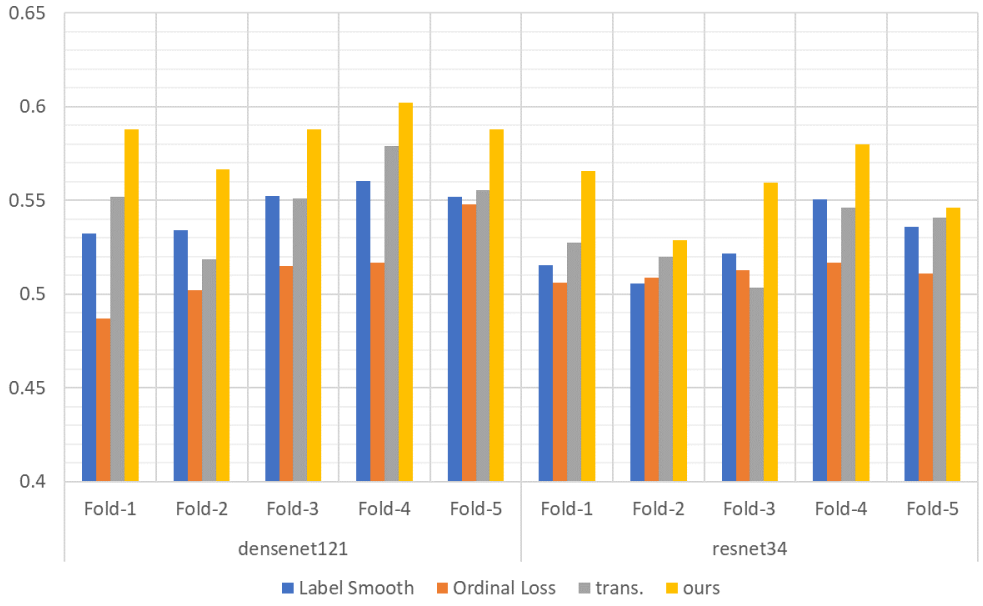
Table 7.1: Accuracy of the five-class task

Method	Backbone	Accuracy
Antony et al. (2017)	VGG-like CNN	61.90%
Górriz et al. (2019)	VGG-16	64.30%
Tiulpin et al. (2018)	resnet34	66.71%
"Ordinal loss" (Chen et al., 2019)	resnet34	63.56%
"Label smooth" (Berrada et al., 2018)	resnet34	65.74%
transfer learning	resnet34	65.91%
Ours (single)	resnet34	67.98%
Ours (bagging)	resnet34	**68.32%**
"Ordinal loss" (Chen et al., 2019)	densenet121	66.34%
"Label smooth" (Berrada et al., 2018)	densenet121	67.00%
transfer learning	densenet121	67.59%
Ours (single)	densenet121	69.25%
Ours (bagging)	densenet121	**70.13%**

and 3.79% (densenet121) in terms of accuracy score, compared to "ordinal loss" (Chen et al., 2019). For "label smoothing", the improvement is 2.58% (resnet34) and 3.13% (densenet121). We observe that ensembling the peer models yields slightly better results. However, the single model makes the main progress. The comparisons to Chen et al. (2019) and Berrada et al. (2018) suggest that the proposed method exploits the high confidence samples. The "ordinal loss" and "label smoothing" solve the label uncertainty by adjusting the loss functions. Chen et al. (2019) apply a weight matrix to the starndard cross-entropy loss. Berrada et al. (2018) adjust the target distributions. However, all samples are still considered equally confident. During training, CNN tries to memorize the low confidence samples, given its powerful representation capability (Han et al., 2018). Such memorization will not contribute to the models' generalization on unseen data. A step forward enabled by our approach is that we separate the high and low confidence samples. In addition, the hybrid loss function handles samples accordingly. By focusing on the high confidence samples, we achieve higher performance.



(a) Accuracy on each fold



(b) MCC on each fold

Figure 7.1: Metrics of 5-class task on each fold. We group the results by the CNN architecture.

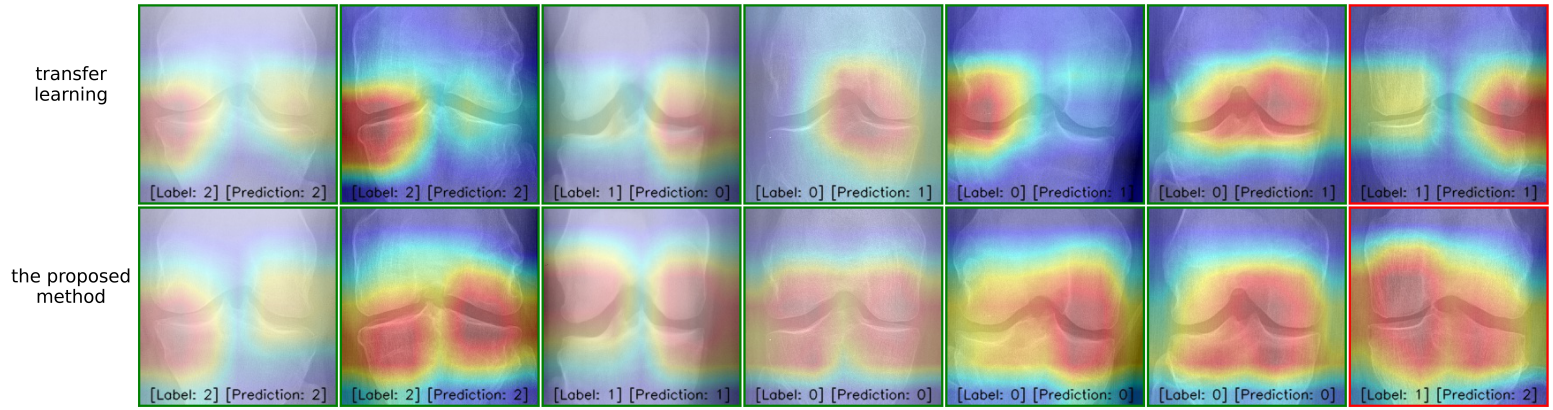


Figure 7.2: Comparison of activated region via GradCAM(Selvaraju et al., 2017). To obtain this figure, we applied GradCAM on the densenet121 model trained by transfer learning baseline and our own method. Ground truth label and prediction are marked on each picture as "Label" and "Prediction". The first row shows the activated regions of the baselines. The second row is obtained by our method. The red color shows the supporting regions for the predictions. Although in some cases we overestimated the severity compared to the baseline method (marked by the red boxed), the comprehensive accuracy is improved.

To examine the performance of each fold, we show the accuracy and MCC scores in Figure. 7.1. Results shown in Figure. 7.1 suggest that our scheme adapts to different folds automatically. Chen et al. (2019) and Berrada et al. (2018) show competitive results on a certain single fold. For example, the best accuracy scores achieved on one single fold by Chen et al. (2019) are 67.6% (resnet34) and 68.50% (densenet121). For Berrada et al. (2018), best accuracy scores achieved are 67.11% (resnet34) and 68.01% (densenet121). However, the lowest accuracy scores for both Chen et al. (2019) and Berrada et al. (2018) are bellow the transfer learning baselines. In previous methods, the training depends on the pre-determined parameters in the loss function. When evaluated on different folds, our data-driven method shows superior performance on every fold. The MCC scores follow the same trends as accuracy regarding our method. Improvement of MCC shows that our method does not favor any specific class, but gains better performance in all classes.

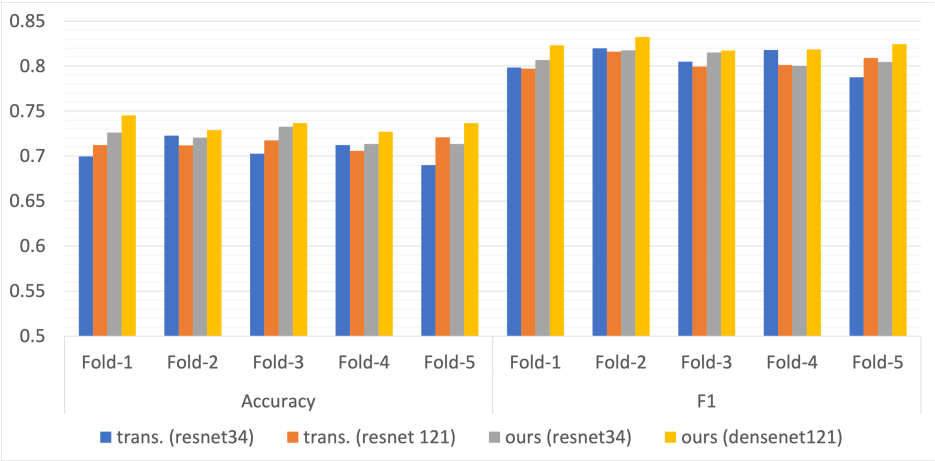
In Figure. 7.2, we use GradCAM(Selvaraju et al., 2017) to illustrate the activated regions of densenet121's classification result. The second row of Figure. 7.2 shows that our training scheme drives the model to extract features from both sides of the knee joint areas. As shown in the green boxes of Figure. 7.2, the comprehensive features obtained from both the lateral side and the medial side lead to a correct prediction. Despite the fact that our method could over-estimate the severity as shown in the red boxes, the overall accuracy is improved.

7.1.2 The early-stage tasks

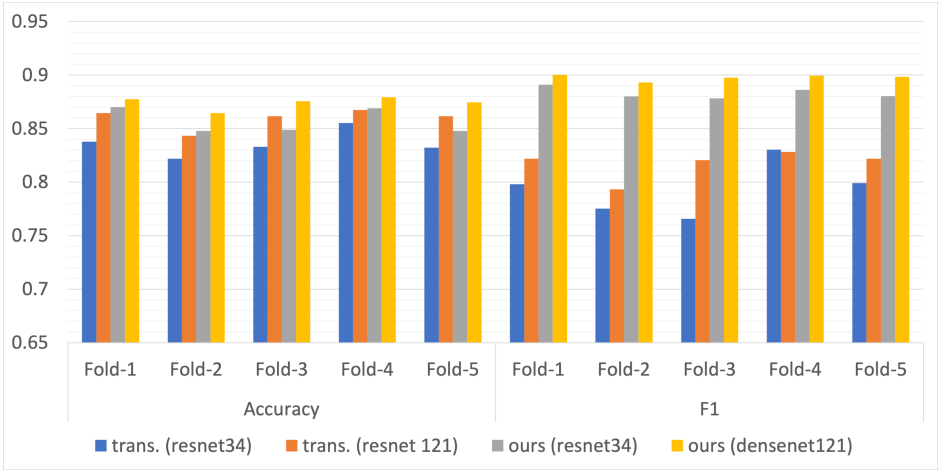
Due to the demands from the clinical environment, we examine our method on the early-stage tasks. For a fair comparison, experiments ran under two conditions. On the one hand, Antony et al. (2016) evaluated the performance using all early stage samples, which maintained an imbalance class distribution. On the other hand, Nasser et al. (2020) re-sampled the data of KL-0, KL-1 and KL2 to obtain a balancing subset. Correspondingly, we also re-sample the data and compare the results under two conditions as in Table 7.2.

Table 7.2: Comparison of the accuracy on early-stage tasks

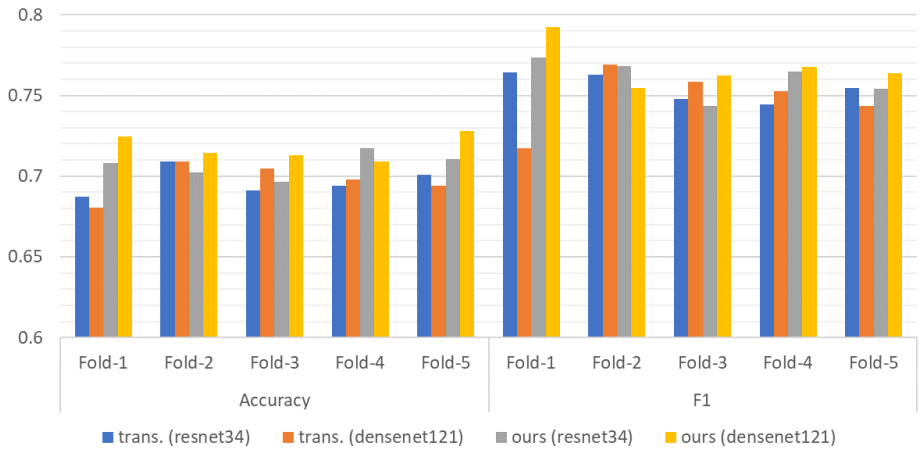
Method	Task	Accuracy
*evaluated on all OAI early-stage samples		
Antony et al. (2016)	KL-0 vs. KL-1	64.70%
transfer learning (resnet34)	KL-0 vs. KL-1	70.55%
transfer learning (densenet121)	KL-0 vs. KL-1	71.37%
ours(resnet34)	KL-0 vs. KL-1	72.12%
ours(densenet121)	KL-0 vs. KL-1	**73.50%**
Antony et al. (2016)	KL-0 vs. KL-2	77.60%
transfer learning (resnet34)	KL-0 vs. KL-2	83.93%
transfer learning (densenet121)	KL-0 vs. KL-2	85.55%
ours(resnet34)	KL-0 vs. KL-2	85.99%
ours(densenet121)	KL-0 vs. KL-2	**87.42%**
Antony et al. (2016)	KL-1 vs. KL-2	65.80%
transfer learning (resnet34)	KL-1 vs. KL-2	69.65%
transfer learning (densenet121)	KL-1 vs. KL-2	69.73%
ours (resnet34)	KL-1 vs. KL-2	70.69%
ours (densenet121)	KL-1 vs. KL-2	**71.78%**
**evaluated on the resampled balancing data		
Nasser et al. (2020)	KL-0 vs. KL-1	69.83%
ours (resnet34)	KL-0 vs. KL-1	65.50%
ours (densenet121)	KL-0 vs. KL-1	65.50%
Nasser et al. (2020)	KL-0 vs. KL-2	82.53%
ours (resnet34)	KL-0 vs. KL-2	83.19%
ours (densenet121)	KL-0 vs. KL-2	84.66%
Nasser et al. (2020)	KL-1 vs. KL-2	77.05%
ours (resnet34)	KL-1 vs. KL-2	70.96%
ours (densenet121)	KL-1 vs. KL-2	72.77%



(a) KL-0 vs. KL-1 metrics on each fold



(b) KL-0 vs. KL-2 metrics on each fold



(c) KL-1 vs. KL-2 metrics on each fold

Figure 7.3: Metrics of different methods on each fold (evaluated on all early-stage samples)

The hyper-parameters are the same as Chapter 7.1.1. For our method, we list the average ensembled results of all five folds. Notably, Nasser et al. (2020) explored multiple categories of classifiers followed by the discriminative regularization auto-encoder (DRAE). For each image, the authors extracted five ROIs and trained the corresponding DRAE and classifiers. The single decision of each image aggregated predictions from five ROIs through the max-voting strategy. Among the classifiers reported in Nasser et al. (2020), SVM-RBF achieved the best performance, which are listed in Table 7.2.

Compared to Antony et al. (2016), the accuracy improves on all three binary classification tasks. We also outperform the transfer learning baseline on these three tasks. In terms of each fold's performance, Figure. 7.3 shows similar trends as five-stage tasks.

On the other hand, we observe that on the KL-0 vs. KL-1 and KL-1 vs. KL-2 tasks, Nasser et al. (2020) reaches higher accuracy than our work. Table 7.2 shows that the classification performance benefits from prior expert knowledge. However, applying the method of Nasser et al. (2020) to a clinical environment is difficult due to the intensive human intervention. The ROI extraction is based on the manually annotated tibial edge, which requires an expert to check every image in the dataset. The advantage of the proposed method is the fully automatic end-to-end procedure for OA assessment. Notably, the experiments in this work are based on the automatic knee segmentation by the YOLO model.

7.2 Label Confidence Estimation

7.2.1 Low Confidence Sample Characterization

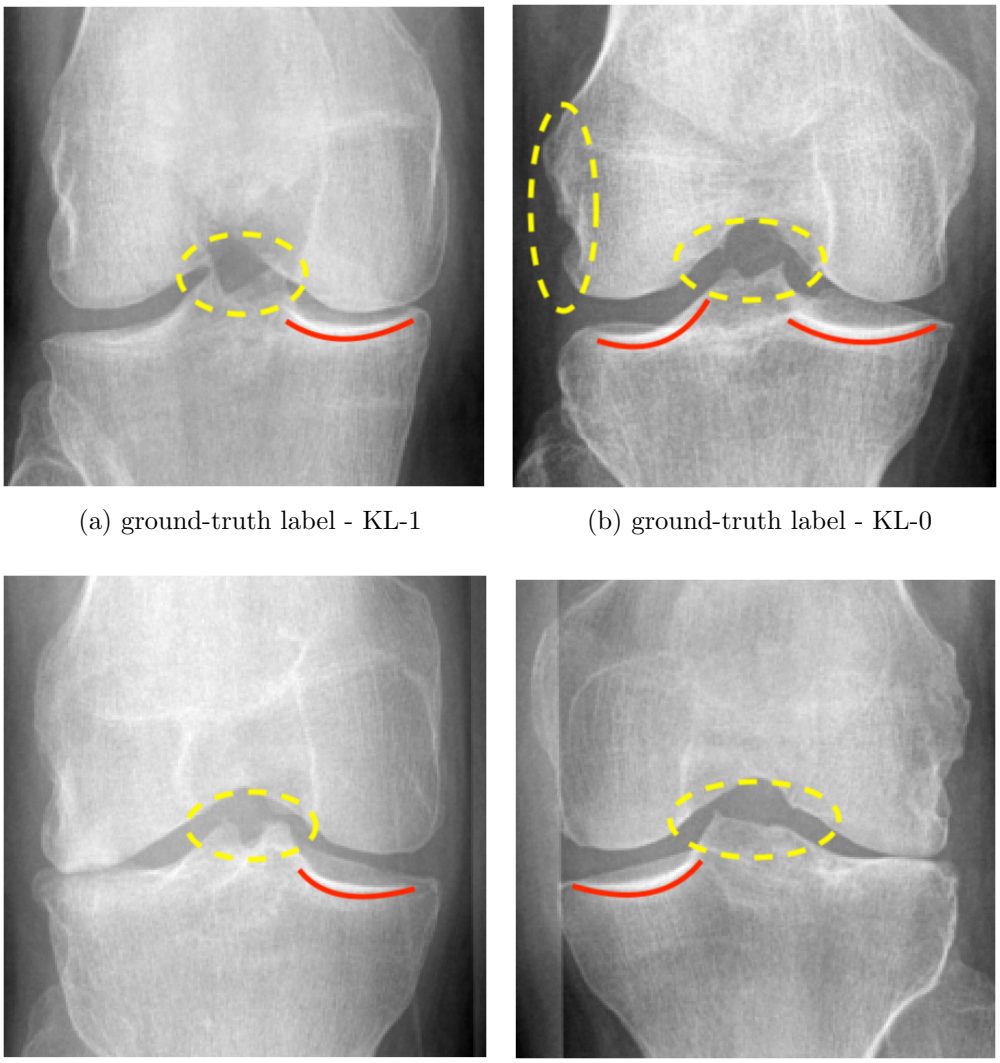
Characterizing low confidence samples is the foundation of estimating label confidence. We verify the characterized low confidence samples from two aspects.

First, we present the low confidence samples characterized by the densenet121 from the validation set to radiologists to re-examine the KL grade. In Figure. 7.4, we show the

ambiguous samples considered by radiologists, who highlighted the suspicious lesions which may affect the decision. The case study confirms the existence of low confidence samples which do not have significant evidence of their KL grade. If treated similar to the high confidence samples, CNN could memorize these samples instead of learning general patterns.

Second, due to the difficulty of verifying large-scale low confidence samples, we use the label-shifting of early-stage samples to simulate the errors made by individual annotators. Particularly, we randomly shift the KL-0, KL-1, and KL-2 labels to its adjacent class. The ratios of label-shifting are 5% and 10% for the training and validation set respectively. Then, we use the densenet121 to verify our method's awareness of label confidence level change. Meanwhile, we keep track of the label-shifting samples to check whether they were characterized during training. The transfer learning method is used as a baseline here. Table 7.3 shows the mean label confidence level and accuracy. As we are adding label noise in the dataset, the accuracy of both methods decreases. However, our training scheme still outperforms the baseline by nearly 2%. On the other hand, the estimated label confidence level also drops from 72.6% o 69.1%, which indicates the awareness of the label noise change. We observe that the estimated label confidence does not strictly follow the label-shifting ratio. For example, when we add 5% label noise, the estimated label confidence drops by 2.4%. With 5% more noise, it further drops by 1.1%. The amount of noisy samples undermines the low confidence sample estimation result. As low confidence data becomes dominant, the CNN cannot distinguish a normal sample from a noisy one, resulting in the noisy samples being categorized as normal. Northcutt et al. (2021) also discuss such an issue by assuming the correctly labeled samples dominating each class.

By tracking the manually added noisy samples, we calculate the average percentage of those found by CNN. 74.22% of the noisy samples are detected under 5% condition and 70.54% under 10% condition. This result suggests that the label confidence estimation is a valid method to detect low confidence samples on a large-scale dataset. Due to the



(a) ground-truth label - KL-1

(b) ground-truth label - KL-0

(c) ground-truth label - KL-4

(d) ground-truth label - KL-4

Figure 7.4: Low confidence samples verified by individual annotator. For each image, we show the label obtained from OAI on the bottom. Highlighted areas are annotated by the radiologists, which may lead to a low label confidence. Evident osteophyte is indicated by yellow circle. Sclerosis is annotated by red lines. In Figure. 7.4a and Figure. 7.4b, these features may lead to a higher KL level assessment. In Figure. 7.4c, and Figure. 7.4d, the joint space narrowing is asymmetrical, which is mainly located on medial side. This is the main reason that an individual annotator may underestimate the severity.

Table 7.3: Comparisons of the Performance under Random Label-Shifting (LS) Conditions (five-class task, the densenet121 model)

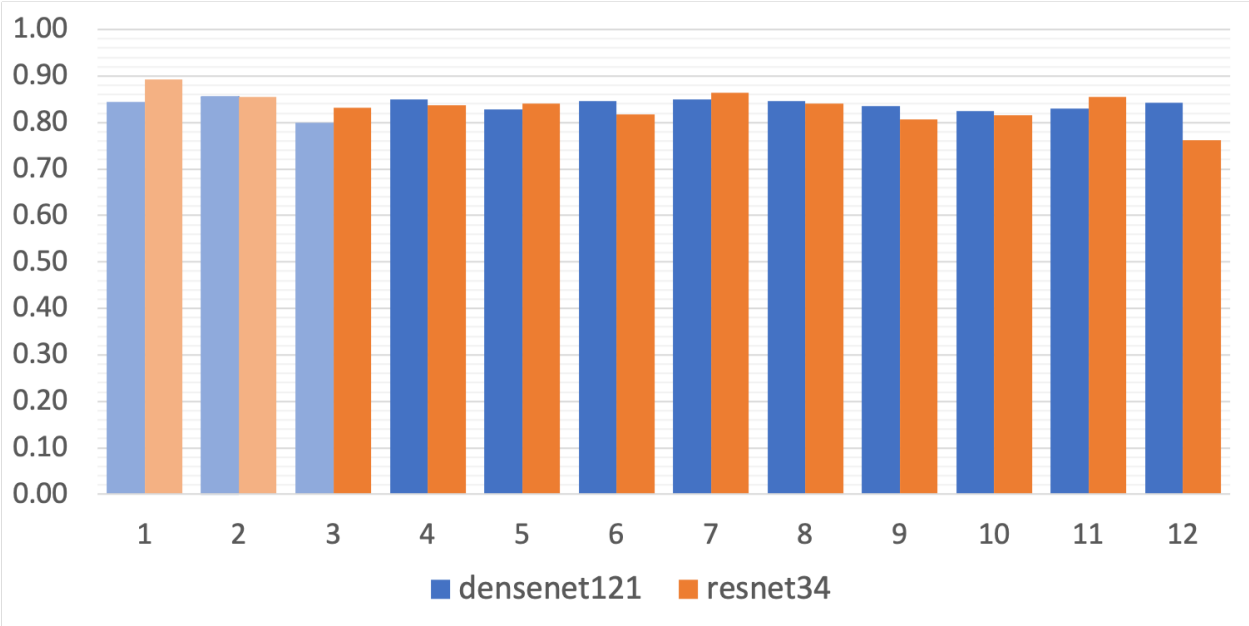
Method	Random LS Ratio	Label Confidence (KL-0,1,2)	Accuracy
baseline	0%	-	67.59%
proposed	0%	0.726	70.13%
baseline	5%	-	66.48%
proposed	5%	0.702	68.79%
baseline	10%	-	66.73%
proposed	10%	0.691	67.87%

uncertainty in the original OAI dataset, our estimation does not perfectly match the manually added noisy samples. However, it provides a good reference for our interactive training and hybrid loss function.

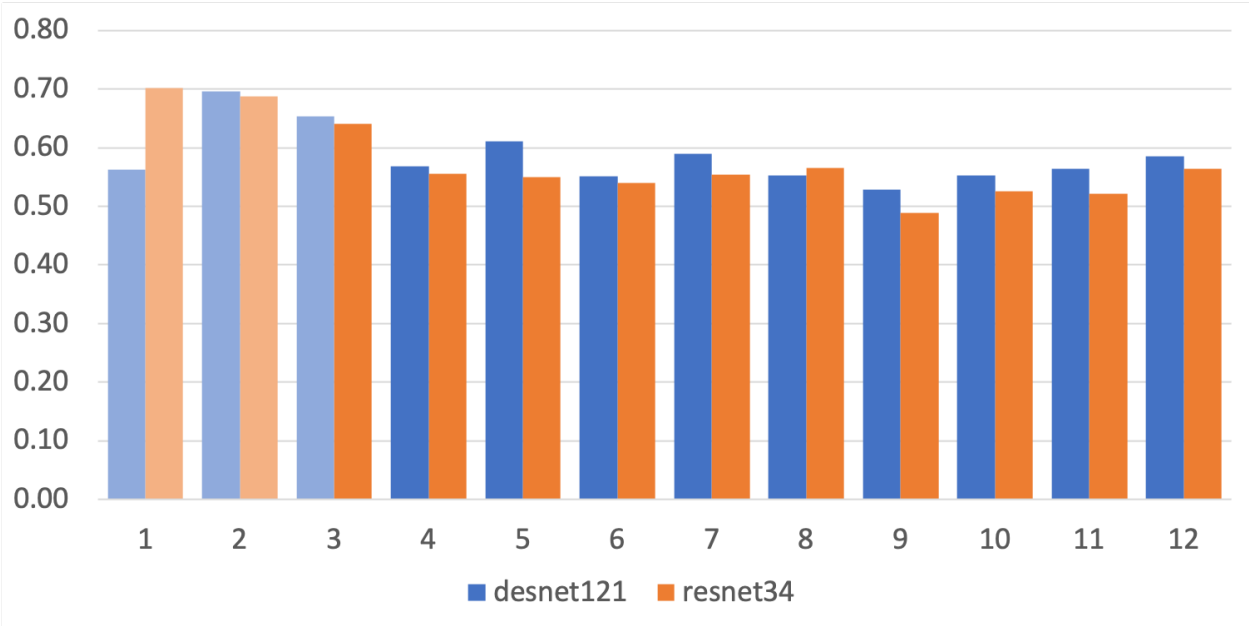
7.2.2 Label Confidence Estimation Process

To unravel the interaction of model training and label confidence estimation, we show the mean of label confidence after each epoch in Figure. 7.5. Figure. 7.5a shows the results for the KL-0, which stands for "no OA". Figure. 7.5b shows the results of the KL-1, which represents the "doubtful OA". As the label confidence estimations of KL-2, KL-3, and KL-4 are similar to the KL-0, we do not show them here.

Throughout the training, we observe a similar trend from KL-0 and KL-1, where the label confidence level is dynamically balanced. It indicates that the training process is consistently pushing CNN learning from the high confidence samples. Moreover, it maintains the stability of the weights used by our hybrid loss. On the other hand, we find the label confidence of KL-1 lower than other classes. This difference could be explained by the fact that the radiographical evidence in KL-1 images is less determinative than others. The similar estimation of two CNN models proves the reliability of our method. For the uncertain data, the annotator's personal experience influences the given label Y, which determines the label confidence. Thus, the estimation results are model-independent. As expected, we



(a) KL-0 label confidence



(b) KL-1 label confidence

Figure 7.5: Estimated confidence level after each epoch. The less saturated colors of the first several bars represent the warm-up epoch. To draw this figure, we take an average over the results of all five-fold training (with $\lambda = 0.01$ for both CNNs).

Table 7.4: Comparisons of Different Warm-up Epoch

Model	Accuracy	MCC
resnet34 (epoch = 1)	67.97%	0.5552
resnet34 (epoch = 2)	68.32%	0.5561
resnet34 (epoch = 3)	67.91%	0.5555
densenet121 (epoch = 1)	69.76%	0.5815
densenet121 (epoch = 2)	69.86%	0.5826
densenet121 (epoch = 3)	70.13%	0.5864

observed no significant differences in Figure. 7.5 regarding the two CNNs, suggesting that our method is reliable.

7.3 Effects of Hyper-parameters

In the proposed method, two hyper-parameters control the learning process. The number of warm-up epochs determines when to apply the label confidence information. And λ in the proposed loss function determines the weights of the low confidence set. We use the 5-class task to examine hyper-parameter effects in this section.

7.3.1 Effects of warm-up epoch

The effects of warm-up epoch are shown in Table 7.4. It suggests that the influence or warm-up epoch is not significant. The difference caused by the warm-up epoch is within 0.4% in terms of accuracy and 0.001 in terms of MCC for both CNNs. Similar to Figure. 7.5, this result suggests that the training scheme reaches a stable state after the first one or two epochs. In this case, the final performance is not sensitive to the warm-up hyper-parameter.

Table 7.5: Comparisons of Different λ

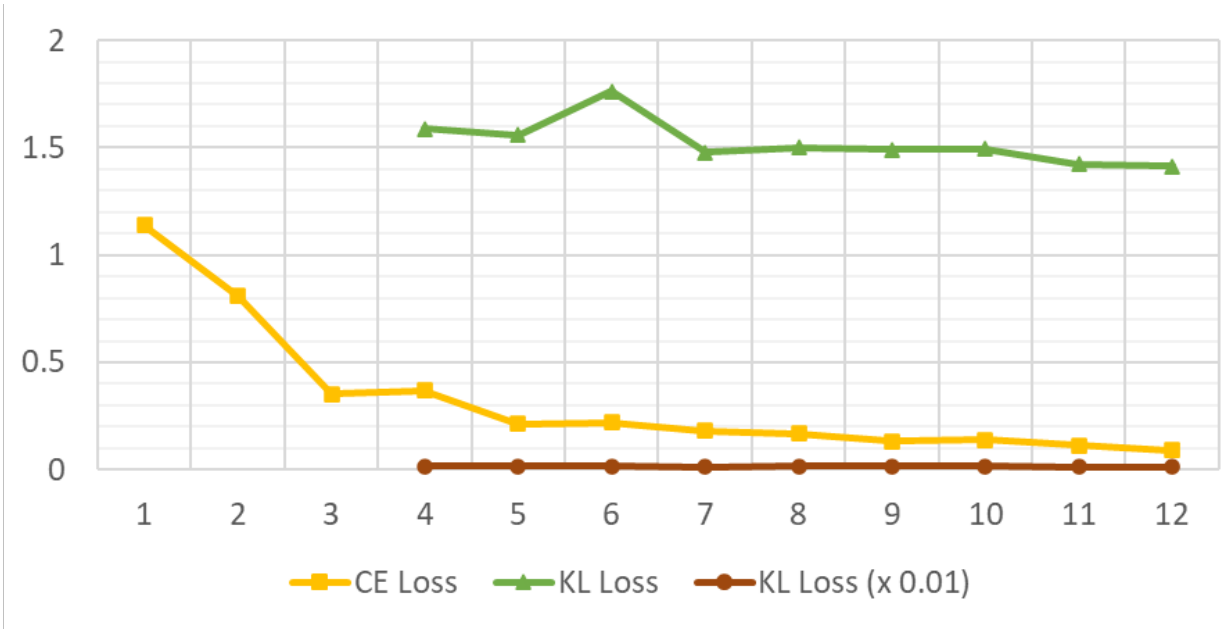
Model	Accuracy	MCC
resnet34 ($\lambda = 0$)	67.86%	0.5612
resnet34 ($\lambda = 0.01$)	68.32%	0.5561
resnet34 ($\lambda = 0.05$)	66.92%	0.5521
densenet121 ($\lambda = 0$)	69.53%	0.5774
densenet121 ($\lambda = 0.01$)	70.13%	0.5864
densenet121 ($\lambda = 0.05$)	68.97%	0.5654

7.3.2 Effects of λ in hybrid loss

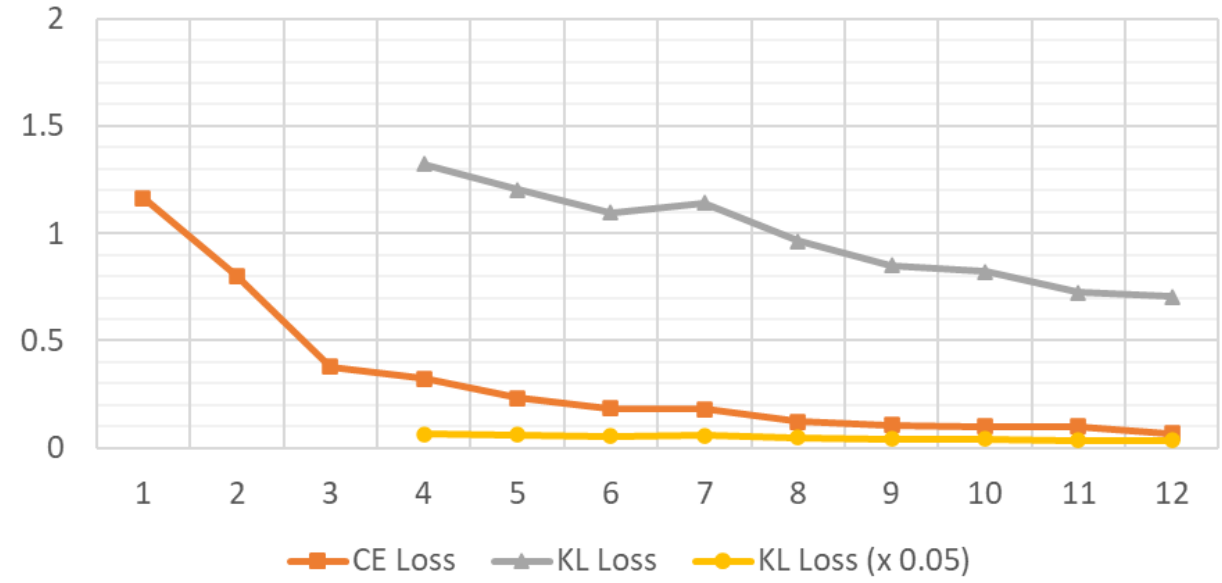
Table 7.5 shows that setting λ as 0.01 yields the best performance. When λ is 0, the accuracy scores decrease by 0.46% for resnet34 and 0.6% for densenet121. On the other hand, when λ is 0.05, the accuracy scores also drop by 1.4% (resnet34) and 1.16% (densene121).

Results in Table 7.5 reflect the impacts of λ. When lambda is 0, it is equivalent to discarding the low confidence set. Compared to the baselines, the resnet34's accuracy increases by 1.95%, and that of densenet121 by 1.84%. CNNs achieve the major improvement by estimating label confidence and learning from high confidence samples. As discussed in Chapter 5.3, machine learning models can make empirical errors on the unseen data. Results in Table 7.5 confirm the benefit of using λ to remedy the empirical errors. However, when λ further increases to 0.05, it overestimates the loss caused by low confidence samples. Thus, the loss function cannot help CNN to learn from reliable samples.

To illustrate the learning process, we plot the average loss of each epoch in Figure. 7.6. Two terms of our hybrid loss function are plotted separately, marked as "CE Loss" and "KL Loss". As shown in Figure. 7.6a when λ is 0.01, it suppresses the impact of "KL Loss". Through the training, CNNs mainly learn from the "CE Loss", which is calculated from a high confidence set. However, in Figure. 7.6b, when λ is 0.05, weighted "KL Loss" is near the "CE Loss". During training, more efforts are made to minimize the "KL Loss" compared to Figure. 7.6a, especially in the later epochs. As shown in Table 7.5, overestimating the



(a) Loss values during training with $\lambda = 0.01$



(b) Loss values during training with $\lambda = 0.05$

Figure 7.6: Average training loss after each epoch. The two terms in the hybrid loss function are marked as "CE Loss" and "KL Loss". We also plot the weighted "KL Loss" with respect to different λ. To plot this figure, we observe the densenet121's training process on five-class task. And we use three epochs for the warm-up training.

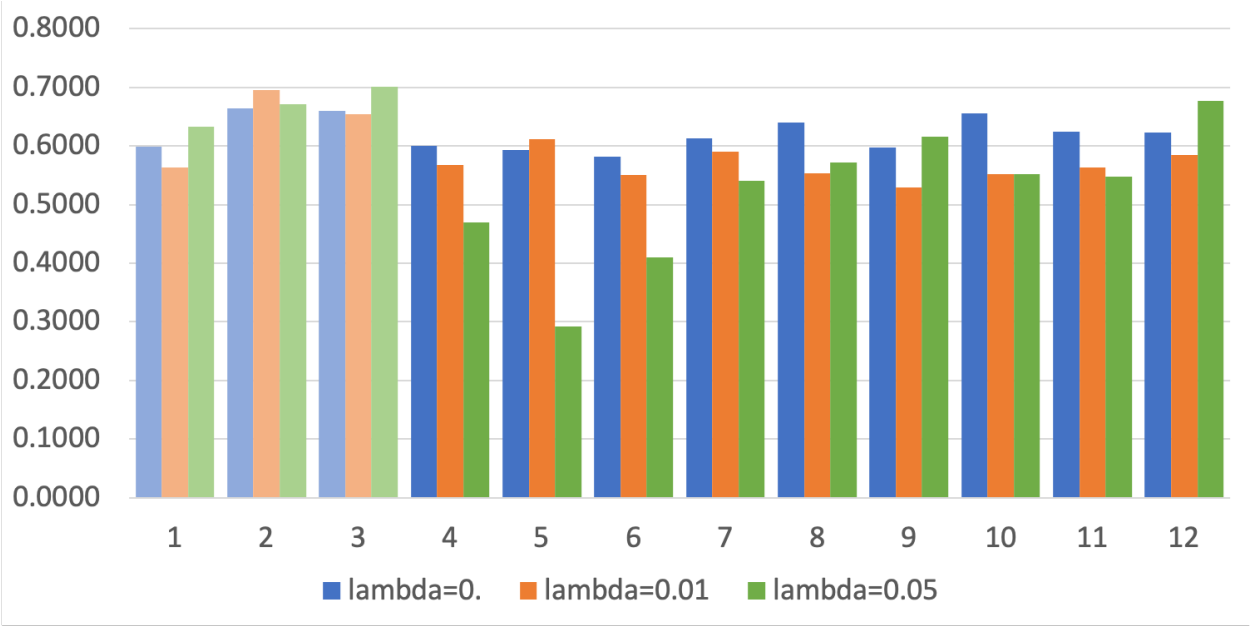


Figure 7.7: Averaged label confidence of KL-1 characterized by the densenet121 using different λ. The less saturated colors of the first several bars represent the warm-up epoch. When lambda increases, the estimated label confidence is becoming unstable.

"KL Loss" leads to a significant drop of the accuracy score. Inappropriate λ also affects the label confidence estimation. In Figure. 7.7, we show the label confidence of KL-1 under the conditions of different λ. Compare to 0 and 0.01, using 0.05 causes the fluctuation during the training. Although CNN manages to stabilize the trends in the later epoch, the overall performance is corrupted. Moreover, when we use 0.1 as λ in the hybrid loss function, the training process does not converge in the end.

CHAPTER 8

CONCLUSION

In this paper, we propose a novel training scheme and a hybrid loss function targeting the label uncertainty in the OA dataset. The proposed training scheme has two stages. First, in the label confidence estimation stage, we extract the label confidence information. In the model training stage, we use it to refine the samples. Moreover, the proposed hybrid loss function emphasizes the high confidence samples and suppresses low confidence ones. We conduct the experiments on two tasks to validate our approach, including five-stage OA assessment and early-stage OA detection. To examine the effect of low confidence sample characterization, we perform a manual case study and large-scale label noise interference experiments. Despite the fact that KL-0 vs. KL-1 and KL-1 vs. KL-2 tasks still benefit from the semi-automatic feature extraction, our approach reaches state-of-art performance on five-stage and KL-0 vs. KL-2 tasks without human intervention. As an object detection CNN is employed for the knee joint area segmentation, our method depends on the standard procedure to collect the X-ray screen data. In a clinical environment, data collection is affected by various factors, like the medical device and lesion area alignment. The impacts brought by the preprocessing method are not explored. We observe that our experiments run on the dataset from a single vendor. In future, we would like to explore the application of the proposed method on data from multiple sources.

To our knowledge, this is the first work to enhance the OA severity assessment from the view of sample confidence. Our work is a fully automatic and data-driven process for data refining, which differs from the previous researches. In future, introducing label confidence to other medical imaging problems holds promise.

Part of this chapter is reprinted from © 2021 IEEE. Reprinted, with permission, from Y. Wang et al., "Learning from Highly Confident Samples for Automatic Knee Osteoarthritis Severity Assessment: Data from the Osteoarthritis Initiative," in IEEE Journal of Biomedical and Health Informatics, doi: 10.1109/JBHI.2021.3102090.

APPENDIX

DATASET ACKNOWLEDGMENT

The OAI is a public-private partnership comprised of five contracts (N01-AR-2-2258; N01-AR-2-2259; N01-AR-2-2260; N01-AR-2-2261; N01-AR-2-2262) funded by the National Institutes of Health, a branch of the Department of Health and Human Services, and conducted by the OAI Study Investigators. Private funding partners include Merck Research Laboratories; Novartis Pharmaceuticals Corporation, Glaxo Smith Kline; and Pfizer, Inc. Private sector funding for the OAI is managed by the Foundation for the National Institutes of Health. This manuscript was prepared using an OAI public use data set and does not necessarily reflect the opinions or views of the OAI investigators, the NIH, or the private funding partners.

REFERENCES

Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Ahishakiye, E., M. B. Van Gijzen, J. Tumwiine, R. Wario, and J. Obungoloch (2021). A survey on deep learning in medical image reconstruction. *Intelligent Medicine*.

Allen, K. D. and Y. M. Golightly (2015). Epidemiology of osteoarthritis: state of the evidence. *Current opinion in rheumatology 27*(3), 276.

Angluin, D. and P. Laird (1988). Learning from noisy examples. *Machine Learning 2*(4), 343–370.

Antony, J., K. McGuinness, K. Moran, and N. E. O'Connor (2017). Automatic detection of knee joints and quantification of knee osteoarthritis severity using convolutional neural networks. In *International conference on machine learning and data mining in pattern recognition*, pp. 376–390. Springer.

Antony, J., K. McGuinness, N. E. O'Connor, and K. Moran (2016). Quantifying radiographic knee osteoarthritis severity using deep convolutional neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 1195–1200. IEEE.

Aprilliani, U. and Z. Rustam (2018). Osteoarthritis disease prediction based on random forest. In *2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pp. 237–240. IEEE.

Arpit, D., S. Jastrzkebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, et al. (2017). A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 233–242. JMLR. org.

Barron, J. T. (2017). Continuously differentiable exponential linear units. *arXiv preprint arXiv:1704.07483*.

Berrada, L., A. Zisserman, and M. P. Kumar (2018). Smooth loss functions for deep top-k classification. In *International Conference on Learning Representations*.

Breiman, L. (1996). Bagging predictors. *Machine learning 24*(2), 123–140.

Chen, P., L. Gao, X. Shi, K. Allen, and L. Yang (2019). Fully automatic knee osteoarthritis severity grading using deep neural networks with a novel ordinal loss. *Computerized Medical Imaging and Graphics 75*, 84–92.

Chen, P., B. B. Liao, G. Chen, and S. Zhang (2019). Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, pp. 1062–1070. PMLR.

Christodoulou, E., S. Moustakidis, N. Papandrianos, D. Tsaopoulos, and E. Papageorgiou (2019). Exploring deep learning capabilities in knee osteoarthritis case study for classification. In *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pp. 1–6. IEEE.

Clevert, D.-A., T. Unterthiner, and S. Hochreiter (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Culvenor, A. G., C. N. Engen, B. E. Øiestad, L. Engebretsen, and M. A. Risberg (2015). Defining the presence of radiographic knee osteoarthritis: a comparison between the kellgren and lawrence system and oarsi atlas criteria. *Knee Surgery, Sports Traumatology, Arthroscopy 23*(12), 3532–3539.

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee.

Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Drory, A., O. Ratzon, S. Avidan, and R. Giryes (2018). The resistance to label noise in k-nn and cnn depends on its concentration. *arXiv preprint arXiv:1803.11410*.

Elkan, C. (2001). The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, Volume 17, pp. 973–978. Lawrence Erlbaum Associates Ltd.

Forman, G. (2005). Counting positives accurately despite inaccurate classification. In *European Conference on Machine Learning*, pp. 564–575. Springer.

Forman, G. (2008). Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery 17*(2), 164–206.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448.

Górriz, M., J. Antony, K. McGuinness, X. Giró-i-Nieto, and N. O'Connor (2019, 08–10 Jul). Assessing knee oa severity with cnn attention-based end-to-end architectures. In *Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning*, Volume 102 of *Proceedings of Machine Learning Research*, pp. 197–214. PMLR.

Han, B., Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama (2018). Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pp. 8527–8537.

He, K., G. Gkioxari, P. Dollár, and R. Girshick (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969.

He, K., X. Zhang, S. Ren, and J. Sun (2016a). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

He, K., X. Zhang, S. Ren, and J. Sun (2016b). Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer.

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

Ioffe, S. and C. Szegedy (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR.

Jiang, L., Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei (2018). Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pp. 2304–2313. PMLR.

Jurman, G., S. Riccadonna, and C. Furlanello (2012, 08). A comparison of mcc and cen error measures in multi-class prediction. *PLOS ONE 7*(8), 1–8.

Karimi, D., H. Dou, S. K. Warfield, and A. Gholipour (2020). Deep learning with noisy labels: exploring techniques and remedies in medical image analysis. *Medical Image Analysis 65*, 101759.

Kellgren, J. and J. Lawrence (1957). Radiological assessment of osteo-arthrosis. *Annals of the rheumatic diseases 16*(4), 494.

Klambauer, G., T. Unterthiner, A. Mayr, and S. Hochreiter (2017). Self-normalizing neural networks. *Advances in neural information processing systems 30*.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems 25*.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*(11), 2278–2324.

Lundervold, A. S. and A. Lundervold (2019). An overview of deep learning in medical imaging focusing on mri. *Zeitschrift für Medizinische Physik 29*(2), 102–127.

Ma, X., Y. Wang, M. E. Houle, S. Zhou, S. Erfani, S. Xia, S. Wijewickrema, and J. Bailey (2018). Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, pp. 3355–3364.

Maiese, K. (2016). Picking a bone with wisp1 (ccn4): new strategies against degenerative joint disease. *Journal of translational science 1*(3), 83.

Mason, D. (2011). Su-e-t-33: pydicom: an open source dicom library. *Medical Physics 38*(6Part10), 3493–3493.

Mirikharaji, Z., Y. Yan, and G. Hamarneh (2019). Learning to segment skin lesions from noisy annotations. In *Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data*, pp. 207–215. Springer.

Nasser, Y., R. Jennane, A. Chetouani, E. Lespessailles, and M. El Hassouni (2020). Discriminative regularized auto-encoder for early detection of knee osteoarthritis: Data from the osteoarthritis initiative. *IEEE Transactions on Medical Imaging*.

Natarajan, N., I. S. Dhillon, P. Ravikumar, and A. Tewari (2017). Cost-sensitive learning with noisy labels. *J. Mach. Learn. Res. 18*(1), 5666–5698.

Natarajan, N., I. S. Dhillon, P. K. Ravikumar, and A. Tewari (2013). Learning with noisy labels. In *Advances in neural information processing systems*, pp. 1196–1204.

Northcutt, C., L. Jiang, and I. Chuang (2021). Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research 70*, 1373–1411.

Oliveira, F. P. and J. M. R. Tavares (2014). Medical image registration: a review. *Computer methods in biomechanics and biomedical engineering 17*(2), 73–93.

Ouali, Y., C. Hudelot, and M. Tami (2020). An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*.

Palazzo, C., C. Nguyen, M.-M. Lefevre-Colau, F. Rannou, and S. Poiraudeau (2016). Risk factors and burden of osteoarthritis. *Annals of physical and rehabilitation medicine 59*(3), 134–138.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc.

Ratner, A., P. Varma, and B. Hancock (2019). Weak supervision: A new programming paradigm for machine learning—sail blog. *Visited on 6*(26), 2020.

Redmon, J., S. Divvala, R. Girshick, and A. Farhadi (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

Redmon, J. and A. Farhadi (2017). Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271.

Ren, S., K. He, R. Girshick, and J. Sun (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99.

Ryan, S. (2020). *Nursing Older People with Arthritis and other Rheumatological Conditions*. Springer.

Salem, N., H. Malik, and A. Shams (2019). Medical image enhancement based on histogram algorithms. *Procedia Computer Science 163*, 300–311.

Selvaraju, R. R., M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626.

Settles, B. (2009). Active learning literature survey.

Shamir, L., S. M. Ling, W. W. Scott Jr, A. Bos, N. Orlov, T. J. Macura, D. M. Eckley, L. Ferrucci, and I. G. Goldberg (2008). Knee x-ray image analysis method for automated detection of osteoarthritis. *IEEE Transactions on Biomedical Engineering 56*(2), 407–415.

Sharma, S., S. S. Virk, and V. Jain (2016). Detection of osteoarthritis using svm classifications. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 2997–3002. IEEE.

Suresha, S., L. Kidziński, E. Halilaj, G. Gold, and S. Delp (2018). Automated staging of knee osteoarthritis severity using deep neural networks. *Osteoarthritis and Cartilage 26*, S441.

Tan, C., F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu (2018). A survey on deep transfer learning. In *International conference on artificial neural networks*, pp. 270–279. Springer.

Tiulpin, A., S. Klein, S. M. Bierma-Zeinstra, J. Thevenot, E. Rahtu, J. van Meurs, E. H. Oei, and S. Saarakkala (2019). Multimodal machine learning-based knee osteoarthritis progression prediction from plain radiographs and clinical data. *Scientific reports 9*(1), 1–11.

Tiulpin, A., J. Thevenot, E. Rahtu, P. Lehenkari, and S. Saarakkala (2018). Automatic knee osteoarthritis diagnosis from plain radiographs: A deep learning-based approach. *Scientific reports 8*(1), 1–10.

van der Woude, J., S. Nair, R. Custers, J. Van Laar, N. Kuchuck, F. Lafeber, and P. Welsing (2016). Knee joint distraction compared to total knee arthroplasty for treatment of end stage osteoarthritis: simulating long-term outcomes and cost-effectiveness. *PloS one 11*(5).

Van Rooyen, B., A. Menon, and R. C. Williamson (2015). Learning with symmetric label noise: The importance of being unhinged. In *Advances in Neural Information Processing Systems*, pp. 10–18.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Wang, R., T. Lei, R. Cui, B. Zhang, H. Meng, and A. K. Nandi (2020). Medical image segmentation using deep learning: A survey. *IET Image Processing*.

Weiss, K., T. M. Khoshgoftaar, and D. Wang (2016). A survey of transfer learning. *Journal of Big data 3*(1), 1–40.

Wittenauer, R., L. Smith, and K. Aden (2013). Background paper 6.12 osteoarthritis. *World Health Organisation*.

Wu, B., C. Xu, X. Dai, A. Wan, P. Zhang, Z. Yan, M. Tomizuka, J. Gonzalez, K. Keutzer, and P. Vajda (2020). Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677*.

Xue, C., Q. Dou, X. Shi, H. Chen, and P.-A. Heng (2019). Robust learning at noisy labeled medical images: applied to skin lesion classification. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pp. 1280–1283. IEEE.

Yamashita, R., M. Nishio, R. K. G. Do, and K. Togashi (2018). Convolutional neural networks: an overview and application in radiology. *Insights into imaging 9*(4), 611–629.

Yu, F. and V. Koltun (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.

Zhou, S. K., H. Greenspan, C. Davatzikos, J. S. Duncan, B. Van Ginneken, A. Madabhushi, J. L. Prince, D. Rueckert, and R. M. Summers (2021). A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. *Proceedings of the IEEE*.

Zou, K. H., S. K. Warfield, A. Bharatha, C. M. Tempany, M. R. Kaus, S. J. Haker, W. M. Wells III, F. A. Jolesz, and R. Kikinis (2004). Statistical validation of image segmentation quality based on a spatial overlap index1: scientific reports. *Academic radiology 11*(2), 178–189.

BIOGRAPHICAL SKETCH

Yifan Wang received his BSc Degree in Microelectronics from the Fudan University, Shanghai, China in 2015. He earned his M.Eng. in Integrated Circuit Engineering from Fudan University, Shanghai, China in 2017. He is currently pursuing his PhD degree in Computer Engineering from The University of Texas at Dallas. His current research interests include the application of machine learning methods on medical image analysis.

CURRICULUM VITAE

Yifan Wang

May 15, 2022

Contact Information:

Department of Electrical & Computer Engineering
The University of Texas at Dallas
800 W. Campbell Rd.
Richardson, TX 75080-3021, U.S.A.

Email: Yifan.Wang9@utdallas.edu

Educational History:

PhD, Computer Engineering, The University of Texas at Dallas, TX, U.S., 2022, Advisor: Dian Zhou
MEng, Integrated Circuit Engineering, Fudan University, Shanghai, China, 2015
BS., Microelectronics, Fudan University, Shanghai, China, 2015

Employment History:

Teaching Assistant, Department of Electrical Engineering, the University o Texas at Dallas, Fall 2017 – present
R&D Intern, Synopsys, Inc., Shanghai, China, June 2016 – Dec. 2016
R&D Intern, Cadence Design Systems, Shanghai, China, Oct. 2015 – Apr. 1026
QA Engineer Intern, Dell EMC, Shanghai, China, Apr. 2015 – Sep. 2015

Research and Projects:

Integrated Circuit (IC) Design Parameter Optimization Framework – Summer 2021

- Built an optimization framework to improve the efficiency of the IC parameter design process using machine learning methods.
- Modeled the IC behavior with the Gaussian process and searched for optimal design parameters with Bayesian optimization.

Knee Osteoarthritis (OA) Assessment By Analyzing Label Noise – Fall 2020 - Spring 2021

- Designed and built a knowledge distill procedure to enhance the CNN's performance on estimating the OA severity.

- Automatically refined the dataset by estimating label confidence through a two-phase interactive training; proposed and implemented a hybrid loss function for the samples of different confidence levels.

Automatic Knee Osteoarthritis (OA) Assessment – Fall 2020 - Spring 2021

- Developed a fully-automatic end-to-end procedure for OA severity assessments.
- Located and extracted the knee joint area through object detection technique; Assessed the OA severity using a two-stage model including CNN backbone and visual transformer.

Medical Segmentation Research: Identify Glomeruli in Human Kidney Tissue Images – Spring 2021

- Developed a deep learning model to detect the functional tissue units from kidney images.
- Implemented the "U-net + visual transformer" model to enhance the segmentation performance; trained the model through two stages, including warm-up training on random cropped data and fine-tuning on hard samples.

Bayesian Optimization with Gaussian Process – Spring 2019

- Implemented the Gaussian process model and Bayesian optimization framework for the black-box optimization problem.
- Built the Gaussian process model with CUDA to improve the kernel computation efficiency; cooperating with the function minimization using C++.

Teaching Experience:

I participated assisting the following class at the University of Texas at Dallas:

Undergraduate level classes

CE/EE3120 Digital Circuits Laboratory
TE3310 Electrical Network Analysis
CE/EE2310 Introduction to Digital System
Graduate level classes
CE/EEDG Application Specific Integrated Circuits Design
CE/EE 4370 Embedded Systems

Colloquium & Presentations:

- "Deep Learning on Biomedical and Health Informatics", Ph.D. Qualifying Examination, The University of Texas at Dallas, March 2018.

- "Learning from Highly Confident Samples for Automatic Knee Osteoarthritis Severity Assessment", Doctoral Proposal, The University of Texas at Dallas, October, 2020.
- "Semi-Supervised Learning with Label Confidence for Automatic Knee Osteoarthritis Severity Assessment", Ph.D. Dissertation Defense, The University of Texas at Dallas, March 2022.

Publications:

- **Wang, Yifan**, Zhaori Bi, Yuxue Xie, Tao Wu, Xuan Zeng, Shuang Chen, and Dian Zhou. "Learning from Highly Confident Samples for Automatic Knee Osteoarthritis Severity Assessment: Data from the Osteoarthritis Initiative." IEEE Journal of Biomedical and Health Informatics (2021).
- **Wang, Yifan**, Xianan Wang, Tianning Gao, Le Du, and Wei Liu. "An automatic knee osteoarthritis diagnosis method based on deep learning: data from the osteoarthritis initiative." Journal of Healthcare Engineering 2021 (2021).
- He Biao, Zhang Shuhan, **Wang Yifan**, Gao Tianning, Yang Fan, Yan Changhao, Zhou Dian, Bi Zhaori, and Xuan Zeng. "A Batched Bayesian Optimization Approach for Analog Circuit Synthesis via Multi-Fidelity Modeling" IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (under minor revision)
- **Yifan, Wang**, Zhaori Bi, Le Du, Tianning Gao, Dian Zhou, and Guoxin Ye. "Medical Segmentation with Swin-Gate Unet" IEEE Journal of Translational Engineering in Health and Medicine (under review)