Multi-Domain Information Fusion for *Plasmodium* Development Stage Recognition

Quoc Khanh Tran[1][0009−0006−2367−7761], Muriel Visani[2,3][0000−0001−7513−4749], Thierry Urruty[4][0000−0003−1339−1920], Océane Delandre[5,6,7,8], and Thi-Oanh Nguyen[1][0000−0002−6166−2011] ⋆

[1] School of Information and Communications Technology (SoICT), Hanoi University of Science and Technology (HUST), Hanoi, Vietnam
[2] French Military Center for Epidemiology and Public Health (CESPA), Marseille, France
[3] Laboratoire L3i, La Rochelle Université, France
[4] Université de Poitiers, Univ. Limoges, CNRS, XLIM, Poitiers, France
[5] Unité Parasitologie et Entomologie, Département Risques Vectoriels, Institut de Recherche Biomédicale des Armées (IRBA), Marseille, France
[6] Aix Marseille Univ, SSA, APHM, RITMES, Marseille, France
[7] IHU Méditerranée Infection, Marseille, France
[8] Centre National de Référence du Paludisme, Marseille, France

Abstract. Malaria is a disease caused by the *Plasmodium* parasite. Automatic recognition of *Plasmodium* infection and its stages of development in human blood reduces the resources required for manual diagnosis and clinical trials. However, with few annotated samples per development stage, the model's ability to distinguish them may not be satisfactory. When obtaining more data is not feasible, leveraging multiple datasets provides an alternative solution. Furthermore, the diversity and variability across multiple datasets can help in learning a more generalizable model. For these reasons, we use an existing, unsupervised Multi-Domain Information Fusion (MDIF) method to train the model using multiple datasets while handling domain gaps, and propose a supervised variant of MDIF (MDIF-C). MDIF consists of two graph convolutional network (GCN) layers to refine the feature representation with information across domains. While the existing MDIF (MDIF-D) fuses the information only at the domain level, our proposed MDIF-C leverages both domain and class-level fusion, thus making the information fusion more beneficial and mitigating the impact of data imbalance. Experiments are conducted on three datasets of two different *Plasmodium* species, namely BBBC041-v1, IML Malaria, and RecoPlasmodiumV1, with an end-to-end pipeline from blood smear images. The pipeline includes red blood cell detection and development stage classification. Experimental results show the superiority of the entire pipeline using our MDIF-C for classification, over both training on multiple datasets without fusion and MDIF-D, even in the presence of different *Plasmodium* species and noise.

⋆ Corresponding author: `oanh.nguyenthi@hust.edu.vn`

Keywords: Malaria disease · *Plasmodium* development stage recognition · Supervised classification · Data scarcity · Multi-domain learning · Domain information fusion

1 Introduction

Malaria disease has strong impacts on public health, societies, and economics [1]–[3]. *Plasmodium* parasites are transmitted to humans through the bite of infected mosquitoes and undergo several development stages in the human liver and blood. Information about development stages in human blood is crucial for the assessment of newly developed medications within the framework of clinical trials. To reduce the time and resources required for manual microscopic examination, we propose a deep learning method to automatically recognize *Plasmodium* development stages from blood smear images. Red blood cells (RBCs) are detected and then classified into the five classes being necessary to estimate per-stage parasitaemia (percentage of infected RBCs per development stage), namely "Healthy", "Ring", "Trophozoite", "Schizont", and "Gametocyte". For examples of these five classes from different datasets (showcasing different species), please refer to Fig. 2.

However, this classification problem is difficult, due to extreme data imbalance (the vast majority of RBCs being healthy), low inter-class variability among parasitized classes, large intra-class variability (some development stages consisting of visually dissimilar sub-stages), and labeled data scarcity (especially for some rarely observed stages, *e.g.* gametocytes). To improve the model's ability to distinguish development stages despite data scarcity, we utilize *multi-domain learning* to make full use of the rich and diverse data from multiple domains[1].

A challenge in multi-domain learning is domain gaps. Different domains usually have different characteristics, *e.g.* in our case: staining agent, magnification scale, or even *Plasmodium* species. Thus, simply training a single classifier from multiple domains may not improve performance. It may even worsen it, as such a classifier might sacrifice performance on the domains with the lowest number of samples [5]. In this paper, we propose to use Multi-Domain Information Fusion (MDIF) [6], designated here MDIF-D, to mitigate the impact of the domain gap.

Even when source datasets are fully labeled, the existing MDIF method (MDIF-D), that performs unsupervised domain adaptation, only transfers information at the domain level, leaving the rich class information unexploited. Moreover, in the presence of class imbalance, such a model may be biased toward the majority classes. For these reasons, we introduce a new, supervised version of MDIF, namely class-level MDIF (MDIF-C), to bridge the domain gap more effectively, and lower the impact of class imbalance.

Our contributions are listed as follows. 1) A two-step pipeline is proposed, for first detecting / localizing RBCs and then classifying them into "Healthy" or one out of four *Plasmodium* development stages. 2) In the classification step,

[1] A domain commonly refers to a dataset where samples follow the same underlying data distribution [4].

we use the existing (unsupervised) MDIF-D [5] to enhance model generalization by taking advantage of data diversity from multiple domains while mitigating the domain gap. 3) We introduce MDIF-C, a new method for supervised MDIF-based domain adaptation, which uses two levels of fusion: domain and class. 4) Experiments results on three datasets show the effectiveness of our end-to-end pipeline using MDIF-C, even in the presence of domains with different *Plasmodium* species and data noise.

2 Related Work

***Plasmodium* Development Stage Recognition.** This task can be handled using different pipelines. Although some methods carry it out in a single step [7]–[9], others use two steps [7], [10]–[12] or even three steps [10], [13]. Two-step approaches perform either infected RBC detection followed by classification, or, like ours, perform RBC detection followed by classification, including the "Healthy" class. Three-step approaches usually consist of RBC detection, infected (or uninfected) RBC classification, and (for infected samples) development stage classification. Several works simply focus on the classification stage, taking as input pre-cropped RBC images available in the ground truth. The best performing method on the public dataset BBBC041-v1 [14], among the ones presented in Hung *et al.* [7] is 2-step, and uses Faster R-CNN [15] for uninfected cell detection and (for negative samples) AlexNet [16] for development stage classification. Zedda *et al.*'s [12] results were reported on the public dataset IML-Malaria [10] using YOLOv8m [17] to detect infected RBCs, then ViT-Large [18] to classify infected RBCs into development stages. Using a similar 2-step approach, these authors develop a YOLOv8-based architecture for infected RBCs detection, followed by a ViT-based classifier [9]. In addition to the whole pipeline, there are a few studies that only handle the classification stage, either for infected RBC development stage classification only [19], or for both development stage and healthy RBC classification [20], [21].

It is important to note that the use of diverse pipelines (as explained above), combined with a lack of consistency in evaluation strategies across current studies, hinders fair comparisons (see Section 5.2). Some studies use private datasets [8], [13], supplement data with self-captured images [20], disregard a few classes from evaluation [7], only carry out the classification stage [19]–[21], or report results for different parts of the pipeline separately [10].

Data Imbalance. Several studies tackling data imbalance in medical imaging include few-shot learning MetaMed [22], one-shot learning [23], and Balanced-MixUp [24]. To the best of our knowledge, Li *et al.* [20] are the only ones addressing this problem in the *Plasmodium* infection and development stages classification by using an additional dataset. Although this method shows effectiveness, their additional data are unlabeled, leaving rich-labeled data unexploited.

Multi-Domain Learning. Multi-domain learning employs data from multiple domains to enhance generalization. Gou *et al.* [25] introduce a shared backbone and a private prediction head for each domain for pose estimation

and tracking. Liu *et al.* [26] use two networks to extract domain-invariant and domain-specific features. Given the lack of training samples for some *Plasmodium* development stages, multi-domain learning provides a potential solution.

Domain Adaptation. In the presence of scarce labeled data, domain adaptation aims to take advantage of multiple datasets while bridging the gap between their respective domains. Maximum mean discrepancy [27]–[30], adversarial training [31], domain normalization [6], [32] and Multi-Domain Information Fusion (MDIF) [6] (designated here MDIF-D) are applied in recent studies on domain adaptation. By taking advantage of a graph structure, MDIF-D can capture the relationship among data samples, providing an advantage over other methods that focus mainly on distribution alignment. But, unsupervised domain adaptation methods (including MDIF-D) leave the rich class information – when available – unexploited. Additionally, most domain adaptation methods are affected by class imbalance. To make use of the whole information available while mitigating the impact of the extreme class imbalance that we have to face in our application, we introduce a new, supervised MDIF-based domain adaptation method: MDIF-C.

The following section details the pipeline we propose to handle *Plasmodium* stage recognition from blood smears, including our proposed MDIF-C method.

3 Methodology

3.1 Overview

To recognize *Plasmodium* development stages from blood smear images, we apply the two-step pipeline shown in Fig. 1, which is original compared to the ones mostly found in the literature [7], [9], [12]. Indeed, by relying on a detection step that provides only infected RBCs, the methods in [7], [9], [12] do not provide the unparasitized RBC counts, and thus are not enough to estimate per-stage parasitaemia, which is a crucial statistics for healthcare professionals. In our proposed pipeline on the other hand, the first step detects all RBCs, while the second classifies them into the five classes that are necessary to estimate per-stage parasitaemia (see Fig. 2): "Healthy", "Ring", "Trophozoite", "Schizont" and "Gametocyte".

In general, the background in blood smear images is not complex, making the detection phase quite straightforward, while classification is more challenging due to the difficulties mentioned in the second paragraph of Section 1. Thus, for the classification step, we propose to leverage data from multiple domains and use MDIF to refine features by incorporating information across domains, comparing our supervised MDIF-C to the existing, unsupervised, MDIF-D method [6].

Data X from all domains is first embedded into a feature space using a feature extractor $f(.)$, resulting in a set of feature representations Z. These features are then refined with global information by MDIF, before being passed to a classifier $g(.)$. The core idea is to facilitate information exchange between groups of data using a GCN. Specifically, the data is divided into *groups of similar instances* G,

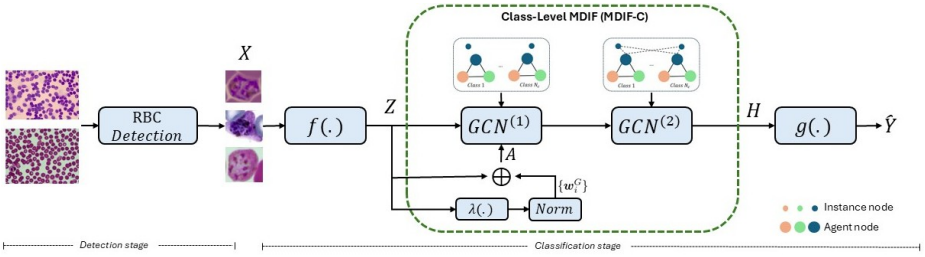

Fig. 1: *Plasmodium* development stage recognition pipeline, including our proposed MDIF-C. Colors represent domains.

each represented by an *agent node*. These agent nodes communicate with each other to share global information and subsequently propagate this information to their associated instance nodes through two GCN layers. Depending on how finely the data is divided into groups G, we either use the existing Domain-Level MDIF (MDIF-D) [6] or our proposed Class-Level MDIF (MDIF-C).

3.2 Agent Node

Suppose we have D domains, where each domain $\mathcal{D}_d = \{(\boldsymbol{x}_i^d, y_i^d) | i \in [1, N_D]\}, d \in [1, D]$ consists of N_d instances (RBCs) $\boldsymbol{x}_i^d$ and their corresponding labels $y_i^d \in [1, C]$. Each *agent node* $\boldsymbol{a}^G$ is determined based on the features of instances in G, $\{\boldsymbol{z}_i = f(\boldsymbol{x}_i), \forall \boldsymbol{x}_i \in G\}$. Since each instance node contributes differently to the representation, a contribution coefficient $\boldsymbol{w}_i^G$ is assigned to each instance $\boldsymbol{x}_i$ in G for the calculation of its corresponding agent node $\boldsymbol{a}^G$: $\boldsymbol{w}_i^G = \lambda(\boldsymbol{z}_i) / \sum_{\boldsymbol{x}_j \in G} \lambda(f(\boldsymbol{x}_j))$, where λ is a fully connected layer. The calculation of an agent node is described in Eq. 1 (where $\cdot$ denotes element-wise multiplication):

$$\boldsymbol{a}^G = \sum_{x_i \in G} \boldsymbol{w}_i^G \cdot f(\boldsymbol{x}_i) \tag{1}$$

A group G consists of all instances from a domain $\mathcal{D}_d$ in MDIF-D, and all instances of a class c within a domain $\mathcal{D}_d$ in MDIF-C. The agent nodes of MDIF-D and MDIF-C are respectively denoted as $\boldsymbol{a}^d$ and $\boldsymbol{a}^{d,c}$. We denote A as the set of agent nodes across all domains, A^d as the set of agent nodes in domain $\mathcal{D}_d$: $A^d = \{\boldsymbol{a}^d\}$ in MDIF-D and $A^d = \{\boldsymbol{a}^{d,c} \mid c \in [1, C]\}$ in MDIF-C.

3.3 Graph Construction

The graph topology defines how the information is propagated and integrated across nodes. There are two types of nodes in the graphs: *agent nodes* (described in Section 3.2), and *instance nodes*, which represent the feature embeddings of individual instances. Let us describe the construction of the MDIF-C graph. Denote $\mathcal{G} = (V, E)$ as the undirected graph where V is the set of all vertices, $V = A \cup Z$, and E defines their connections. The first layer focuses solely on

exchanging global information between groups; thus, only agent-to-agent connections are considered, as shown in Eq. 2. In contrast, the second layer enables each agent node to distribute information to its associated instance nodes by incorporating both agent-to-agent and agent-to-instance connections (Eq. 3).

$$\text{Layer 1: } E = \{(u, v) | u, v \in A^c; u \neq v; c \in [1, C]\} \tag{2}$$

$$\text{Layer 2: } E = \begin{cases} \{(u, v) | u, v \in A^c; u \neq v; c \in [1, C]\} \bigcup \\ \{(u, v) | u \in A^d; v \in Z^d; d \in [1, D]\} \end{cases} \tag{3}$$

where $Z^d = \{\boldsymbol{z}_i^d \mid i \in [1, N_d]\}$, $A^c = \{\boldsymbol{a}^{d,c} \mid d \in [1, D]\}$.

In MDIF-D, because each domain corresponds to a single agent node, the index c is omitted, $A^c \equiv A \equiv \{\boldsymbol{a}^d \mid d \in [1, D]\}$. In the second layer of MDIF-C, agent-to-instance connection weights are determined by feature similarity instead of being set to 1 as in MDIF-D: $s(\boldsymbol{z}_i^d, \boldsymbol{a}^{d,c}) = \frac{1/\|\boldsymbol{z}_i^d - \boldsymbol{a}^{d,c}\|_2}{\sum_{c'=1}^{C} 1/\|\boldsymbol{z}_i^d - \boldsymbol{a}^{d,c'}\|_2}$. Intuitively, the more similar an instance node is to an agent node, the higher the weight is.

4 Experiment Settings

Datasets. We use three datasets, BBBC041-v1[2] [14] (images contributed by Hung *et al.* [7]), IML Malaria[3] [10], and RecoPlasmodiumV1 for the main experiments. The latter has been developed by us, is currently under publication and will be made available publicly by January 2026. It is important to note that, while BBBC041-v1 and IML Malaria both contain images of *Plasmodium vivax*, RecoPlasmodiumV1 contains *Plasmodium falciparum*, whose stages of development are quite visually dissimilar to *vivax*.

Since we perform the complete detection and classification pipeline and there is no predefined train/test split for the IML Malaria dataset, we randomly partitioned the data at the image level using a 70/20/10 split for training, testing, and validation. The RecoPlasmodiumV1 dataset was partitioned similarly. At the RBC level, the data is approximately stratified following the five classes (see Table 1). For BBBC041-v1 on the other hand, we use the original train and test sets, and randomly pick 10% of the train set for validation. Samples from these datasets and dataset statistics are respectively shown in Fig. 2 and Table 1. To make our research reproducible while complying with the MMM double-blind reviewing process, we will provide a link to these splits in this paper's camera-ready version.

Implementation Details We train our detection and classification models separately. For detection, YOLOv11n [33] pretrained on COCO [34] is fine-tuned with the SGD optimizer [35] and with a learning rate 10^{-2} over 100 epochs. For classification, feature extractor ResNet50 [36] pretrained on ImageNet [37] and MDIF implemented with PyTorch Geometric [38] are trained with the Adam

[2] Available at https://bbbc.broadinstitute.org/BBBC041
[3] Available at https://www.kaggle.com/datasets/qaziammararshad/iml-malaria1

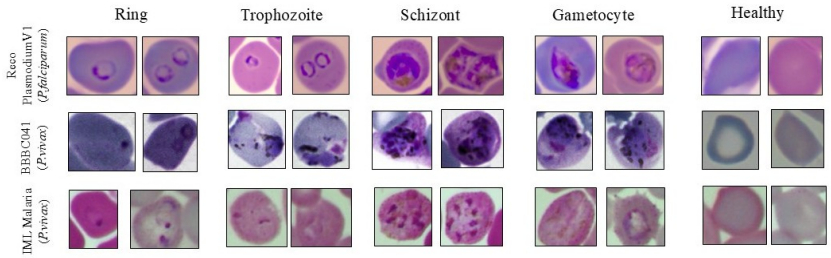

Fig. 2: Samples of healthy RBCs and development stages. From top to bottom, RecoPlasmodiumV1 (*P. falciparum*), BBBC041-v1 (*P. vivax*), IML Malaria (*P. vivax*).

Class	**RecoPlasmodiumV1**			**BBBC041-v1**			**IML Malaria**		
	Train	*Valid*	*Test*	*Train*	*Valid*	*Test*	*Train*	*Valid*	*Test*
Ring	482	78	115	317	36	169	121	15	28
Trophozoite	117	26	41	1339	134	111	57	7	13
Schizont	215	22	63	164	15	11	18	4	5
Gametocyte	101	18	41	125	19	12	169	33	59
Healthy	33066	4972	9742	69452	7968	5614	26423	3736	7740

Table 1: Number of instances for train, validation, and test sets across three datasets.

optimizer [39] and with a learning rate of 10^{-4}, reduced by 10^{-1} after half of a total of 50 epochs. Experiments are conducted on an NVIDIA-P100 GPU.

Evaluation Strategy. In section 5.1, we evaluate four training strategies to assess the effectiveness of our proposed method, MDIF-C. The first is *Individual Training (IT)*, where the model is trained on each domain separately. The second is *Joint Training (JT)*, where data from all domains are combined to train a single model. The last two strategies apply the MDIF approach to transfer information across domains: *Domain-Level MDIF (MDIF-D)*, adapted from Bai et al. [6], and *Class-Level MDIF (MDIF-C)*, our proposed method. In section 5.2, we compare our experimental results to the few approaches from the literature that also perform the full pipeline, even though differences in the pipeline architectures and evaluation metrics hinder a fair quantitative comparison.

Evaluation Metrics. To evaluate the performance of our full pipeline (two-stage method), we employ standard metrics for multiclass detection, including mean Average Precision over all classes (mAP) and over the four parasitized classes ($mAP@4$) at an IoU threshold of 0.5. In addition, we report accuracy (Acc), macro F1-score ($F1$) across all classes, and weighted F1-score over the four parasitized classes ($wF1@4$), computed at an IoU threshold of 0.5 and a confidence threshold of 0.3.

Particularly, given the F1 scores of each of the four parasitized classes (denoted as $F1_c, c \in \{R, T, S, G\}$) and of healthy RBCs (denoted as $F1_H$), and

the number of instances for each class $N_c, c \in \{R, T, S, G, H\}$, the macro F1-score $F1$ and the weighted F1-score over the four parasitized classes $wF1@4$ are respectively calculated as follows: $F1 = \frac{\sum_{c\in\{R,T,S,G,H\}} F1_c}{5}$ and $wF1@4 = \sum_{c\in\{R,T,S,G\}} \frac{N_c}{N_R+N_T+N_S+N_G} F1_c$

5 Experiment Results

5.1 Effectiveness of MDIF-C for the whole Recognition Pipeline

To assess the effectiveness of the proposed method, MDIF-C, we evaluate four training strategies applied to the second (classification) step in the pipeline, and report the performances of the complete recognition pipeline. Table 2 reports the mean Average Precision (mAP), accuracy (Acc), and macro F1-score ($F1$) over all classes, together with the performance on the four parasitized classes, measured using $mAP@4$ and $wF1@4$, for each experiment on the three datasets. We found that incorporating additional data from different domains does not always yield positive effects if the information transfer is not carefully designed. As shown in Table 2, the performance achieved with joint training (JT) or with domain-level fusion (MDIF-D) is not consistently superior to individual training (IT). This can be explained by the loss of class-specific information during unsupervised cross-domain knowledge transfer and/or data imbalance. In contrast, our proposed method, MDIF-C, reaches the highest scores for most metrics across all datasets, particularly for parasitized classes. Significant gains compared to MDIF-D are observed on both *P. vivax* datasets: BBBC041-v1 (roughly +4% in $mAP@4$ and +8.3% in $wF1@4$), and IML Malaria (+11.2% in $mAP@4$ and +3.6% in $wF1@4$). On the *P. falciparum* dataset RecoPlasmodiumV1, MDIF-C improves $mAP@4$ by roughly 2.2% and $wF1@4$ by about 0.8% compared to MDIF-D.

Method	RecoPlasmodiumV1					BBBC041-v1					IML Malaria				
	Acc	*mAP*	*F1*	*mAP@4*	*wF1@4*	*Acc*	*mAP*	*F1*	*mAP@4*	*wF1@4*	*Acc*	*mAP*	*F1*	*mAP@4*	*wF1@4*
IT	**93.62**	82.73	85.42	79.44	83.61	82.99	29.51	37.91	13.69	30.90	75.95	64.88	69.87	57.06	74.55
JT	93.58	79.97	84.19	75.25	82.60	83.40	32.85	35.08	17.69	31.59	75.94	60.07	66.87	51.06	73.22
MDIF-D	93.57	82.38	85.29	78.27	83.26	83.37	33.52	40.57	18.57	38.26	75.93	60.05	66.12	51.02	74.44
MDIF-C	93.59	**84.16**	**85.89**	**80.49**	**84.03**	**83.78**	**36.69**	**41.33**	**22.50**	**46.51**	**75.99**	**69.02**	**73.53**	**62.23**	**78.00**

Table 2: Performance comparison of four training methods for the full recognition pipeline on three datasets, reported using the total accuracy Acc, mAP, macro F1-score over all classes ($F1$), and the weighted F1-score over the four parasitized classes ($wF1@4$) (all in %). Bold and underlined values indicate the best and second-best results, respectively.

Fig. 3 shows the confusion matrices of the models trained with IT and MDIF-C on the BBBC041-v1 test set. Compared to IT, MDIF-C correctly recognizes more parasitized samples, particularly rings and trophozoites. It also reduces the number of ring and trophozoite samples misclassified as healthy RBCs.

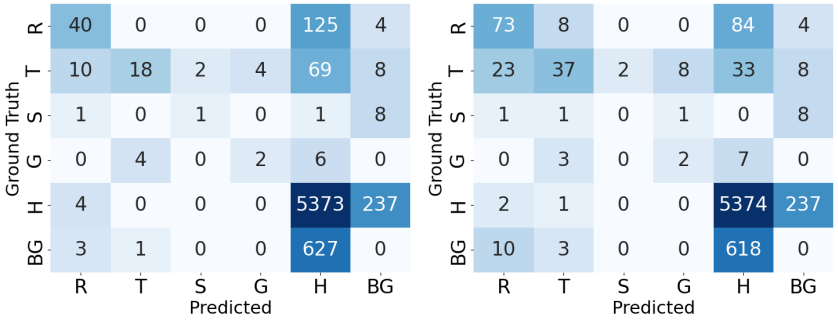

Fig. 3: Confusion matrices for IT (left) and MDIF-C (right) on the BBBC041-v1 test set. R, T, S, G, H, BG respectively denote "Ring", "Trophozoite", "Schizont", "Gametocyte", "Healthy" and "Background". MDIF-C achieves higher recognition of parasitized samples, especially rings and trophozoites, and reduces their misclassification as healthy RBCs.

5.2 Comparison with Existing Studies

To compare with existing methods, we performed experiments on public datasets using the published train/test splits whenever available. Although different studies have reported results on BBBC041-v1 and the IML dataset, most focus on infected RBC detection or infected RBC classification solely, with only very few addressing the recognition of the stage of parasites from whole blood smear images, *i.e.* the whole pipeline that can be used in real-life scenarios [7], [9], [12]. However, making a fair comparison with these works in realistic scenarios remains challenging.

The first difficulty stems from the different pipeline architectures used. Indeed, as existing pipelines rely on a detection step that provides either non-healthy cells (including parasitized RBCs and white blood cells) [7] or parasitized RBCs [9], [12], the subsequent classification models are trained without the class "Healthy", and thus cannot recognize it. On the other hand, given that our detection step provides all RBCs (including healthy ones, which are necessary to estimate per-stage parasitaemia), we include the class "Healthy" for training and evaluating our classifier. Although we provide a performance evaluation on the four parasitized classes only, our results are hardly comparable to the ones reported in [7], [9], [12]. Indeed, for their performance evaluations, these works not only ignore healthy RBCs misdetected as parasitized and parasitized RBCs misdetected as healthy, they also disregard the "Background" and "Difficult" classes.

The second difficulty arises from differences in train/test splitting, since each work employs its own partitioning strategy. More specifically, the images in BBBC041-v1 [14], provided by Jane Hung (first author of [7]), appear to have been used in [7], though the dataset is not explicitly specified in the paper. However, their test set differs from the one available on the official BBBC041-v1 website; therefore, we cannot directly compare our results with those of Hung *et al.* [7]. For the IML Malaria dataset, no official test set is provided. Zedda *et al.* [9], [12] evaluated their methods using random 70/20/10 splits for training,

validation, and testing. Based on their confusion matrix and the scores reported in [12], we find that the evaluation metrics were calculated in the same manner as in [7]. To the best of our knowledge, the studies by Zedda *et al.* in [9], [12] achieved the best results on this dataset.

Since our work employs the same split ratio as Zedda *et al.*, we provide an approximate comparison with their results by recalculating the metrics similarly as [9], [12], from our final confusion matrix. The metrics reported include accuracy ($Accuracy^{\dagger}$), macro F1 score ($F1^{\dagger}$), macro precision ($Precision^{\dagger}$), and macro recall (sensitivity) ($Recall^{\dagger}$) in the four parasitized classes, considering only true parasite detections. The comparison between our MDIF-C and existing methods is presented in Table 3, while Fig. 4 shows the confusion matrices of the method in [12] and our MDIF-C on the IML Malaria test set. Our MDIF-C makes less confusion between rings and trophozoites: there is only one ring recognized as trophozoite, and two samples of trophozoites recognized as rings, while the corresponding numbers in [12] are four and nine, respectively. Our MDIF-C outperforms the others in terms of accuracy ($Accuracy^{\dagger}$), macro F1 score ($F1^{\dagger}$), and sensitivity ($Recall^{\dagger}$).

Method	**IML Malaria**			
	$Accuracy^{\dagger}$	$F1^{\dagger}$	$Precision^{\dagger}$	$Recall^{\dagger}$
Zedda *et al.* [12]	80.0	76.0	**85.0**	73.0
Zedda *et al.* [9]	75.8	72.7	81.3	68.8
MDIF-C	**89.5**	**82.1**	83.0	**81.4**

Table 3: Performance comparison (all in %) with existing studies on the four parasitized classes, using the same evaluation strategy as in [9], [12], *i.e.* disregarding background, healthy RBCs misclassified as parasites, and parasites misclassified as healthy RBCs.

5.3 Discussion

Label noise in the ground truth of public datasets. During our experiments, we observed low performance on the two public datasets, even with individual training (IT). We further investigated by closely examining samples from BBBC041-v1 and IML Malaria, and found out that label noise may be a contributing factor. Some examples are shown in Fig. 5. In both datasets, many bounding boxes labeled as "Healthy" actually contain parasites, though many of them could be corrected by shrinking their bounding boxes (Fig. 5 (a) and (b)). There are also parasitized RBCs annotated as "Healthy", as illustrated in Fig. 5 (c). Such noise can confuse the models during the training process.

Robustness of MDIF-C against label noise. To assess the robustness of MDIF-C against label noise in the training data, we conducted an ablation study with different dataset combinations for the classification phase of parasite stage

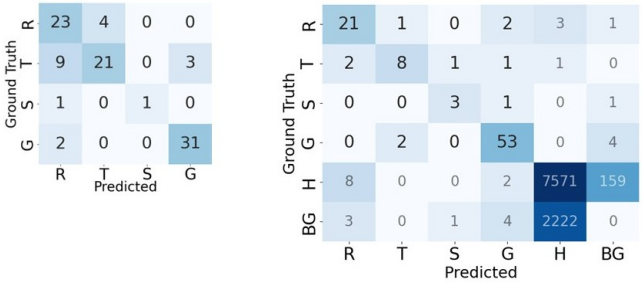

Fig. 4: Confusion matrices of Zedda *et al.* [12] (left, disregarding background, healthy RBCs misclassified as parasites, and parasites misclassified as healthy RBCs) and our MDIF-C across the full pipeline (right). R, T, S, G, H, and BG respectively denote "Ring", "Trophozoite", "Schizont", "Gametocyte", "Healthy" and "Background". Results are obtained from IML Malaria using the same train/validation/test split ratio, but not the same splits, which, combined with the fact that the confusion matrix in [12] (disregarding "background" as a predicted class) only provides partial information, explains the observed differences in the number of samples per class. MDIF-C gives fewer confusions among parasitized classes, especially between rings and trophozoites.

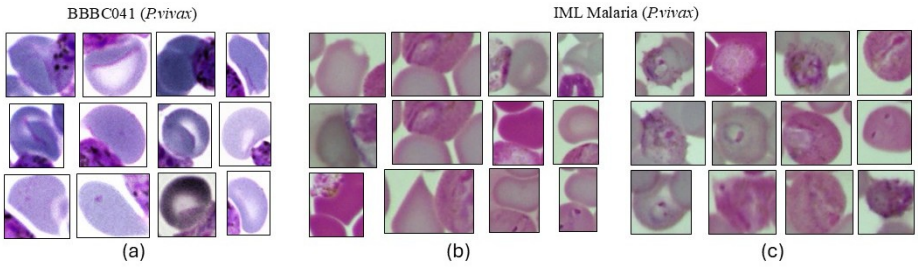

Fig. 5: Examples of noisy labels from our training subsets of BBBC041-v1 and IML Malaria. Panels (a) and (b) show bounding boxes annotated as "Healthy" but containing parasites, which in many cases could be corrected by reducing the bounding box size. Panel (c) shows RBCs mislabeled as "Healthy" despite being parasitized.

recognition. The first experiment used {BBBC041-v1, IML Malaria}, and the second used {BBBC041-v1, IML Malaria, RecoPlasmodiumV1}. Note that RecoPlasmodiumV1 has been collegially annotated by a pool of five expert microscopists, offering higher label quality than the other two datasets. The weighted F1-scores over the four parasitized classes ($wF1@4$) from these experiments are reported in Table 4. The results confirm that MDIF-C consistently achieves the best performance across both combinations. In particular, we observe substantial improvements over individual training (IT) and Joint Training (JT) on BBBC041-v1 and IML Malaria, despite their label noise. Moreover, the inclusion of RecoPlasmodiumV1 further yields significant gains on BBBC041-v1, with $wF1@4$ increasing from 33.85% under IT to 43.04% and 51.74% with MDIF-C when using {BBBC041-v1, IML Malaria} and {BBBC041-v1, IML Malaria, RecoPlasmodiumV1}, respectively. This shows the ability of MDIF-C to yield better performances in one domain (*P. vivax* in BBBC041-v1 and IML Malaria) by adding carefully annotated samples from another domain (*P. falciparum* in RecoPlasmodiumV1).

Dataset Group/Method		RecoPlasmodiumV1	BBBC041-v1	IML Malaria
IT (baseline)		88.83	33.85	72.93
{BBBC041-v1 + IML Malaria}	JT	-	20.75	79.20
	MDIF-D	-	40.14	75.47
	MDIF-C	-	**43.04**	**81.05**
{BBBC041-v1 + IML Malaria + RecoPlasmodiumV1}	JT	88.36	37.69	74.58
	MDIF-D	**88.87**	42.60	77.11
	MDIF-C	**88.87**	**51.74**	**80.39**

Table 4: Performance ($wF1@4$) of different strategies (IT, JT, MDIF-D, MDIF-C) for the classification phase on two dataset groups: {**BBBC041-v1, IML Malaria**} and {**BBBC041-v1, IML Malaria, RecoPlasmodiumV1**}. Values in bold indicate the best result within each group.

6 Conclusion

In this study, a two-step pipeline is proposed for recognizing *Plasmodium* development stages and healthy RBCs (which are necessary to estimate the per-stage parasitaemia). In the presence of data scarcity, we leverage the diversity of multiple datasets to train the model. In particular, we propose a new, supervised MDIF-based supervised domain adaptation method: MDIF-C, to enhance the model's generalization. Experiment results indicate that when combining multiple datasets for training, the performance gain is not always ensured without an appropriate domain fusion method. With the advantage of taking into account class labels on top of the domain during information fusion, our MDIF-C achieves the highest scores on most metrics across the three experimented datasets. Besides, we also find that data quality plays an important role in multi-domain learning, and show the robustness of MDIF-C to data noise.

Beyond this specific application, MDIF-C can be used to solve other fine-grained recognition tasks in the presence of data scarcity. An interesting direction to explore would be to adapt MDIF-C to semi-supervised learning when a large amount of data is available but mostly unlabeled.

References

[1] World Health Organization, *Malaria*, https://www.who.int/news-room/fact-sheets/detail/malaria, Accessed: July 29, 2025.

[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, 2020.

[4] T. Xiao, H. Li, W. Ouyang, and X. Wang, "Learning deep feature representations with domain guided dropout for person re-identification," in *Proceedings of the IEEE conference on CVPR*, 2016, pp. 1249–1258.

[5] 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.

[6] 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 CVPR*, 2021, pp. 12 914–12 923.

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

[8] G. Wang, G. Luo, H. Lian, L. Chen, W. Wu, and H. Liu, "Application of deep learning in clinical settings for detecting and classifying malaria parasites in thin blood smears," in *Open forum infectious diseases*, Oxford University Press US, vol. 10, 2023.

[9] L. Zedda, A. Loddo, and C. Di Ruberto, "A deep architecture based on attention mechanisms for effective end-to-end detection of early and mature malaria parasites in a realistic scenario," *Computers in Biology and Medicine*, vol. 186, p. 109 704, 2025.

[10] Q. A. Arshad, M. Ali, S. 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.

[11] 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.

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

[13] 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, 2021.

[14] V. Ljosa, K. L. Sokolnicki, and A. E. Carpenter, "Annotated high-throughput microscopy image sets for validation," *Nature methods*, vol. 9, no. 7, p. 637, 2012.

[15] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, Curran Associates, Inc., 2012.

[17] G. Jocher, A. Chaurasia, and J. Qiu, *Ultralytics yolov8*, version 8.0.0, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[19] 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.

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

[21] 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.

[22] R. Singh, V. Bharti, V. Purohit, A. Kumar, A. K. Singh, and S. K. Singh, "Metamed: Few-shot medical image classification using gradient-based meta-learning," *Pattern Recognition*, vol. 120, p. 108 111, 2021.

[23] L. Gao, L. Zhang, C. Liu, and S. Wu, "Handling imbalanced medical image data: A deep-learning-based one-class classification approach," *Artificial intelligence in medicine*, vol. 108, p. 101 935, 2020.

[24] A. Galdran, G. Carneiro, and M. A. González Ballester, "Balanced-mixup for highly imbalanced medical image classification," in *Medical Image Computing and Computer Assisted Intervention - MICCAI*, Springer, 2021, pp. 323–333.

[25] 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 ECCV Workshops*, 2018.

[26] 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 CVPR*, 2019, pp. 7193–7201.

[27] 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 CVPR*, 2017, pp. 2272–2281.

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

[29] 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.

[30] 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.

[31] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the 32nd ICML - Volume 37*, ser. ICML'15, Lille, France: JMLR.org, 2015, pp. 1180–1189.

[32] 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 CVPR*, 2019, pp. 7354–7362.

[33] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *arXiv preprint arXiv:2410.17725*, 2024.

[34] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: Common objects in context," in *European conference on computer vision*, Springer, 2014, pp. 740–755.

[35] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE CVPR*, 2016, pp. 770–778.

[37] 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 CVPR*, IEEE, 2009, pp. 248–255.

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

[39] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.