HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

GRADUATION THESIS

Multi-Domain Information Fusion for Plasmodium Life Cycle Development Classification

TRAN QUOC KHANH
khanh.tq190085@sis.hust.edu.vn

Major: Data Science

Supervisor: Doctor Nguyen Thi Oanh _____

Signature

Department: Computer Science

School: School of Information and Communications Technology

HANOI, 1/2025

ACKNOWLEDGMENT

I would express my deepest gratitude for my family, without them this thesis would not have been possible. My heartfelt thanks go to Dr. Nguyen Thi Oanh for invaluable supervision. Special thanks to members of the CESPA Project, Assoc.Prof. Muriel Visani and Asst.Prof. Thierry Urruty, for constructive feedback and suggestions. I would like to extend my appreciation to Dr. Sun Yajuan for great experience at Singapore Institute of Manufacturing Technology; Dr. Michel Toulouse and Assoc.Prof. Muriel Visani for great lectures at HUST; Dr. Ajren Verhoeff at OTH Regensburg for kindness and encouragement. I am grateful to my beloved friends Hannah Stücke, Farzan Tajbakhsh, Zhao Lan, Le Minh Nghia, Nguyen Nhu Ngoc, Nguyen Hoang Anh, Pham Viet Hoang, and Nguyen Mai Phuong. Last but not least, I am proud of my younger self for courage to embark on this academic journey.

ABSTRACT

Malaria is a fatal disease causes by plasmodium parasite, which is transmitted to human by mosquito. The disease burdens societies with medical as well as socioeconomic issues, especially in developing countries. Early diagnosis of malaria disease can reduce the impact of the disease. However, requirement of human resources for microscopy diagnosis raises challenges, especially in developing countries, which are mostly affected by the disease. Computer aided diagnosis with deep learning model can be a solution for this issue. Despite the fact that diagnosis of life cycle development is important for clinical testing of new medicine, current studies only focused on detecting presence of plasmodium, leaving life cycle classification nearly untouched. Additionally, despite the severe data imbalance problem, there are only a few studies worked on this issue of life cycle development classification. To this end, this thesis study takes advantages of multiple datasets to enrich minor class training data. Due to different data acquisition methods, simply adding training data from multiple datasets might cause domain distribution gap, thus negatively affect model's performance. For this reason, Multi-Domain Information Fusion (MDIF) has been proposed to effectively integrate information across domains. In particular, MDIF consists of graph convolutional network, knowledge graph (adjacency matrix) construction, and agent nodes. Knowledge graph is designed to model the domain relationship at domain and class level. Agent node is a global representation to reduce negative impact of noises. Quantitative, qualitative evaluation, and analyses have been carried out to show the effectiveness of our proposed module.

Student

(Signature and full name)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Definition
CNN	Convolutional Neural Network
GCN	Graph Convolutional Network
KNN	K-nearest Neighbors
MLP	Multi-layer Perceptron
RBC	Red Blood Cell
SMV	Support Vector Machine

CHAPTER 1. INTRODUCTION

The introduction chapter presents an overview of the thesis study. First, impacts of the malaria disease and the significance of automatic diagnosis of malaria parasite are discussed in Section 1.1. Medical aspect of plasmodium parasite is explained in Section 1.2 to provide basic knowledge. The following Section 1.3 briefly provides current state of research on plasmodium classification task and points out the research gap. Scope of research is then defined in this Section 1.4. Objective and contributions of the thesis are explained in the Section 1.5. Finally, Section 1.6 shows an overview of the thesis's structure.

1.1 Problem Statement

Malaria is a disease caused by plasmodium parasite. It has a profound impact on societies. According to World Health Organization, there was an estimation of 263 million malaria cases and 597 thousands malaria deaths in 83 countries in 2023 [1]. The burden of malaria goes beyond the realm of healthcare. A study on economic and social burden of malaria found that the rate of economic growth per year in countries highly effected by malaria disease was 1.3% lower than in other countries from 1965 to 1990 [2]. In Africa, it was estimated that a half of school days missed among schoolchildren are due to malaria disease [3].

The parasite is transmitted to human from infected female mosquitoes. Malaria disease causes symptoms, such as fever, headache, and chills in patients within 10-15 days of getting parasite transmitted by mosquito. Some cases may cause severe illness and even deaths. Because some malaria symptoms are not specific, early diagnosis of the malaria disease is important to reduce the fatality and prevents deaths [1]. Methods for parasite diagnosis testing includes microscopy and rapid diagnosis test. Nowadays, manual microscopic examination of stained thin and thick blood smears remains standard. However, microscopy method is time-consuming and highly-skilled microscopists are required [4]. This condition poses the challenges in remote areas, especially in Africa, where carries 94% of malaria cases and 95% of deaths [1].

Recent advancement in deep learning and computer vision provides a potential solution for this issue. Automatic detection and recognition model can be a low-cost and quick diagnosis method for malaria disease. The computer-aided diagnosis thus can enhance accuracy and efficiency of treatment, then reduce the disease burden for developing countries. Beside diagnosing presence of plasmodium parasite, stage of life cycle development is also important for clinical testing

of new medicine to treat malaria disease. In particular, the thesis study is part of a project collaborated with French Armed Forces Center for Epidemiology and Public Health (CESPA) and The French Armed Forces Biomedical Research Institute (IRBA) with the aim of developing an automatic clinical testing for malaria medicine. To this end, this thesis study is dedicated to the task of plasmodium life cycle classification.

1.2 Medical Background of Plasmodium

To provide a better understands of medical basis for this thesis study, this section briefly introduces general information about plasmodium parasite. There are five species of plasmodium, namely Plasmodium falciparum (P. falciparum), Plasmodium vivax (P. vivax), Plasmodium ovale (P. ovale), Plasmodium malariae (P.malariae), and Plasmodium knowlesi (P.knowlese). Among them, P. falciparum and P. vivax are the most threatening species, and P. falciparum is the deadliest [1].

All of the these species share the same life cycle development, which involves two major stages, in vector (mosquito) and host (such as human, monkey, and chimpanzee). The stage in host can be divided into two sub-stages, exoerythrocytic (liver) and erythrocytic (blood) stage. Figure 1.1 illustrates the life cycle development of plasmodium.

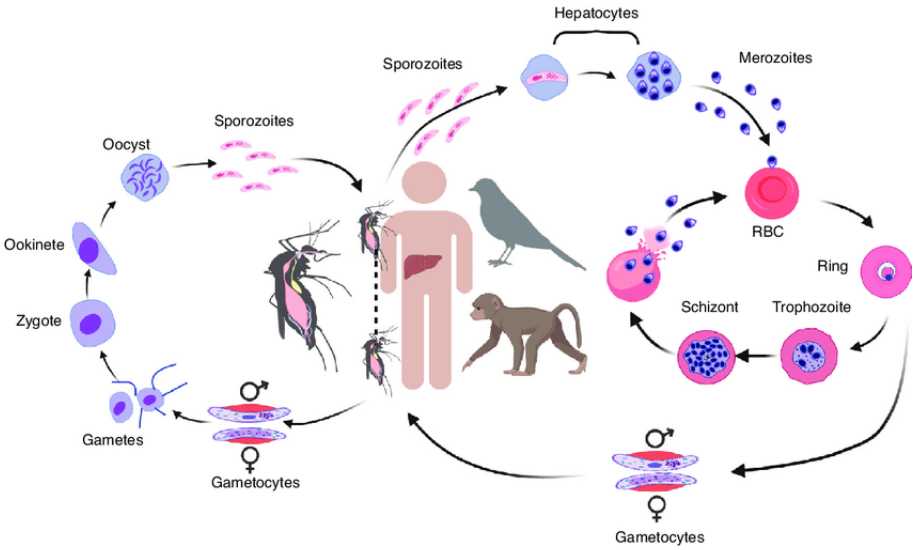


Figure 1.1: Illustration of life cycle development of plasmodium parasite. Figure extracted from Su et al. [5].

The development stage of plasmodium in host begins when the sporozoites are transmitted to host by infected vector. Sporozoites first invade hepatocytes (liver cells) and remain for 9-16 days in liver [6]. There, they undergo a multiplication process and each parasite multiplies into tens of thousands cells, named merozoites. When the liver cells are ruptured, merozoites are released back into blood.

Merozoites enter RBC (red blood cell) and start an asexual reproduction process. The first development stage of plasmodium in human blood is early trophozoite. Because of its morphological appearance, it is also known with the name ring. At trophozoite stage, the parasites grow and enlarge within the blood cells. After enlarging, trophozoites develop into schizonts. Each schizont has around 20 merozoites inside, being ready to invade other RBCs. A new cycle is repeated when the parasites erupt the occupied RBCs, release the merozoites back into blood. The duration of one cycle in blood depends on type of species. This time corresponds to symptom recurrence in malaria patients. P. knowlesi takes 24 hours, P. falciparum, P. ovale, and P.vivax take 48 hours, and P. malariae takes 72 hours to finish one cycle in human blood [7]. Figure 1.2 shows some examples of the plasmodium parasites in their life cycle development in human RBC.

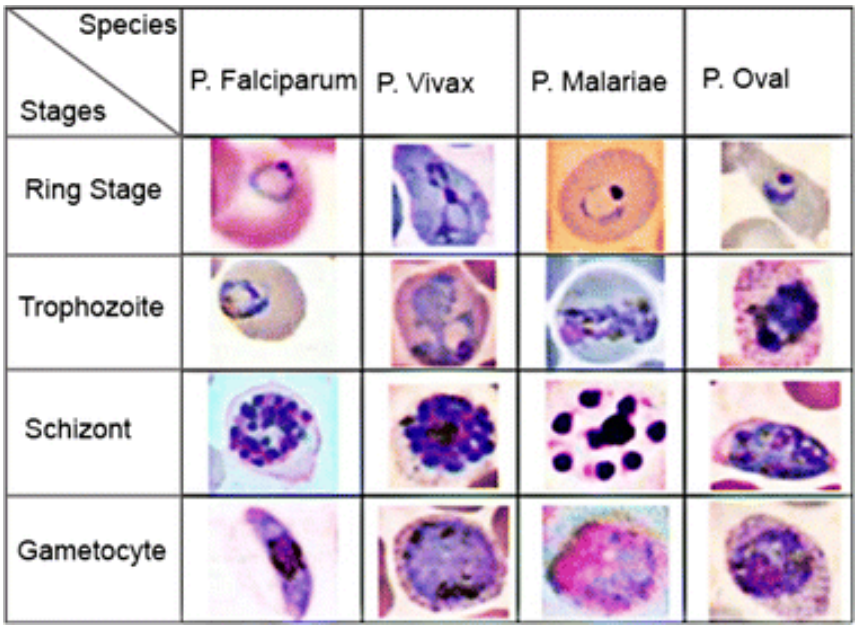


Figure 1.2: Samples of four different malaria parasite species and their life cycle development stages. Figure extracted from Jan et al. [4]

The responses of the hosts to the parasites, such as the release of immune effectors, may trigger a small number of parasites differentiating into male and female gametocytes [5]. This process is essential for the sexual reproduction of the parasites in vectors and the transmission of the disease to other hosts. More detailed information for life cycle development in vector could be found in the studies [6], [7].

Recently, microscopy examination of blood is the standard for diagnosis of malaria disease. World Health Organization provided a handbook for malaria diagnosis using microscopy method [8]. Blood sample is first applied a chemical dyes in a process called staining. The chemical helps coloring RBC and highlights

structures under microscope. Giemsa is a popular and low cost staining chemical. Then blood slide is prepared by one of two methods, named thin and thick blood smear. Thick blood smear is a drop of blood on grass slide dried for 30 minutes. With thin blood smear, blood drop is spread across grass slide and left dried for 10 minutes. The blood image is then captured by a microscope, and diagnosis can be carried out either by experts or automatically by a deep learning model. Example of procedure for diagnosis of plasmodium parasite from blood smear images are shown in Figure 1.3. Since the concentration of RBC in thick blood smear is higher than thin blood smear, thick smear method is often used to detect presence of parasite, while thin blood smear is more helpful to recognize the species and life cycle of plasmodium [4].

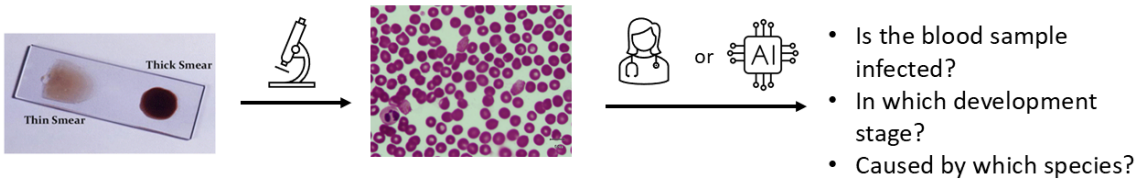


Figure 1.3: Plasmodium diagnosis from blood smear, carried out either manually by experts or semi-automatically by deep learning model.

1.3 State of Research

Recently, some studies have utilized machine learning to automatically detect plasmodium from blood smear images and have achieved good results [9]–[11]. However, these works only focused on detecting of presence of the parasites, let the life cycle detection unexplored. Although a few studies worked on the plasmodium life cycle detection and classification task [12]–[15], their evaluation strategies are not compatible with each other, making it hard to draw conclusion on state of search.

The main research gap in plasmodium life cycle classification is data imbalance issue. Since class imbalance is a nature of plasmodium life cycle, i.e. RBCs are naturally dominated by healthy RBC class, a few studies utilized a specific network to separate healthy and infected RBC [16], [17]. However, this strategy increases computational demand, making it hard to deploy in hardware-limited environment. To the best of our knowledge, Li et al. [14] are the only ones that employed additional dataset to add more minor instances. It should be noticed that the additional data is unlabeled, thus the quality might negatively affect model performance. The integration of multiple labeled data remains unexplored.

1.4 Scope of Research

There are several pipelines to detect and classify plasmodium life cycle from blood smear images. Because detection of RBCs from blood smear images have achieved good result [9], [10], within the study time frame, detection from blood smear images was excluded, only life cycle development classification from RBC was considered. In particular, we focused solely on developing a method for effective integration of data from multiple domains to tackle data imbalance. Experiments and evaluations to access the effectiveness of the proposed method were carried out. In addition to quantitative evaluations, qualitative analyses for further and more accurate assessment were also included.

1.5 Research Objectives and Contributions

With the main purpose of improving performance on minor classes, and given the aforementioned research gap, this thesis study introduced the concept of multi-domain learning to enrich minor classes with data from multiple domains to address severe data imbalance issue of plasmodium life cycle classification task. Since data from different domains are obtained in different ways, such as different microscopes, magnification scales, and staining agents, "physically" adding data for training does not always ensure better performance [18], [19]. To this end, a module named Multi-Domain Information Fusion (MDIF) was proposed to effectively integrate data from different datasets.

Figure 1.4 illustrates our proposed framework. The model is trained jointly with data from multiple domains. A module named Multi-Domain Information Fusion (MDIF) is placed on top of model backbone to integrate information from multiple domains at feature level. The refined features output of MDIF are then fed to a classifier for life cycle prediction.

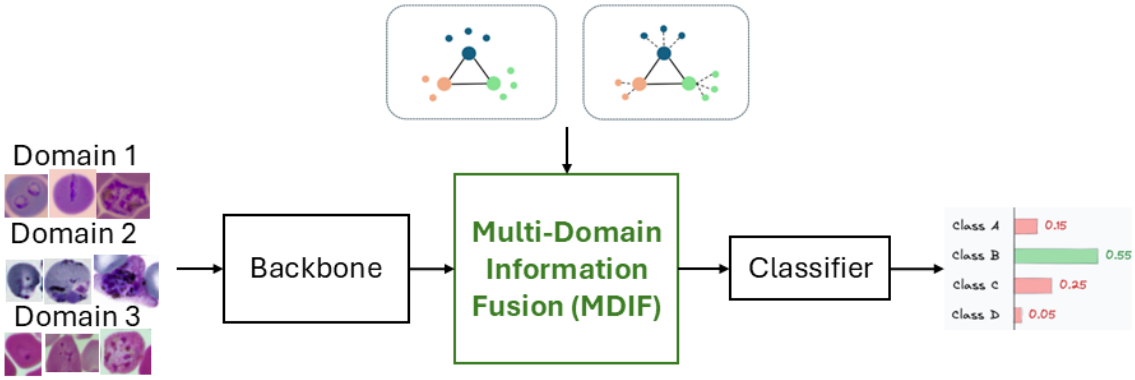


Figure 1.4: Overall classification framework of plasmodium life cycle classification with Multi-Domain Information Fusion.

The contributions of this thesis study can be summarized in two aspects. 1) We

introduced the concept of multi-domain learning into plasmodium life cycle classification field. 2) Multi-Domain Information Fusion was proposed to effectively integrate data from multiple domains.

1.6 Organization of Thesis

The rest of the thesis is structured as follows. Chapter 2 provides an overview of the current state of research in plasmodium-related recognition tasks. Some deep learning related fields, such as Domain Adaptation, are also presented in the chapter as foundation for this thesis study. Based on related knowledge provided in previous chapter, in Chapter 3, life cycle classification framework with details of our proposal to integrate multi-domain data to tackle class imbalance will be explained in detail. We show our implementation details in Chapter 4 to facilitate results reproduction. Effectiveness of our method will be presented in Chapter 5, with quantitative, qualitative results, and ablation studies. Chapter 6 summarizes the thesis and outlines potential future work.

CHAPTER 2. LITERATURE REVIEW

This chapter aims at presenting related works and foundational basis for this thesis study. First, state of research on plasmodium-related tasks is summarized in Section 2.1 and 2.2. In Section 2.3, drawbacks of these existing studies on life cycle development classification are analyzed to reason our motivation to employ data from multiple domains to address data imbalance issue. Then, related fields that is essential for this study, object detection, domain adaptation, multi-domain learning, graph convolutional network, with their recent application in research will be presented in the following Section 2.4.

2.1 Plasmodium Detection

The aim of this section is to briefly present recent studies on plasmodium detection from blood smear images. There are two groups of works, infection detection and life cycle development detection. The former detects and classifies RBCs to infected/uninfected, and in the latter, detected RBCs from blood smears are recognized into life cycle developments. Their detection and classification pipeline can be either in one, two, or three stages. In one-stage pipeline, a model, such as YOLO [20] and R-CNN [21], is responsible for both detection and classification. In two-stage approach, one model detects the location of RBC in blood smear images, and the other recognized their labels. In there-stage, the first model separates cells from blood smear images, the next one classifies cells to infected/uninfected, and the last one recognizes life cycle development.

Object detection: Object detection is a task that localizes and classifies objects in images. R-CNN [21] is the pioneering work on employing CNN for object detection. The proposed framework consists of a region proposal module, which localizes potential regions that may contain objects, and a CNN model to classify proposed regions. The selective search [22] region proposal module in R-CNN is computationally demanding, thus making it a bottleneck for running time. Faster R-CNN [23] tackled the computational demand of region proposal component by integrating it with detection network. The proposed region proposal network is a CNN network, thus could be accelerated by graphics processing unit, enabling computationally lower-cost detection.

Despite attempt to reduce computational time, R-CNN family models were still slow, since each proposed region is processed independently. First version of YOLO [20] handled this issue by unifying detection pipeline into a single network, which is responsible for detecting bounding boxes and class probabilities. Although the

new pipeline reported 10mAP lower than Faster R-CNN with VGG-16, it was 2.5 times faster computational time [20]. YOLOv3 [24] added more detection heads at different feature level to detect multi-scale objects. This strategy improved accuracy on small object, compared with its predecessors , despite being slightly slower.

Infection detection: One of the pioneers on applying deep learning on plasmodium infection detection is Rajaraman et al. [9]. Prior to this work, the studies [25]–[27] used Otsu's threshold method [28] to segment RBC from blood smear images, then texture and morphological features are input to a MLP or SVM to classify development stages. Nanoti et al. [29] used K-mean clustering on Lab color space to segment RBC. Extracted features are dimensionally reduced and input to KNN.

Rajaraman et al. [9] tackled the demand of malaria-specific expertise knowledge on selecting hand-craft features by using CNN to facilitate automatic end-to-end classification of plasmodium. First, RBCs are segmented from blood smear image using multi-scale Laplacian of Gaussian filter. This method gave sensitivity and F1-score of 99.4% and 96.2% correspondingly. Then a CNN model classifies segmented RBC to healthy and infected cells. Among experimented CNN architectures, ResNet50 gave the higher performance with 95.7% accuracy on NIH [30].

In recent years, a few studies utilized object detection deep learning model to detect plasmodium from blood smear images. Abdurahman et al. [10] proposed modified version of YOLOv3 [24] and YOLOv4 [31], named YOLOV3-MOD1, YOLOV3-MOD2, and YOLOV4-MOD. Experiments with other detection models, Faster R-CNN [23] and SSD [32] in particular, were conducted. Among these models, YOLOV4-MOD reported the hightest performance with mAP of 96.32%, 0.5% higher than YOLOv4. Results for YOLOV3-MOD1 and YOLOV3-MOD2 are mAP of 96.14% and 95.46% respectively. Nakasi et al. [11] investigated different detection architectures with specific application on mobile devices. The results revealed that Faster R-CNN is the best model in term of accuracy, while SSD is the best option for deployment with computational constraint.

Life cycle development detection: The next group of works in plasmodium detection is life cycle development detection. Hung et al. [12] utilized Faster R-CNN for single-stage detection and classification of life cycle and achieved accuracy of 59% on BBBC041 [12]. The reason for this poor performance was class imbalance with a large number of healthy RBC, whereas infected classes only have a small portion. The authors addressed this issue by using two-stage approach, Faster R-CNN to detect and classify RBC into healthy and infected, while AlexNet predicts

life cycle from infected RBC. This approach improved performance with roundly 40% increase in accuracy, to 98%. However, it should be noticed that the results were reported with disregarding background, RBC, and difficult cells.

The other works that used two-stage detection and classification similar to Hung et al. [12] includes the studies of Zedda et al. [13], [33]. In their former work [13], the authors employed YOLOv5 and DarkNet53, while in the latter [33], a pipeline using FastSAM [34] to extract proposal regions and ConvNeXt [35] for life cycle classification was proposed. Two studies achieved 96.05% and 80% accuracy on MP-IDB Falciparum [36] and IML Malaria [17], respectively.

Being affected by class imbalance, Arshad et al. [17] proposed three-stage approach. The first stage is a detection model to recognize RBC from blood smear image. The second stage does binary detection to classify RBC to healthy and infected cells, and the last one classifies infected RBC to life cycle. To deal with false positive cases in the first stage, i.e. healthy RBC that are misclassified as infected, healthy RBC was also considered as a class in the last stage. The watershed algorithm [37] was utilized to extract RBC in first stage, and two ResNet50v2 [38] were used in second and third stage. This three-stage pipeline achieved 79.61% accuracy on IML Malaria [17], 5% better than two-stage. Davidson et al. [39] considered life cycle as a numeric value and conducted regression task. With similar three-stage pipeline, they achieved root mean squared error of 0.23 on their own data set by employing R-CNN, ResNet50 and Resnet34 [40] for three corresponding stages.

2.2 Plasmodium Life Cycle Classification

This section presents studies that recognize life cycle of plasmodium. These studies take the cropped RBC from blood smear images as input and classify into life cycle development. It is noticed that detection of RBC from blood smear images is not considered in these works. Their classification pipeline either includes one or two stages. In one-stage, RBCs (including both infected and healthy cells) or infected RBCs are classified into life cycle using one classification model. In two-stage, RBCs (including both infected and healthy cells) are firstly classified as infected/uninfected cells, then life cycle developement is recognized from infected cells.

An example of one-stage classification of infected RBCs is the study of Chaudhry et al. [15]. The authors introduced a light-weight CNN architecture which has less than 0.4 million parameters. Despite of small number of parameters, the proposed model gave better result than large CNN architecture, such as DenseNet121 [41] and ResNet18 [40]. This advantage made the model a good option to implement

on mobile devices, where computational power is limited. With similar classification pipeline, Loddo et al. [42] investigated eleven popular CNN architectures to classify life cycle of plasmodium, among them Densenet201 gave the best result on MP-IDP Falciparum dataset [36] with accuracy of 94.15%.

The study of Li et al. [14] is the one that utilized one-stage life cycle classification. The authors employed GCN and unsupervised learning with additional unlabeled data to handle class imbalance. In the study [16], classification pipeline was divided into two stages to handle dataset imbalance. The first network is responsible to classify RBC into infected and healthy cells, then infected RBC are classified into life cycle by the second network. EfficientNetB7 achieved the best result. An accuracy of 87.95% and F1-score of 87.80% were reported on BBBC041 [12].

Multi-Domain Learning for Life Cycle Classification: The only study that introduced multiple datasets to handle class imbalance on the task of life cycle classification is the work of Li et al. [14]. Beside BBBC041 [12], they used NIH [30] as an other dataset. Since NIH did not have life cycle information, unsupervised learning methods were utilized. Specifically, a GCN module was placed on top of CNN backbone to bridge the domain gap. In knowledge graph, instances from two datasets are connected with each other by clustering strategy. This method achieved 98.3% accuracy on BBBC041.

2.3 Summary of Existing Studies on Plasmodium Detection and Classification

Although there are a few studies that dedicated to detection and classification of plasmodium life cycle, and their reported results are high, [12]–[15], there has not been a standard strategy to evaluate these methods. Each study used their own evaluation strategy, e.g. dataset split, disregarding difficult cells. Additionally, they conducted experiments only on small number of datasets among the ones that have been published. Thus, these conditions make it hard to compare these methods with each other and draw conclusion about state of research on plasmodium life cycle detection and classification task.

With class imbalance as a nature of plasmodium, a few studies have tried to handle this issue by introducing an other network to separate infected and healthy RBC. However, there are still more instances toward healthy RBC than infected class. Furthermore, introducing an other network requires more computational power, which may limit the future employment of model on constrained environment.

An other way to tackle data imbalance is adding more minor classes data. To

the best of our knowledge, the work of Li et al. [14] is the only study that takes advantage of available public datasets to handle this issue. However, this study did not utilize minor class instances effectively. While many instances of minor classes were used for testing, there were only a few instances left for training. This strategy might lead to poor performance on minor classes.

More importantly, the additional data Li et al. used in their study [14], NIH, does not have life cycle development labels. As a consequence, unsupervised technique was required to integrate this dataset to life cycle classification task. While several public datasets on life cycle classification task are available, the integration of them remains unexplored. With the aim of filling the research gap on addressing data imbalance by integrating labeled data from multiple datasets, the following sections will introduce related fields as foundational basis for this thesis study.

2.4 Related Fields for Multi-Domain Learning Plasmodium Life Cycle Development Classification

With the research gap in employing data from multiple domains for plasmodium life cycle development classification, this chapter aims at providing state of research on related fields as a foundation for this study. Applications of multi-domain learning in computer vision are introduced in Section 2.4.1. Since domain adaptation is slightly different from multi-domain learning in objective, while their principal can be employed from one to the other, current studies on domain adaptation is presented in Section 2.4.2. Finally, because of advantage of graph convolutional network in modeling relationship across domains, its principle and a few applications are introduced in Section 2.4.3.

2.4.1 Multi-Domain Learning

With a little different objective compared with domain adaptation, multi-domain learning provides methods to train model with multiple domains. The goal is to improve generalization across all domains, with the principal of taking advantage of available data and knowledge from multiple domains.

Gou et al. [43] introduced Multi-Domain Pose Network for the pose estimation and tracking task. The network consists of a shared backbone to extract features from multiple domains, and separates prediction head for each domain. The backbone and all classification heads were trained jointly with data from all domains. To further improve performance on individual domain, classification head was then fine-tuned on each domain.

Seeing that domains share common knowledge, while each domain has specific ones, Liu et al. [19] proposed a framework consists of private and a shared network.

The private network is specific for each domain, with the aim of learning domain-specific features. The other shared networks are used in all domains to extract domain-invariant features. Adversarial loss is applied on features of the shared network. Feature orthogonal regularization is applied between domain-specific and domain-invariant features to minimize mutual information. Finally, the features compose of domain-specific and domain-invariant features.

2.4.2 Domain Adaptation

Domain adaptation is a subfield of machine learning that deals with domain shift between source and target data. Model is often trained with one (or more) source and one target data, with the objective of improving performance on target domain. Usually, domain adaptation techniques are helpful when source data is abundant (e.g. synthetic data) and target data is scarce or unlabeled (e.g. real world data). Although the domains have similarity in nature, they come from different distributions, which pose distribution shift problem.

Maximum mean discrepancy (MMD) is a popular and simple technique to align features of different domains. MMD is the measurement of distance between source and target features. In addition to classification loss, model is trained jointly with MMD loss, thus domain gap in feature space is reduced. Few examples of studies that employed MMD technique are the studies [44]–[47].

Ganin and Lempitsky [48] introduced a method to align domain distribution at feature level with the idea of utilizing adversarial training. The authors added a domain classifier on top of backbone, dedicated to distinguish which domain the data comes from. Gradient from domain classifier is reversed when being back-propagated to the backbone with the purpose of fooling the domain classifier. This method results in a domain-invariant feature, thus reduces the domain gap in feature space.

To reduce domain gap, input-level alignment techniques are methods to make source data similar to target data. Since source and target data have different distributions, one way to make them similar is to normalize data to a unified distribution. This technique is called domain-specific normalization. Based on this idea, Chang et al. [49] introduced Domain-Specific Batch Normalization (DSBN) module for unsupervised domain adaptation.

Bai et al. [50] approached domain adaptation for person re-identification task from two aspects, domain-specific view and domain-fusion view. The authors improved DBSN [49] by a rectified version, named Rectified Domain-Specific Batch Normalization (RDSBN). From domain-fusion view, they developed a GCN mod-

ule to fuse information at feature level across domains.

Since domain adaption works with multiple domains, it has many aspects in common with multi-domain learning. With the aim of reducing domain gap and learning domain-invariant features, domain adaptation techniques have the potential to apply to multi-domain learning. The proposed methodology in this thesis study was inspired by the study [50]. We adapted Multi-Domain Information Fusion to the multi-domain learning for plasmodium life cycle classification task.

2.4.3 Graph Convolutional Network

GCN is a type of neural network that operates on graph-structured data. The idea of GCN was first proposed in the study [51]. The work of Kipf et al. [52] is regarded as the first work that utilized GCN for graph-based classification. Similar to CNN, node features are aggregated among graph by their neighbors feature. The only difference is that in GCN neighbors are defined by graph topology, instead of spatial structure in CNN. This character makes GCN suitable for the graph-structured problems, such as social network, traffic network, energy grid.

In computer vision tasks, GCN was widely applied to human pose estimation [53]–[55], skeleton action recognition [10], [56]–[58]. Since the physical structure of human skeleton might not capture relationship of joints in action, Li et al. [59] proposed actional link and structural link to capture joint-dependency. With similar reasoning, Chan et al. [58] introduced Channel-wise Topology Refinement module. This module dynamically includes more edges to effectively model dependency of joints in human skeleton.

In recent years, GCN has been applied to domain adaptation by modeling relationship between source and target data. In the study [60], knowledge from source data is transferred to target data to recognize unlabeled instances via graph-based label propagation. Ma et al. [61] bridged source and target domain in there perspectives, domain label, class label, and data structure. Among the perspectives, domain label information is modeled and processed by GCN. Pseudo-edge with GCN was introduced in the study [62] to effectively enhance pseudo label quality in semi-supervised domain adaptation task.

This chapter has presented state of research on plasmodium-related tasks, including infection classification, life cycle development detection, and life cycle development classification. With the research gap in employing multiple datasets from other several domains to handle data imbalance in life cycle classification, we have presented the related fields to apply multi-domain learning to the plasmod-

ium life cycle development classification. Based on the foundational related fields presented in this chapter, the following chapter is dedicated to describe in detail our proposed method.ology

CHAPTER 3. METHODOLOGY

One of the challenges in multi-domain learning is domain shift. This phenomenon is caused by different methods of data acquisition. In particular for plasmodium task, staining agent and microscope are a few examples. The differences in obtaining techniques results in a different characteristics of data, such as color and illumination, and different distributions at feature as shown in Figure 3.1. As a consequence, directly adding training data from multiple dataset might have limited improvement and even poorly affect performance of model [18], [50]. For this reason, the MDIF is proposed with the aim of mitigating domain shift at feature level. The distributions are expected to be well-aligned as illustrated in Figure 3.1.

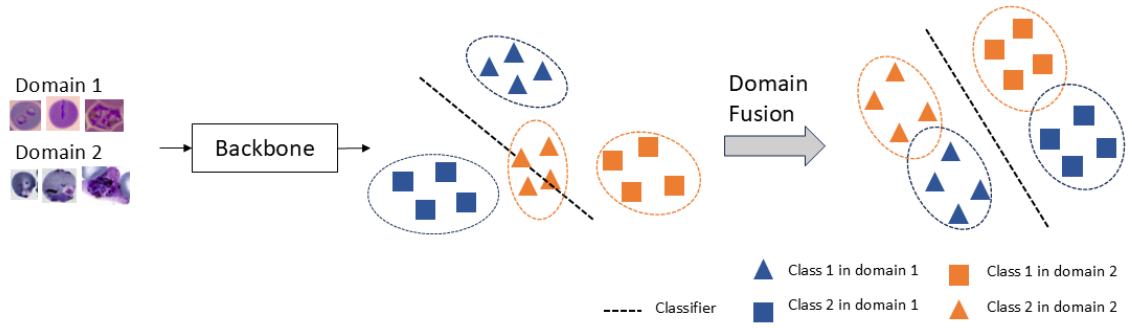


Figure 3.1: Illustration of bridging domain distributions at feature level with MDIF. Colors represent domains.

This chapter is dedicated to describe in detail our proposed solution frame work with Multi-Domain Information Fusion (MDIF). Overview of the solution framework is first demonstrated in Section 3.1. Mathematical notations are also introduced for clear explanation. Section 3.2 describes in detail the integration of information across domains in MDIF. In the two sub-sections, two main components of MDIF, agent node and knowledge graph construction to represent domains relationship are described, respectively.

3.1 Overall Framework

With inspiration from the study [50], which utilized GCN-based module to fuse information at feature level for the unsupervised domain adaptation for person re-identification task, similar strategy with refinement for the classification task was proposed. Figure 3.2 shows the Multi-Domain Information Fusion (MDIF) in the plasmodium life-cycle classification pipeline. First, cell images are input to a backbone to extract features. Then MDIF operates on the features to fuse information across domains. There are two GCN layers in MDIF. The former is responsible for

information globally across agent nodes, while the latter ensures all instance nodes receive global information. The output of MDIF refined features is classified into life cycle by a classifier. In training phase, cross-entropy loss is utilized for gradient back-propagation. Because instance nodes are connected with only agent nodes, samples can be classified independently with each other in inference phase, with agent node values calculated from training phase. For this reason, inference of one sample can be carried out without other samples.

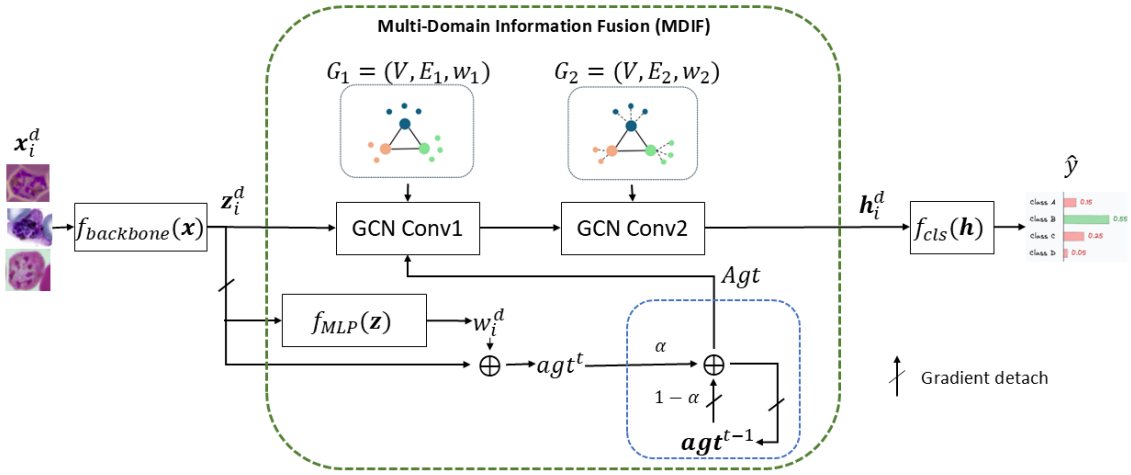


Figure 3.2: Classification framework of plasmodium life cycle classification with Mutli-Domain Information Fusion.

To facilitate a clear explanation, we introduce some notations. The over all solution framework is mathematically described as follows. Let N_c be the number of classes. Suppose that there are D domains, each domain has N_d samples $(\boldsymbol{x}_i^d, y_i^d)$, where $\boldsymbol{x}_i^d$ represents input image and y_i^d represents its class label. Let $X^d = \{\boldsymbol{x}_i^d \mid i \in \{1, ..., N_d\}\}$ be the set of all images of domain d. The total number of samples is calculated as $N = \sum_{d=1}^{D} N_d$. Denote the backbone as a function $f_{backbone}(\boldsymbol{x})$, then the feature of one image is output of the backbone function $\boldsymbol{z}_i^d = f_{backbone}(\boldsymbol{x}_i^d)$. MDIF can be formulated as a function $g(\boldsymbol{z})$. The fused feature $\boldsymbol{h}_i^d = g(\boldsymbol{z}_i^d)$ will be classified as a life cycle development by a classifier $\hat{y}_i^d = f_{cls}(\boldsymbol{h}_i^d)$. Cross-entropy loss function $\mathcal{L}(y_i^d, \hat{y}_i^d)$ updates the parameters by gradient backpropagation.

A subtle component in training phase is the sampling strategy. Since the classes in life cycle development task are extremely imbalanced, employing MDIF with random sampling may pose a negative impact on the performance. In particular, agent node (which will be explained in Section 3.2.1) is calculated mostly by dominant class, making it less representative. For this reason, we employ weighted sampling strategy. Each sample is assigned a weight, which is inversely propor-

tional to its class frequency. The sample selecting probability is then defined by its weight. Basically, the weight of a sample is defined as one over number of samples of its class. Hence, the more frequent a class is, the less likely it is sampled. Due to specific settings of our life cycle classification, sampling weight for each specific class will be described in Section 4.2.

3.2 Multi-Domain Information Fusion

Multi-Domain Information Fusion (MDIF) integrates information across domains at feature level $\boldsymbol{z}_i^d = f_{backbone}(\boldsymbol{x}_i^d)$. The set of all features extracted by backbone of domain d is denoted as $Z^d = \{\boldsymbol{z}_i^d | i \in 1, ..., N_d\}$. We proposed two versions of MDIF, namely Domain-level and Class-level. MDIF Domain-level integrates information across multiple domains, regardless of class information, while the MDIF Class-level refines the fusion process of MDIF Domain-level by taking into account class information. In MDIF Class-level, information of each class in each domain is separated by having specific agent node for each class, with the expectation of better influence on instance features. Domain-level and Class-level are different in agent nodes definition and graph construction. Two core components of MDIF with two versions will be described in the following sub-sections.

3.2.1 Agent Node

A simple approach to model the relationship instances across domains is directly connect one instance with other instances in different domains. In the study [14], Liu et al. connected a target instance with all source instances, whose class center is the closest to the target instance. However, this method might be negatively affected by noises. For this reason, agent node is introduced as a global representation. In MDIF Domain-level, one agent node represents one domain, while in MDIF Class-level, it corresponds to one class of one domain. The number of agent nodes is therefore D in the former, and $D \cdot N_c$ in the latter.

Domain-level agent node:

$$\boldsymbol{agt}^d = \sum_{i=1}^{N_d} \boldsymbol{w_i^d} \cdot \boldsymbol{z}_i^d \tag{3.1}$$

Class-level agent node:

$$\boldsymbol{agt}^{d,c} = \sum_{i=1}^{N_d} (y_i^d == c) \cdot \boldsymbol{w_i^d} \cdot \boldsymbol{z}_i^d \tag{3.2}$$

Seeing that each instance may have different representative degree, an agent

node is the weighted combination of instances. Equations 3.1 and 3.2 show the calculation for Domain-level and Class-level, respectively. Each instance has a specific contribution to the agent node, its contribution weight is first calculated by a learnable linear transformation $\boldsymbol{w}_i^d = f_{MLP}(\boldsymbol{z}_i^d)$. It is noticed that the purpose of the transformation $f_{MLP}(\boldsymbol{z})$ is to define a contribution weight for each specific instance, hence its gradient is not back-propagated to the backbone as shown in Figure 3.2 . The contribution weight is then normalized with all other weight of instances that contribute to the same agent node, in particular, all instances of the same domain in Domain-level and all instances of the same class and same domain in Class-level. Mathematically described, $\boldsymbol{w}_i^d = \boldsymbol{w_i^d} / \sum_{j=1}^{N_d} \boldsymbol{w}_j^d$ in Domain-level, and $\boldsymbol{w}_i^d = \boldsymbol{w_i^d} / \sum_{j=1}^{N_d} (y_j^d == c) \boldsymbol{w}_j^d$ in Class-level.

To further stabilize agent node, exponential moving average is utilized. The values of agent nodes are then updated as Equation 3.3, in which t represents time step and α is the smoothing factor.

$$(\boldsymbol{agt})^t = (1-\alpha)(\boldsymbol{agt})^{t-1} + \alpha(\boldsymbol{agt})^t \tag{3.3}$$

3.2.2 Graph Construction

In this sub-section, the construction of two graphs $G_1 = (V, E_1, w_1)$ and $G_2 = (V, E_2, w_2)$ is discussed. $w_1 : E_1 \to \mathbb{R}$ and $w_2 : E_2 \to \mathbb{R}$ are functions that map an edge to its weight. Let Agt^d denote the set of agent nodes of domain d. Then the set of all agent nodes is $Agt = \bigcup_{d=1}^{D} Agt^d$. In Domain-level, there is only one agent node represents the whole domain, $Agt^d = \{\boldsymbol{agt}^d\}$. While in Class-level, each class in each domain has one agent node, then $Agt^d = \{\boldsymbol{agt}^{d,c} \mid c \in \{1, ..., N_c\}\}$. The set of vertices composes of all instance nodes and agent nodes, $V = (\bigcup_{d=1}^{D} Z^d) \cup (\bigcup_{d=1}^{D} Agt^d)$.

a, Domain-level Graph Construction

Since there are two proposed versions of MDIF, Domain-level graph and Class-level graph, domain-level graph construction is first explained. Figure 3.3 illustrates graph construction of Domain-level. Because each domain has one represented agent node, the total number of vertices is $\mid V \mid = \sum_{d=1}^{D} N_d + D$. In the first layer, only information among agent nodes is exchanged. Thus, all agent nodes are connected with each other by inter-domain edge as shown in Equation 3.4. In second layer, instance nodes receive information from its domain agent node. The instance node is connected to its corresponding agent node. Similar to the first layer, agent nodes are connected with each other in the second layer. Equation 3.5

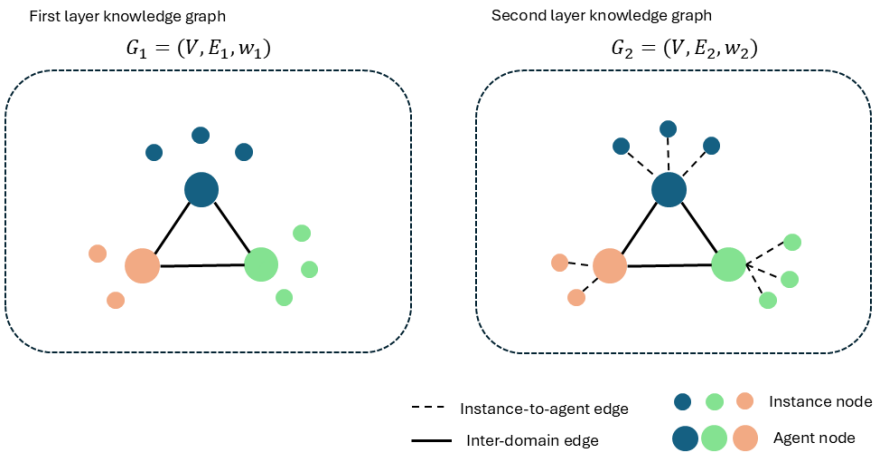


Figure 3.3: Illustration of Domain-level knowledge construction. Color represents domains.

shows the graph construction of the second layer.

First layer knowledge graph construction:

$$w_1(\boldsymbol{u}, \boldsymbol{v}) = \begin{cases} w_1(\boldsymbol{agt}^{d_1}, \boldsymbol{agt}^{d_2}) = 1 \quad \forall d_1, d_2 \in \{1, ..., D\}, d_1 \neq d_2 \\ 0 \quad otherwise \end{cases} \tag{3.4}$$

Second layer knowledge graph construction:

$$w_2(\boldsymbol{u}, \boldsymbol{v}) = \begin{cases} w_2(\boldsymbol{agt}^{d_1}, \boldsymbol{agt}^{d_2}) = 1 \quad \forall d_1, d_2 \in \{1, ..., D\}, d_1 \neq d_2 \\ w_2(\boldsymbol{z}_i^d, \boldsymbol{agt}^d) = 1 \quad \forall d \in \{1, ..., D\}, i \in \{1, ..., N_d\} \\ 0 \quad otherwise \end{cases} \tag{3.5}$$

b, Class-level Graph Construction

The next strategy to integrate information across domain is Class-level. In Class-level, each class in each domain is represented by an agent node. This finally makes the total number of nodes equals to $\mid V \mid = \sum_{d=1}^{D} N_d + D \cdot N_c$. The calculation of agent node is described in Equation 3.2. Figure 3.4 illustrates the construction of knowledge graph for Class-level. Similar to Domain-level, the first layer only integrates information at global level. For that reason, only agent nodes of the same class are connected across all domains. This connection is named inter-domain edge. It is mathematically described in Equation 3.6.

Because in the second layer, global information flows to instance nodes, instance

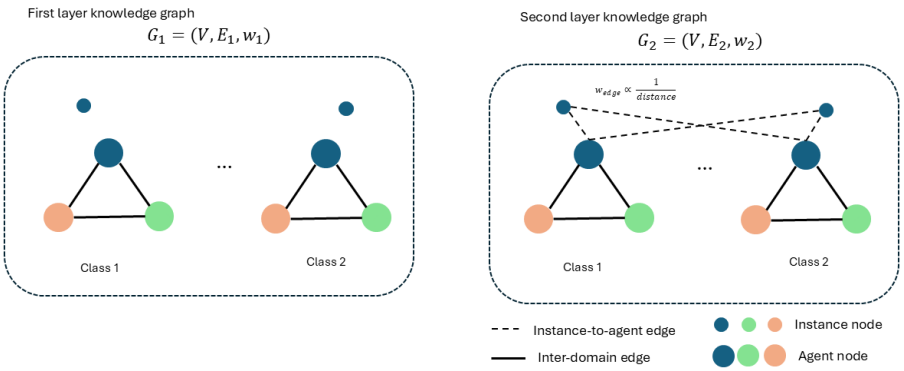


Figure 3.4: Illustration of Class-level knowledge graph construction. Color represents domains. For simplicity, only two classes and two instance nodes are displayed.

nodes are then connected with agent nodes of its domain. The weight of connection to each class is inversely proportional to the Euclidean distance to this class agent node. Intuitively, the closer an instance node is to the class agent node, the more information it receives. Equation 3.7 mathematically explains the calculation of edge weight, in which the weight equals to the inverse distance to class agent node normalized by the sum of inverse distances to all class agent nodes in this domain. The construction of knowledge graph for second layer is defined in Equation 3.8. All inter-domain connections have weight equals to 1, and instance-to-agent connection weight is described previously in Equation 3.7.

First layer knowledge graph construction:

$$w_1(\boldsymbol{u}, \boldsymbol{v}) = \begin{cases} w_1(\boldsymbol{agt}^{d_1,c}, \boldsymbol{agt}^{d_2,c}) = 1 \quad \forall c \in \{1, ..., N_c\}, d_1, d_2 \in \{1, ..., D\}, d_1 \neq d_2 \\ 0 \quad otherwise \end{cases} \tag{3.6}$$

Second layer knowledge graph construction:

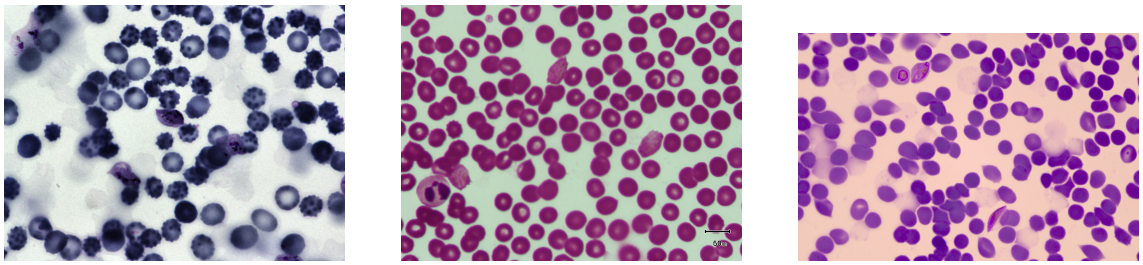
$$w(\boldsymbol{z}_i^d, \boldsymbol{agt}^{d,c}) = \frac{\frac{1}{\|\boldsymbol{z}_i^d - \boldsymbol{agt}^{d,c}\|_2}}{\sum_{c'=1}^{N_c} \frac{1}{\|\boldsymbol{z}_i^d - \boldsymbol{agt}^{d,c'}\|_2}} \tag{3.7}$$

$$w_2(\boldsymbol{u}, \boldsymbol{v}) = \begin{cases} w_1(\boldsymbol{agt}^{d_1,c}, \boldsymbol{agt}^{d_2,c}) = 1 \quad \forall c \in \{1, ..., N_c\}, d_1, d_2 \in \{1, ..., D\}, d_1 \neq d_2 \\ w(\boldsymbol{z}_i^d, \boldsymbol{agt}^{d,c}) \quad \forall d \in \{1, ..., D\}, c \in \{1, ..., N_c\}, i \in \{1, ..., N_d\} \\ 0 \quad otherwise \end{cases} \tag{3.8}$$

This chapter has presented our proposed solution framework for plasmodium life cycle classification with MDIF to integrate domain information at feature-level. Details of the two proposed MDIF, Domain-level and Class-level, has been descried. Agent nodes has been introduced to globally represent domains in Domain-level and classes in each domain in Class-level. Two constructions of knowledge, which define how data flows across domains have been explained. The following chapter will describe the implementation detail of the classification framework with MDIF.

4.1 Dataset

We employed three datasets for our multi-domain learning life cycle classification. Two of them are datasets published by other studies, BBBC041 [12] and IML Malaria [17]. The last is the dataset that was obtained by researchers at Institut de Recherche Biomédicale des Armées (IRBA) and Centre d'Epidémiologie et de Santé Publique des Armées (CESPA). For convenience, we named the data Our Plasmodium. Figure 4.1 and 4.2 show blood smear images and samples of cropped cells of there datasets. It is noticed that our framework works on life cycle classification task, thus blood smear images are cropped into cells in pre-processing phase.



(a) **(b)** **(c)**

Figure 4.1: Blood smear images sampled from there datasets. (a) BBBC041, (b) IML Malaria, (c) Our Plasmodium.

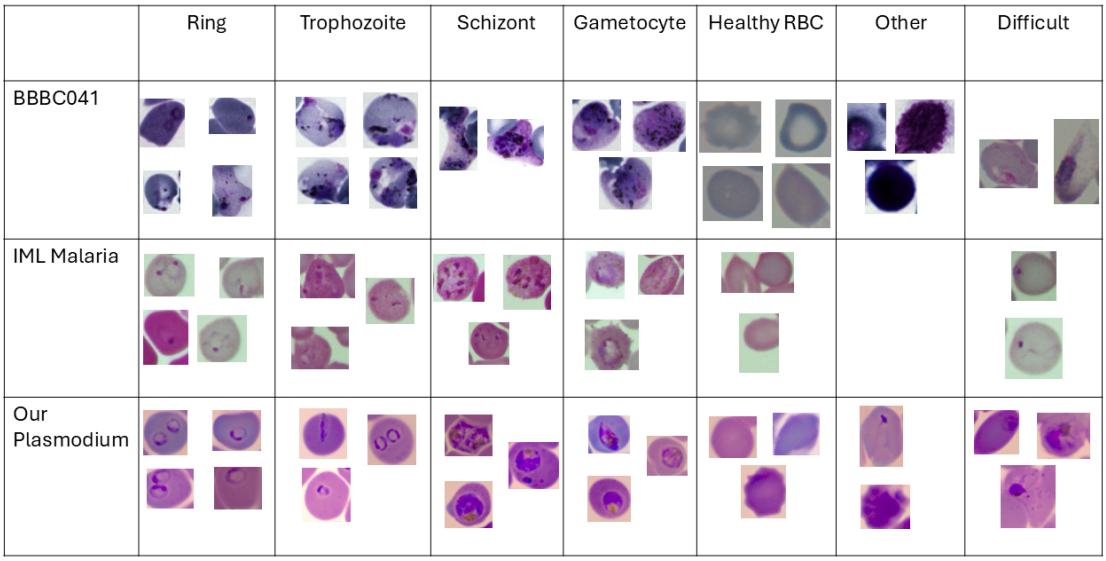


Figure 4.2: Examples of cells in there datasets.

4.1.1 BBBC041

The BBBC041 dataset [12] was collected by three different researchers in different locations, specifically from patients in Brazil and Southeast Asia. The species is P. Vivax. The blood smears were obtained with Giemsa chemical staining agent. There are 1364 images in the dataset, with more than 85,000 annotated cells. Ground truth is the bounding boxes around annotated cells. Annotated classes include RBCs, leukocyte (white blood cell), ring, trophozoite, schizont, gametocyte, and difficult. Cells are marked as difficult if the researcher could not clearly categorize it to one of the classes. BBBC041 dataset is heavily imbalanced toward RBC class, with more than 83,000 samples, making up 95% of the dataset. The dataset was originally divided into train and test set by the authors. We use 10% of the train set for validation. Statistical information for train, test, and validation is included in Table 4.2.

4.1.2 IML Malaria

IML Malaria dataset [17] was obtained from blood samples of patients in the province of Punjab, Pakistan. This area was highly affected by malaria disease, with average of 500,000 infection cases and 50,000 deaths reported every year [63]. The dataset consists of 345 images of P. Vivax genus, captured with XSZ-107 series microscope at 100x objective magnification. Average number of cells in each image is 111. The staining method is thin blood smear with Giemsa staining agent. Annotation includes bounding boxes categorized into ring, trophozoite, schizont, gametocyte, healthy RBC, and difficult. The dataset contains totally nearly 38,000 annotated cells, with 98.6% healthy RBCs. We randomly split the dataset with ratio of 7/2/1 for train, test, and validation, correspondingly. Detailed information for number of instances in these sets is provided in Table 4.2.

4.1.3 Our Plasmodium

In Our Plasmodium, the blood smear images were captured from P. Falciparum infected culture blood samples, i.e. the blood samples are preserved in low temperature condition. There were three projects to create the dataset, each one was responsible by different experts and was prepared in different conditions. For this reason, quality of the image annotation largely depends on which project it comes from. It should be noticed that Our Plasmodium is still in progress of refinement for publishing, hence there may be still some mistakes in the dataset.

In the dataset, infected cells are categorized into six classes, specifically ring, trophozoite, schizont stage 1, schizont stage 2, gametocyte stage 1, gametocyte stage 2-5. Healthy RBC includes three categories, unparasitized, unparasitized with

dead kernel, and unparasitized with artefact. The first class is pure healthy RBC, while the second one used to be parasitized, but the parasites have been died and the cell is no longer affected by parasite. The artefact is healthy RBC, but looks like parasitized cells, due to factors such as staining agent. There are two out of cell classes, merozoite and artefact. The former is plasmodium in merozoite life cycle stage and the latter is caused by technical factors. Beside, cells can be labeled as difficult if the microscopist is unsure about its stages. In total, there are twelve classes in Our Plasmodium data. The dataset then contains roundly 45,000 cells, in which 42,000 are healthy RBC. Split ratio for train, test, and validation is 7/2/1. We provide more detailed information in Table 4.2.

Figure 4.3 shows the number of instances in each class for there dataset. The size of BBCC041 is twice as the size of IML Malaria and Our Plasmodium. One apparent issue in all there datasets is that they are extremely all dominated by healthy RBC class. While there are only 173 samples of gametocyte in Our Plasmodium, healthy RBC has 42,030 samples. The same observation can be derived from BBBC041 and IML Malaria dataset.

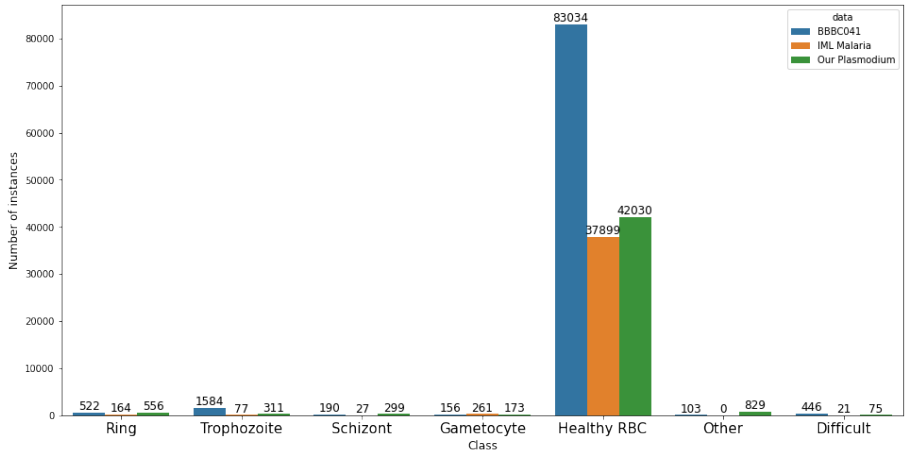


Figure 4.3: Number of instances by each class in three datasets.

Because the annotations of three datasets are slightly different, we define classes for classification framework and the corresponding classes in each data set as shown in Table 4.1. Since subclasses of parasites are usually not needed for medical reason, we group Our Plasmodium into seven classes, particularly ring, trophozoite, schizont, gametocyte, healthy RBC, difficult, and other. Schizont includes schizont stage 1 and stage 2; gametocyte contains gametocyte stage 1 and stage 2-5; unparasitized dead kernel and artefact form other class. Due to their different appearance with pure unparasitized cells, they are not grouped with healthy class. Out of cell classes are not considered in the out task. In BBBC041, other class is

equivalent to leukocyte, while IML Malaria does not have instances of other class. With the purpose of employing classification part to the whole detection and classification pipeline in future work, difficult class is still considered as a class in our classification framework. With the definition of classes for life cycle classification framework, statistical information of train, test, and validation set is provided in Table 4.2.

Classification Class	**Corresponding Class**		
	BBBC041	**IML Malaria**	**Our Plasmodium**
Ring	Ring	Ring	Ring
Trophozoite	Trophozoite	Trophozoite	Trophozoite
Schizont	Schizont	Schizont	Schizont stage 1 Schizont stage 2
Gametocyte	Gametocyte	Gametocyte	Gametocyte stage 1 Gametocyte stage 2-5
Healthy	Healthy	Healthy	Healthy
Other	Leukocyte		Healthy - dead kernel Healthy - artefact
Difficult	Difficult	Difficult	Difficult

Table 4.1: Classification classes and their dataset corresponding.

Classification Class	**BBBC041**			**IML Malaria**			**Our Plasmodium**		
	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test
Ring	317	36	169	121	15	28	380	58	118
Trophozoite	1339	134	111	57	7	13	219	33	59
Schizont	164	15	11	18	4	5	213	27	59
Gametocyte	125	19	12	169	33	59	112	16	45
Healthy	69452	7968	5614	26423	3736	7740	29285	4182	8563
Other	90	13	0	0	0	0	549	97	183
Difficult	389	52	5	17	0	4	48	12	15

Table 4.2: Number of instances by class for train, validation, and test set.

4.2 Experiment Settings

Our Mutli-Domain Information Fusion module is implemented based on MMPretrain [64]. Source code is available at github. ResNet50 [40] pre-trained with ImageNet dataset [65] is adopted as backbone. Graph convolutional network in MDIF module is implemented with PyTorch Geometry [66]. Experiment are carried out on NVIDIA-P100 GPU provided by Kaggle [67].

Cell images are resized to 224×224. In training phase, they are augmented by random crop with random ratio in $(0.6, 1.0)$, and horizontal and vertical flip with

probabilities of 0.25 and 0.25, respectively. Model is trained in 50 epochs. Evaluation on validation set is carried every 5 epochs. Adam [68] with weight decay 10^{-4} is utilized as optimizer. We set initial learning rate to 10^{-4}, and reduce by $1/10$ at 25-th epoch. Smoothing coefficient in exponential moving average formular to update agent nodes value is set to 0.6.

As mentioned in Section 3.1, sampling strategy is important for MDIF. To reduce bias toward healthy RBC class, weighted sampling strategy is utilized. Specifically, sampled weights for healthy RBC, ring, trophozoite, schizont, and gametocyte classes are set to one over number of instances in each class. This strategy ensures equal number of instances of life cycle classes and healthy class are sample every epoch. Since other and difficult classes are considered as medically unimportant, their sampled weights are set as one over number of instances in each class multiplied by 20.

5.1 Evaluation Strategy

To evaluate the effectiveness of the proposed MDIF, we set up the experiments as follows. Two baselines and two approaches are experimented. The first baseline is trained and tested on individual dataset, and the second baseline is trained jointly on all datasets without module MDIF. We call them Individual Training (baseline 1) and Joint Training (baseline 2). In first approach, model with MDIF Domain-level placed on top of backbone is trained with merged dataset. In second approach, MDIF Class-level is employed.

Our experiment design is illustrated in Figure 5.1 and briefly summarized as follows:

- Individual Training: model is trained on individual dataset without MDIF.
- Joint Training: model is trained jointly on all datasets without MDIF.
- MDIF Domain-level: model is trained jointly on all datasets with MDIF Domain-level.
- MDIF Class-level: model is trained jointly on all datasets with MDIF Class-level.

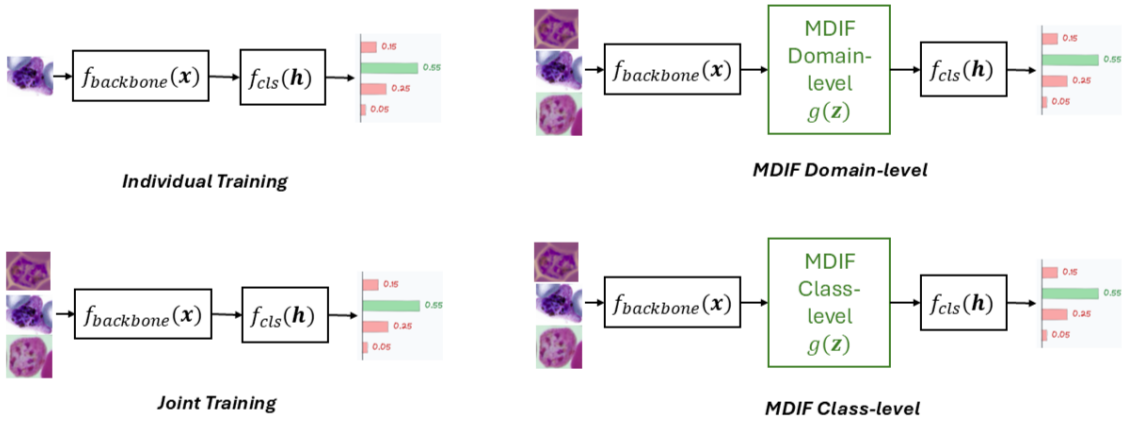


Figure 5.1: Illustration of our experiment design.

In addition to accuracy, three other evaluation metrics are introduced to access the performance on minor classes, namely life cycle micro-averaged recall ($R_{LifeCycle}$), life cycle micro-averaged precision ($P_{lifecycle}$), and life cycle micro-average F1 score. They are calculated over four life cycle classes, i.e. ring, trophozoite, schizont, and gametocyte. Let TP_R, FN_R, FP_R be number of instances that are correctly predicted as ring (true positives), number of ring instances that are

misclassified as other classes (false negatives), and number of instances of other classes that are misclassified as ring (false positives), respectively. Similarly, TP_T, FN_T, FP_T, TP_S, FN_S, FP_S, TP_G, FN_G, and FP_G are true positives, false negatives, and false positives of the corresponding trophozoite, schizont, and gametocyte classes. Evaluation metrics for life cycle classes are calculated as following Equation 5.1, 5.2, and 5.3. Beside quantitative evaluation metrics, qualitative analysis of confusion matrices will be carried out to access the performance of the proposed method.

$$R_{LifeCycle} = \frac{\sum_{c\in\{R,T,S,G\}} TP_c}{\sum_{c\in\{R,T,S,G\}} TP_c + FN_c} \tag{5.1}$$

$$P_{LifeCycle} = \frac{\sum_{c\in\{R,T,S,G\}} TP_c}{\sum_{c\in\{R,T,S,G\}} TP_c + FP_c} \tag{5.2}$$

$$F1_{LifeCycle} = 2 \cdot \frac{R_{LifeCycle} \cdot P_{LifeCycle}}{R_{LifeCycle} + P_{LifeCycle}} \tag{5.3}$$

Results with three model selection strategies are reported, best accuracy, best recall, and last epoch. In best accuracy, models with highest accuracy in validation are selected. Best recall means models are chosen based on micro-averaged recall over four life cycle classes $R_{LifeCycle}$. The last strategy is to test models on the last training epoch. With initial purpose of selecting model with highest performance on Our Plasmodium dataset, in experiments with joint training datasets, i.e. Joint Training, MDIF Domain-level, and MDIF Class-level, selection metric is derived in Our Plasmodium.

5.2 Effectiveness of MDIF on BBBC041

Experiment results on BBBC041 are reported in Table 5.1. This dataset generally benefits from additional data. Joint Training improves performance on all metrics in best accuracy and last epoch model selection with around 0.6% of accuracy and 10% of $F1_{LifeCycle}$. Employing MDIF Domain-level and Class-level further enhances results. MDIF Class-level achieves the highest accuracy 97.20% with best accuracy selection. Regarding performance of life cycle classes, MDIF Domain level is the best model in all there selection modes with highest $F1_{LifeCycle}$ of 58.64%. Although Joint Training reports best life cycle micro-averaged precision $P_{LifeCycle}$ on best accuracy and last epoch selection, it should be noticed that their micro-averages recall $R_{LifeCycle}$ is relatively lower than MDIF, thus their general performance on minor classes is not as good.

Method		Accuracy	Life Cycle Classes		
			$R_{LifeCycle}$	$P_{LifeCycle}$	$F1_{LifeCycle}$
Best Accuracy	Individual Training	96.34	36.30	53.14	43.14
	Joint Training	96.93	44.22	**68.72**	53.81
	MDIF Domain-level	97.04	**53.80**	64.43	**58.64**
	MDIF Class-level	**97.20**	52.81	62.02	57.04
Best Recall	Individual Training	96.34	36.30	53.14	43.14
	Joint Training	90.86	30.36	56.44	39.48
	MDIF Domain-level	**96.50**	53.45	**60.45**	**56.73**
	MDIF Class-level	96.12	**61.93**	40.26	48.80
Last Epoch	Individual Training	96.27	41.91	47.04	44.33
	Joint Training	96.93	44.22	**68.72**	53.81
	MDIF Domain-level	**97.04**	**53.80**	64.43	**58.64**
	MDIF Class-level	96.08	46.86	59.17	52.30

Table 5.1: Results on BBBC041.

We further analyze performances of the proposed MDIF by examining confusion matrices shown in Figure 5.2. They are results of the model selected with best accuracy strategy. In Individual Training, samples are mostly predicted as healthy RBC or trophozoite. There is no schizont and gametocyte predicted. Adding more datasets in Joint Training improves overall performance. MDIF Domain-level and Class-level further enhance recognition among life cycle classes. There are 24 ring and 7 trophozoite more instances that are recognized in MDIF Domain-level compared with Joint Training. It should also be noticed that 9 healthy RBCs that are classified as ring in MDIF Domain-level might not be model's mistakes, but rather annotation errors. These wrong classification cases are presented in Appendix A.

5.3 Effectiveness of MDIF on IML Malaria

Table 5.2 shows experiment results on the IML Malaria dataset. There is a clear improvement of MDIF on life cycle classes evaluation metrics, while accuracy of all methods are slightly similar. In best recall model selection, MDIF Class-level achieves nearly 14% higher than Individual Training and 10% higher than Joint Training in micro-averaged F1 score of four life cycle classes. The two versions of MDIF reports 5% $R_{LifeCycle}$ more than the two baselines in last epoch evaluation.

Confusion matrices on IML Malaria are presented in Figure 5.3. MDIF increases the number of life cycle class instances correctly classified by model. With Class-level, there are 1 ring, 2 trophozoites, and 2 schizonts more that are recognized, compared with Individual Training. Although the number of correctly classified healthy RBC instances reduces, it may also be because of annotation errors. These cases are shown in Section 5.5 and Appendix A.

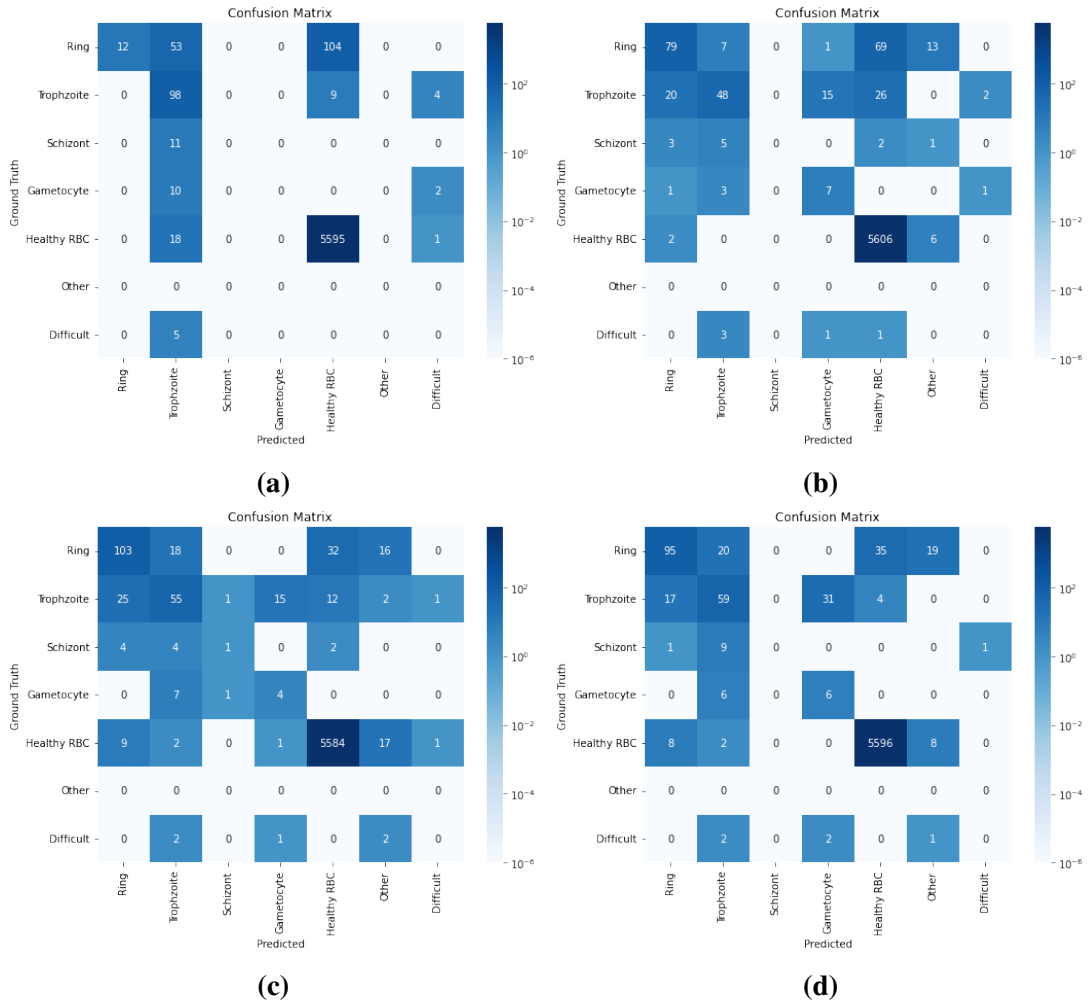


Figure 5.2: Confusion matrices, BBBC041 dataset. Model is selected with best accuracy strategy. (a) Individual Training; (b) Joint Training; (c) MDIF Domain-level; (d) MDIF Class-level

Method		Accuracy	Life Cycle Classes		
			$R_{LifeCycle}$	$P_{LifeCycle}$	$F1_{LifeCycle}$
Best Accuracy	Individual Training	99.44	82.86	69.60	75.65
	Joint Training	99.43	83.81	69.29	75.86
	MDIF Domain-level	**99.45**	**88.57**	**70.99**	**78.81**
	MDIF Class-level	99.36	87.62	68.66	76.99
Best Recall	Individual Training	98.94	83.81	52.38	64.47
	Joint Training	98.93	83.81	58.67	69.02
	MDIF Domain-level	99.32	**85.71**	65.22	74.07
	MDIF Class-level	**99.45**	**85.71**	**72.58**	**78.60**
Last Epoch	Individual Training	99.44	82.86	69.60	75.65
	Joint Training	99.43	83.81	69.29	75.86
	MDIF Domain-level	**99.45**	**88.57**	**70.99**	**78.81**
	MDIF Class-level	99.35	**88.57**	67.88	76.86

Table 5.2: Results on IML Malaria.

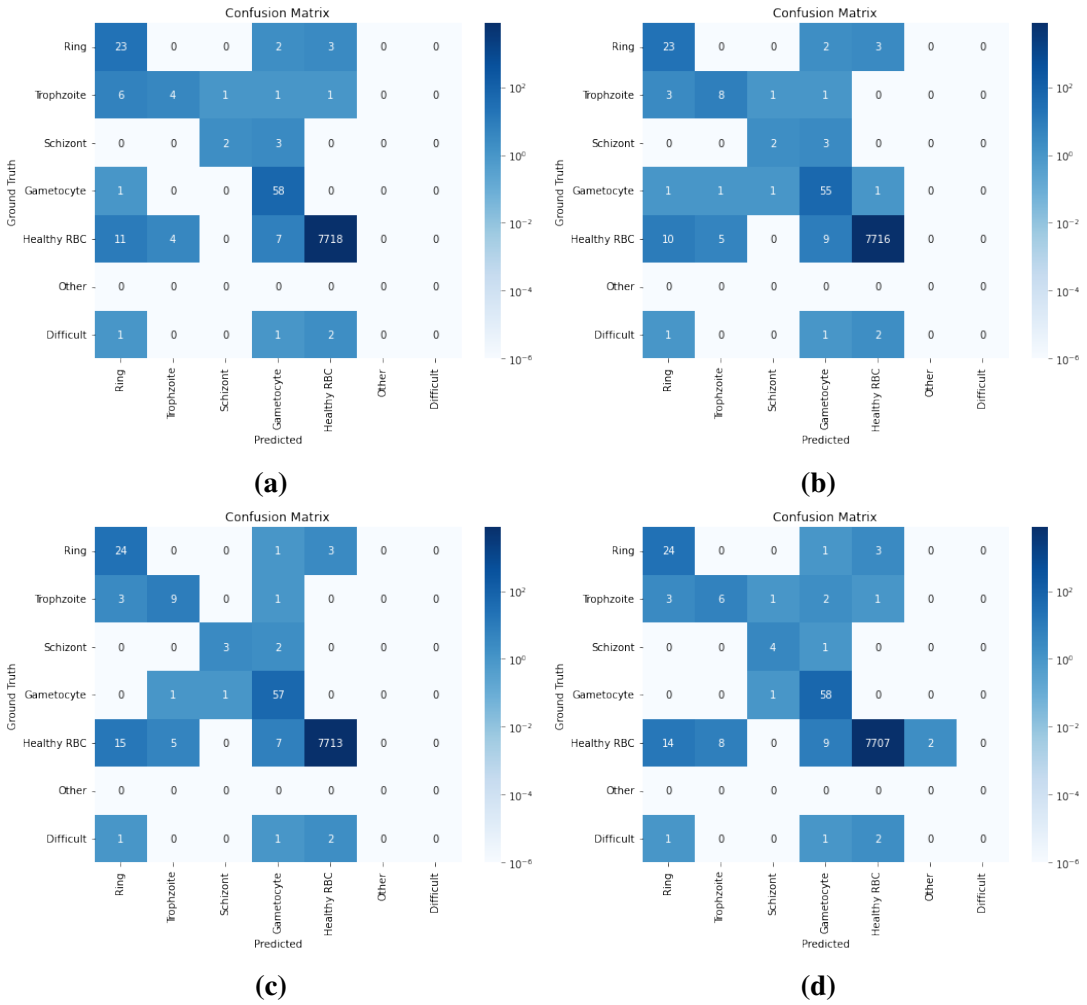


Figure 5.3: Confusion matrices, IML Malaria dataset. Model is selected with best accuracy strategy. (a) Individual Training; (b) Joint Training; (c) MDIF Domain-level; (d) MDIF Class-level

5.4 Effectiveness of MDIF on Our Plasmodium

Result on Our Plasmodium dataset in Table 5.3 shows a slightly better performance of MDIF on life cycle classes, while accuracies of Individual Training and Joint Training are generally higher than MDIF. In best accuracy model selection, the highest micro-averaged F1 score of life cycle classes is 79.39% of MDIF Domain-level, around 0.8% higher than the two baselines, while $F1_{LifeCycle}$ of MDIF Class-level is slightly lower. Similar trend of F1 score on life cycle classes can be observed in last epoch model selection. In best recall selection, MDIF Class-level achieves the highest $F1_{LifeCycle}$ of 80.49%, while Individual Training and MDIF Domain-level shares the best micro-averaged life cycle recall of 83.27%.

We present the confusion matrices on Our Plasmodium dataset with model selected with best accuracy strategy in Figure 5.4. Ring and trophozoite classes are less confused in MDIF Domain-level. While with schizont and gametocyte,

Method		Accuracy	Life Cycle Classes		
			$R_{LifeCycle}$	$P_{LifeCycle}$	$F1_{LifeCycle}$
Best Accuracy	Individual Training	95.39	81.85	75.41	78.50
	Joint Training	**95.66**	81.49	**75.83**	78.56
	MDIF Domain-level	95.31	**83.63**	75.56	**79.39**
	MDIF Class-level	93.14	82.56	74.60	78.38
Best Recall	Individual Training	92.23	**83.27**	77.23	80.14
	Joint Training	78.61	79.72	75.17	77.37
	MDIF Domain-level	**93.44**	**83.27**	74.52	78.66
	MDIF Class-level	92.77	82.56	**78.52**	**80.49**
Last Epoch	Individual Training	95.39	81.85	75.41	78.50
	Joint Training	**95.66**	81.49	**75.83**	78.56
	MDIF Domain-level	95.30	**83.63**	75.56	**79.39**
	MDIF Class-level	91.84	**83.63**	73.90	78.46

Table 5.3: Results on Our Plasmodium.

the recognition ability can be considered similar between Individual Training and MDIF Domain-level. In Class-level, performance of gametocyte is better than the other methods. While its performance in healthy RBC class is the worst, i.e. more healthy RBC instances are incorrectly recognized as other class. However, the evaluation may not reflect quality of the model, because some misclassification cases might be correctly classified. They are shown in Section 5.5 and Appendix A.

5.5 Importance of Domains Fusion

A major question in Multi-Domain Learning is that whether "physically" adding more training data will lead to better performance, since training data might come from different distributions. To answer this question in the field of plasmodium life cycle classification, we examine the performance of Individual Training and Joint Training on three datasets, BBBC041, IML Malaria, and Our Plasmodium. Generally, BBBC and IML Malaria benefit from Joint Training, except in best recall model selection of BBBC041, Joint Training has worse performance than Individual Training. However, in Our Plasmodium, Joint Training reports a significantly lower accuracy and 3% less in $F1_{LifeCycle}$ than Individual Training.

With these observations, we can come to a conclusion that "adding more training datasets does not necessarily ensure a better performance". Therefore, employment of domain fusion for multi-domain learning of plasmodium life cycle classification is necessary. MDIF Domain-level shows a better quantitative performance than Individual Training and Joint Training. MDIF Domain-level F1 score of life cycle class $F1_{LifeCycle}$ is the highest in most cases. Detailed analyses has been provided in the previous sections.

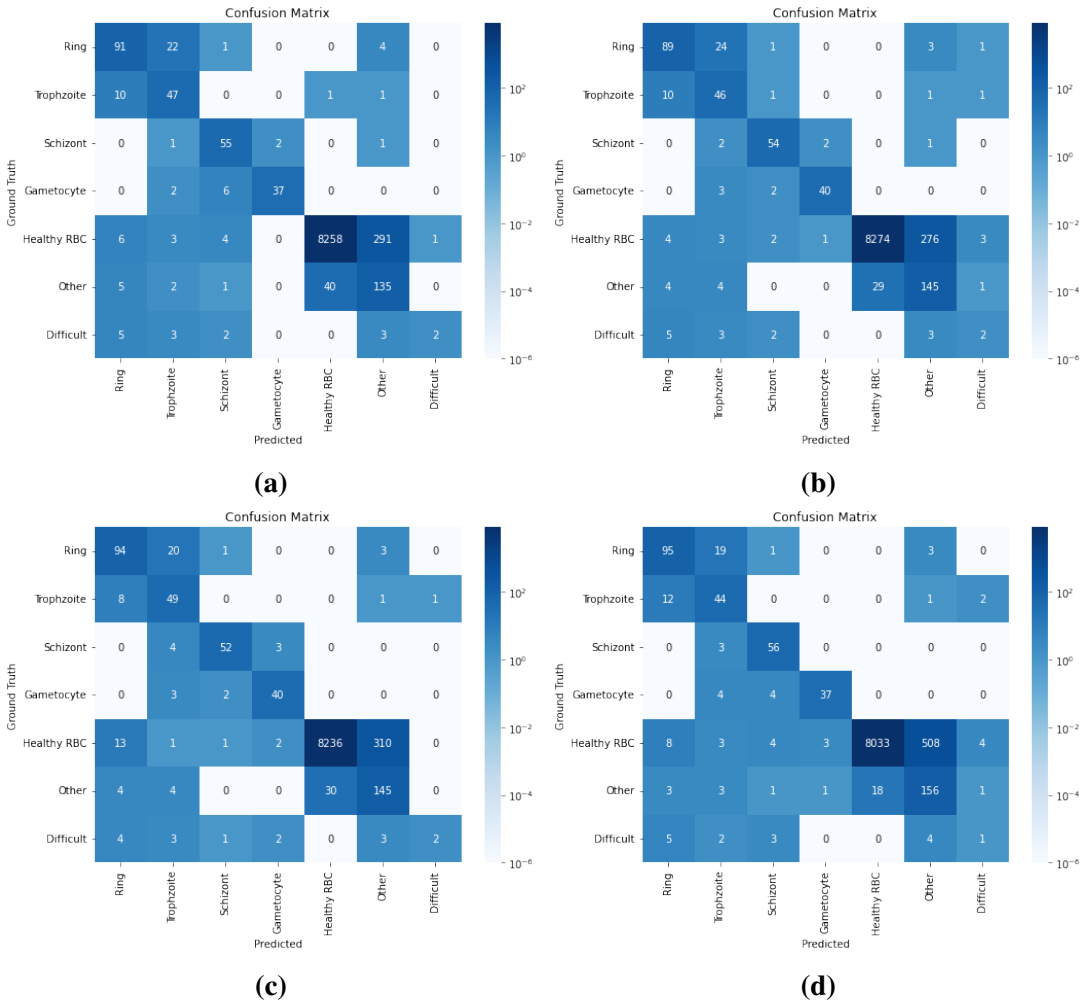


Figure 5.4: Confusion matrices, Our Plasmodium dataset. Model is selected with best accuracy strategy. (a) Individual Training; (b) Joint Training; (c) MDIF Domain-level; (d) MDIF Class-level

MDIF Class-level is designed as a more refined version of MDIF Domain-level with agent node for each class in each domain. We expected the employment of class agent node would reduce the impact of data imbalance to a lower level, i.e. instance nodes receive information from class agent nodes proportionally to their similarity with class agent node. Thus it would enhance performance of the model further, especially among life cycle classes. However, the quantitative results of MDIF Class-level are lower than MDIF Domain-level in most of the cases, as shown in Table 5.1, 5.2, and 5.3. We did further investigation with qualitative analyses and found that the reason might be dataset quality.

Figure 5.5 shows 13 healthy RBC instances that are misclassified by MDIF Domain-level as ring in Our Plasmodium dataset. However, their appearances are abnormal. Some were apparently incorrectly annotated. While the others look healthy, but the bounding boxes contain part of other infected cells. Similarly, Figure 5.6 presents cases that healthy RBC samples are incorrectly recognized as trophozoite

and schizont in IML Malaria data. Beside aforementioned issues, effect of artefact such as color of staining agent can be observed. Some other misclassification cases are provided in Appendix A. The errors in data might not only affect the quantitative evaluation, but also the model in training phase. Since MDIF Class-level operates on specific class, annotation quality might have stronger impact on it than MDIF Domain-level, making its quantitative result poorer.

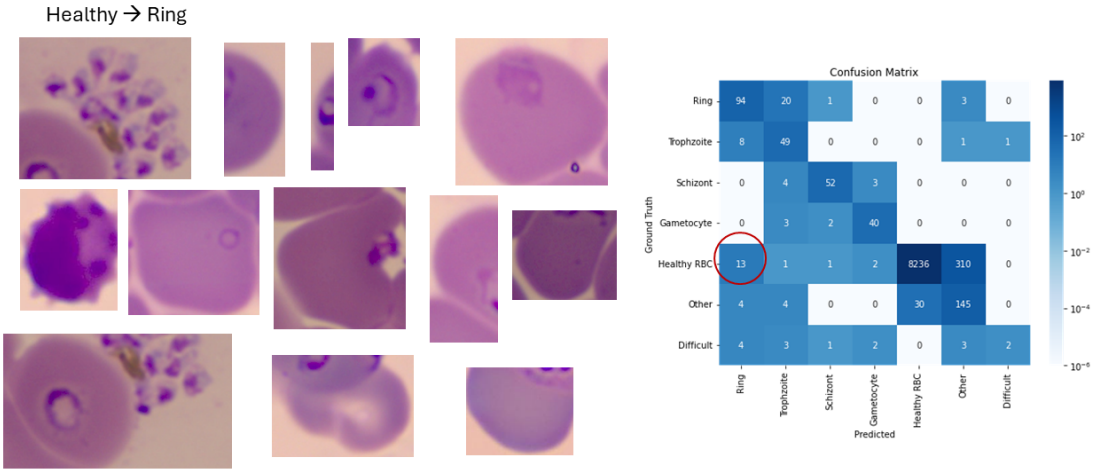


Figure 5.5: Wrong classification cases in Our Plasmodium, MDIF Domain-level. Healthy RBC samples are misclassified ad ring. Annotation mistakes?

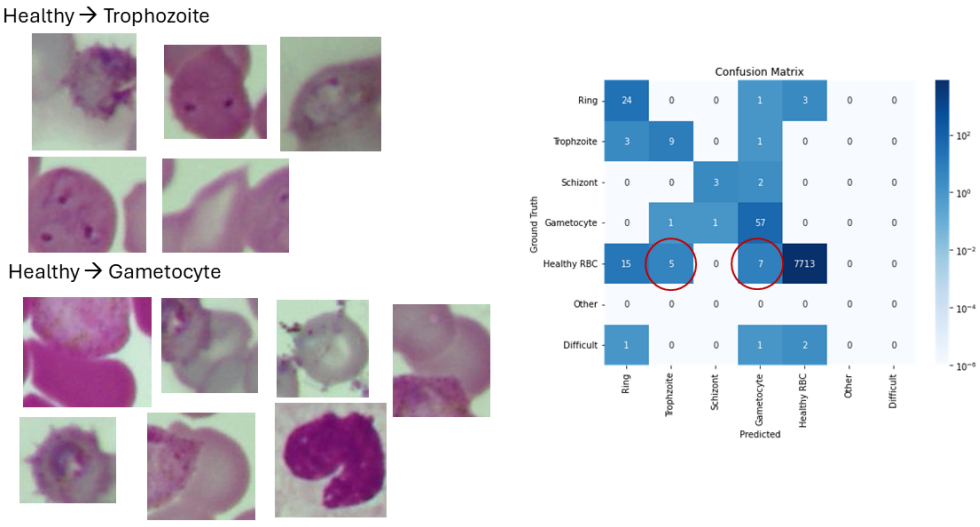


Figure 5.6: Wrong classification cases in IML Malaria, MDIF Domain-level. Healthy RBC samples are misclassified as trophozoite and gametocyte. Annotation mistakes?

5.6 Ablation Studies

5.6.1 Necessity of Weighted Sampling

A subtle component in our framework is weighted sampling. The component is responsible for sampling an equal number of instances each training epoch by over-sampling minor classes and under-sampling major classes. We employ weighted

sampling strategy to our classification framework with the argument that domain agent node in MDIF Domain-level may be biased by major class, i.e. healthy RBC, in random sampling setting. To validate our hypothesis, we run Individual Training, Joint Training, and MDIF Domain-level, with random sampling strategy, i.e. all instances are sampled in each training epoch with equal probability regardless of their class. Model is selected with best accuracy selection strategy. Results are reported in Table 5.4, bold text means worst result. Except BBBC041, MDIF Domain-level trained with random sampling significantly decreases performance on life cycle classes, with 6% and 8% lower in micro-averaged F1 score on life cycle classes in IML Malaria and Our Plasmodium correspondingly, compared with Individual Training. MDIF Domain-level also shows the lowest accuracy on IML Malaria.

Dataset	Method	Accuracy	Life Cycle Classes		
			$R_{LifeCycle}$	$P_{LifeCycle}$	$F1_{LifeCycle}$
BBBC041	Individual Training	**94.85**	**1.00**	27.27	**1.93**
	Joint Training	96.29	33.99	56.86	42.55
	MDIF Domain-level	95.19	8.58	**23.85**	12.62
	MDIF Domain-level*	97.04	53.80	64.43	58.64
IML Malaria	Individual Training	99.44	73.33	75.49	74.39
	Joint Training	99.40	73.33	74.04	73.68
	MDIF Domain-level	**99.31**	**66.67**	**70.00**	**68.29**
	MDIF Domain-level*	99.45	88.57	70.99	78.81
Our Plasmodium	Individual Training	97.71	80.07	77.59	78.81
	Joint Training	**97.16**	79.36	73.60	76.37
	MDIF Domain-level	97.39	**68.68**	**71.48**	**70.05**
	MDIF Domain-level*	95.31	83.63	75.56	79.39

* Trained with weighted sampling

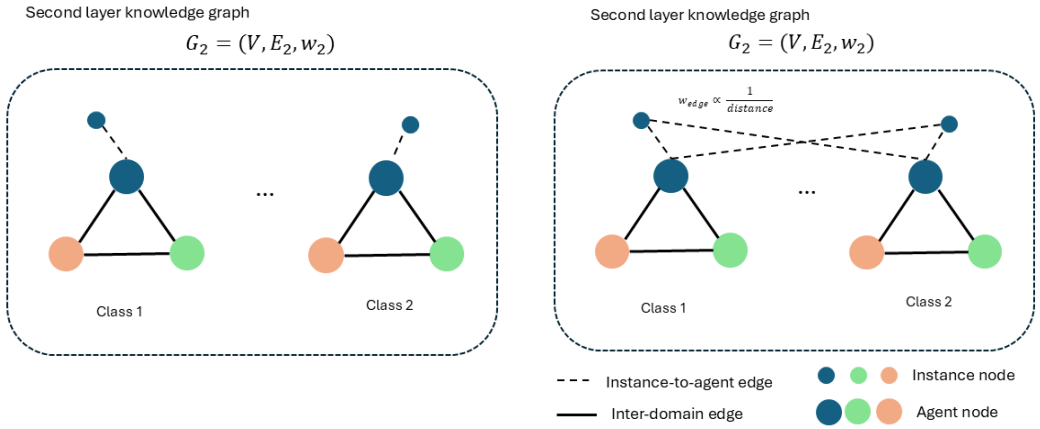
Table 5.4: Result on BBBC041, IML Malaria, and Our Plasmodium with random sampling.

Utilization of weighted sampling solves the issue of bias by major class. In Table 5.4, MDIF Domain-level trained with weighted sampling strategy is significantly better than the one trained with random sampling in almost all evaluation metrics. $F1_{LifeCycle}$ increases 46% in BBBC041, and approximately 10% in IML and Our Plasmodium. Although accuracy on Our Plasmodium is 2% lower, it should be noticed that test set is dominated by healthy RBC class, thus this evaluation metric does not effectively represent the performance of the method.

5.6.2 Necessity of Proper Graph Construction

Graph construction is an important part to model the relationship of domains. It defines how information is integrated across domains. Prior to the current construc-

tion of Class-level knowledge graph, we had experimented with slightly different graph construction. In training phase, instance nodes are connected with only agent node of its class, with the expectation that it would receive only global information within the class. In inference phase, the construction strategy is similar to the current graph construction, instances are connected to all agent nodes of its domain, with edge weight being inversely proportional to distance. Figure 5.7 illustrates the prior construction of Class-level graph in training and inference phase.



(a) **(b)**

Figure 5.7: Illustration of prior Class-level graph construction (a) Training phase; (b) Inference phase.

A major issue with this graph construction is that training and inference are different. During inference, model does not know how to handle new connections, thus performance is significantly reduced. It can be clearly observed in IML Malaria and Our Plasmodium dataset. Table 5.5 shows the result of MDIF Class-level with prior graph construction in comparison with other methods. Results in bold text mean worst result.

In this chapter, effectiveness of MDIF in integrating domain information has been shown in there datasets, BBBC041, IML Malaria, and Our Plasmodium. Quantitative and qualitative analyses have been provided to support our conclusion. Since there are only two studies on life cycle development of plasmodium [14], [16], and the differences in classification pipelines and dataset split with our implementation, comparison was not conducted. Ablation studies have also been carried out. Particularly, the necessity of proper domain integration strategy has been reasoned, and the probable reason that MDIF Class-level did not achieve performance as good as expected has been explained.

Dataset	Method	Accuracy	Life Cycle Classes		
			$R_{LifeCycle}$	$P_{LifeCycle}$	$F1_{LifeCycle}$
BBBC041	Individual Training	96.34	**36.30**	53.14	**43.14**
	Joint Training	96.93	44.22	68.72	53.81
	MDIF Domain-level	97.04	53.80	64.43	58.64
	MDIF Class-level*	**95.63**	42.57	**46.24**	44.33
IML Malaria	Individual Training	99.44	82.86	69.60	75.65
	Joint Training	99.43	83.81	69.29	75.86
	MDIF Domain-level	99.45	88.57	70.99	78.81
	MDIF Class-level*	**99.04**	**80.95**	**59.44**	**68.55**
Our Plasmodium	Individual Training	95.39	81.85	75.41	78.50
	Joint Training	95.66	81.49	75.83	78.56
	MDIF Domain-level	95.31	83.63	75.56	79.39
	MDIF Class-level*	**89.54**	**73.67**	**69.00**	**71.26**

* Old knowledge graph construction version

Table 5.5: Result on BBBC041, IML Malaria, and Our Plasmodium. MDIF Class-level is the version with old knowledge graph construction.

CHAPTER 6. CONCLUSION

6.1 Summary

Malaria disease has a strong impact on not only medical but also socioeconomic aspects to community, especially developing countries. In this work, we have proposed a framework for automatic diagnosis of the disease, which can contribute to the intervention to reduce malaria fatality. Particularly, multi-domain learning methods have been employed for the classification of plasmodium life cycle task to address the data imbalance issue by taking advantages of data from multiple datasets. A module named Multi-Domain Information Fusion is placed on top of feature extractor backbone to integrate information across domains, thus reduce domain shift. We conducted experiments on three datasets, BBBC041, IML Malaria, and Our Plasmodium. Experiment results and ablation studies have shown the effectiveness of the proposed MDIF.

6.2 Future Works

The proposed MDIF module operates on the feature space, thus it is responsible for domains alignment at feature level. On the other hand, methods for domains alignment at data level could be coupled with feature-level alignment with the expectation of further improve performance. To name a few, domain-specific normalization and style transfer are widely applied. In addition, more advanced methods, such as Domain-Specific Batch Normalization [49] with learnable parameters, could provide potential improvement.

Since data imbalance is an severe issue as the nature of the plasmodium life cycle development, methods to tackle this problem are important for the life cycle classification task. In this work, classification of RBC (includes both healhty and infected) is still effected by the domination of the major class healthy RBC. An other way to reduce its impact is separating the classification framework into two stages. The first stage is responsible for classifying healthy and infected RBC, and the second stage will recognize life cycle development from infected RBC input. The integration of multi-domain data is then applied to the second stage with MDIF module placed on top of feature extractor backbone for the same purpose.

REFERENCE

[1] World Health Organization, *Malaria*, https://www.who.int/news-room/fact-sheets/detail/malaria, Accessed: December 28, 2024, 2024.

[2] J. Sachs and P. Malaney, "The economic and social burden of malaria," *Nature*, vol. 415, no. 6872, pp. 680–685, 2002.

[3] K. E. Halliday, S. S. Witek-McManus, C. Opondo, *et al.*, "Impact of school-based malaria case management on school attendance, health and education outcomes: A cluster randomised trial in southern malawi," *BMJ global health*, vol. 5, no. 1, e001666, 2020.

[4] Z. Jan, A. Khan, M. Sajjad, K. Muhammad, S. Rho, and I. Mehmood, "A review on automated diagnosis of malaria parasite in microscopic blood smears images," *Multimedia Tools and Applications*, vol. 77, pp. 9801–9826, 2018.

[5] X.-z. Su and J. Wu, "Zoonotic transmissions and host switches of malaria parasites," *Zoonoses (Burlington, Mass.)*, vol. 1, no. 1, 2021.

[6] R. Tuteja, "Malaria- an overview," *The FEBS journal*, vol. 274, no. 18, pp. 4670–4679, 2007.

[7] S. Sato, "Plasmodium—a brief introduction to the parasites causing human malaria and their basic biology," *Journal of physiological anthropology*, vol. 40, no. 1, p. 1, 2021.

[8] W. H. Organization, *Basic malaria microscopy*. World Health Organization, 2010.

[9] S. Rajaraman, S. K. Antani, M. Poostchi, *et al.*, "Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images," *PeerJ*, vol. 6, e4568, 2018.

[10] F. Abdurahman, K. A. Fante, and M. Aliy, "Malaria parasite detection in thick blood smear microscopic images using modified yolov3 and yolov4 models," *BMC bioinformatics*, vol. 22, pp. 1–17, 2021.

[11] R. Nakasi, E. Mwebaze, A. Zawedde, J. Tusubira, B. Akera, and G. Maiga, "A new approach for microscopic diagnosis of malaria parasites in thick blood smears using pre-trained deep learning models," *SN Applied Sciences*, vol. 2, pp. 1–7, 2020.

[12] J. Hung and A. Carpenter, "Applying faster r-cnn for object detection on malaria images," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 56–61.

[13] L. Zedda, A. Loddo, and C. Di Ruberto, "A deep learning based framework for malaria diagnosis on high variation data set," in *International Conference on Image Analysis and Processing*, Springer, 2022, pp. 358–370.

[14] S. Li, Z. Du, X. Meng, and Y. Zhang, "Multi-stage malaria parasite recognition by deep learning," *GigaScience*, vol. 10, no. 6, giab040, 2021.

[15] H. A. H. Chaudhry, M. S. Farid, A. Fiandrotti, and M. Grangetto, "A lightweight deep learning architecture for malaria parasite-type classification and life cycle stage detection," *Neural Computing and Applications*, vol. 36, no. 31, pp. 19 795–19 805, 2024.

[16] F. Araujo, N. Colares, U. Carvalho, C. F. Costa Filho, and M. G. Costa, "Plasmodium life cycle-stage classification on thick blood smear microscopy images using deep learning: A contribution to malaria diagnosis," in *2023 19th International Symposium on Medical Information Processing and Analysis (SIPAIM)*, IEEE, 2023, pp. 1–4.

[17] Q. A. Arshad, M. Ali, S.-u. Hassan, *et al.*, "A dataset and benchmark for malaria life-cycle classification in thin blood smear images," *Neural Computing and Applications*, vol. 34, no. 6, pp. 4473–4485, 2022.

[18] B. Chen, Z. Yan, K. Li, *et al.*, "Variational attention: Propagating domain-specific knowledge for multi-domain learning in crowd counting," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 065–16 075.

[19] Y. Liu, X. Tian, Y. Li, Z. Xiong, and F. Wu, "Compact feature learning for multi-domain image classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7193–7201.

[20] J Redmon, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[22] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, pp. 154–171, 2013.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[24] J. Redmon, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[25] S. Kanafiah, M. Mashor, Z Mohamed, *et al.*, "An intelligent recognition procedure for trophozoite stages of plasmodium knowlesi malaria," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 1-16, pp. 31–35, 2018.

[26] D. Setyawan, R. Wardoyo, M. E. Wibowo, and E. E. H. Murhandarwati, "Classification of plasmodium falciparum based on textural and morphological features," *Int. J. Electr. Comput. Eng*, vol. 12, no. 5, pp. 5036–5048, 2022.

[27] S. Savkare and S. Narote, "Automatic system for classification of erythrocytes infected with malaria and identification of parasite's life stage," *Procedia Technology*, vol. 6, pp. 405–410, 2012.

[28] S. Savkare and S. Narote, "Automated system for malaria parasite identification," in *2015 international conference on communication, information & computing technology (ICCICT)*, IEEE, 2015, pp. 1–4.

[29] A. Nanoti, S. Jain, C. Gupta, and G. Vyas, "Detection of malaria parasite species and life cycle stages using microscopic images of thin blood smear," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, IEEE, vol. 1, 2016, pp. 1–6.

[30] A. Maqsood, M. S. Farid, M. H. Khan, and M. Grzegorzek, "Deep malaria parasite detection in thin blood smear microscopic images," *Applied Sciences*, vol. 11, no. 5, p. 2284, 2021.

[31] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[32] W. Liu, D. Anguelov, D. Erhan, *et al.*, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, Springer, 2016, pp. 21–37.

[33] L. Zedda, A. Loddo, C. Di Ruberto, *et al.*, "Sammi: Segment anything model for malaria identification," *VISAPP*, vol. 3, pp. 367–374, 2024.

[34] X. Zhao, W. Ding, Y. An, *et al.*, "Fast segment anything," *arXiv preprint arXiv:2306.12156*, 2023.

[35] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 976–11 986.

[36] A. Loddo, C. Di Ruberto, M. Kocher, and G. Prod'Hom, "Mp-idb: The malaria parasite image database for image processing and analysis," in *Processing and Analysis of Biomedical Information: First International SIPAIM Workshop, SaMBa 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Revised Selected Papers 1*, Springer, 2019, pp. 57–65.

[37] J. B. Roerdink and A. Meijster, "The watershed transform: Definitions, algorithms and parallelization strategies," *Fundamenta informaticae*, vol. 41, no. 1-2, pp. 187–228, 2000.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, Springer, 2016, pp. 630–645.

[39] M. S. Davidson, C. Andradi-Brown, S. Yahiya, *et al.*, "Automated detection and staging of malaria parasites from cytological smears using convolutional neural networks," *Biological imaging*, vol. 1, e2, 2021.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

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

[42] A. Loddo, C. Fadda, and C. Di Ruberto, "An empirical evaluation of convolutional networks for malaria diagnosis," *Journal of Imaging*, vol. 8, no. 3, p. 66, 2022.

[43] H. Guo, T. Tang, G. Luo, R. Chen, Y. Lu, and L. Wen, "Multi-domain pose network for multi-person pose estimation and tracking," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.

[44] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2272–2281.

[45] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.

[46] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International conference on machine learning*, PMLR, 2015, pp. 97–105.

[47] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *International conference on machine learning*, PMLR, 2017, pp. 2208–2217.

[48] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*, PMLR, 2015, pp. 1180–1189.

[49] W.-G. Chang, T. You, S. Seo, S. Kwak, and B. Han, "Domain-specific batch normalization for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2019, pp. 7354–7362.

[50] Z. Bai, Z. Wang, J. Wang, D. Hu, and E. Ding, "Unsupervised multi-source domain adaptation for person re-identification," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 914–12 923.

[51] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[52] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[53] H. Ci, C. Wang, X. Ma, and Y. Wang, "Optimizing network structure for 3d human pose estimation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2262–2271.

[54] B. X. Yu, Z. Zhang, Y. Liu, S.-h. Zhong, Y. Liu, and C. W. Chen, "Glagcn: Global-local adaptive graph convolutional network for 3d human pose estimation from monocular video," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8818–8829.

[55] T. Xu and W. Takano, "Graph stacked hourglass networks for 3d human pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 16 105–16 114.

[56] H. Duan, Y. Zhao, K. Chen, D. Lin, and B. Dai, "Revisiting skeleton-based action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 2969–2978.

[57] X. Zhang, C. Xu, and D. Tao, "Context aware graph convolution for skeleton-based action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 14 333–14 342.

[58] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, "Channel-wise topology refinement graph convolution for skeleton-based action recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 13 359–13 368.

[59] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3595–3603.

[60] Z. Ding, S. Li, M. Shao, and Y. Fu, "Graph adaptive knowledge transfer for unsupervised domain adaptation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 37–52.

[61] X. Ma, T. Zhang, and C. Xu, "Gcan: Graph convolutional adversarial network for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8266–8276.

[62] B. H. Ngo, Y. J. Chae, J. E. Kwon, J. H. Park, and S. I. Cho, "Improved knowledge transfer for semi-supervised domain adaptation via trico training strategy," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 214–19 223.

[63] W. Khan, A. U. Rahman, S. Shafiq, H. Ihsan, and K. Khan, "Malaria prevalence in malakand district, the north western region of pakistan," *JPMA*, vol. 69, no. 946, 2019.

[64] M. Contributors, *Openmmlab's pre-training toolbox and benchmark*, `https://github.com/open-mmlab/mmpretrain`, 2023.

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

[66] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[67] Kaggle, *Kaggle: Your machine learning and data science community*, Accessed: December 28, 2024, n.d. [Online]. Available: `https://www.kaggle.com`.

[68] D. P. Kingma, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

APPENDIX

A. SOME MISCLASSIFICATION CASES

Some instances that are classified by MDIF Domain-level. These might not due to model's mistakes, but rather data annotation errors. Figure A.1 and A.2 shows healthy RBC instances that are classified as ring, trophozoite, and gametocyte. Samples of healthy RBC class that are recognized as ring of IML Malaria are presented in Figure A.3. In Figure A.4, instances with healthy RBC labels that are classified as trophozoite, schizont, and gametocyte are shown.

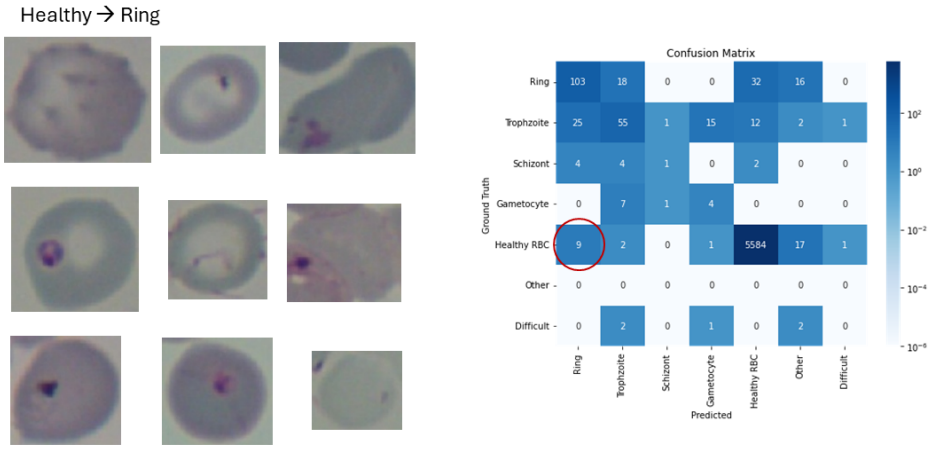


Figure A.1: Wrong classification cases in BBBC041, MDIF Domain-level. Healthy RBC samples are misclassified as ring. Annotation mistakes?

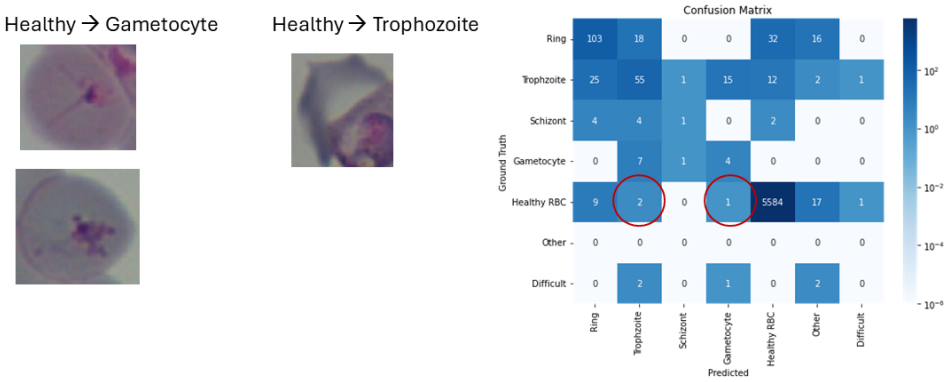


Figure A.2: Wrong classification cases in BBBC041, MDIF Domain-level. Healthy RBC samples are misclassified as trophozoite and gametocyte. Annotation mistakes?

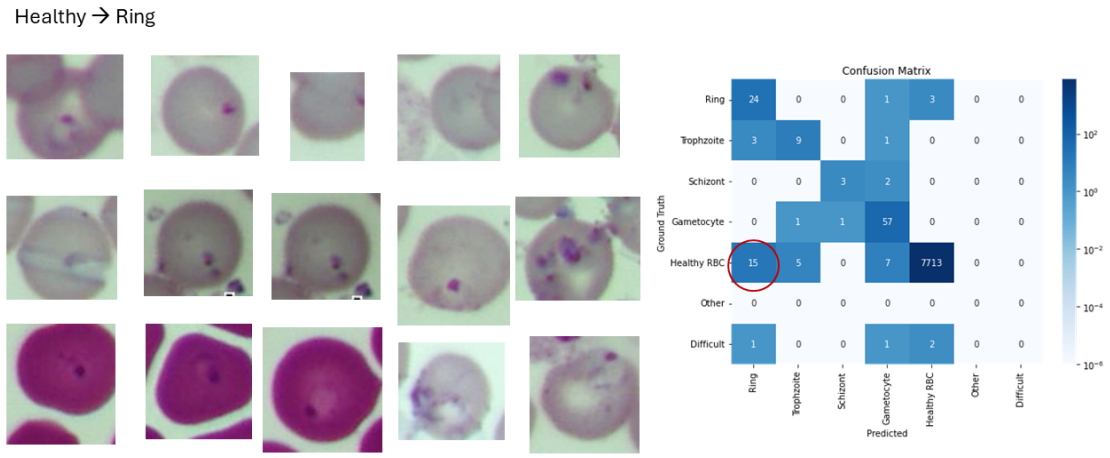


Figure A.3: Wrong classification cases in IML Malaria, MDIF Domain-level. Healthy RBC samples are misclassified as ring. Annotation mistakes?

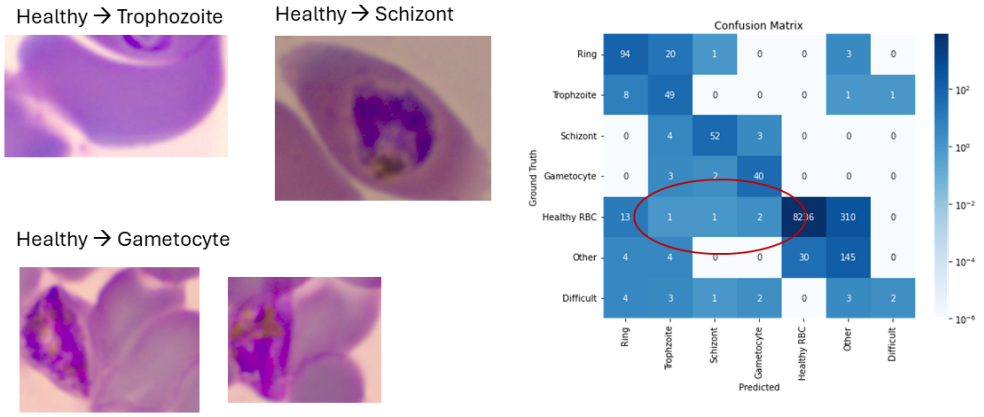


Figure A.4: Wrong classification cases in Our Plasmodium, MDIF Domain-level. Healthy RBC samples are misclassified as trophozoite, schizont, and gametocyte. Annotation mistakes?