



REPUBLIC OF CAMEROON  
Peace – Work – Fatherland  
◇ ◇ ◇ ◇ ◇  
MINISTRY OF HIGHER EDUCATION  
◇ ◇ ◇ ◇ ◇  
THE UNIVERSITY OF DOUALA  
◇ ◇ ◇ ◇ ◇  
National Higher Polytechnic School of Douala (NHPSD)



Laboratory of Computer Engineering,  
Data Science and Artificial Intelligence (LDSAI)

Academic Year: 2024–2025

---

# Multi-Task Transfer Learning

## And Application to Malaria Detection

---

Presented by  
**Gwade Steve**

Submitted in partial fulfillment of the requirements  
for the degree of

**Master of Science**  
in Data Science and Artificial Intelligence

**Supervised by:**  
Prof. Dr. habil. P. Njionou Sadjang

**Co-supervised by:**  
Dr. Armieille Noulapeu

National Higher Polytechnic School of Douala  
University of Douala, Cameroon

November 22, 2025

---

## Declaration

---

I, **Gwade Steve**, hereby solemnly declare that this Master's thesis entitled "*Multi-Task Transfer Learning and Application to Malaria Detection*" is the result of my own research and investigations carried out under the guidance and supervision of Prof. Dr. habil. P. Njionou Sadjang and Dr. Armielle Noulapeu at the National Higher Polytechnic School of Douala (NHPSD), University of Douala, Cameroon.

I declare that, to the best of my knowledge, this work has not been submitted, either in whole or in part, for any other degree, diploma, or certification at any other academic institution.

I further declare that all sources of information, ideas, and data used in this thesis have been fully acknowledged. Any contributions from other researchers, authors, or institutions are clearly referenced in accordance with academic conventions.

I accept full responsibility for the integrity and authenticity of the research and the results presented herein. I affirm that any errors or omissions are entirely my own and do not reflect the work of my supervisors or any other individual.

Signed: \_\_\_\_\_

Date: \_\_\_\_\_

Place: \_\_\_\_\_

---

## Acknowledgments

---

First, I place my gratitude to God for the strength, health, and guidance granted to me throughout this path.

I would like to express my deepest appreciation to my supervisor, **Prof. Dr. habil. PATRICK NJIONOU SADJANG**. Thank you for your trust, your patience, and the wisdom you shared at every stage of this research. Your mentorship was vital in shaping not just this work, but my growth as an engineer and as a researcher.

A special thanks goes to my co-supervisor, **Dr. ARMIELLE NOULAPEU**, for her sharp eye and honest feedback. Her critiques significantly elevated the quality of this work. I am also indebted to **LIONEL OWONO**, **ALEX BRUNO KENFACK**, **YVAN NGNEUNMEU** and **Dr. HIOL NICOLAS** for taking the time to review my drafts and offering insights that helped me see things more clearly.

To my colleagues and labmates, thank you for the conversations and debates that helped refine the ideas presented here. I also want to give a specific shout-out to my friends **NYEMB NDJEM EONE ANDRE KEVIN** and **NJIMEYUP HAROLD FRANCOIS** for their support and feedback.

Most importantly, my heart goes out to my family. To my mother, **NGO HIOLL VERONIQUE**, and my father, **GWADE ELIAS**, thank you for your endless love and for the sacrifices you made to get me here. Finally, to my girlfriend, thank you for your understanding, and for keeping me motivated during the hardest moments of this journey.

---

# Abstract

---

Automated malaria diagnosis remains a major challenge for researchers. While many deep learning methods exist, most focus on simplified goals like binary classification or just finding infected cells in a blood smear. These specialized models often achieve high scores on benchmarks, but they struggle in real diagnostic settings that require identifying cells, locating parasites, and measuring infection levels all at once.

This thesis proposes a **Multi-Task Transfer Learning (MTTL)** framework designed to perform a complete diagnostic analysis from single thin blood smear images. The system uses Low-Rank Adaptation (LoRA) and partial fine-tuning to learn four tasks together: multi-class detection, cell segmentation, infection localization, and instance-level classification.

Our experiments show that standard Single-Task Learning (STL) works well for simple detection but fails as the complexity increases. We observed the F1 score drop from 0.82 in simple one-class tests to 0.38 in three-class settings. In contrast, our best MTTL model, using RoI Classification, achieved a 103% relative improvement in detecting infected cells. It also reduced the error in parasitemia estimation from 3.30% down to 1.08%, a 67% reduction, while producing fewer false alarms and sharper localization results.

These findings suggest that combining Transfer Learning, Multi-Task Learning, and Parameter-Efficient Fine-Tuning creates a strong foundation for complex medical imaging tasks. This framework improves reliability and makes results easier to interpret, offering a practical strategy for developing diagnostic models that can be adapted to other biomedical fields. The complete implementation and trained models have been released publicly.

**Keywords:** Malaria Diagnosis, Multi-Task Learning, Transfer Learning, Computational Pathology, Deep Learning, Blood Smear Analysis, Object Detection, LoRA, Fine-Tuning

**Code Repository:** <https://github.com/GwadeSteve/MTTL-Research-Malaria>

---

## Summary

---

**Context and Motivation.** Malaria remains one of the most persistent infectious diseases in the world and continues to affect millions of people each year, particularly in tropical regions such as Sub-Saharan Africa. Microscopic examination of thin blood smears is still considered the reference method for diagnosis because it allows direct visualization of parasites. However, this process is slow, requires highly skilled technicians, and can produce inconsistent results in low-resource environments. In recent years, machine learning methods have been explored to support and automate malaria diagnosis. Despite promising progress, most models have been developed for very specific objectives such as binary detection of infected cells or parasite classification, without considering the complete diagnostic process.

**Problem Statement and Objectives.** This observation reveals an important research gap, the lack of unified computational models able to handle the full complexity of malaria diagnosis within a single framework. In practice, diagnosis involves several interrelated steps such as identifying infected cells, segmenting their contours, locating the parasite, and estimating the level of infection. Existing approaches usually address these tasks separately, which limits their robustness when applied to new data or varied clinical conditions. The objective of this work is therefore to design, implement, and evaluate a comprehensive diagnostic framework based on **Multi-Task Transfer Learning (MTTL)**. The goal is to develop a system that learns several related tasks at the same time and produces reliable and complete diagnostic predictions from a single blood smear image.

**Methodology.** We begin by defining a mathematical framework that combines three existing concepts: **Transfer Learning (TL)**, **Multi-Task Learning (MTL)**, and **Parameter-Efficient Fine-Tuning (PEFT)**. This theoretical basis provides a structured way to adapt pre-trained models to multiple interdependent tasks. Based on this formulation, we implemented a hybrid approach using Low-Rank Adaptation (LoRA) together with partial fine-tuning of the backbone network. The proposed system learns four related tasks at once: cell detection, binary cell segmentation, parasite localization, and instance-level cell classification. Experiments were conducted using the public NLM thin falciparum blood smear dataset in both single-task and multi-task configurations, making it possible to directly compare the performance of **Single-Task Learning (STL)** and the proposed **MTTL** approach.

---

**Results and Analysis.** The results clearly show the advantage of the multi-task strategy. Single-task models perform well on simple detection but lose stability when the problem becomes more complex. For example, the F1 score for detecting infected cells decreases from 0.82 in the single-class case to 0.38 when three classes are considered. The multi-task model, regularized by auxiliary tasks, remains much more stable and consistent. The best MTTL configuration, augmented with RoI Classification, achieves an improvement of 103% in the F1 score for infected cells compared with STL, and the mean absolute error in parasitemia estimation decreases from 3.30% to 1.08%, representing a **67%** reduction. Visual inspection also shows that MTTL produces cleaner detection maps, fewer false positives on healthy slides, and more accurate localization of infected regions.

**Conclusion and Perspectives.** These findings confirm that combining Transfer Learning, Multi-Task Learning, and Parameter-Efficient Fine-Tuning forms a strong foundation for complex medical image analysis. The proposed MTTL model improves both diagnostic accuracy and interpretability by linking related visual cues across tasks. Although this study focuses on malaria detection, the same framework could be extended to other biomedical imaging domains that require the joint learning of related objectives. Future work may include validation on additional datasets, evaluation under domain shift conditions, and a more detailed theoretical study of task transferability and generalization.

**Keywords:** Malaria Diagnosis, Multi-Task Learning, Transfer Learning, Deep Learning, Parameter-Efficient Fine-Tuning, LoRA, Computational Pathology

**Code Repository:** <https://github.com/GwadeSteve/MTTL-Research-Malaria>

# Contents

<b>Declaration</b> . . . . .	i
<b>Acknowledgments</b> . . . . .	ii
<b>Abstract</b> . . . . .	iii
<b>Summary</b> . . . . .	iv
<b>List of Acronyms</b> . . . . .	ix
<b>List of Figures</b> . . . . .	x
<b>List of Tables</b> . . . . .	xii
<b>List of Equations</b> . . . . .	xiii
<b>Introduction</b> . . . . .	1
<b>1 Theoretical Foundations and Literature Review</b> . . . . .	5
1.1 Deep Neural Networks as Composite Functions . . . . .	5
1.1.1 Hierarchical Representations . . . . .	6
1.1.2 Convolutional Neural Networks . . . . .	6
1.2 CNN Architecture Components . . . . .	7
1.3 Limitations of Training CNNs from Scratch . . . . .	10
1.4 Transfer Learning Paradigm . . . . .	11
1.4.1 Definition and Typology . . . . .	11
1.4.2 Common Transfer Learning Techniques in Deep Learning . . . . .	12
1.4.3 General Benefits and Limitations of Transfer Learning . . . . .	13
1.5 Multitask Learning Paradigm . . . . .	14
1.5.1 Definition and Formulation . . . . .	14
1.5.2 MTL Architectural Approaches . . . . .	15
1.5.3 Optimization Challenges . . . . .	15
1.5.4 General Benefits and Limitations of Multitask Learning . . . . .	16
1.6 Prior Work in Malaria Detection . . . . .	16
1.6.1 Advanced AI Paradigms for Malaria Detection . . . . .	16
1.6.2 Multitask and Multitask Transfer Learning Relevance . . . . .	19
1.6.3 Observed Gaps in the Literature . . . . .	20

<b>2 Multi-Task Transfer Learning Framework</b>	<b>21</b>
2.1 Mathematical Preliminaries and Notation	21
2.2 MTTL Definition and Typology	22
2.3 The MTTL Problem Formulation	23
2.3.1 Derivations by Fine-Tuning Strategy	23
2.3.2 Derivations by Task-Balancing Strategy	25
<b>3 Methodological Application to Malaria Diagnosis</b>	<b>26</b>
3.1 Instantiated MTTL Objective for Malaria Diagnosis	26
3.1.1 Dataset	27
3.1.2 Image Preprocessing Pipeline	28
3.2 Tasks Formulation	28
3.3 MTTL Model Architecture	31
3.3.1 Shared Backbone with Integrated LoRA Adapters	33
3.3.2 Task-Specific Heads	33
3.4 Experimental Protocol	37
3.4.1 Phase 1: Establishing Optimal STL Baselines	37
3.4.2 Phase 2: Evaluating Diagnostic Completeness and MTTL Synergy	37
3.5 Evaluation Protocol and Metrics	38
3.5.1 Diagnostic Test Datasets	38
3.5.2 Task-Specific Performance Metrics	38
3.6 Implementation Details and Experimental Setup	39
3.6.1 Hardware and Software Environment	39
3.6.2 Training Configuration	40
<b>4 Results and Discussion</b>	<b>41</b>
4.1 Single-Task Learning Detection Baselines	41
4.2 Evaluating STL Baselines and Their Limitations	42
4.2.1 STL Baseline Performance Across All Diagnostic Tasks	42
4.2.2 Training Dynamics of STL Models	42
4.2.3 STL Performance Under Increasing Complexity	44
4.3 MTTL Performance and Synergy Analysis	44
4.3.1 Analysis of MTTL Synergy	45
4.3.2 Training Dynamics for Top MTTL Models	47
4.4 Final Model Evaluation and Robustness Analysis	48
4.4.1 Detection Performance Leaderboard	48
4.4.2 Diagnostic Robustness on Specialized Subsets	48
4.4.3 Parasitemia Quantification on Multi-Class Detection Models	49
<b>5 Discussion</b>	<b>51</b>
5.1 The Specialist vs. The Generalist	51
5.2 Answering the Central Research Question	51
5.3 Implications of Key Findings	52

5.4 Limitations of the Study . . . . .	52
<b>6 Conclusion . . . . .</b>	<b>54</b>
<b>References . . . . .</b>	<b>56</b>
<b>A Experimental Configurations . . . . .</b>	<b>64</b>
A.1 Training and Optimization Strategy . . . . .	64
A.2 Model Configurations . . . . .	64
<b>B Performance Leaderboards for All Tasks . . . . .</b>	<b>66</b>
B.1 Detection Performance and Synergy . . . . .	66
B.2 Auxiliary Task Leaderboards . . . . .	67
<b>C Qualitative Examples . . . . .</b>	<b>69</b>
C.1 Auxiliary Segmentation and Localization . . . . .	69
C.2 Auxiliary RoI Classification . . . . .	71
C.3 Detection Performance . . . . .	72

---

# List of Acronyms

---

<b>AI</b>	Artificial Intelligence . . . . .	2
<b>AMP</b>	Automatic Mixed Precision . . . . .	40
<b>ANN</b>	Artificial Neural Network . . . . .	5
<b>AUC</b>	Area Under the Curve . . . . .	10
<b>BCE</b>	Binary Cross-Entropy . . . . .	9
<b>CNN</b>	Convolutional Neural Network . . . .	2
<b>CV</b>	Computer Vision . . . . .	2
<b>DL</b>	Deep Learning . . . . .	2
<b>FC</b>	Fully Connected . . . . .	9
<b>FN</b>	False Negative . . . . .	2
<b>FP</b>	False Positive . . . . .	2
<b>FPN</b>	Feature Pyramid Network . . . . .	32
<b>HOG</b>	Histogram of Oriented Gradients . .	6
<b>IoU</b>	Intersection over Union . . . . .	38
<b>LBP</b>	Local Binary Pattern . . . . .	6
<b>LoRA</b>	Low-Rank Adaptation . . . . .	32
<b>MAE</b>	Mean Absolute Error . . . . .	39
<b>mAP</b>	mean Average Precision . . . . .	38
<b>ML</b>	Machine Learning . . . . .	4
<b>MTL</b>	Multi-Task Learning . . . . .	3
<b>MTTL</b>	Multi-Task Transfer Learning . . . .	3
<b>NLM</b>	National Library of Medecine . . . .	27
<b>PEFT</b>	Parameter-Efficient Fine-Tuning . . .	3
<b>RBC</b>	Red Blood Cell . . . . .	3
<b>RDT</b>	Rapid Diagnostic Test . . . . .	1
<b>ReLU</b>	Rectified Linear Unit . . . . .	8
<b>ROC</b>	Receiver Operating Characteristic .	10
<b>RoI</b>	Region of Interest . . . . .	33
<b>RPN</b>	Region Proposal Network . . . . .	33
<b>STL</b>	Single-Task Learning . . . . .	3
<b>TL</b>	Transfer Learning . . . . .	3
<b>ViT</b>	Vision Transformer . . . . .	7
<b>WBC</b>	White Blood Cell . . . . .	3
<b>WSI</b>	Whole Slide Imaging . . . . .	17

---

# List of Figures

---

1	Common methods for malaria diagnosis. . . . .	2
1.1	Hierarchical feature learning in a deep network. . . . .	6
1.2	Some notable Convolutional Neural Network (CNN) architectures. . . . .	7
1.3	Basic building blocks of a Convolutional Neural Network (CNN) . . . . .	7
1.4	Principle of the convolution operation. . . . .	8
1.5	Max pooling with 2x2 windows and stride s=2 from input to output. . . . .	8
1.6	Conceptual overview of three common fine-tuning strategies. . . . .	13
1.7	Main architectural approaches for MTL. . . . .	15
3.1	Original Samples from NLM Thin Falciparum dataset. . . . .	29
3.2	Preprocessed version of the images from Figure 3.1. . . . .	29
3.3	Visualization of Ground Truth Data for the Multi-Task Framework. . . . .	31
3.4	High-level diagram of our MTTL model architecture. . . . .	32
3.5	Architectural details of the shared backbone, illustrating (a) the core LoRA module, (b) its integration into a standard ResNet block, and (c) the overall adapted backbone structure. . . . .	34
3.6	Custom Faster R-CNN Detection Head Architecture. . . . .	35
3.7	Segmentation Head Architecture. . . . .	35
3.8	Infection Localization Head Architecture. . . . .	36
3.9	MLP Classification Head Architecture. . . . .	36
4.1	Training dynamics for 1-Class and 2-Class optimal STL detection models. . . .	43
4.2	Training dynamics for 3-Class detection and 2-Class RoI classification models. .	43
4.3	Training dynamics for 3-Class RoI classification and Segmentation models. . .	43
4.4	Training dynamics for the Heatmap Localization model. . . . .	43
4.5	STL detector performance degrades with complexity. . . . .	44
4.6	Best 1-Class Detection MTTL model (+Seg, F1=0.7654). . . . .	47
4.7	Best 2-Class Detection MTTL model (+RoI, F1=0.7810). . . . .	47
4.8	Best 3-Class Detection MTTL model (+RoI, F1=0.7710). . . . .	47
4.9	Parasitemia estimation for all best multi-class detection models. . . . .	49
C.1	Qualitative comparison for Cell Segmentation. . . . .	69

---

C.2	Qualitative comparison for Heatmap Localization. . . . .	70
C.3	Qualitative comparison for Heatmap Localization. . . . .	70
C.4	Qualitative comparison for 3-Class RoI Classification. . . . .	71
C.5	Qualitative comparison for 1-Class Detection. . . . .	72
C.6	Qualitative comparison for 2-Class Detection. . . . .	73
C.7	Qualitative comparison for 3-Class Detection. . . . .	74

---

# List of Tables

---

1.1	Selected TL Applications in Medical Imaging and Malaria Detection . . . . .	17
3.1	Class Distribution of Annotated Cells in the Dataset. . . . .	27
3.2	Summary of Evaluation Metrics for Each Task. . . . .	39
4.1	1-Class Detection Baselines (Ranked by F1(Inf)) . . . . .	41
4.2	2-Class Detection Baselines (Ranked by F1(Inf)) . . . . .	41
4.3	3-Class Detection Baselines (Ranked by F1(Inf)) . . . . .	41
4.4	STL baseline performance for Detection. . . . .	42
4.5	STL baseline performance across tasks. . . . .	42
4.6	MTTL experiments results for <b>1-Class Detection</b> . . . . .	45
4.7	MTTL experiments results for <b>2-Class Detection</b> . . . . .	45
4.8	MTTL experiments results for <b>3-Class Detection</b> . . . . .	46
4.9	Best performers by detection task complexity. . . . .	48
4.10	Performance of best detection models on specialized test subsets. . . . .	48
4.11	Parasitemia estimation performance of the champion models on the full test set.	49
A.1	Single-Task Learning (STL) Champion Model Configurations. . . . .	64
A.2	Multi-Task Transfer Learning (MTTL) Configurations. . . . .	65
A.3	Detailed training configurations for MTTL task combinations. . . . .	65
B.1	MTTL results for <b>1-Class Detection</b> . . . . .	66
B.2	MTTL results for <b>2-Class Detection</b> . . . . .	66
B.3	MTTL results for <b>3-Class Detection</b> . . . . .	67
B.4	Segmentation Performance Leaderboard (Ranked by Dice). . . . .	67
B.5	Heatmap Localization Performance Leaderboard (Ranked by Dice). . . . .	67
B.6	RoI Classification Performance Leaderboard (Ranked by F1-Macro). . . . .	68

---

# List of Equations

---

1.1	Iterative Layer Transformation . . . . .	5
1.2	Deep Network Composite Function . . . . .	5
1.3	LoRa Update equation. . . . .	13
1.4	Multitask Learning optimization formulation. . . . .	14
2.1	Domain Definition. . . . .	21
2.2	Task Definition. . . . .	21
2.3	Model Decomposition. . . . .	21
2.4	Empirical Risk Minimization. . . . .	21
2.5	STL Optimization Objective. . . . .	22
2.6	Transfer Learning Optimization Objective. . . . .	22
2.7	Multi-task Learning Optimization Objective. . . . .	22
2.8	Source Backbone Parameter Extraction. . . . .	23
2.9	MTTL Master Objective Function. . . . .	23
2.10	MTTL Objective with Uncertainty Weighting. . . . .	25
3.3	Instantiated MTTL Objective. . . . .	27
3.5	Total Multi-Class Detection Loss. . . . .	28
3.6	Smooth L1 Loss component. . . . .	30
3.7	Cell Segmentation Loss. . . . .	30
3.8	Infection Localization Loss. . . . .	30
3.9	RoI Classification Loss. . . . .	30
4.1	Performance gain formula. . . . .	45

---

# Introduction

---

Malaria remains a persistent global public health challenge. Transmitted by infected female *Anopheles mosquitoes*, this parasitic disease places a disproportionate burden on low and middle-income countries, particularly in sub-Saharan Africa. The 2024 *World Malaria Report* estimates there were nearly **263 million** cases and over **597,000 deaths** in 2023, with children under five and pregnant women bearing the highest risk (World Health Organization 2023). Beyond direct health consequences, malaria slows economic development and strains healthcare systems, perpetuating cycles of poverty (Sachs et al. 2001). Despite decades of eradication efforts including insecticide-treated nets and preventive therapies the disease continues to pose a major obstacle to global health.

Effective control relies on early, reliable, and widely available diagnosis. Missed or delayed diagnoses lead to incorrect treatment, increased mortality, and continued transmission (World Health Organization 2023). However, diagnostic consistency remains a significant hurdle. Microscopy is highly accurate but demands trained technicians, stable electricity, and equipped laboratories resources often scarce in rural areas (Wongsrichanalai et al. 2007). Rapid diagnostic tests (RDTs) offer an accessible alternative, yet their performance is inconsistent. Field studies in Cameroon highlight this variability, some tests show high specificity but low sensitivity (Ngalame et al. 2025), others demonstrate the reverse, leading to high false-positive rates (Missoup et al. 2025). Furthermore, factors such as storage conditions and parasite genetic variations, including *pfhrp2/3* deletions, jeopardize RDT reliability (Gatton et al. 2020; Prosser et al. 2021).

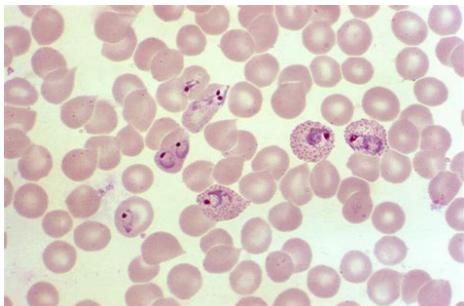
These limitations urges the need for diagnostic tools that balance accuracy, speed, and field suitability. Digital microscopy and AI-based image analysis show promise in reducing reliance on specialized infrastructure while making precise diagnosis accessible (Poostchi et al. 2018; Maturana et al. 2022; World Health Organization 2023).

## The Malaria Diagnostic Challenge

Traditional diagnosis relies on microscopy and Rapid Diagnostic Tests (RDTs), both of which face operational constraints. Microscopy (Figure 1) is the gold standard (World Health Organization 2016). When performed by experts, it identifies parasite species and infection

---

severity. However, its efficacy declines in real-world settings due to technician shortages, variable slide preparation, and the time-intensive nature of manual examination.



(a) Manual microscopy of a blood smear.



(b) Rapid Diagnostic Test (RDT).

Figure 1: Common methods for malaria diagnosis.

Conversely, RDTs are cost-effective and easy to deploy in underserved regions. Yet, they lack the ability to quantify parasitemia or identify specific species and may fail to detect low-density infections or specific strains (Eyong et al. 2022; Jacobs et al. 2014). The resulting False Negatives (FNs) and False Positives (FPs) contribute to untreated cases or drug resistance through misuse. Combined with supply chain interruptions and high seasonal caseloads, these limitations leave large populations vulnerable to inaccurate diagnosis.

## The Potential of AI in Medical Imaging

Advances in Artificial Intelligence (AI), particularly Computer Vision (CV) and Deep Learning (DL), offer new pathways for medical imaging. Models based on Convolutional Neural Networks (CNNs) and transformer architectures can analyze data with consistency and precision that rivals human experts. Systems based on these architectures have reached near expert performance in tasks such as detecting tumors in X-rays, grading retinal diseases in fundus photographs (Gulshan et al. 2016), and classifying skin lesions from dermoscopic images (Esteva et al. 2017). In the context of malaria, AI offers distinct advantages over traditional methods.

First, AI models can automatically detect parasites in blood smear images, reducing dependency on human expertise and accelerating turnaround times. Second, unlike standard RDTs, AIs can classify species and estimate parasite density metrics essential for appropriate treatment. Third, these systems can be embedded into low-cost digital microscopes or mobile applications, extending reach to low-resource environments. While currently limited to proof-of-concept studies (Rajaraman et al. 2018), AI offers scalability and adaptability, models can be retrained to recognize new strains or regional variations. By combining automated analysis with telemedicine, AI can serve as a powerful complementary tool to strengthen malaria control programs.

---

## Research Problem and Motivation

While DL and Transfer Learning (TL) have shown promise in detecting *parasitized cells*, current systems often fall short of clinical utility. Most approaches focus on binary classification (infected vs. uninfected) or parasitized cell detection, whereas a reliable diagnosis requires locating parasites, distinguishing cell types (e.g., Red Blood Cells (RBCs), White Blood Cells (WBCs)), estimating parasitemia and even distinguishing parasite specie. The current literature of Single-Task Learning (STL) approaches do not deliver this comprehensive information from a single smear using a single model.

Multi-Task Learning, a paradigm designed to learn related tasks simultaneously, has been largely overlooked in this domain. We go even further and propose that combining MTL with the prior knowledge of TL creates a superior hybrid strategy. To our knowledge, this combined paradigm **Multi-Task Transfer Learning (MTTL)** has not been formally applied to malaria diagnosis. This thesis addresses two central problems:

1. **Practical Problem:** Existing works in the literature address only isolated aspects of diagnosis (e.g., infected cell detection, binary classification), providing clinicians with partial outputs rather than a diagnostic system.
2. **Theoretical Problem:** There is no clear formulation in current literature unifying TL, MTL, and modern Parameter-Efficient Fine-Tuning (PEFT) techniques for medical image analysis. Our work closes this gap by proposing a unified formulation and using malaria detection as a representative case.

## Objectives and Research Questions

The aim of this research is to design and empirically validate a MTTL framework for automated malaria diagnosis.

- **Main Objective:** Design a MTTL approach and validate its effectiveness in malaria diagnosis.
- **Specific Objectives:**
  - Formalize and propose a mathematical formulation of MTTL.
  - Apply the proposed framework to malaria diagnosis tasks.
  - Compare the performance of MTTL against STL baselines.
  - Evaluate the contribution of auxiliary tasks to identify optimal configurations.

**Research Question:** *Does an MTTL approach provide superior efficacy and robustness compared to STL for automated malaria diagnosis?*

This research question guided our work and led us to the conclusions drawn at the end of this work.

---

# Hypotheses

## Main Hypothesis:

We hypothesize that initializing a shared backbone with relevant prior knowledge and jointly learning related tasks forces the model to learn a more powerful, generalizable shared representation, improving performance on individual tasks compared to independent training.

## Supporting Hypotheses

- **H1 (MTTL vs. STL):** The MTTL paradigm will outperform STL baselines in diagnostic completeness and generalization, specifically in multi-class cell detection.
- **H2 (Auxiliary Task Synergy):** Auxiliary tasks will act as effective regularizers.
- **H3 (Robustness):** MTTL models will demonstrate higher robustness on specialized subsets (Infected-Only and Healthy-Only) compared to STL baselines.

# Contributions and Thesis Outline

This work contributes to medical imaging and Machine Learning (ML) by providing:

- **An MTTL Framework:** A mathematical formulation combining TL, MTL, and PEFT for medical image analysis.
- **A Diagnostic System:** The design and validation of a comprehensive, multi-task system for malaria diagnosis.
- **Empirical Evidence:** A rigorous study demonstrating the superiority of the MTTL paradigm over STL for complex diagnostics.

The rest of this document is structured as follows. **Chapter 1** establishes the theoretical foundations of DL, CNNs, TL, MTL, PEFT, and the current state of literature for AI applied to medical imaging and malaria specifically. **Chapter 2** presents our core contribution, the mathematical framework for MTTL. **Chapter 3** details the methodological application to malaria diagnosis, the experimental implementation, including dataset and architecture. **Chapter 4** reports the empirical findings and comparative analysis. Finally, **Chapter 5** interprets the implications and limitations of the study, and **Chapter 6** summarizes findings and outlines future directions.

## Theoretical Foundations and Literature Review

---

Deep Learning (DL), a branch of Machine Learning (ML), has revolutionized how machines analyze complex data, especially visual information like medical images (Litjens et al. 2017). Its fundamental strength, as highlighted by reviews such as Alzubaidi et al. (2021), is the ability to learn representations of features directly from raw data, obviating the need for manual, domain-specific feature engineering. This chapter covers the foundational principles of DL and Convolutional Neural Networks (CNNs). We will define their core components and mathematical foundations, describe the learning process, and discuss the principal challenges encountered when applying these models to medical data. This foundation is essential to understand why advanced learning paradigms like Transfer and Multitask Learning are not merely advantageous, but often necessary.

### 1.1 Deep Neural Networks as Composite Functions

DL models are a class of Artificial Neural Networks (ANNs) distinguished by their depth, typically referring to a significant number of layers. Mathematically, a deep network  $h(x; \theta)$  is a composite function that maps an input  $x$  to an output through a series of  $L$  nested transformations.

This process can be defined iteratively. Let the input be the zeroth-layer activation,  $A^{(0)} = x$ . The activation  $A^{(i)}$  of the  $i$ -th layer is computed by applying a transformation  $h^{(i)}$  (parameterized by  $\theta_i$ ) to the activation of the preceding layer:

$$A^{(i)} = h^{(i)}(A^{(i-1)}; \theta_i) \quad \text{for } i = 1, \dots, L \tag{1.1}$$

The final output of the network is the activation of the last layer,  $y = A^{(L)}$ . This unrolled, iterative definition results in the fully composite function:

$$h(x; \theta) = h^{(L)}(h^{(L-1)}(\dots h^{(1)}(x; \theta_1) \dots; \theta_{L-1}); \theta_L) \tag{1.2}$$

The complete set of model parameters,  $\theta = \{\theta_1, \dots, \theta_L\}$ , is learned from data via an optimization process. The hierarchical, multi-layered structure enables the model to discover and represent intricate structures within the data at multiple, increasing levels of abstraction.

### 1.1.1 Hierarchical Representations

Traditional ML workflows depended on handcrafted features (e.g., SIFT (Lowe 2004), Histogram of Oriented Gradients (HOG) (Dalal et al. 2005), Local Binary Pattern (LBP) (Ojala et al. 2002)), which required significant domain expertise and were often brittle. DL models, and CNNs in particular, automate this process of feature discovery (LeCun, Bengio, et al. 2015). The hierarchical composition of functions enables a layer-wise abstraction of features:

- **Early Layers:** Learn to detect primitive features such as edges, corners, and color gradients.
- **Intermediate Layers:** Combine these primitives into more complex motifs, textures, and parts of objects.
- **Later Layers:** Assemble motifs into high-level, task relevant representations of entire objects or concepts.

Automated learning of a feature hierarchy is the primary advantage of deep learning models, especially in medical imaging where the relevant visual patterns may be too subtle for human-engineered features to capture effectively.

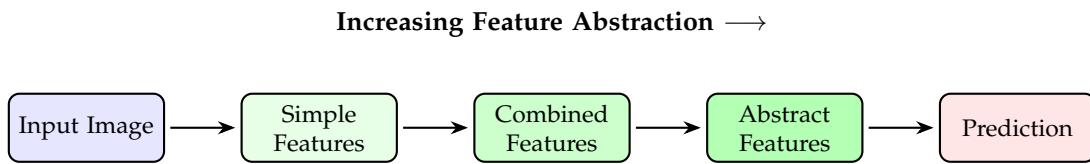


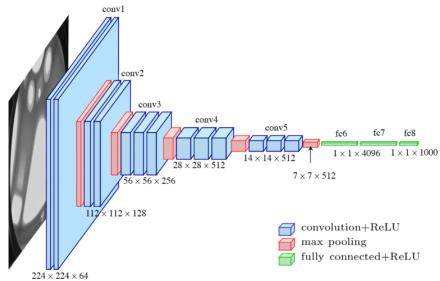
Figure 1.1: Hierarchical feature learning in a deep network.

### 1.1.2 Convolutional Neural Networks

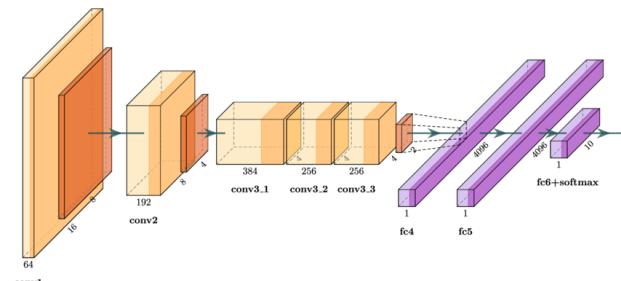
CNNs are a specialized class of neural networks designed for processing grid-like data, such as images (LeCun, Bengio, et al. 2015). Their architecture is inspired by the human visual cortex and incorporates strong **inductive biases** that make them exceptionally effective for visual tasks:

- **Local Receptive Fields:** Neurons are connected only to small, local regions of the input, exploiting the spatial locality of features in images.
- **Parameter Sharing:** The same set of weights (a filter or kernel) is applied across all spatial locations of the input. This drastically reduces the number of learnable parameters and builds in **translation equi-variance** the ability to detect a feature regardless of where it appears in the image.
- **Spatial Sub-sampling:** Pooling layers progressively reduce the spatial dimensions of the data, creating representations that are more robust to small shifts and distortions.

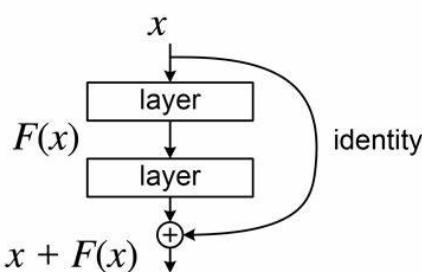
Notable CNN architectures that build upon these principles include LeNet-5 (LeCun, Bottou, et al. 1998), AlexNet (Krizhevsky et al. 2012), VGGNet (Simonyan et al. 2014), GoogLeNet (Szegedy, Liu, et al. 2015a), ResNet (He et al. 2016a), and U-Net (Ronneberger et al. 2015). More recently, Vision Transformers (Vision Transformers (ViTs)) (Dosovitskiy et al. 2020) have emerged as a powerful alternative, which uses attention mechanisms instead of convolutions.



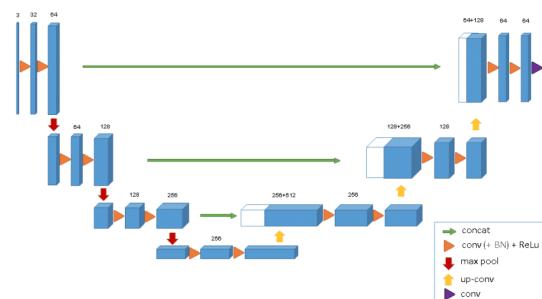
(a) LeNet-5 (LeCun, Bottou, et al. 1998)



(b) AlexNet (Krizhevsky et al. 2012)



(c) ResNet (Residual Block) (He et al. 2016a)



(d) U-Net (Ronneberger et al. 2015)

Figure 1.2: Some notable CNN architectures.

## 1.2 CNN Architecture Components

A modern CNN is constructed by composing a sequence of fundamental layers.

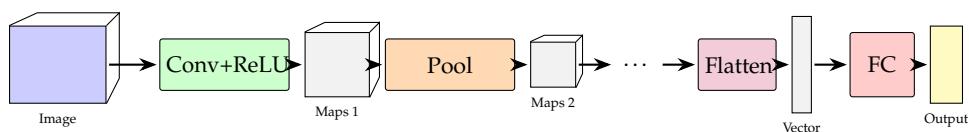


Figure 1.3: Basic building blocks of a CNN

### Convolutional Layers

This is the primary building block of a CNN, it works by sliding a small, learnable filter over an input volume. At each position, it computes a dot product between the filter's weights and the corresponding input patch. This process, illustrated in Figure 1.4, generates a feature

map that highlights specific patterns such as a vertical edge or a particular texture detected by the filter.

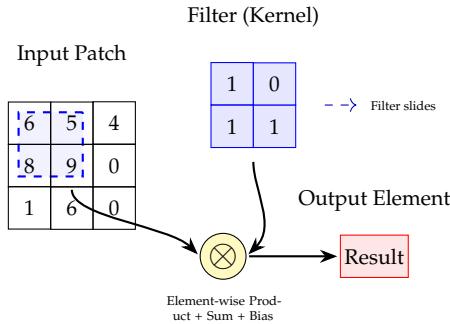


Figure 1.4: Principle of the convolution operation.

## Pooling Layers

Pooling layers perform non-linear down-sampling to reduce the spatial dimensions of feature maps. This decreases computational load, reduces the number of parameters, and provides a degree of translation invariance. The two most common forms are:

- **Max Pooling:** Selects the maximum element from a local region  $R$ :

$$\text{MaxOut}(R) = \max_{(p,q) \in R} (a_{pq})$$

- **Average Pooling:** Computes the average of elements in region  $R$ :

$$\text{AvgOut}(R) = \frac{1}{|R|} \sum_{(p,q) \in R} a_{pq}$$

Pooling is usually applied with a fixed window size (e.g., 2x2) and stride (e.g., 2).

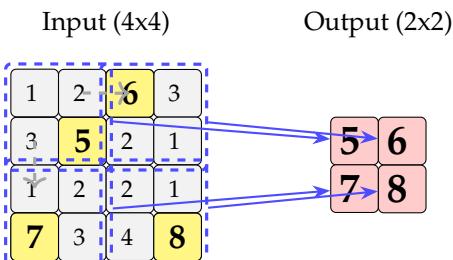


Figure 1.5: Max pooling with 2x2 windows and stride  $s=2$  from input to output.

## Activation Functions

An activation function  $\sigma(\cdot)$  is applied element-wise to the output of a linear operation (like convolution or a fully-connected layer 1.2). It introduces non-linearity, which is critical for the network to learn complex mappings. The choice of activation function significantly impacts model performance and training dynamics. Common choices include the Sigmoid, Hyperbolic Tangent (Tanh), Rectified Linear Unit (Rectified Linear Unit (ReLU)), and Leaky ReLU functions, each with distinct properties.

## Fully Connected Layers and Output Layers

After several stages of convolution and pooling, the high-level feature maps are flattened into a vector and passed to one or more Fully Connected (FC) layers. In a FC layer, every input neuron is connected to every output neuron. The final FC layer produces the model's output logits,  $z$ . A final activation function transforms these logits into a desired output format, such as class probabilities using the Softmax function for multi-class classification:

$$P(y = j|x) = \text{Softmax}(z_j) = \frac{e^{z_j}}{\sum_{k=1}^C e^{z_k}}$$

Here,  $z_j$  is the logit for class  $j$  out of  $C$  total classes.

## Supervised Learning in CNNs

CNNs are most commonly trained through supervised learning. Given a dataset of labeled examples  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ , where  $x_i$  is an input image and  $y_i$  is its corresponding label, the goal is to learn the network parameters  $\theta$  that best map inputs to outputs.

The training process is usually framed as an optimization problem where the objective is to find the set of parameters  $\theta^*$  that minimizes the **empirical risk**, the average loss over the training dataset, often with an added regularization term  $R(\theta)$ :

$$\theta^* = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(f(x_i; \theta), y_i) + \lambda R(\theta)$$

Here,  $\hat{y}_i = f(x_i; \theta)$  is the model's prediction for input  $x_i$ , and the **loss function**  $\mathcal{L}(\cdot, \cdot)$  quantifies the difference between the prediction  $\hat{y}_i$  and the true label  $y_i$ . Common loss functions include:

- **Cross-Entropy Loss:** For classification tasks. Binary Cross-Entropy (Binary Cross-Entropy (BCE)) for two-class problems:

$$\mathcal{L}_{BCE} = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

- **Mean Squared Error (MSE):** For regression tasks:

$$\mathcal{L}_{MSE} = \frac{1}{2}(\hat{y} - y)^2$$

- **Dice Loss:** Commonly used for segmentation tasks to measure overlap between predicted ( $P$ ) and ground truth ( $G$ ) masks:

$$\mathcal{L}_{Dice} = 1 - \frac{2|P \cap G|}{|P| + |G|}$$

## 1.3 Limitations of Training CNNs from Scratch

Although Convolutional Neural Networks (CNNs) have shown remarkable success, training them from random initialization poses significant challenges, especially in medical imaging.

### Data Scarcity

Deep models are **data-hungry**, requiring large, diverse annotated datasets. In medical domains, acquiring such data is often impractical due to costly, time-intensive expert labeling, privacy regulations (e.g., HIPAA, GDPR), and the rarity of certain conditions, which also induces severe class imbalance. Consequently, many medical datasets are orders of magnitude smaller than standard computer vision benchmarks, making training from scratch unreliable.

### Overfitting and Limited Generalization

High-capacity models trained on small datasets are prone to **overfitting**, memorizing training samples rather than learning generalizable patterns. Techniques like regularization (L1/L2 (Hoerl et al. 1970; Tibshirani 1996)), Dropout (Srivastava et al. 2014), and Batch Normalization (Ioffe et al. 2015) help but cannot fully compensate for scarce data.

### Domain Shift

Supervised learning assumes  $P_{train}(X, Y) = P_{test}(X, Y)$ . In clinical practice, this rarely holds (**domain shift** (Kouw et al. 2021)) due to differences in imaging devices, acquisition protocols, sample preparation, and patient populations. Domain sensitivity limits the deployability of models trained purely on local datasets.

### Class Imbalance

Medical datasets often exhibit extreme imbalance, where rare but clinically critical cases are vastly outnumbered. Standard losses bias training toward majority classes, degrading minority-class performance. Remedies include weighted losses (e.g., Focal Loss (Lin, Goyal, et al. 2017)), resampling (e.g., SMOTE (Chawla et al. 2002)), and robust metrics (e.g., Area Under the Curves (AUCs), Receiver Operating Characteristic (ROC) (Fawcett 2006)).

These limitations—scarce data, overfitting, domain shift, and class imbalance—make training CNNs from scratch impractical in real-world medical applications like malaria detection. This motivates advanced paradigms designed to enhance robustness and generalization, notably:

- **Transfer Learning (TL):** Leveraging knowledge from large-scale datasets to improve performance with limited labeled data (Chapter 1.4).
- **Multitask Learning (MTL):** Learning related tasks simultaneously to improve efficiency and generalization (Chapter 1.5).

## 1.4 Transfer Learning Paradigm

Transfer Learning (TL) is a paradigm where knowledge learned in a source domain ( $D_S, T_S$ ) is transferred to a target domain ( $D_T, T_T$ ) to improve performance on a target task. This approach is particularly useful in medical imaging where large annotated datasets are difficult to obtain due to privacy concerns and the high cost of expert labeling. General benchmarks like ImageNet (Deng et al. 2009) provide millions of examples compared to medical data which is often limited. TL allows models to perform effectively in these low resource settings by reusing pre-learned features rather than training from scratch.

### 1.4.1 Definition and Typology

The most widely cited formal definition of TL is provided by Pan et al. (2010), who defined it in terms of a **domain** and a **task**:

- A **domain**  $D$  consists of a feature space  $\mathcal{X}$  and a marginal probability distribution  $P(X)$ , where  $X = \{x_1, \dots, x_n\} \in \mathcal{X}$ .
- A **task**  $T$  is composed of a label space  $\mathcal{Y}$  and a predictive function  $f(\cdot)$  learned from the training data.

Given a source domain  $D_S$  with task  $T_S$ , and a target domain  $D_T$  with task  $T_T$ , the goal of **transfer learning** is to improve the learning of the target predictive function  $f_T(\cdot)$  in  $D_T$  using knowledge from  $D_S$  and  $T_S$ , under the condition that  $D_S \neq D_T$  or  $T_S \neq T_T$ .

### Typology of Transfer Learning

TL can be categorized along several complementary dimensions. Zhuang et al. (Zhuang et al. 2020) proposed a more structured and modern classification of transfer learning methods. They categorized these methods along several complementary dimensions, reflecting the evolution of the field, especially with the rise of DL:

- **By label availability between source and target:**
  - *Inductive TL*: Target domain has labeled data. The source may or may not be labeled.
  - *Transductive TL*: Target domain is unlabeled, but labeled source data is available.
  - *Unsupervised TL*: Neither source nor target domains contain labels, and transfer relies on unsupervised representation learning.
- **By feature space consistency:**
  - *Homogeneous TL*: Source and target share the same feature space (e.g., RGB pixel space), though distributions differ. This is the most common case in medical imaging.

- *Heterogeneous TL*: Feature spaces differ (e.g., transferring between 2D fundus photography and 3D OCT scans). Specialized methods are required to align the modalities.

- **By what is transferred:**

- *Instance-based TL*: Source samples are reweighted or reused directly in the target training process.
- *Feature-based TL*: Shared or mapped feature representations are constructed to connect domains (e.g., domain-invariant embeddings).
- *Parameter-based TL*: Source trained parameters are reused as initialization or partially shared (e.g., fine-tuning CNN backbones).
- *Relational-based TL*: Structural knowledge such as inter-class relations or graph structures is transferred.

### 1.4.2 Common Transfer Learning Techniques in Deep Learning

Several practical techniques facilitate parameter-based and feature-based TL within DL.

#### Fine-tuning Pre-trained Models

Fine-tuning is the most common parameter-based TL technique. It adapts a model pre-trained on a large source dataset like Deng et al. (2009) for a target task by continuing the training process on the target data. The primary strategies, shown in Figure 1.6 are:

- **Full Fine-tuning**: All layers of the pre-trained network are updated using a small learning rate. This is suitable when the target dataset is sufficiently large and similar to the source.
- **Feature Extraction (Frozen Layers)**: The convolutional base of the pre-trained model is frozen, acting as a fixed feature extractor. Only a new, task-specific classifier head is trained. This is ideal for very small target datasets to prevent overfitting.
- **Partial Fine-tuning**: A hybrid approach where early layers (capturing generic features like edges) are frozen, while later layers (capturing more abstract features) are fine-tuned along with the new head. This balances feature reuse and task-specific adaptation.

#### Learning Rate Management

Effective fine-tuning requires careful learning rate management to avoid catastrophic forgetting of valuable pre-trained features. A common practice is using **differential learning rates**, where different parts of the network have different update speeds. Let  $\theta = \{\theta_1, \dots, \theta_L\}$  be the parameters of the model's layers. The update rule for layer  $i$  becomes:

$$\theta_i \leftarrow \theta_i - \eta_i \nabla_{\theta_i} \mathcal{L}$$

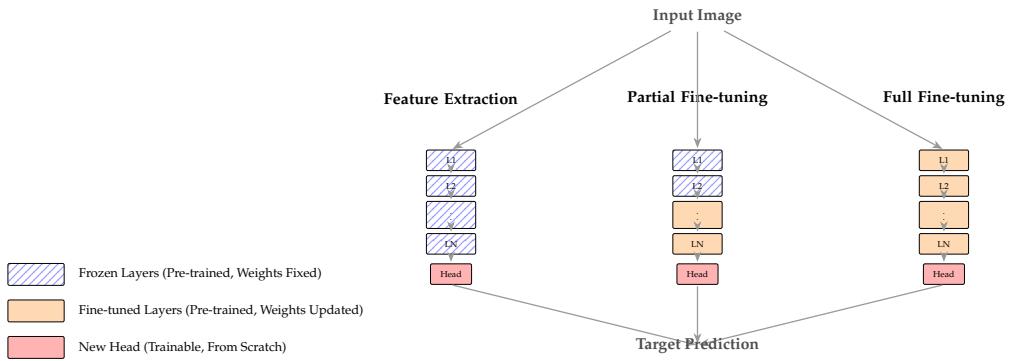


Figure 1.6: Conceptual overview of three common fine-tuning strategies.

where  $\eta_i$  is the learning rate for the  $i$ -th layer. Typically, earlier layers are assigned a smaller learning rate than later layers ( $\eta_i < \eta_j$  for  $i < j$ ), allowing the model to preserve general features while adapting task-specific ones more aggressively.

### Parameter-Efficient Fine-Tuning (PEFT)

Parameter-Efficient Fine-Tuning (PEFT) adapts large pretrained models by training only a small set of extra parameters while freezing the majority, enabling efficient transfer with minimal storage and compute (Houlsby et al. 2019). A key example is **adapter modules**, precisely **Low-Rank Adaptation (LoRA)** (Hu et al. 2021) which inserts trainable low-rank matrices into the weight update path. For an input  $\mathbf{z}$  (Fig. 3.5a):

$$\mathbf{z}' = \mathbf{z} + \frac{\alpha}{r} W_B W_A \mathbf{z}. \quad (1.3)$$

This keeps parameter growth minimal while maintaining strong performance.

### 1.4.3 General Benefits and Limitations of Transfer Learning

TL provides clear advantages for model development, particularly in data-scarce domains, but it also comes with important caveats that must be considered.

#### Key Benefits

- **Improved Performance with Limited Data:** Using pre-trained models allows achieving high performance on a target task with substantially fewer labeled samples than training from scratch, as the model already captures generalizable feature representations.
- **Faster Convergence:** Fine-tuning starts from a well-initialized weight set, reducing the number of epochs needed to reach optimal performance and speeding up training.
- **Better Generalization:** Features learned from large, diverse source datasets (e.g., ImageNet) often provide robust representations that act as a regularizer, lowering the risk of overfitting on small target datasets.

- **Access to Advanced Architectures:** Transfer learning enables the use of state-of-the-art architectures without the computational cost of training them from scratch.

### Limitations and Challenges

- **Negative Transfer:** If the source and target domains/tasks are too dissimilar, transferred knowledge can bias the model and degrade performance compared to training from scratch (Wang et al. 2019).
- **Domain Shift:** Differences in data distribution (e.g., natural images vs. medical microscopy) may limit transfer effectiveness, as source features might be irrelevant or misleading for the target task (Kouw et al. 2021).
- **Inherited Source Bias:** Any biases present in the source dataset such as demographic or equipment-related can propagate to the target model, reducing robustness to out-of-distribution samples.
- **Fine-tuning Complexity:** Selecting which layers to freeze, choosing learning rates, or deciding on partial vs. full fine-tuning often requires extensive experimentation (Raffel et al. 2020).

Overall, TL is a powerful tool for accelerating learning and improving generalization, but it must be applied carefully, with attention to domain alignment and task relevance.

## 1.5 Multitask Learning Paradigm

MTL is a paradigm in which a single model is trained to perform multiple related tasks simultaneously. The central idea, introduced in early work by Caruana (1997), is that learning tasks in parallel with a shared representation allows the model to exploit commonalities across tasks, improving generalization. This joint learning acts as a form of inductive transfer, providing regularization that reduces overfitting, particularly in data scarce settings (Ruder 2017).

### 1.5.1 Definition and Formulation

In an MTL setting, we consider  $K$  related tasks  $\{T_k\}_{k=1}^K$ , each with dataset  $\mathcal{D}_k = \{(x_i^{(k)}, y_i^{(k)})\}_{i=1}^{N_k}$  and loss  $\mathcal{L}_k$ . A model is designed with both shared parameters  $\theta_{sh}$  and task-specific parameters  $\{\theta_k\}_{k=1}^K$ . The overall objective is a weighted sum of task losses:

$$\mathcal{L}_{MTL}(\theta_{sh}, \{\theta_k\}_{k=1}^K) = \sum_{k=1}^K \lambda_k \mathcal{L}_k(f(x; \theta_{sh}, \theta_k), y_i^{(k)}) \quad (1.4)$$

where  $\lambda_k$  balances the contribution of each task. The challenge is to optimize both the parameters and the task weights such that knowledge sharing through  $\theta_{sh}$  promotes positive transfer rather than interference.

### 1.5.2 MTL Architectural Approaches

Architectural design is central to how MTL takes advantage of shared representations. The two most widely adopted approaches are **hard parameter sharing** and **soft parameter sharing**:

- **Hard Parameter Sharing:** A shared encoder (backbone) captures common features, followed by task-specific heads. This forces the backbone to learn representations useful across all tasks and provides strong regularization.
- **Soft Parameter Sharing:** Each task has its own parameters, but similarity between task models is encouraged through explicit regularization (e.g., penalizing large distances between parameter sets).

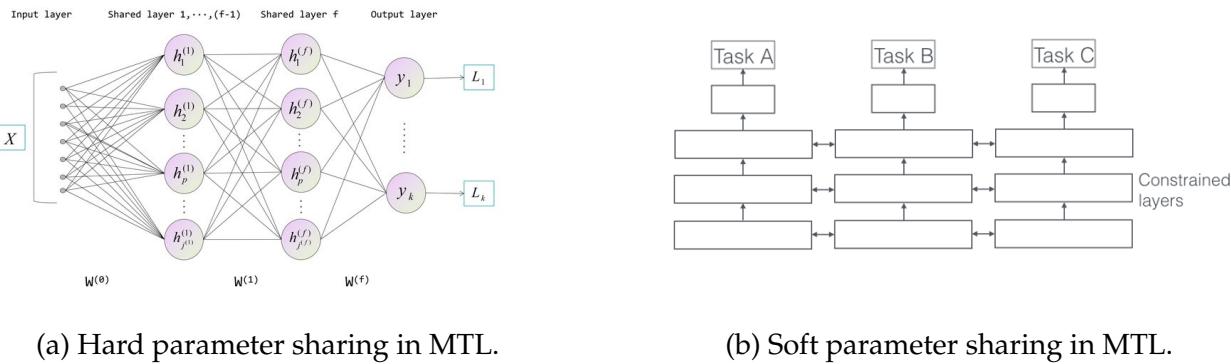


Figure 1.7: Main architectural approaches for MTL.

More advanced architectures mentioned by Ruder (2017), such as Deep Relationship Networks, Cross-stitch Networks, and Sluice Networks, offer more nuanced mechanisms for controlling knowledge sharing between tasks.

### 1.5.3 Optimization Challenges

Unlike Single-Task Learning (STL), MTL introduces unique training difficulties:

- **Task Balancing:** Uniform task weights ( $\lambda_k = 1$ ) often lead to dominance by tasks with larger losses. Dynamic schemes, such as uncertainty weighting (Kendall et al. 2018) or GradNorm (Chen et al. 2018), adaptively rescale contributions during training.
- **Negative Transfer and Gradient Conflicts:** Sharing across weakly related tasks may harm performance (Wang et al. 2019). Conflicting gradients from different tasks can pull shared parameters in opposite directions. Recent methods mitigate this by modifying or projecting gradients to reduce destructive interference (Yu et al. 2020).

### 1.5.4 General Benefits and Limitations of Multitask Learning

#### Key Benefits

- **Implicit Data Augmentation:** Joint training exposes the shared backbone to more varied data distributions, improving representation robustness.
- **Regularization:** Shared parameters constrain the model, reducing overfitting and improving generalization.
- **Feature Relevance:** Auxiliary tasks can guide attention toward informative features that a primary task might underutilize.
- **Efficiency:** A single MTL model supports multiple tasks in one forward pass, saving inference cost relative to training separate models.

#### Limitations and Challenges

- **Negative Transfer:** The main risk, arising when tasks are insufficiently related, degrading performance relative to single-task baselines.
- **Task Balancing Complexity:** Optimal weighting of  $\lambda_k$  is non-trivial, and poor choices can destabilize training.
- **Architecture Design:** Determining the right degree of sharing between tasks requires careful experimentation.
- **Dependence on Task Relatedness:** Benefits assume relatedness between tasks. If this assumption fails, MTL may be counterproductive.

## 1.6 Prior Work in Malaria Detection

This section reviews prior research in malaria blood smear analysis with a particular emphasis on advanced AI paradigms. We first discuss how transfer learning and related methods have been applied to malaria detection across different imaging modalities. We then examine the emerging but underexplored potential of Multi-Task Learning (MTL) and Multi-Task Transfer Learning (MTTL), which defines our own research focus.

### 1.6.1 Advanced AI Paradigms for Malaria Detection

Transfer learning (TL) has been the most widely adopted paradigm in malaria detection, largely due to its effectiveness in overcoming limited annotated datasets. Pre-trained convolutional neural networks (CNNs) such as VGG (Simonyan et al. 2014), ResNet (He et al. 2016b), DenseNet (G. Huang et al. 2017), Inception (Szegedy, Liu, et al. 2015b), and EfficientNet (Tan et al. 2019) appear frequently in the literature, with ImageNet as the primary source domain (Deng et al. 2009). Fine-tuning these models on malaria datasets has consistently yielded

## 1.6. PRIOR WORK IN MALARIA DETECTION

---

high accuracy in the narrow task of binary classification between parasitized and uninfected cells. Table 1.1 summarizes representative works, spanning thin blood smears, thick smears, and smartphone-based microscopy.

Table 1.1: Selected TL Applications in Medical Imaging and Malaria Detection

Study	Year	Task (Dataset/Type)	Method/Model	Key Result/Note
<b>2016</b>				
Gulshan et al. (2016)	2016	Diabetic Retinopathy Grading (Fundus Images)	InceptionV3 TL	Ophthalmologist-level Acc.
<b>2017</b>				
Rajpurkar et al. (2017)	2017	Pneumonia Det. (ChestX-ray14)	DenseNet-121 TL	Radiologist-level Perf.
Esteva et al. (2017)	2017	Skin Cancer Classif. (Dermoscopy)	InceptionV3 TL	Dermatologist-level Acc.
Bejnordi et al. (2017)	2017	Breast Cancer Metastasis Det. (CAMELYON16 Whole Slide Imaging (WSI))	GoogLeNet TL	High Acc. lymph node
<b>2018</b>				
Rajaraman et al. (2018)	2018	Malaria: Binary Cell Classif. (NIH, Thin Giemsa)	VGG, ResNet, etc. TL	ResNet-50: 95.9% Acc.
<b>2021</b>				
Abdurahman et al. (2021)	2021	Malaria: Parasite Det. (Makerere U., Thick N/S)	Modified YOLOv3/v4	YOLO for thick smears
Ufuktepe et al. (2021)	2021	Malaria: RBC Det. & Extraction (NLM Thin Smears)	Modified CFPNet-M (Distance Transform Regression)	92.2% Cell Count Acc.
<b>2022</b>				
Banerjee et al. (2022)	2022	Malaria: Binary Cell Classif. (NIH, Thin N/S)	Falcon DCNN, TL	Custom DCNN+TL; 95.2% Acc.
D. Li et al. (2022)	2022	Malaria: Binary Cell Classif. (NIH, Thin N/S)	Residual Attn. CNN + SVM	Hybrid CNN-SVM; 99.7% Acc.
Alonso-Ramírez et al. (2022)	2022	Malaria: Binary Cell Classif. (NLM NIH, Thin N/S)	CNN + LSTM/BiLSTM	CNN-RNN hybrid; 99.89% Acc.
Jameela et al. (2022)	2022	Malaria: Cell Classif. (NIH, Thin Smear)	ResNet50, ResNet34, VGG-16, VGG-19 TL/FT	Best VGG-19 Acc.
<b>2023</b>				
Sukumarran et al. (2023)	2023	Malaria Det. (MP-IDB/ Giemsa Thin)	YOLOv4, Faster R-CNN, SSD300	YOLOv4: Prec. 83%, Rec. 95%, F1 89%, mAP 93.87%
Khan et al. (2023)	2023	Malaria: Binary Cell Classif. (NLM NIH, Thin N/S)	Inception-ResNet TL	95% Acc.

*Continued on next page*

## 1.6. PRIOR WORK IN MALARIA DETECTION

**Table 1.1 – continued from previous page**

Study	Year	Task (Dataset/Type)	Method/Model	Key Result/Note
Omaer Faruq Goni et al. (2023)	2023	Malaria: Binary Cell Classif. (Kaggle NIH derived, Thin N/S)	Light CNN + DELM	CNN feat. + DELM; 99.66% Acc.
Silka et al. (2023)	2023	Malaria: Cell Det. (BBBC041, Thin Giemsa)	Custom Semantic Seg. CNN	99.68% Acc. (Det. via Seg.); 97.1% pixel Acc.
<b>2024</b>				
Hoyos-Villegas et al. (2024)	2024	Malaria: Parasite/Leukocyte Det. (NIH, Thin N/S)	YOLOv8	95% Acc. (Parasite)
Ramos et al. (2024)	2024	Malaria: Plasmodium vivax ROI Classif. (BBBC+FIOCRUZ, Thin Giemsa)	DenseNet201, MobileNetV2, VGG19, etc. TL	DenseNet201: 99.41% AUC
Ohdar et al. (2024)	2024	Malaria: Multi-Species Classif. (MP-IDB, Thin N/S)	Xception, EfficientNet, MobileNet, DenseNet121 Ensemble TL	98.9% Acc. (4 species)
Boit et al. (2024)	2024	Malaria: RBC Classif. (NIH, Thin Giemsa)	EDRI (Hybrid DL Model)	97.68% Acc.
<b>2025</b>				
Ramos-Briceño et al. (2025)	2025	Malaria: Species Classif. (NLM, Thick Giemsa)	Custom CNN (7-ch. input, preprocessing)	99.51% Acc. (P. falciparum, P. vivax, uninfected WBC)
Sora-Cardenas et al. (2025)	2025	Malaria: Parasite/Leukocyte Det. & Classif. (Thick Romanowsky)	SVM (Quality), OTSU (Leukocyte), Custom CNN (Parasite)	Parasite Det. F1: 82.10%; Stage Classif. F1: 83-88%
Mmileng et al. (2025)	2025	Malaria: Parasite Classif. (NLM Thin Smear)	ConvNeXt Tiny (V1/V2 mod.) TL/DA	ConvNeXt V2 Tiny: 98.1% Acc.
Nasra et al. (2025)	2025	Malaria: Parasite Det. (Lacuna, Thick/Thin)	Optimized CNN Architecture	Approx. 98% Acc.

**Note:** Gray background indicates malaria studies. Key datasets: NIH (Rajaraman et al. 2018), Lacuna (Lab et al. 2023) MP-IDB Loddo et al. 2019.

**Abbreviations:** Acc.: Accuracy; Attn.: Attention; Classif.: Classification; CNN: Convolutional Neural Network; Det.: Detection; F1: F1-Score; FT: Fine-Tuning; mAP: Mean Average Precision; N/S: Not Specified; NLM: National Library of Medicine; Perf.: Performance; Prec.: Precision; RBC: Red Blood Cell; Rec.: Recall; ROI: Region of Interest; Seg.: Segmentation; TL: Transfer Learning; WBC: White Blood Cell; WSI: Whole Slide Imaging.

Beyond simple fine-tuning, researchers have expanded into ensemble strategies (Nayak et al. 2022; Bibin et al. 2017), hybrid pipelines merging CNN features with classifiers like SVMs (D. Li et al. 2022), and object detection using YOLO variants (Abdurahman et al. 2021; Hoyos-Villegas et al. 2024). Given the scarcity of annotated samples, data augmentation is also standard practice. Despite these methodological advances, recurring limitations persists. Models outputs are partial and not enough for a full diagnosis in clinical settings, also models that perform well on benchmarks often degrade when faced with variations in clinics,

staining protocols, or imaging devices. Additionally, inconsistent reporting of experimental details, such as smear types and dataset curation, often complicates reproducibility and the assessment of true robustness.

While transfer learning has built a strong foundation for automated classification, the focus has largely remained on the binary distinction between infected and uninfected cells. A complete clinical diagnosis usually requires broader insights, including species identification, life-stage classification, and parasitemia estimation. Since standard TL frameworks typically optimize for single-task objectives, there is a clear opportunity to explore methods that better align with the multi-faceted nature of malaria diagnosis.

### 1.6.2 Multitask and Multitask Transfer Learning Relevance

In contrast to single-task paradigms, multitask learning (MTL) and its extension, multitask transfer learning (MTTL), provide a framework for jointly addressing multiple outputs that share visual structure. Tasks such as distinguishing *Plasmodium* species, identifying parasite life stages, and estimating parasitemia density are interrelated. A shared backbone guided by pre-trained weights could capture these correlations more effectively than training separate models. Despite this promise, no systematic application of MTL or MTTL to malaria imaging has yet been reported.

Outside the malaria domain, MTL and MTTL have shown substantial utility in medical imaging tasks. For example, Yadav et al. (2024) explored multitask transfer learning in an active learning pipeline for medical image segmentation, demonstrating improved annotation efficiency. H. Huang et al. (2022) employed a multi-task framework for organ segmentation and anatomical prior prediction, achieving robust semi-supervised learning performance. Similarly, Kim et al. (2023) introduced a cross-task attention mechanism to enhance multi-task learning for diverse medical imaging applications. Chandani et al. (2025) and Dheer et al. (2025) investigated MTTL strategies for segmentation tasks, showing that shared feature representations improve accuracy across related tasks. Schoenpflug et al. (2023) applied multitask learning to simultaneously segment tissues and detect tumors in colorectal cancer histology slides, while W. Li et al. (2023) leveraged MTL to classify histologic subtypes and tumor grade in non-small cell lung cancer CT scans. Even in robotic surgery, Islam et al. (2021) developed a spatio-temporal MTL model to jointly predict instrument tracking paths and surgeon scanpaths. Beyond medical vision, MTTL has also been applied to non-medical domains such as natural language processing and multi-view autonomous driving perception (Zhang et al. 2019; Raffel et al. 2020), demonstrating both efficiency and robustness benefits.

These studies collectively illustrate that multitask and multitask transfer paradigms can capture shared structure across related tasks, improving generalization and reducing the need for separate models an opportunity that remains largely unexplored in malaria diagnosis.

### 1.6.3 Observed Gaps in the Literature

While research has advanced, significant gaps remain that limit clinical utility. These issues become clear when viewing malaria diagnosis as a complex, multi-step process rather than a simple classification task. Key limitations include:

- **Focus on binary outputs:** Many studies concentrate primarily on distinguishing parasitized from uninfected cells. While useful, this binary output does not provide the complete diagnostic profile, such as species identification or parasitemia estimation, which is required for effective treatment decisions.
- **Absence of multitask frameworks:** To date, there has been no exploration of MTL or MTTL for malaria detection, despite how well these approaches align with the natural workflow of microscopic diagnosis.
- **Theoretical limitations:** Current literature lacks a formalized framework that adapts multitask transfer principles specifically for malaria or similar diagnostic challenges.
- **Reporting inconsistencies:** Reproducibility is often difficult because studies frequently omit key details regarding smear types, staining protocols, or preprocessing steps.
- **Dataset constraints:** Public datasets are often small and homogeneous, which limits how well models generalize across different clinics or imaging equipment.

Prior research highlights the potential of transfer learning but often lacks a holistic approach. The limited scope of existing studies, combined with the absence of multitask strategies, points to a clear need for more comprehensive methods. This gap motivates our focus on developing a multitask transfer learning framework for malaria diagnosis.

## Multi-Task Transfer Learning Framework

---

This chapter presents the theoretical contribution of this work, a mathematical framework for Multi-Task Transfer Learning (MTTL) built on foundational concepts from Pan et al. (2010), Caruana (1997), and Ruder (2017). We first establish the necessary mathematical preliminaries. From there, we formulate Single-Task, Transfer, and Multi-Task Learning as distinct optimization problems. This leads to our proposed MTTL formulation, which integrates pre-trained knowledge and parameter-efficient fine-tuning into a joint adaptation process across multiple related tasks.

### 2.1 Mathematical Preliminaries and Notation

A **domain**  $\mathcal{D}$  is a tuple defining the feature space and its underlying data distribution:

$$\mathcal{D} = \{\mathcal{X}, P(\mathbf{x})\} \quad (2.1)$$

where  $\mathcal{X}$  is the input feature space and  $P(\mathbf{x})$  is the marginal probability distribution over  $\mathcal{X}$ . A **task**  $\mathcal{T}$  defines a predictive objective within a domain:

$$\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}, \quad \text{where } f : \mathcal{X} \rightarrow \mathcal{Y} \quad (2.2)$$

where  $\mathcal{Y}$  is the label space and  $f(\cdot)$  is the true, unknown target function.

A **model** is a parametric function  $h(\mathbf{x}; \theta)$  with parameters  $\theta \in \Theta$ , designed to approximate  $f(\cdot)$ . For modern deep architectures, the model is a composition of a feature-extracting **backbone**  $\phi(\cdot; \theta_\phi)$  and a task-specific **head**  $g(\cdot; \theta_g)$ :

$$h(\mathbf{x}; \theta) = g(\phi(\mathbf{x}; \theta_\phi); \theta_g), \quad \text{where } \theta = \{\theta_\phi, \theta_g\} \quad (2.3)$$

Given a dataset  $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$  drawn from a joint distribution  $P(\mathbf{x}, y)$ , learning in a supervised setting is framed as finding the optimal parameters  $\theta^*$  that minimize the **empirical risk** over a loss function  $\mathcal{L}(\cdot, \cdot)$ :

$$\theta^* = \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(h(\mathbf{x}_i; \theta), y_i) \quad (2.4)$$

Using the established notation, we now formally define the optimization objectives for STL,

TL, MTL, and MTTL.

In the standard Single-Task Learning (STL) paradigm, a model with randomly initialized parameters  $\theta = \{\theta_\phi, \theta_g\}$  is trained from scratch to learn a single task  $\mathcal{T}$  using its corresponding dataset  $D$ . The objective is a direct application of Equation 2.4:

$$(\theta_\phi^*, \theta_g^*) = \arg \min_{\theta_\phi, \theta_g} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(g(\phi(\mathbf{x}_i; \theta_\phi); \theta_g), y_i) \quad (2.5)$$

Transfer Learning (TL) uses knowledge acquired from a source task  $\mathcal{T}_S = \{\mathcal{Y}_S, f_S(\cdot)\}$  on a source domain  $\mathcal{D}_S = \{\mathcal{X}_S, P_S(\mathbf{x})\}$  to a target task  $\mathcal{T}_T = \{\mathcal{Y}_T, f_T(\cdot)\}$  on a target domain  $\mathcal{D}_T = \{\mathcal{X}_T, P_T(\mathbf{x})\}$ . A pre-trained backbone  $\phi(\cdot; \theta_\phi^*)$  from  $\mathcal{T}_S$  is used. A new head  $g_T(\cdot; \theta_{g_T})$  is trained for  $\mathcal{T}_T$ .

$$\theta_T^* = \arg \min_{\theta_T} \frac{1}{N_T} \sum_{i=1}^{N_T} \mathcal{L}_T(g_T(\phi(\mathbf{x}_i; \theta_\phi^*); \theta_{g_T}), y_i) \quad (2.6)$$

where  $\theta_T = \{\theta_\phi^*, \theta_{g_T}\}$  and parameters  $\theta_\phi^*$  may be frozen or fine-tuned.

Multi-Task Learning (MTL) on the other hand trains a single model for multiple related tasks  $\{\mathcal{T}_k = \{\mathcal{Y}_k, f_k(\cdot)\}\}_{k=1}^M$  on a shared domain  $\mathcal{D} = \{\mathcal{X}, P(\mathbf{x})\}$ . It employs a shared backbone  $\phi(\cdot; \theta_\phi)$  and task-specific heads  $\{g_k(\cdot; \theta_{g_k})\}_{k=1}^M$ . The model  $h(\mathbf{x}; \theta)$  outputs for all the tasks  $\{g_1(\phi(\mathbf{x}; \theta_\phi); \theta_{g_1}), \dots, g_M(\phi(\mathbf{x}; \theta_\phi); \theta_{g_M})\}$ .

$$\theta^* = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^M \alpha_k \mathcal{L}_k(g_k(\phi(\mathbf{x}_i; \theta_\phi); \theta_{g_k}), y_{i,k}) \quad (2.7)$$

where  $\theta = \{\theta_\phi, \{\theta_{g_k}\}_{k=1}^M\}$  and  $\alpha_k$  are task weighting coefficients. All parameters in  $\theta$  are typically trainable.

## 2.2 MTTL Definition and Typology

By combining the principles of Transfer Learning (TL) and Multi-Task Learning (MTL), we describe Multi-Task Transfer Learning (MTTL) as a unified strategy that combines the strengths of both. The idea is to use shared representations learned from pre-trained models and train multiple related tasks together. We define MTTL as follows:

**Formal Definition:** **Multi-Task Transfer Learning (MTTL)** is a learning paradigm designed to solve a set of  $M$  target tasks  $\{\mathcal{T}_k\}_{k=1}^M$  by optimizing a joint objective function that is conditioned on the parameters  $\theta_{\phi_S}^*$  of a backbone model pre-trained on a source task  $\mathcal{T}_S$ . A specific MTTL method is an instantiation of this paradigm, defined by its **Typology**, a selected **Fine-Tuning Strategy**, and a **Task-Balancing Strategy**.

**Typology** MTTL can be categorized along multiple axes, we propose a typology based on the flow of knowledge and tasks, which is particularly relevant to medical imaging:

- **Source-to-Target Cardinality:** This describes the relationship between the number of source and target tasks. Common settings include *One-to-Many* (e.g., ImageNet pre-training for multiple diagnostic tasks), *Many-to-Many*, and *Many-to-One*. Our framework is formulated in the *One-to-Many* setting.
- **Fine-Tuning Formulation:** This specifies how the pre-trained knowledge is utilized and adapted during the multi-task optimization phase. Key strategies, which will be formalized in Section 2.3.1, include Frozen Backbone, Full Fine-Tuning, and Parameter-Efficient Fine-Tuning (PEFT).

## 2.3 The MTTL Problem Formulation

We formulate our MTTL problem in a *One-to-Many* setting, where a single source task  $\mathcal{T}_S$  provides a pre-trained backbone, which is then fine-tuned on  $K$  target tasks  $\{\mathcal{T}_k\}_{k=1}^K$ .

**Stage 1: Source Representation Learning.** Let  $h_S(\cdot; \theta_S) = g_S(\phi(\cdot; \theta_\phi); \theta_{g_S})$  be the source model. We obtain the optimal source backbone parameters  $\theta_\phi^*$  by solving the standard STL problem on the source data:

$$\theta_\phi^* = \left( \arg \min_{\theta_\phi, \theta_{g_S}} \hat{R}_S(\theta_\phi, \theta_{g_S}) \right)_{\theta_\phi} \quad (2.8)$$

where  $(\cdot)_{\theta_\phi}$  denotes projection onto the parameter space of the backbone.

**Stage 2: Target Multi-Task Optimization.** We transfer the backbone  $\phi(\cdot; \theta_\phi^*)$  and attach  $K$  new heads  $\{g_k(\cdot; \theta_{g_k})\}$ . The foundation of our framework is the following master objective function for the target stage:

$$\min_{\Theta_{\text{train}}} \mathcal{J}(\Theta_{\text{train}}, \{\lambda_k\}) = \sum_{k=1}^K \lambda_k \hat{R}_k(\Theta_{\text{train}}) + \Omega_{\text{reg}}(\Theta_{\text{train}}) \quad (2.9)$$

Finally, to define a concrete learning problem or in other words to formalize the learning objective, we specify two key components the **Fine-Tuning Strategy** to determine the set of trainable parameters, regularization scope for the shared backbone ( $\Theta_{\text{train}}$ ,  $\Omega_{\text{reg}}$ ) and the **Task-Balancing Strategy** ( $\{\lambda_k\}$ ), which controls the relative weighting of individual task losses.

### 2.3.1 Derivations by Fine-Tuning Strategy

The fine-tuning strategy defines the set of trainable parameters  $\Theta_{\text{train}}$  and the regularization term  $\Omega_{\text{reg}}$  in (2.9). This choice specifies which parameters are updated and how they are constrained.

**Formulation 1: Frozen Backbone (Feature Extraction).** The backbone  $\phi(\cdot; \theta_\phi^*)$  is used as a fixed feature extractor. The trainable parameters consist only of the new heads.

$$\begin{aligned} \text{Let } \Theta_{\text{train}} &:= \{\theta_{g_k}\}_{k=1}^M \quad \text{and} \quad \Omega_{\text{reg}} := 0 \\ \implies \mathcal{J} &= \sum_{k=1}^M \lambda_k \hat{R}_k(\theta_\phi^*, \theta_{g_k}) \end{aligned}$$

**Formulation 2: Full Fine-Tuning.** All layers of the pre-trained backbone are adapted. We define this as learning a dense update matrix  $\Delta\theta_\phi$  for the initial backbone weights  $\theta_\phi^*$ .

$$\begin{aligned} \text{Let } \Theta_{\text{train}} &:= \{\Delta\theta_\phi, \{\theta_{g_k}\}_{k=1}^M\} \quad \text{and} \quad \Omega_{\text{reg}} := \gamma \|\Delta\theta_\phi\|_2^2 \\ \implies \mathcal{J} &= \sum_{k=1}^M \lambda_k \hat{R}_k(\theta_\phi^* + \Delta\theta_\phi, \theta_{g_k}) + \gamma \|\Delta\theta_\phi\|_2^2 \end{aligned}$$

**Formulation 3: Partial Fine-Tuning.** A subset of the backbone layers, typically the deeper, more task-specific ones, are unfrozen and fine-tuned. Let the backbone parameters be partitioned into a frozen set  $\theta_{\phi,\text{frozen}}^*$  and a trainable set  $\theta_{\phi,\text{tune}}^*$ . We learn an update  $\Delta\theta_{\phi,\text{tune}}$ .

$$\begin{aligned} \text{Let } \Theta_{\text{train}} &:= \{\Delta\theta_{\phi,\text{tune}}, \{\theta_{g_k}\}_{k=1}^M\} \quad \text{and} \quad \Omega_{\text{reg}} := \gamma \|\Delta\theta_{\phi,\text{tune}}\|_2^2 \\ \implies \mathcal{J} &= \sum_{k=1}^M \lambda_k \hat{R}_k(\{\theta_{\phi,\text{frozen}}^*, \theta_{\phi,\text{tune}}^* + \Delta\theta_{\phi,\text{tune}}\}, \theta_{g_k}) + \gamma \|\Delta\theta_{\phi,\text{tune}}\|_2^2 \end{aligned}$$

**Formulation 4: Parameter-Efficient Fine-Tuning (PEFT).** This approach, exemplified by LoRA (Hu et al. 2021), keeps the entire backbone  $\theta_\phi^*$  frozen but injects a small set of new, trainable parameters  $\theta_\psi$  (the adapters).

$$\begin{aligned} \text{Let } \Theta_{\text{train}} &:= \{\theta_\psi, \{\theta_{g_k}\}_{k=1}^M\} \quad \text{and} \quad \Omega_{\text{reg}} := 0 \\ \implies \mathcal{J} &= \sum_{k=1}^M \lambda_k \hat{R}_k(\phi_{\text{adapted}}(\cdot; \theta_\phi^*, \theta_\psi), \theta_{g_k}) \end{aligned}$$

**Formulation 5: Hybrid Tuning (Our Approach).** This strategy, central to our experimental investigation, combines Partial Fine-Tuning with PEFT. A subset of the backbone layers is fully fine-tuned while the remaining frozen layers are adapted using PEFT.

$$\begin{aligned} \text{Let } \Theta_{\text{train}} &:= \{\Delta\theta_{\phi,\text{tune}}, \theta_\psi, \{\theta_{g_k}\}_{k=1}^M\} \quad \text{and} \quad \Omega_{\text{reg}} := \gamma \|\Delta\theta_{\phi,\text{tune}}\|_2^2 \\ \implies \mathcal{J} &= \sum_{k=1}^M \lambda_k \hat{R}_k(\phi_{\text{adapted}}(\cdot; \{\theta_{\phi,\text{frozen}}^*, \theta_{\phi,\text{tune}}^* + \Delta\theta_{\phi,\text{tune}}\}, \theta_\psi), \theta_{g_k}) + \gamma \|\Delta\theta_{\phi,\text{tune}}\|_2^2 \end{aligned}$$

The choice of which backbone layers to unfreeze is critical. Based on established principles of transfer learning (Yosinski et al. 2014), earlier layers of a CNN learn generic, low-level features (e.g., edges, textures), while later layers learn more abstract, task-specific features.

To preserve the generic features while still tuning the high-level representations to our specific domain, we chose to unfreeze only the final residual block (layer4) of the ResNet-50 backbone. The remaining, earlier layers were kept frozen but were adapted using LoRA. This hybrid approach aims to achieve the best of both worlds deep feature adaptation with maximal parameter efficiency.

### 2.3.2 Derivations by Task-Balancing Strategy

This component defines the task weights  $\{\lambda_k\}$  and can be applied to any of the fine-tuning formulations above.

**Formulation A: Manual Weighting.** The weights are set as fixed hyperparameters,  $\lambda_k = c_k$ . A common constraint is to enforce a convex combination such that

$$\lambda_k \geq 0 \quad \text{and} \quad \sum_{k=1}^M \lambda_k = 1$$

**Formulation B: Uncertainty Weighting.** This approach re-frames the objective based on maximizing a multi-task Gaussian likelihood, where each task's uncertainty  $\sigma_k^2$  is a learnable parameter (Kendall et al. 2018). This corresponds to setting  $\lambda_k = (2\sigma_k^2)^{-1}$  and adding an uncertainty regularization term. The master objective becomes a joint minimization over model and uncertainty parameters:

$$\min_{\Theta_{\text{train}}, \{\sigma_k\}} \quad \sum_{k=1}^M \left( \frac{1}{2\sigma_k^2} \hat{R}_k(\Theta_{\text{train}}) + \log \sigma_k \right) + \Omega_{\text{reg}}(\Theta_{\text{train}}) \quad (2.10)$$

**Formulation C: Gradient Normalization (GradNorm).** This method (Chen et al. 2018) dynamically tunes  $\{\lambda_k(t)\}$  at each training step  $t$  to equalize learning speeds across tasks. Let  $\theta_{sh} \subseteq \Theta_{\text{train}}$  be the trainable shared parameters. The weights  $\{\lambda_k\}$  are updated to minimize a gradient loss  $\mathcal{L}_{\text{grad}}(t, \{\lambda_k\})$  that measures the disparity in gradient magnitudes relative to each task's training rate.

## Methodological Application to Malaria Diagnosis

---

The previous chapter established the theoretical framework for Multi-Task Transfer Learning. In this chapter, we apply the defined framework to the practical challenge of automated malaria diagnosis from Giemsa stained thin blood smears. We begin by defining the specific MTTL objective function used in this study, including the selected fine-tuning and task balancing strategies. Then we describe the data curation and preprocessing steps required to ensure model stability. This is followed by an overview of the model architecture, which features a shared backbone adapted via Low Rank Adaptation (LoRA) and specialized decoder heads. Finally, we outline the experimental protocols designed to benchmark the system against Single-Task Learning (STL) baselines and evaluate its performance across different data subsets.

### 3.1 Instantiated MTTL Objective for Malaria Diagnosis

Our framework adopts a **One-to-Many** topology, using a single source model pre-trained on ImageNet to learn multiple tasks relevant to malaria diagnosis. To achieve this, we combine two specific strategies:

- **Fine-Tuning Strategy:** We employ both **Parameter-Efficient Fine-Tuning (PEFT)** using LoRA adapters and a **Hybrid Tuning** approach. This enables a comparison between highly efficient adaptation and a more comprehensive fine-tuning of deeper backbone layers to balance performance against computational cost.
- **Task-Balancing Strategy:** We utilize **Uncertainty Weighting** to manage the losses of heterogeneous tasks. This automated approach allows the model to dynamically learn the optimal contribution of each task during training, avoiding the difficulties of manual hyperparameter tuning.

These choices define the final optimization problem addressed in our experiments.

Let  $\theta_\phi^*$  represent the parameters of the pre-trained source backbone. By integrating the chosen fine-tuning strategy with Uncertainty Weighting, the objective function is expressed as:

$$\min_{\Theta_{\text{train}}, \{\theta_{g_k}\}, \{\sigma_k\}} \sum_{k=1}^M \left( \frac{1}{2\sigma_k^2} \hat{R}_k(\Theta_{\text{train}}, \theta_{g_k}) + \log \sigma_k \right) + \Omega_{\text{reg}}(\Theta_{\text{train}}) \quad (3.1)$$

where the empirical risk for task  $k$  is defined as:

$$\hat{R}_k(\Theta_{\text{train}}, \theta_{g_k}) = \frac{1}{N_k} \sum_{i=1}^{N_k} \mathcal{L}_k(g_k(\phi_{\text{adapted}}(\mathbf{x}_i; \theta_\phi^*, \Theta_{\text{train}}); \theta_{g_k}), y_{ik}). \quad (3.2)$$

Substituting (3.2) into (3.1) yields the explicit double-sum formulation:

$$\min_{\Theta_{\text{train}}, \{\theta_{g_k}\}, \{\sigma_k\}} \sum_{k=1}^M \left[ \frac{1}{2\sigma_k^2 N_k} \sum_{i=1}^{N_k} \mathcal{L}_k(g_k(\phi_{\text{adapted}}(\mathbf{x}_i; \theta_\phi^*, \Theta_{\text{train}}); \theta_{g_k}), y_{ik}) + \log \sigma_k \right] + \Omega_{\text{reg}}(\Theta_{\text{train}}) \quad (3.3)$$

### 3.1.1 Dataset

We utilize the publicly available **Malaria Dataset from the National Library of Medicine (NLM)** Medicine (NLM) 2018, curated by the Lister Hill National Center for Biomedical Communications (Rajaraman et al. 2018; Kassim et al. 2021). The dataset contains Giemsa-stained thin blood smear images from 193 patients (148 *P. falciparum*-infected and 45 healthy), captured using a smartphone attached to a light microscope. For this study, we focus on three primary annotations essential for diagnosis:

- **Parasitized:** RBCs containing one or more malaria parasites.
- **Uninfected:** Healthy red blood cells.
- **WBCs:** Leukocytes, a critical negative class that the system must distinguish from parasites.

As shown in Table 3.1, the data reflects the reality of clinical samples with severe **class imbalance**. Uninfected cells vastly outnumber parasitized ones, while white blood cells are extremely rare. This disparity motivates the loss-balancing and data-sampling strategies used in our experiments in order to fix this.

Table 3.1: Class Distribution of Annotated Cells in the Dataset.

Cell Class	Total Annotations
Uninfected	155,640
Parasitized	6,810
White Blood Cell	220

To ensure unbiased evaluation, we partitioned the dataset into training (80%), validation (10%), and test (10%) sets at the **patient level**. This prevents data leakage by guaranteeing that the model never encounters images from the same patient in both training and testing.

### 3.1.2 Image Preprocessing Pipeline

Microscopy images frequently vary in staining, illumination, and acquisition hardware. To mitigate these domain shifts, we designed a sequential preprocessing pipeline. Figure 3.1 displays raw samples, while Figure 3.2 shows the corresponding results after processing.

**Contrast Enhancement (CLAHE).** To highlight intracellular structures like ring-stage parasites, we applied **Contrast Limited Adaptive Histogram Equalization (CLAHE)** (Pizer et al. 1987). Images were converted to the LAB color space, and CLAHE was applied only to the Luminance channel to improve contrast while preserving color fidelity.

**Resizing.** All images were standardized to  $512 \times 512$  pixels using Lanczos interpolation. We favored this method over padding to ensure the entire image region contributes useful features, accepting a minor trade-off in aspect ratio preservation.

**Data Augmentation and Normalization.** We applied on-the-fly augmentations during training, including random flips,  $90^\circ$  rotations, color adjustments, and noise simulation. Finally, pixel intensities were normalized using standard ImageNet statistics to ensure compatibility with the pretrained backbone:

$$\mu = (0.485, 0.456, 0.406), \quad \sigma = (0.229, 0.224, 0.225). \quad (3.4)$$

Normalization was applied only during training and inference, not for the visualization grids (Figures 3.1 and 3.2).

## 3.2 Tasks Formulation

We formulate malaria diagnosis as a set of  $M \leq 4$  synergistic tasks, centered on Multi-Class Cell Detection. Each task is defined by its objective, label space, and a loss function.

- **Task 1: Multi-Class Cell Detection ( $\mathcal{T}_{\text{det}}$ ):** The primary task.
  - **Objective:** Localize every cell of interest with a bounding box and assign it a class label.
  - **Label Space ( $\mathcal{Y}_{\text{det}}$ ):**  $y_{\text{det}} = \{(\mathbf{b}_j, c_j)\}_{j=1}^{N_{\text{obj}}}$ , where  $\mathbf{b}_j \in \mathbb{R}^4$  are coordinates and  $c_j \in \{1, \dots, C\}$  is the class label.
  - **Loss Function ( $\mathcal{L}_{\text{det}}$ ):** Aggregates four equally weighted components from Faster R-CNN (Ren et al. 2017):

$$\mathcal{L}_{\text{det}} = \mathcal{L}_{\text{RPN\_cls}} + \mathcal{L}_{\text{RPN\_reg}} + \mathcal{L}_{\text{RoI\_cls}} + \mathcal{L}_{\text{RoI\_reg}} \quad (3.5)$$

The Region Proposal Network (RPN) uses Binary Cross-Entropy to distinguish objects from background. The final Region of Interest (RoI) classification employs

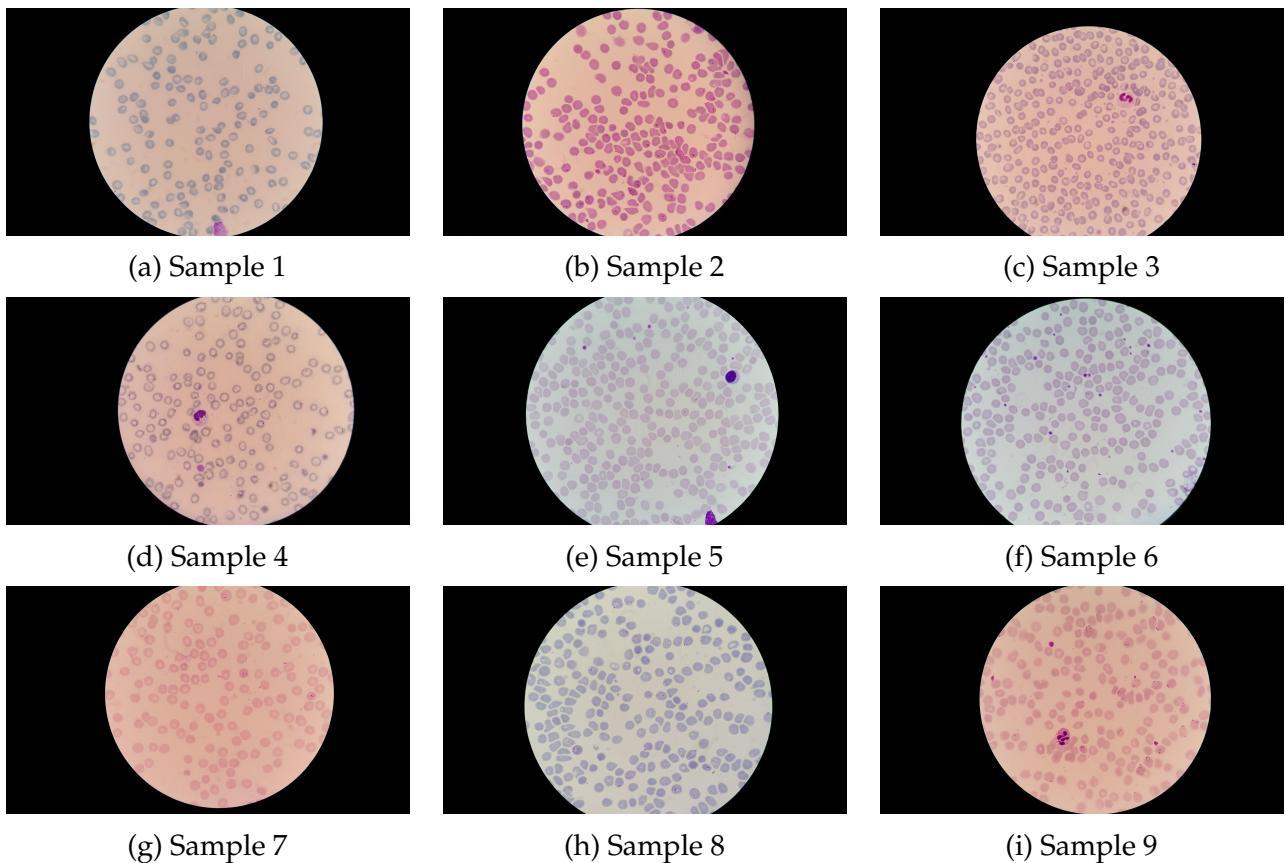


Figure 3.1: Original Samples from NLM Thin Falciparum dataset.

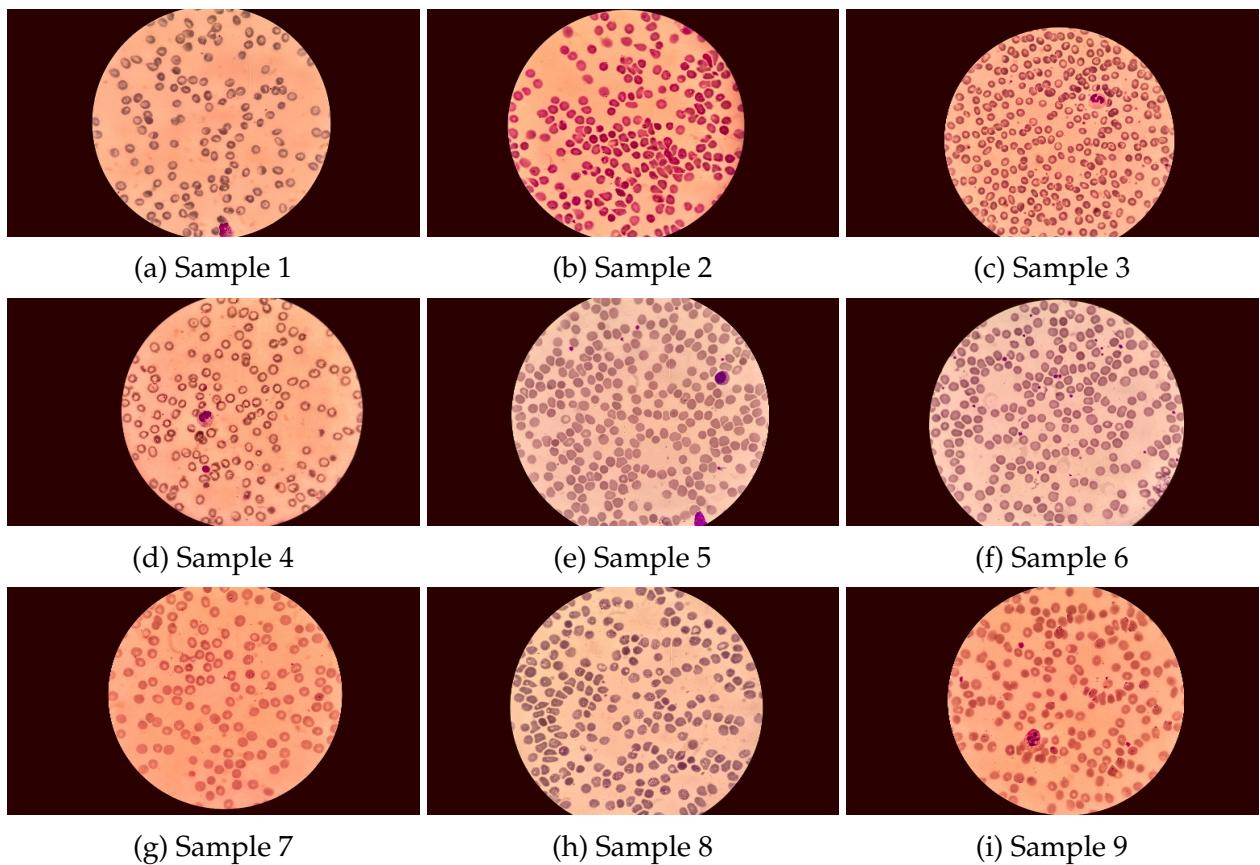


Figure 3.2: Preprocessed version of the images from Figure 3.1.

**Focal Loss** (Lin, Goyal, et al. 2017) ( $\alpha_t = 0.25, \gamma = 2.0$ ) to handle severe class imbalance. Bounding box regression uses the **Smooth L1 Loss** for stability:

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \quad (3.6)$$

- **Task 2: Cell Segmentation ( $\mathcal{T}_{\text{seg}}$ ):** An auxiliary dense prediction task providing shape priors.

- **Objective:** Generate a binary mask for all cells.
- **Label Space ( $\mathcal{Y}_{\text{seg}}$ ):** Binary mask  $\mathbf{M} \in \{0, 1\}^{64 \times 64}$ .
- **Loss Function ( $\mathcal{L}_{\text{seg}}$ ):** An equally weighted sum ( $w_{\text{bce}} = w_{\text{dice}} = 0.5$ ) of Binary Cross-Entropy and Dice Loss:

$$\mathcal{L}_{\text{seg}} = \frac{1}{2} \left( -\frac{1}{N} \sum_i [M_i \log \hat{M}_i + (1-M_i) \log(1-\hat{M}_i)] + 1 - \frac{2 \sum \hat{M}_i M_i + \epsilon}{\sum \hat{M}_i + \sum M_i + \epsilon} \right) \quad (3.7)$$

- **Task 3: Infection Localization ( $\mathcal{T}_{\text{loc}}$ ):** An auxiliary regression task serving as an attention prior.

- **Objective:** Produce a heatmap highlighting parasite centers.
- **Label Space ( $\mathcal{Y}_{\text{loc}}$ ):** Continuous heatmap  $\mathbf{H} \in [0, 1]^{64 \times 64}$ .
- **Loss Function ( $\mathcal{L}_{\text{loc}}$ ):** A composite loss designed for imbalance and boundary alignment inspired by Abraham et al. (2019):

$$\mathcal{L}_{\text{loc}} = 0.8 \cdot (1 - \text{TI})^\gamma + 0.2 \cdot \|\nabla \hat{\mathbf{H}} - \nabla \mathbf{H}\|_1$$

where  $\text{TI} = \frac{\sum \hat{H}_i H_i + \epsilon}{\sum \hat{H}_i H_i + \alpha \sum \hat{H}_i (1 - H_i) + \beta \sum (1 - \hat{H}_i) H_i + \epsilon}$  (3.8)

We use Focal Tversky Loss ( $\alpha = 0.7, \beta = 0.3, \gamma = 0.75$ ) to penalize false negatives, supplemented by a Boundary Loss on Sobel gradients.

- **Task 4: RoI Classification ( $\mathcal{T}_{\text{roi}}$ ):** Provides direct supervisory signals to the backbone.
- **Objective:** Classify pooled features from ground-truth RoIs.
- **Label Space ( $\mathcal{Y}_{\text{roi}}$ ):** Class label  $c \in \{1, \dots, C\}$ .
- **Loss Function ( $\mathcal{L}_{\text{roi}}$ ):** **Weighted Cross-Entropy Loss** to address imbalance:

$$\mathcal{L}_{\text{roi}}(\mathbf{z}, c) = -w_c \log \left( \frac{\exp(z_c)}{\sum_{j=1}^C \exp(z_j)} \right) \quad (3.9)$$

Weights  $w_c$  are set to [23.0, 1.0, 100.0] for {Infected, Healthy, WBC} respectively.

Figure 3.3: Visualization of Ground Truth Data for the Multi-Task Framework.

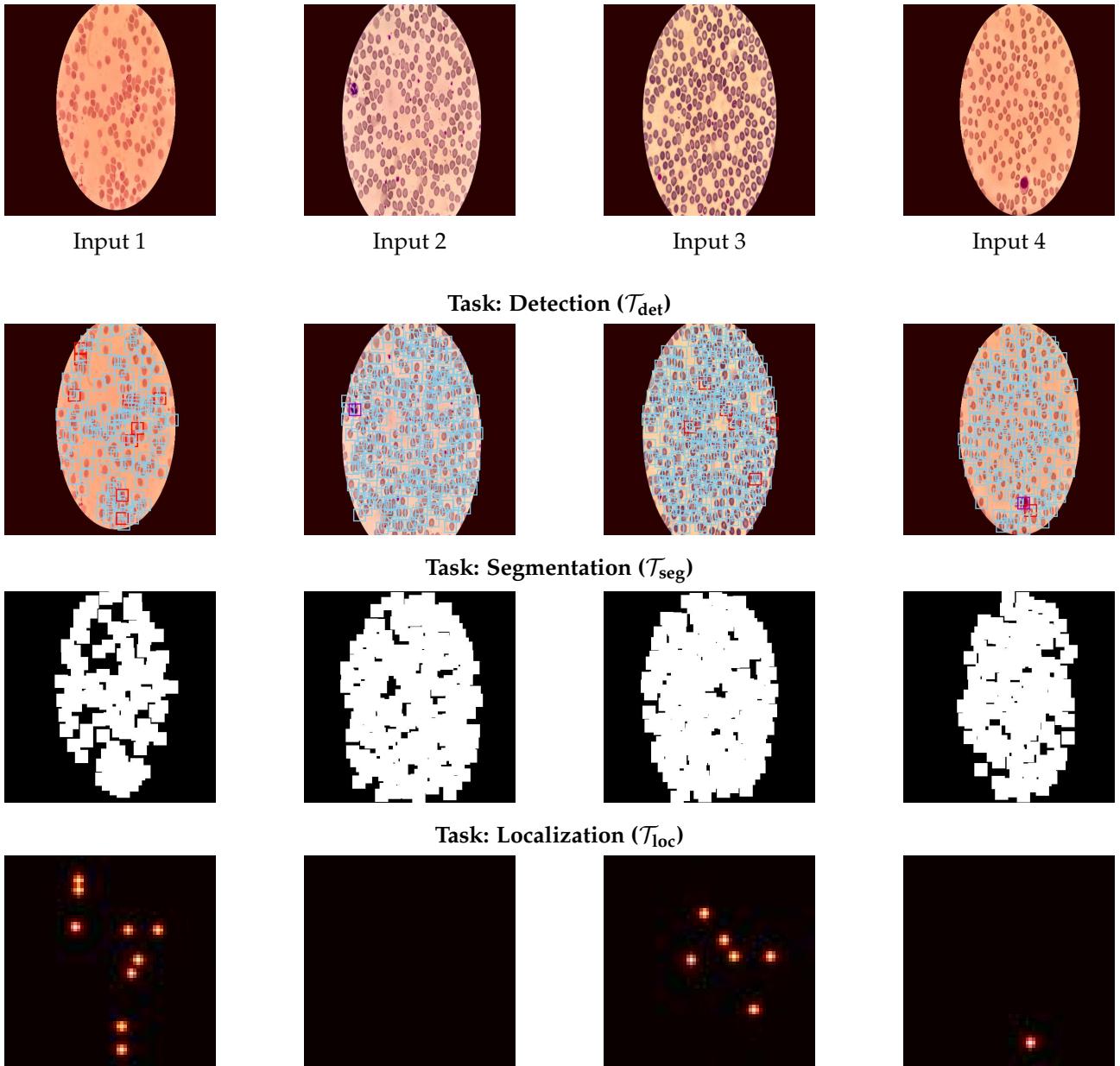


Figure 3.3 visualizes these tasks on a sample image. Detection identifies cell types (Infected: red, Healthy: blue, WBC: lime), while Segmentation outlines boundaries. Localization highlights parasite regions, and ROI Classification distinguishes cell features. Together, these tasks mirror the step-wise reasoning of a human expert.

### 3.3 MTTL Model Architecture

The architecture of our proposed system is a modular and flexible instantiation of the MTTL framework, designed for both rigorous experimentation and high performance. It adheres to the prevalent hard parameter sharing paradigm, comprising two primary parts a single, powerful shared backbone that learns a common feature representation, and a set of specialized, task-specific decoders (heads) that interpret these features.

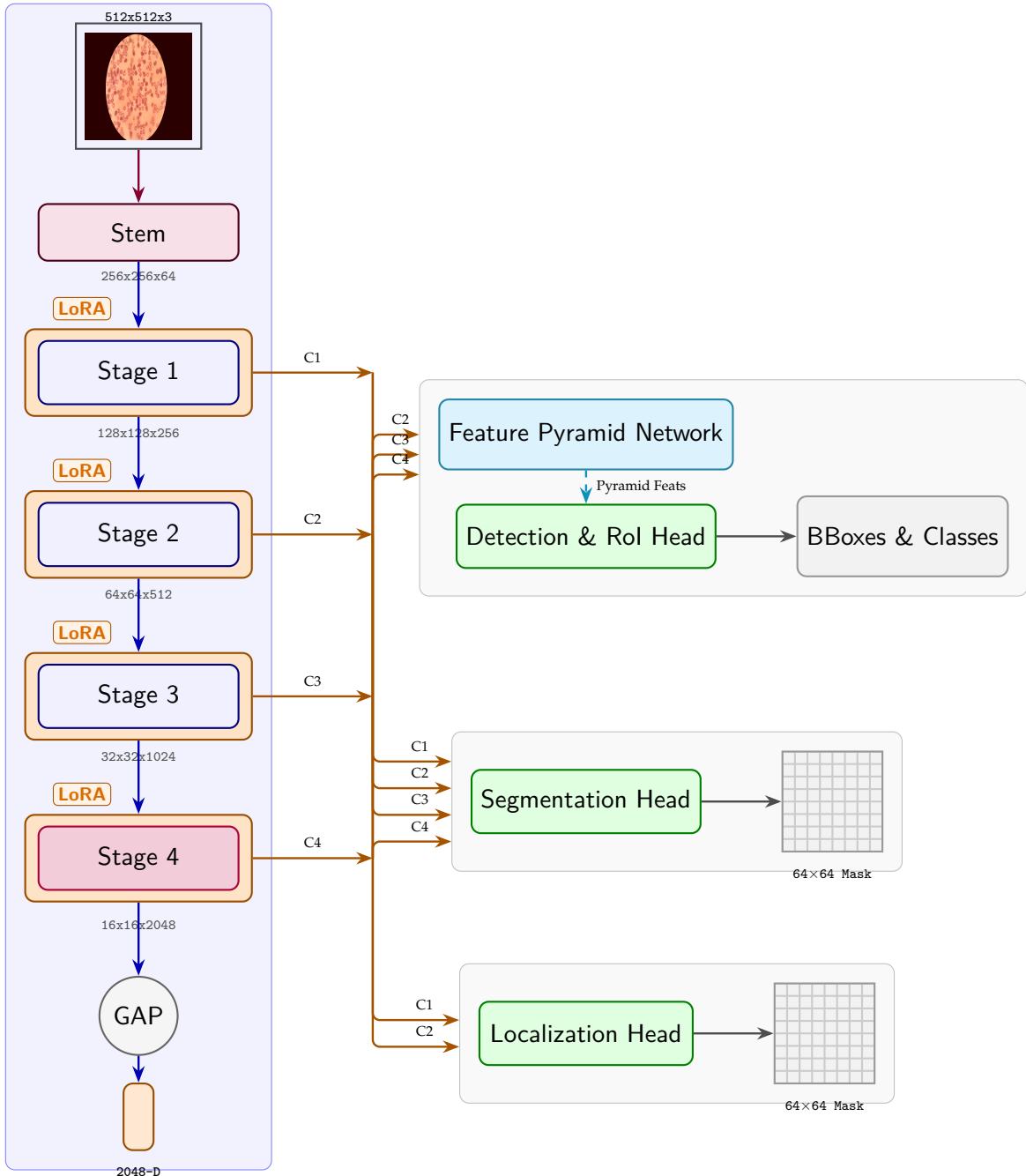


Figure 3.4: High-level diagram of our MTTL model architecture.

The design is shown conceptually in Figure 3.4, a  $512 \times 512 \times 3$  input image passes through the backbone stages (blue), with Stage 4 fine-tuned (purple) alongside Low-Rank Adaptations (LoRAs) adapters (orange). Multi-scale features C1 to C4 are extracted from the backbone and passed to task-specific heads: C2 to C4 feed the Feature Pyramid Network (FPN) for object detection, C1 to C4 are used for segmentation, and C1 to C2 support localization. This modular architecture enables efficient transfer learning through selective parameter updates while maintaining the representational capacity of the pretrained backbone. The design also facilitates exploration of different task combinations during training. Further component details are provided in the following sections.

### 3.3.1 Shared Backbone with Integrated LoRA Adapters

We use a **ResNet-50** backbone (He et al. 2016a) pre-trained on ImageNet. We selected this architecture because it captures rich features effectively while avoiding the vanishing gradient issues found in other deep networks. It is more efficient than older models like VGG (Simonyan et al. 2014) and offers a cleaner, more modular structure than Inception designs (Szegedy, Vanhoucke, et al. 2016). This modularity makes it ideal for integrating adaptation layers.

To adapt the model, we insert **Low-Rank Adaptation (LoRA)** modules into the residual blocks. As shown in Figure 3.5, these modules add small trainable matrices ( $W_A, W_B$ ) alongside the frozen layers. This forms our **Adapted ResNet Backbone**. During training, the original weights  $\theta_\phi^*$  remain frozen; only the LoRA parameters  $\theta_\psi$  are updated. This approach preserves the pre-trained knowledge while allowing the model to learn the specific requirements of malaria diagnosis. In our **Hybrid Tuning** strategy, we also unfreeze the final residual block (*layer4*). We fine-tune this layer with a very low learning rate to refine the most abstract features without disrupting the overall hierarchy.

### 3.3.2 Task-Specific Heads

The shared backbone produces multi-scale features, which are then directed into specialized decoder heads, each designed to solve a specific diagnostic task.

**Feature Pyramid Network (FPN).** To provide the detection and RoI heads with rich multi-scale information, we pass the backbone features through an Feature Pyramid Network (FPN) (Lin, Dollár, et al. 2017). This structure mixes coarse and fine features via a top-down pathway, ensuring the network operates with feature maps that are both semantically rich and spatially detailed.

**Detection Head.** For the primary detection task ( $\mathcal{T}_{\text{det}}$ ), we utilize a customized Faster R-CNN head (Figure 3.6) built on the FPN outputs. It comprises the Region Proposal Network (RPN), Multi-Scale RoI Align, and a box head. We replaced the standard, heavy MLP box head with a lighter design using separable convolutions. Early tests indicated the default head was prone to overfitting and computationally expensive during multi-task training. This lightweight version reduces parameters and maintain sufficient capacity to model fine cell structures.

**RoI Classification Head.** We initially explored two classification strategies. The first, cell-level classification, processed pre-cropped  $64 \times 64$  cells. However, early experiments showed this caused conflicting gradients with the detection task. We therefore discarded it in favor of Region of Interest (RoI) classification ( $\mathcal{T}_{\text{roi}}$ ). This approach extracts features from ground-truth boxes on the full image using RoI Align and feeds them into the shared `clf_head` (Figure 3.9).

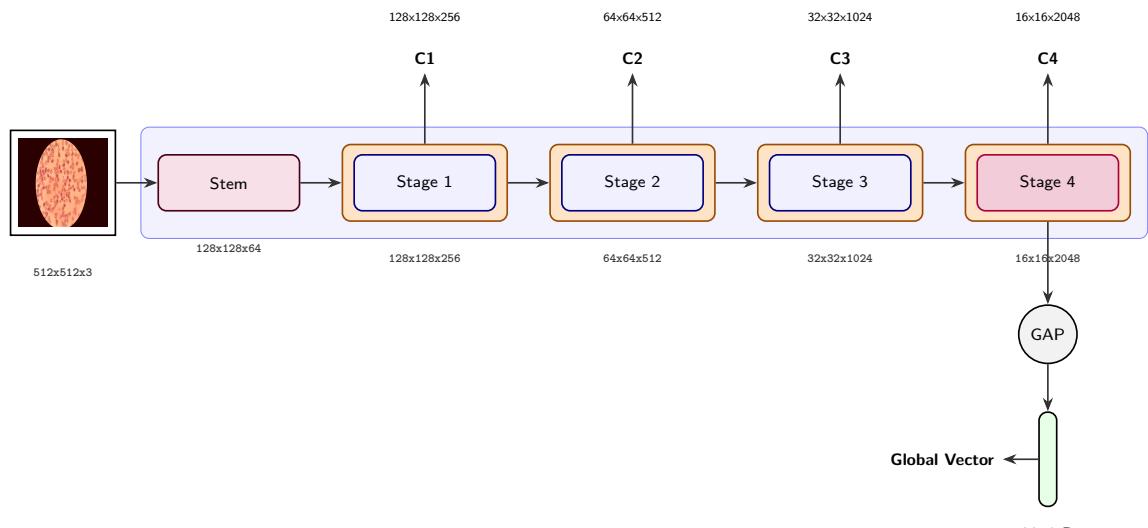
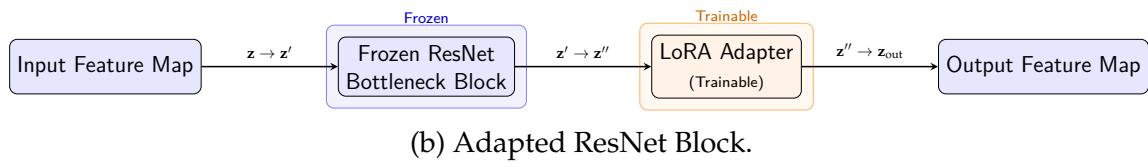
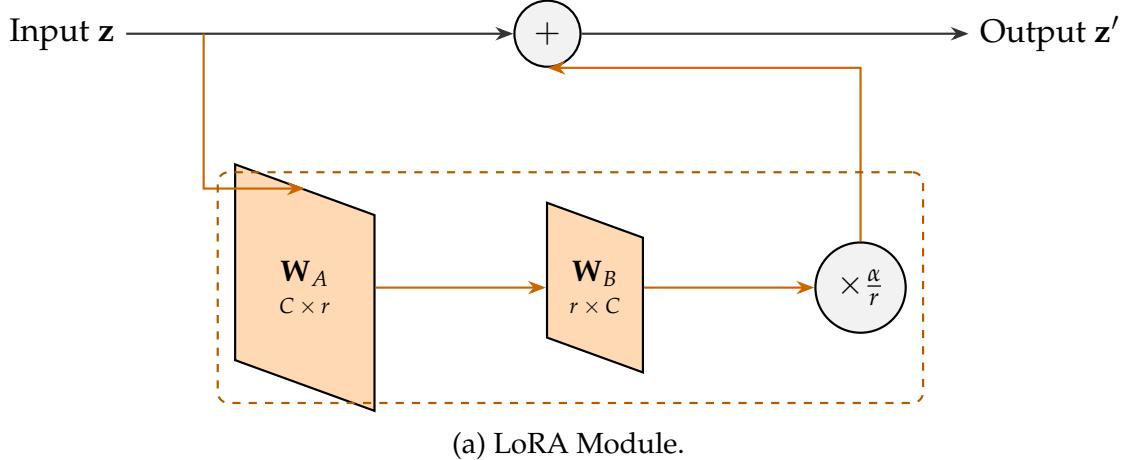


Figure 3.5: Architectural details of the shared backbone, illustrating (a) the core LoRA module, (b) its integration into a standard ResNet block, and (c) the overall adapted backbone structure.

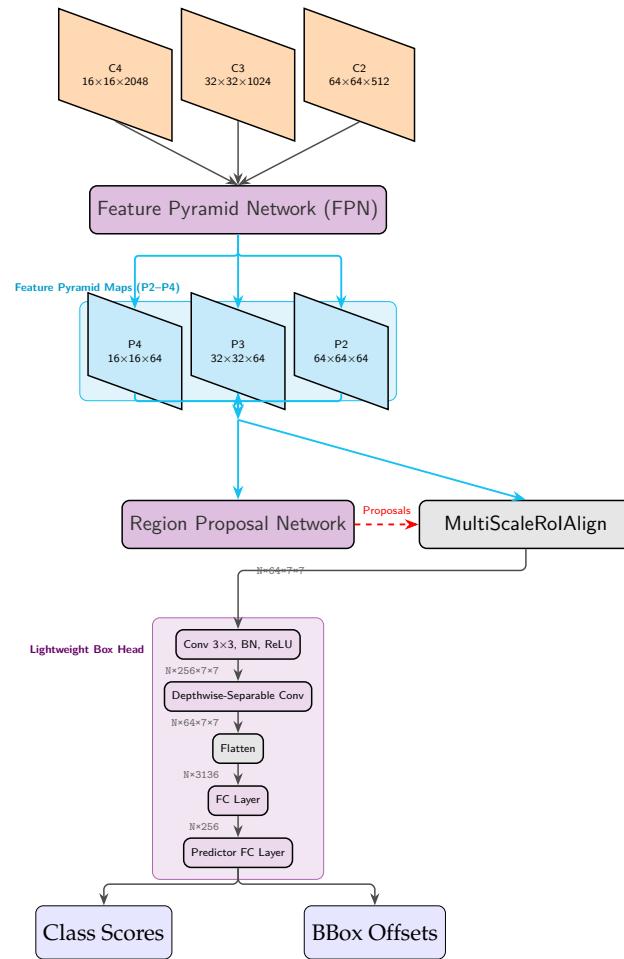


Figure 3.6: Custom Faster R-CNN Detection Head Architecture.

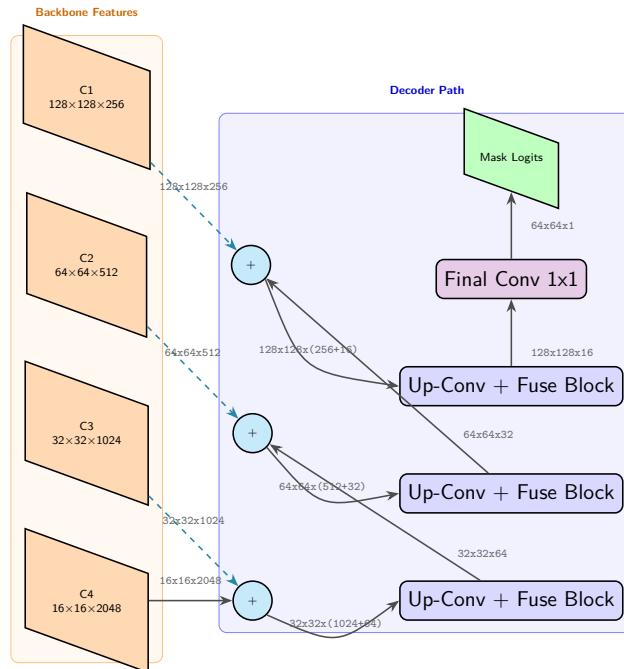


Figure 3.7: Segmentation Head Architecture.

### 3.3. MTTL MODEL ARCHITECTURE

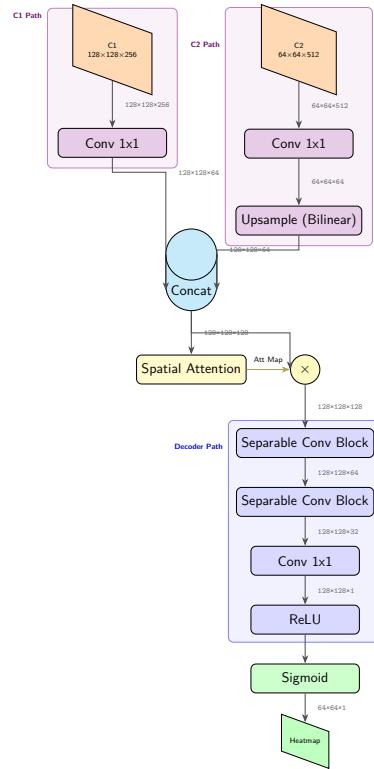


Figure 3.8: Infection Localization Head Architecture.

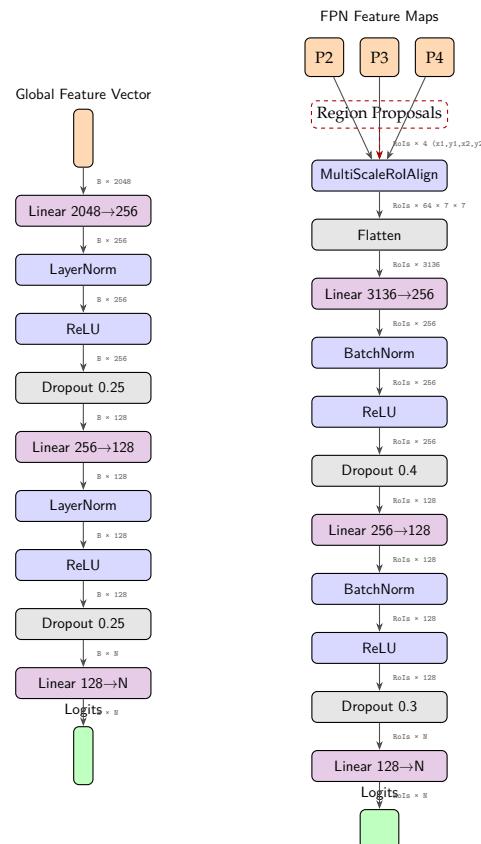


Figure 3.9: MLP Classification Head Architecture.

This method proved synergistic, encouraging clean class separation on well-localized regions without disrupting the detection objective.

**Segmentation Head.** The segmentation head ( $\mathcal{T}_{\text{seg}}$ ) employs a lightweight decoder (Figure 3.7) inspired by U-Net (Ronneberger et al. 2015). It aggregates features from multiple backbone stages and uses upsampling with skip connections to reconstruct a full-resolution mask.

**Infection Localization Head.** The localization head ( $\mathcal{T}_{\text{loc}}$ ) is an attention-based decoder that generates a heatmap of infection centers (Figure 3.8). It fuses high-resolution features with a spatial attention block and uses depth-wise separable convolutions to produce the final map.

*Note: The architectural diagrams referenced in this section were created specifically to illustrate the exact implementation used in this work.*

## 3.4 Experimental Protocol

To rigorously test whether MTTL yields a more robust diagnostic system than STL, we designed a two-phase experimental protocol. This structure isolates key variables to ensure our insights on tuning, data handling, and task interactions are reproducible. All models were trained using the AdamW optimizer (Loshchilov et al. 2019) with a cosine annealing scheduler and linear warm-up.

### 3.4.1 Phase 1: Establishing Optimal STL Baselines

The goal of this first phase is to identify the strongest possible Single-Task Learning (STL) baseline models. We investigate two main design choices:

- **Tuning Strategy:** We compare **LoRA-Only** against **Hybrid Tuning** to determine the best method for knowledge transfer.
- **Data Strategy:** We assess how best to handle class imbalance by contrasting a **Class-Balanced Sampler** with a standard sampler.

The best performing configuration from these tests is then fixed as the foundation for all subsequent experiments.

### 3.4.2 Phase 2: Evaluating Diagnostic Completeness and MTTL Synergy

The second phase addresses our research question that *Does an MTTL approach provide superior efficacy and robustness compared to STL?* We break this down into two analyses.

**(1) Task Complexity.** We progressively increase the number of detection classes to stress-test the STL paradigm. We start with **1-Class:** to detect only *Infected* cells, then **2-Class:** to detect both *Infected* and *Healthy* cells, and finally **3-Class:** to detect *Infected*, *Healthy*, and *WBC* cells.

**(2) MTTL Synergy.** We then test whether auxiliary tasks can overcome STL limitations by providing additional inductive signals. We evaluate various combinations:

- **Pairwise:** Adding Segmentation (shape prior), Localization (attention prior), or RoI Classification (feature refinement) to the detector.
- **Triple:** Combining two auxiliary tasks with detection (e.g., Segmentation and Localization) to merge their benefits.
- **Full Combination:** Integrating all four tasks (Detection, Segmentation, Localization, RoI Classification) to create the most complete MTTL setup.

## 3.5 Evaluation Protocol and Metrics

Our evaluation protocol is designed to ensure fair comparison, clinical interpretability, and deep diagnostic analysis beyond simple aggregate scores.

### 3.5.1 Diagnostic Test Datasets

We evaluate each model on three distinct, patient-disjoint test sets to profile its capabilities:

1. **The Full Test Set:** Reflects the natural distribution of the original data and serves as the primary benchmark for real-world performance.
2. **The Infected-Only Test Set:** Contains only images from infected patients. This target-rich environment rigorously evaluates **sensitivity and recall**.
3. **The Healthy-Only Test Set:** Contains only images from healthy patients. This pure negative dataset evaluates **specificity and precision**, measuring the ability to avoid false positives.
4. **Parasitemia Quantification:** We also assess the ability to quantify disease severity on the **Full Test Set**. For each slide, parasitemia ( $\mathcal{P}$ ) is estimated as:

$$\mathcal{P} = \frac{N_{\text{infected}}}{N_{\text{infected}} + N_{\text{healthy}}} \times 100\% \quad (3.10)$$

where  $N$  represents the predicted cell counts. Accuracy is measured against ground truth using **Mean Absolute Error (MAE)** and the **Pearson Correlation Coefficient (r)**.

### 3.5.2 Task-Specific Performance Metrics

**Detection.** The primary metric is **mean Average Precision (mAP)** at an Intersection over Union (IoU) threshold of 0.5 (mAP@50), following PASCAL VOC standards (Everingham et al. 2010). We also report class-specific **Precision**, **Recall**, and **F1-Score** at an optimal confidence threshold derived from the Precision-Recall curve (Davis et al. 2006). The F1-Score for the *Infected class* is a key performance indicator.

**RoI Classification.** We use the **F1-Macro** score, which is robust to class imbalance, along with overall **Accuracy**.

**Segmentation.** We report the **Dice Similarity Coefficient** (Dice 1945) and the **Intersection over Union (IoU)** on binarized output masks.

**Infection Localization.** This heatmap task is evaluated using the **Dice Score**, **Pearson Correlation**, and **Mean Absolute Error (MAE)**. On pure-negative datasets where Dice is undefined, we report the **Average Prediction Value** to measure false positive activation.

Table 3.2: Summary of Evaluation Metrics for Each Task.

Task	Primary Evaluation Metrics
$\mathcal{T}_{\text{det}}$	mAP@50, Precision, Recall, F1-Score (per-class)
$\mathcal{T}_{\text{loc}}$	Dice Score, Pearson Correlation, MAE, MSE
$\mathcal{T}_{\text{seg}}$	Dice Score, Intersection over Union (IoU)
$\mathcal{T}_{\text{roi}}$	Accuracy, F1-Macro, F1-Score (per-class)

## 3.6 Implementation Details and Experimental Setup

All experiments were conducted on the Kaggle platform (<https://www.kaggle.com>), utilizing its cloud-based computational resources. This section provides a detailed account of the hardware, software, and training configurations employed, ensuring that our experiments are reproducible and systematically controlled.

### 3.6.1 Hardware and Software Environment

Training and evaluation of all models were performed on a Kaggle Notebook instance with the following specifications:

- **Hardware:**
  - **GPU:** Dual NVIDIA Tesla T4 GPUs, each with 16 GB of VRAM.
  - **CPU:** Intel Xeon processors, with 4–8 cores.
  - **RAM:** 30 GB of system memory.
- **Software Environment:**
  - **Operating System:** Debian-based Linux distribution provided by Kaggle.
  - **Programming Language:** Python (Version 3.11.13).

- **Deep Learning Framework:** PyTorch (Version 2.6.0+cu124), with CUDA support on 2 NVIDIA Tesla T4 GPUs
- **Key Scientific Libraries:** NumPy for numerical operations, Pandas for data handling, Scikit-learn for additional metric calculations, and Matplotlib, Seaborn, and Plotly for visualization. A complete list of package versions is available in the project's dependency manifest.

### 3.6.2 Training Configuration

To ensure fair and controlled comparisons, all experiments followed a consistent training protocol, unless specific variables were intentionally altered.

- **Random Seed and Reproducibility:** A fixed seed of 12 was used for all experiments to ensure reproducibility. This seed was applied to Python's random module, NumPy, and PyTorch (including CUDA operations and worker initialization). CUDNN deterministic settings were enforced and benchmark mode disabled to guarantee consistent results across runs.
- **Optimizer:** All experiments used the **AdamW** optimizer (Loshchilov et al. 2019), preferred over standard Adam for its improved weight decay, which often leads to better generalization.
- **Learning Rate Schedule:** A linear warm-up followed by cosine annealing schedule was applied. The learning rate gradually increased from a low starting value to its target over the warm-up epochs, then decayed following a cosine curve. The primary learning rate for heads and adapters was set to  $1 \times 10^{-3}$ .
- **Discriminative Fine-Tuning:** For Hybrid Tuning experiments, backbone layers were unfrozen with a significantly lower learning rate to avoid catastrophic forgetting. Preliminary experiments indicated an optimal backbone learning rate of  $1 \times 10^{-6}$ , maintaining a 1000:1 ratio with the head learning rate.
- **Batch Size and Gradient Accumulation:** Due to GPU memory limitations, MTTL models used a base batch size of 8. To simulate an effective batch size of 16 (comparable to STL experiments), **gradient accumulation** over 2 steps was applied.
- **Automatic Mixed Precision (AMP):** PyTorch's **AMP** was employed to accelerate training and reduce memory usage. AMP leverages 16-bit floating-point precision for selected operations while preserving 32-bit precision where necessary, ensuring numerical stability with faster computation.
- **Early Stopping:** To prevent overfitting and reduce computation, training was halted when the primary validation metric (e.g., mAP@50 for detection) showed no improvement for a predefined number of epochs.

With these settings, we ensure that our training process is consistent, reliable, and directly supports the experimental results that follow.

## Results and Discussion

---

This chapter presents the empirical results of our Multi-Task Transfer Learning (MTTL) study for malaria detection. We first establish strong Single-Task Learning (STL) baselines and track how their performance changes as the diagnostic setup becomes more complex. We then report the results of our MTTL models, analyze the role of each auxiliary task, and evaluate the best configurations through quantitative metrics and visual examples. Extended results and full training details are provided in the Appendices.

### 4.1 Single-Task Learning Detection Baselines

To build reliable STL baselines, we evaluated detection in 1, 2, and 3 classes using four fine-tuning setups: LoRA-Only and Hybrid Tuning, each tested with and without a sampler. This provides a solid reference point for comparing STL performance with our MTTL approach.

Table 4.1: 1-Class Detection Baselines (Ranked by F1(Inf))

Run	mAP@50	mAP@75	F1(Inf)	R(Inf)	P(Inf)
STL_Det_1cls_Hybrid_NoSampler	<b>0.8260</b>	0.5020	<b>0.8182</b>	<b>0.8146</b>	<b>0.8218</b>
STL_Det_1cls_LoRA_WithSampler	0.8012	0.4930	0.8007	0.7991	0.8023
STL_Det_1cls_Hybrid_WithSampler	0.8201	0.4865	0.7994	0.8011	0.7978

Table 4.2: 2-Class Detection Baselines (Ranked by F1(Inf))

Run	mAP@50	mAP@75	F1(Inf)	R(Inf)	P(Inf)
STL_Det_2cls_Hybrid_WithSampler	<b>0.5834</b>	<b>0.4144</b>	<b>0.5990</b>	<b>0.4631</b>	<b>0.8478</b>
STL_Det_2cls_LoRA_WithSampler	0.4442	0.2628	0.3043	0.1847	0.8642
STL_Det_2cls_Hybrid_NoSampler	0.3457	0.1965	0.0000	0.0000	0.0000

Table 4.3: 3-Class Detection Baselines (Ranked by F1(Inf))

Run	mAP@50	mAP@75	F1(Inf)	R(Inf)	P(Inf)
STL_Det_3cls_LoRA_WithSampler	<b>0.3341</b>	<b>0.2304</b>	<b>0.3791</b>	<b>0.2533</b>	<b>0.7529</b>
STL_Det_3cls_Hybrid_WithSampler	0.3191	0.1802	0.0401	0.0205	0.7273
STL_Det_3cls_Hybrid_NoSampler	0.2145	0.1262	0.0000	0.0000	0.0000

**Observations:** Ranking by the infected-cell F1 score shows that the best STL setup changes with task complexity. Hybrid without a sampler works best for 1-class detection, Hybrid with a sampler for 2 classes, and LoRA-Only with a sampler for 3 classes, where it offers the most stable training. STL delivers strong specialists, but performance drops quickly as more classes are introduced. The best 3-class configuration, LoRA-Only with a sampler, serves as the main reference for MTTL comparisons.

## 4.2 Evaluating STL Baselines and Their Limitations

### 4.2.1 STL Baseline Performance Across All Diagnostic Tasks

We first trained specialist models for each task using the best STL settings. The results in Tables 4.4–4.5 give the reference performance for single-task approaches on this dataset. Figures 4.1 through 4.4 illustrates the stable training dynamics of these models.

Table 4.4: STL baseline performance for Detection.

Task	mAP@50	mAP@75	F1(Inf)	F1-Macro	P-Macro	R-Macro
1-Class	<b>0.8260</b>	<b>0.502</b>	<b>0.8182</b>	0.8182	0.8218	0.8146
2-Class	0.5834	0.4144	0.5990	<b>0.9274</b>	<b>0.9742</b>	<b>0.8848</b>
3-Class	0.3341	0.2304	0.3791	0.8977	0.9517	0.8496

Table 4.5: STL baseline performance across tasks.

Task	Dice	IoU	Pearson	MAE	F1-Macro	Accuracy
Segmentation	0.9656	0.9468	–	–	–	–
Heatmap Localization	0.5422	–	0.7866	0.9719	–	–
ROI Classification (2-Class)	–	–	–	–	0.9029	0.9859
ROI Classification (3-Class)	–	–	–	–	0.8047	0.9373

**Observations:** Segmentation, Localization Heatmap and RoI Classification STL models perform very well when each task is learned on its own. Detection on the other hand reaches strong infected-cell scores in the one-class case, but performance decreases as the number of classes grows. The high F1-Macro values in multi-class detection reflect the prevalence of the healthy class. This suggests that STL works for focused tasks but becomes less reliable as diagnostic complexity increases.

### 4.2.2 Training Dynamics of STL Models

Figures 4.1 to 4.4 show the training and validation curves for the best STL models in each task. The losses and main validation metrics evolve smoothly, indicating stable training and no signs of overfitting. This supports the reliability of the baseline results reported above.

## 4.2. EVALUATING STL BASELINES AND THEIR LIMITATIONS

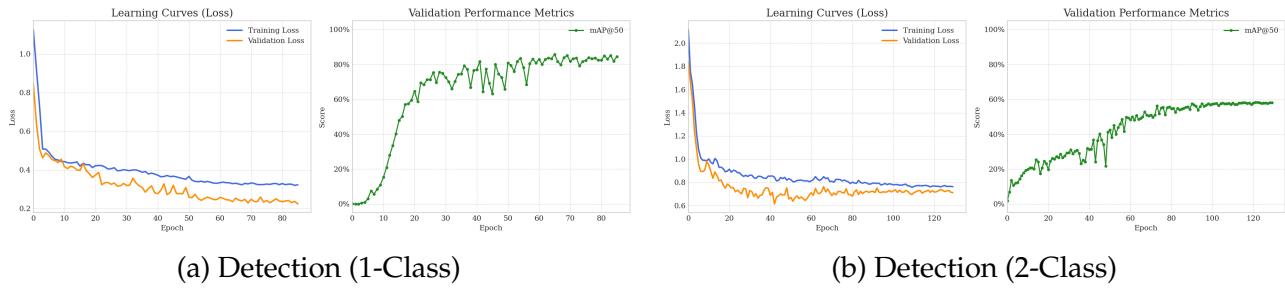


Figure 4.1: Training dynamics for 1-Class and 2-Class optimal STL detection models.

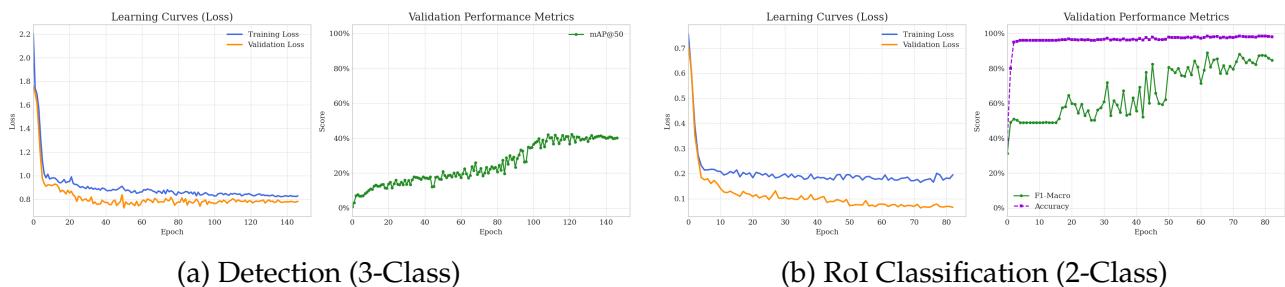


Figure 4.2: Training dynamics for 3-Class detection and 2-Class RoI classification models.

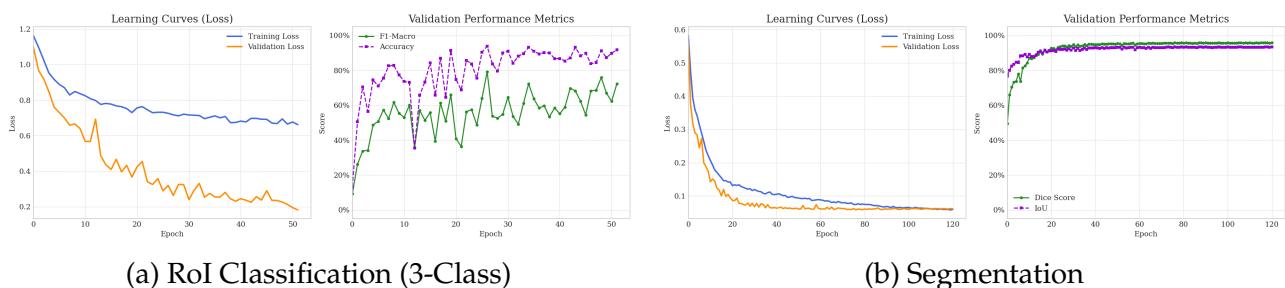


Figure 4.3: Training dynamics for 3-Class RoI classification and Segmentation models.

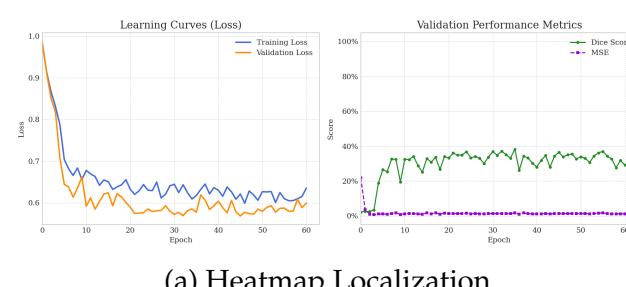


Figure 4.4: Training dynamics for the Heatmap Localization model.

### 4.2.3 STL Performance Under Increasing Complexity

A key finding from our baseline analysis is the STL detector’s inability to scale with increasing diagnostic complexity. As shown in Figure 4.5, F1-Score (Infected class), mAP@50 and mAP@75 show substantial performance drops as the model is required to distinguish between more cell classes.

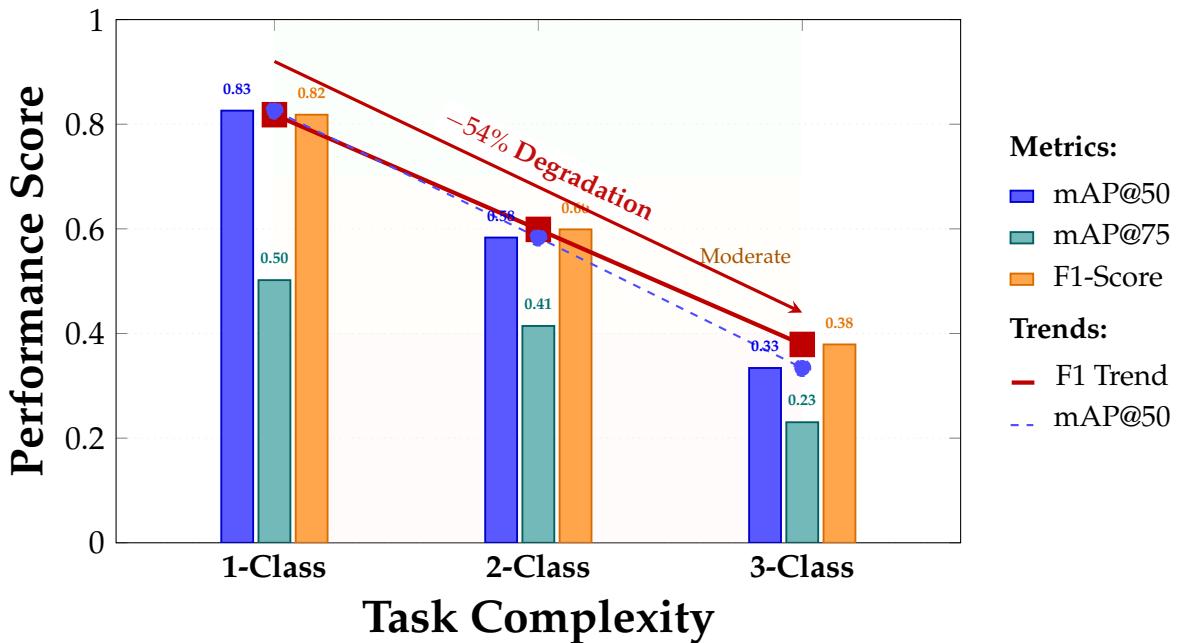


Figure 4.5: STL detector performance degrades with complexity.

**Observations:** As detection moves from 1 to 3 classes, the F1 score on the infected class drops noticeably, showing how STL struggles to keep focus on the rare but clinically important class. The high F1-Macro values in the multi-class setup mostly reflect the model’s confidence on the dominant uninfected class, which remains easy to learn. STL therefore performs well in simple, focused settings but becomes less reliable once the diagnostic problem gains complexity. This motivates the move toward a multi-task transfer learning approach that can handle richer and more realistic conditions.

## 4.3 MTTL Performance and Synergy Analysis

We now turn to our central hypothesis that a multi-task model, guided by auxiliary supervision, should counter the degradation observed in STL as detection becomes more complex. All experiments in this section use Hybrid Tuning without a sampler, since class imbalance is mild in the 1-Class setting and sampling does not bring meaningful benefits. We then assess the effect of adding segmentation (Seg), ROI classification (RoI) and heatmap localization (Loc) as auxiliary tasks. To measure how much each task contributes, we report both the raw F1-score on the infected class and the relative gain over the STL baseline for the same class configuration. For a metric  $X$ , this gain is defined as:

$$\Delta X(\%) = \frac{X_{MTTL} - X_{STL}}{X_{STL}} \times 100 \quad (4.1)$$

A positive  $\Delta$  indicates improvement, whereas a negative  $\Delta$  reflects a decline. Tables 4.6–4.8 present these values for the top STL baselines alongside the MTTL models for each modality.

Table 4.6: MTTL experiments results for **1-Class Detection**.

<b>Task(s)</b>	<b>mAP@50</b>	<b>mAP@75</b>	<b>F1(Inf)</b>	<b>R(Inf)</b>	<b>P(Inf)</b>	<b><math>\Delta F1(Inf)</math></b>
<i>STL 1-Class Baseline</i>	<b>0.8260</b>	<b>0.5020</b>	<b>0.8182</b>	<b>0.8218</b>	<b>0.8146</b>	–
+Seg	0.7885	0.6067	0.7654	0.6953	0.8514	-6.45%
+Seg+Loc	0.7709	0.5805	0.7497	0.7388	0.7609	-8.38%
+Loc	0.7367	0.5363	0.7285	0.7098	0.7483	-10.96%

Table 4.7: MTTL experiments results for **2-Class Detection**.

<b>Task(s)</b>	<b>mAP@50</b>	<b>mAP@75</b>	<b>F1(Inf)</b>	<b>F1-M</b>	<b>Prec-M</b>	<b>Rec-M</b>	<b><math>\Delta F1(Inf)</math></b>
<i>STL 2-Class Baseline</i>	0.5834	0.4144	0.5990	0.9274	0.9742	0.8848	–
+RoI	<b>0.6452</b>	<b>0.4588</b>	<b>0.7810</b>	<b>0.9411</b>	0.9761	<b>0.9085</b>	+30.38%
+Seg	0.6281	0.4462	0.7464	0.9330	0.9734	0.8959	+24.60%
+Loc	0.6398	0.4562	0.7415	0.9282	0.9737	0.8867	+23.80%
+Seg+RoI	0.6113	0.3851	0.6976	0.9256	<b>0.9785</b>	0.8780	+16.46%
+Seg+Loc	0.6025	0.4203	0.6120	0.9025	0.9550	0.8555	+2.17%
+RoI+Loc	0.5958	0.4180	0.6065	0.8958	0.9492	0.8481	+1.25%
+Seg+RoI+Loc	0.5872	0.4091	0.6030	0.8897	0.9453	0.8402	+0.67%

### 4.3.1 Analysis of MTTL Synergy

**Observations:** Across our detection experiments, the impact of auxiliary tasks depends on the complexity of the primary task. For 1-Class Detection, adding auxiliary tasks reduces performance, with F1(Inf) drops from 6% to 11%. For 2-Class Detection, RoI Classification provides the largest gain (+30.38%), followed by Segmentation (+24.60%) and Localization (+23.80%), while combinations reduce incremental improvements (+16.46% to +0.67%). For 3-Class Detection, every MTTL variant improves over the STL baseline, with F1(Inf) gains from +44% to +103.39%, and the strongest single-task improvements coming from RoI and Segmentation.

Our experiments results provide evidence for our central hypothesis that auxiliary tasks, while sometimes detrimental to highly specialized single-task models, become increasingly

Table 4.8: MTTL experiments results for 3-Class Detection.

Task(s)	mAP@50	mAP@75	F1(Inf)	F1-M	Prec-M	Rec-M	$\Delta F1(Inf)$
<i>STL 3-Class Baseline</i>	0.3341	0.2304	0.3791	0.8977	0.9517	0.8496	–
+Seg	0.7013	0.5543	0.7230	0.9351	0.9780	0.8958	+90.74%
+Loc	0.6994	0.5217	0.7034	0.9195	0.9595	0.8827	+85.55%
+RoI	<b>0.7402</b>	0.5475	<b>0.7710</b>	<b>0.9409</b>	0.9739	<b>0.9101</b>	<b>+103.39%</b>
+Seg+RoI	0.7089	0.4375	0.6662	0.9289	0.9729	0.8888	+75.73%
+Loc+RoI	0.7289	0.5128	0.6722	0.9375	<b>0.9803</b>	0.8983	+77.32%
+Seg+Loc	0.6913	0.4853	0.5468	0.9362	0.9739	0.9013	+44.24%
+Seg+RoI+Loc	0.7324	<b>0.5538</b>	0.7360	0.9364	0.9761	0.8997	+94.15%

beneficial as the complexity of the primary task grows. The benefits provided by auxiliary tasks depends on the complexity of the primary detection problem. As the task becomes more nuanced, auxiliary supervision can provide meaningful guidance, though the choice and combination of tasks.

For **1-Class Detection**, adding auxiliary tasks consistently hurts performance. The STL baseline dominates, and all MTTL variants show relative F1(Inf) drops from 6% to 11%. The task is simple enough that extra objectives introduce interference rather than useful signal. In this case, dedicating the model solely to detecting infected cells is most effective.

For **2-Class Detection**, auxiliary tasks begin to provide clear gains. RoI Classification is the most synergistic, giving +30.38% in F1(Inf), likely due to its region level feature refinement aiding distinction between infected and healthy cells. Segmentation and Localization also help, with +24.60% and +23.80% gains, respectively, showing moderate complementary effects. When multiple tasks are combined, the benefits diminish, +Seg+RoI drops to +16.46%, and combinations like +Seg+Loc or +RoI+Loc give only +2.17% and +1.25%, while the full three-task setup barely improves (+0.67%). This suggests that a few well chosen auxiliary tasks are synergistic, adding too many can create conflicts, possibly due to competing gradients or overlapping supervision signals.

For **3-Class Detection**, auxiliary supervision provides the largest advantage. All MTTL variants outperform the weak STL baseline, with relative F1(Inf) improvements from +44% to +103%. RoI Classification leads (+103.39%), followed by Segmentation (+90.74%) and Localization (+85.55%), each contributing complementary strengths: RoI refines local discriminative features, Segmentation provides structural guidance, and Localization helps with spatial attention. Paired combinations like Seg+RoI (+75.73%) and Loc+RoI (+77.32%) remain strong, whereas Seg+Loc (+44.24%) shows weaker synergy, suggesting overlap in the

signals they provide. The full three-task configuration recovers to +94.15%, indicating that ROI stabilizes the learning when combined with multiple auxiliaries. Overall, Segmentation and ROI are the most consistently synergistic, while Localization is helpful but can conflict when stacked with multiple tasks.

#### 4.3.2 Training Dynamics for Top MTTL Models

We illustrate the training dynamics of the best-performing MTTL models for each detection category Figure 4.6 to Figure 4.8. Plots correspond to loss curves, detection metrics, and auxiliary metrics.

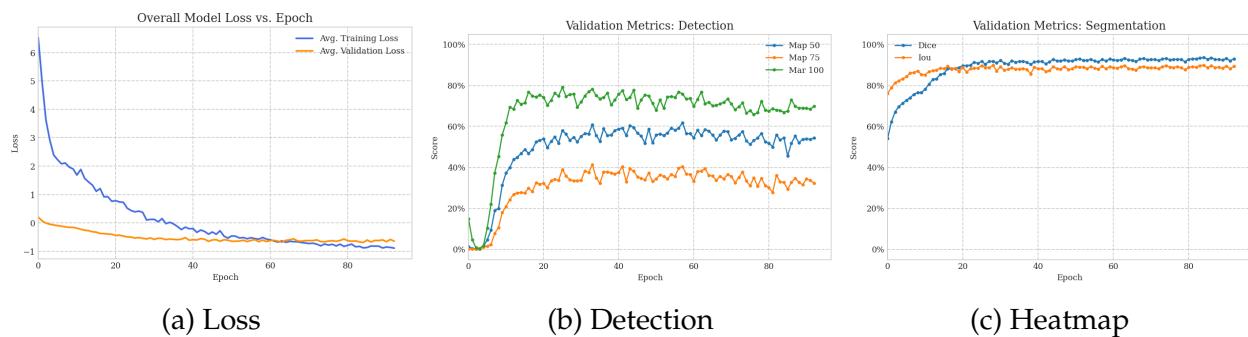


Figure 4.6: Best 1-Class Detection MTTL model (+Seg, F1=0.7654).

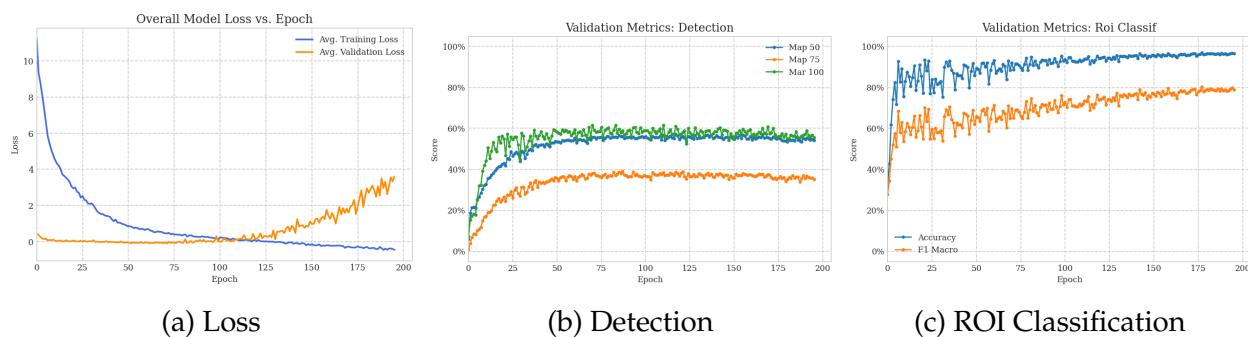


Figure 4.7: Best 2-Class Detection MTTL model (+RoI, F1=0.7810).

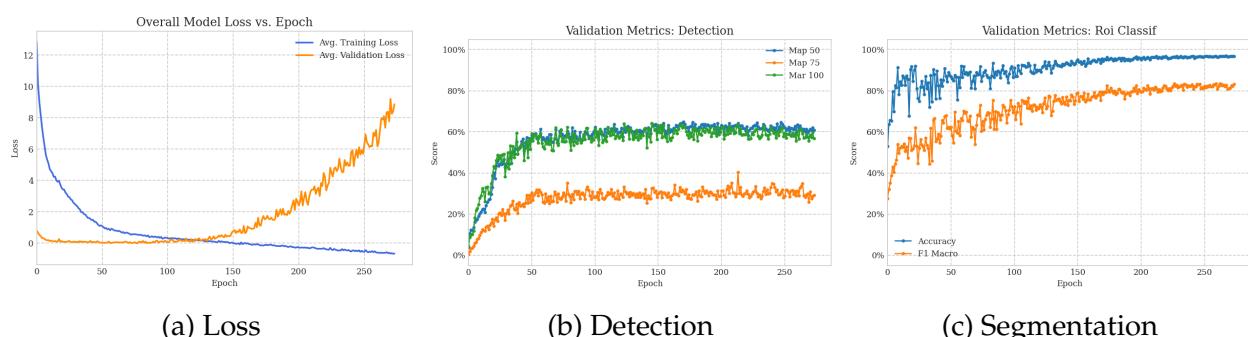


Figure 4.8: Best 3-Class Detection MTTL model (+RoI, F1=0.7710).

## 4.4 Final Model Evaluation and Robustness Analysis

Following the baseline evaluation, we now focus on clinical robustness using the specialized subsets defined in subsection 3.5.1. We assess **sensitivity** on infected-only slides and **specificity** on healthy-only slides. Furthermore, we extend the analysis to **parasitemia quantification**, estimating infection load via the count-based method in Equation 3.10. These tests provide a complete view of reliability and clinical utility.

### 4.4.1 Detection Performance Leaderboard

Table 4.9 presents a leaderboard of the top performing models. Champions are selected based on the highest F1 score for the infected class, enabling a direct comparison across 1, 2, and 3-class tasks. The results illustrate a clear trend: while STL dominates the simplest setting, MTTL becomes increasingly superior as diagnostic complexity grows.

Table 4.9: Best performers by detection task complexity.

Model Configuration	Paradigm	Classes	mAP@50	F1-Macro	F1 (Inf.)
Det(1)	STL	1	<b>0.8260</b>	<b>0.8182</b>	<b>0.8182</b>
Det(1)+Seg	MTTL	1	0.7885	0.7654	0.7654
Det(2)	STL	2	0.5834	0.9274	0.5990
Det(2)+RoI	MTTL	2	<b>0.6452</b>	<b>0.9411</b>	<b>0.7810</b>
Det(3)	STL	3	0.3341	0.8977	0.3791
Det(3)+RoI	MTTL	3	<b>0.7402</b>	<b>0.9409</b>	<b>0.7710</b>

### 4.4.2 Diagnostic Robustness on Specialized Subsets

To test hypothesis H3 (Robustness), we evaluated the champion models on clinically relevant subsets. We measured F1, Recall, and Precision on infected-only slides, and counted total false positives on healthy-only slides (Table 4.10).

Table 4.10: Performance of best detection models on specialized test subsets.

Model	Paradigm	Classes	F1(Inf)	R(Inf)	P(Inf)	$\sum \text{FPs}(H)$
Det(1)	STL	1	0.7725	<b>0.8491</b>	0.7086	12
Det(2)	STL	2	0.6021	0.4631	0.8603	8
Det(3)	STL	3	0.2376	0.1385	0.8333	6
Det(1)+Seg	MTTL	1	0.7250	0.8193	0.6503	27
Det(2)+RoI	MTTL	2	<b>0.7907</b>	0.7401	0.8487	10
Det(3)+RoI	MTTL	3	0.7359	0.6214	<b>0.9023</b>	3

Consistent with previous findings, STL excels in the 1-class setting with high sensitivity ( $F1 = 0.7725$ , Recall=0.8491). However, performance degrades sharply with complexity; 3-class STL drops to an F1 of 0.2376 and Recall of 0.1385. In contrast, MTTL models sustain performance. The 3-class MTTL model achieves an F1 of 0.7359 and Recall of 0.6214, while generating only 3 false positives on healthy slides. This confirms that MTTL offers a superior balance of sensitivity and specificity for complex diagnostics.

#### 4.4.3 Parasitemia Quantification on Multi-Class Detection Models

A key measure of diagnostic utility is accurate slide-level parasitemia estimation on the **full test set**. We therefore evaluated the champion 2-Class and 3-Class models from subsection 4.4.1, deriving parasitemia using the count-based formulation in Equation 3.10. The results are summarized below.

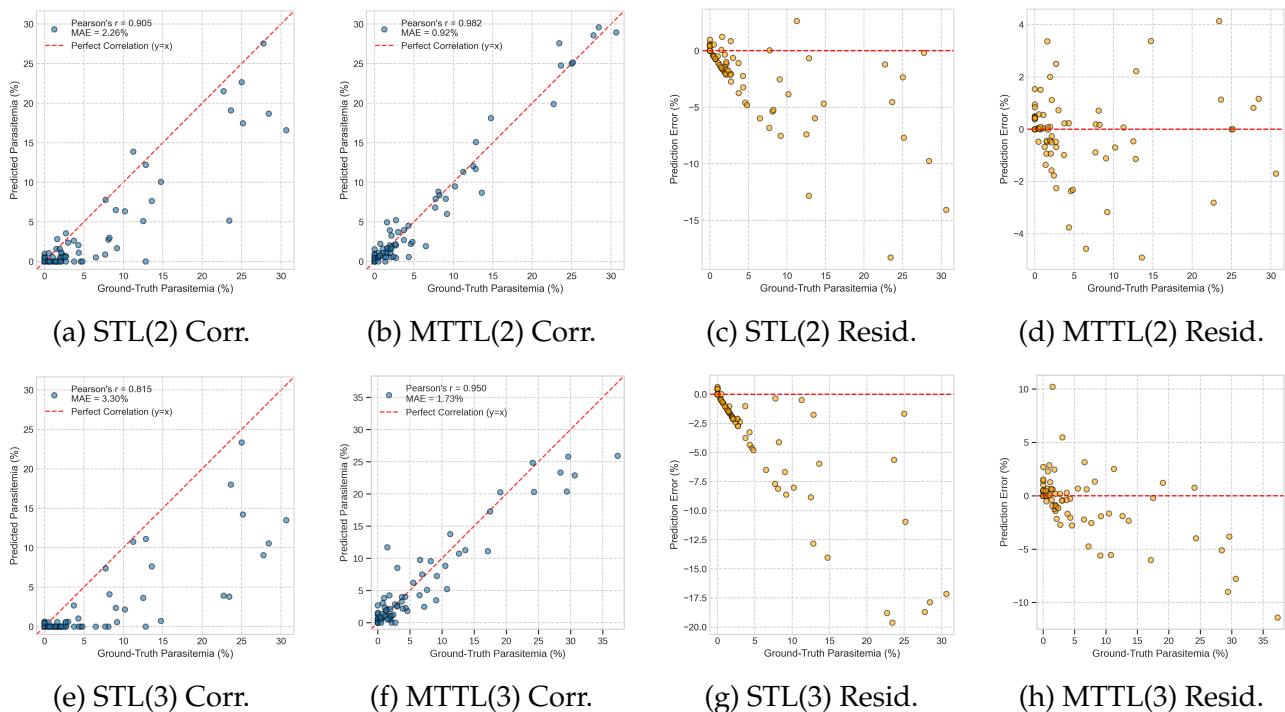


Figure 4.9: Parasitemia estimation for all best multi-class detection models.

Table 4.11: Parasitemia estimation performance of the champion models on the full test set.

Model Configuration	Paradigm	MAE (%)	Pearson Correlation (r)
Det(1)	STL	—	—
Det(1)+Seg	MTTL	—	—
Det(2)	STL	2.2628	0.9048
Det(2)+RoI	MTTL	<b>0.9211</b>	<b>0.9824</b>
Det(3)	STL	3.2963	0.8145
Det(3)+RoI	MTTL	<b>1.0805</b>	<b>0.9739</b>

The quantitative results are presented in Table 4.11. Single-class detection models are excluded from this analysis (indicated by dashes) as they lack the class distinction necessary for parasitemia estimation. Visual assessment is provided through correlation and residual plots in Figure 4.9, enabling direct evaluation of prediction accuracy and error distribution. Lower MAE and higher Pearson's  $r$  indicate superior estimation performance.

The results demonstrate substantial clinical advantages of MTTL models for parasitemia quantification. For 2-class detection, MTTL achieves MAE = 0.92% versus STL's 2.26% (a 59% reduction), while improving correlation from  $r = 0.90$  to  $r = 0.98$ . For 3-class detection, MTTL attains MAE = 1.08% compared to STL's 3.30% (a 67% reduction), with correlation increasing from  $r = 0.81$  to  $r = 0.97$ . These improvements confirm that multi-task learning improves not only detection robustness but also precise infection load quantification.

The correlation plots reveal that MTTL predictions align a little more tightly with ground-truth parasitemia across the full range of infection levels. Correspondingly, the residual plots show that MTTL errors are smaller, more uniformly distributed, and exhibit less systematic bias than STL models. This combination of improved accuracy, reduced variance, and minimized bias highlights the practical reliability of MTTL for slide-level clinical assessments.

## Discussion

---

This chapter interprets our results in relation to the initial hypotheses. We discuss the trade-off between specialized and generalized models, the answer to our central research question, the implications for medical AI, and the limitations of the study.

### 5.1 The Specialist vs. The Generalist

A key finding from our experiments is the clear trade-off between specialization and generalization. The 1-Class STL detector, trained solely to find infected cells, acted as a formidable specialist. On the full test set, it achieved the highest F1-Score for the infected class (0.8182) and an outstanding Recall of 0.8491 on the challenging infected-only subset. However, this aggressive sensitivity reduced specificity, resulting in 12 false positives on healthy-only slides. More critically, the specialist model was brittle. When we expanded the task to the realistic 3-class setting, the STL framework collapsed, with the infected-class F1-Score dropping to a clinically inadequate 0.3791.

Conversely, the 3-Class MTTL model augmented with ROI Classification proved to be a superior generalist. While it did not quite match the specialist's peak score on the simple task, it dramatically outperformed the STL baselines in the complex setting. It maintained a strong infected-class F1-Score of 0.7710 in the 3-class scenario, a relative improvement of over 103% compared to the STL equivalent. Importantly, it achieved this while improving specificity, committing only 3 false positive errors on the healthy-only set. This ability to trade a small amount of narrow-task performance for reliable, broad, and safe diagnostic coverage aligns better with clinical requirements.

### 5.2 Answering the Central Research Question

We sought to determine *whether a multi-task transfer learning framework provides greater efficacy and robustness than single-task learning for automated malaria diagnosis*. Our findings support the conclusion: **Yes, for diagnostically realistic settings, multi-task transfer learning is the superior paradigm.** This is supported by four pillars of evidence:

- **Performance Scaling:** MTTL prevented the severe performance degradation that STL suffered when scaling from 1-class to complex 3-class detection (Hypothesis H1).

- **Auxiliary Task Synergy:** The performance gains from specific auxiliary tasks confirm that adding relevant inductive biases benefits this domain (Hypothesis H2).
- **Robustness:** The 3-Class MTTL model achieved a far better balance of sensitivity (Recall 0.6214) and specificity (low false positives) compared to the STL models, which struggled with complexity (Hypothesis H3).
- **Clinical Utility:** The improved representations translated to better downstream utility. The 3-Class MTTL model reduced the Parasitemia Mean Absolute Error from 3.30% (STL) to 1.08%, a 67% reduction in estimation error.

## 5.3 Implications of Key Findings

These results have practical implications for the design of diagnostic systems in computational pathology.

First, benchmarking on simple, binary tasks is risky. A model that appears state-of-the-art on a narrow benchmark may fail in a real-world workflow that requires multiple, interdependent outputs. The collapse of our STL models in the multi-class setting demonstrates this danger.

Second, specific auxiliary tasks act as powerful regularizers. RoI Classification was the most effective, delivering performance gains of over 103% for 3-class detection. This suggests that forcing the model to refine features on localized cell instances is highly beneficial. Segmentation also proved synergistic, providing shape priors that improved performance by roughly 90%. However, we observed negative transfer in the simple 1-class setting, showing that synergy is not automatic, it requires matching auxiliary objectives to the complexity of the primary task.

Finally, the success of our Hybrid Tuning strategy offers a blueprint for future work. It demonstrates that combining the efficiency of PEFT methods like LoRA with targeted fine-tuning of deep backbone layers is a computationally feasible way to adapt large, pre-trained models to complex medical domains.

## 5.4 Limitations of the Study

While this research provides strong evidence for the benefits of MTTL, we must acknowledge specific limitations that define the scope of our conclusions.

- **Dataset Scope:** We used only the NLM dataset, which contains Giemsa-stained thin blood smears of *P. falciparum*. We have not validated these models on other species (e.g., *P. vivax*) or thick blood smears. True clinical robustness requires evaluation on multi-center datasets that account for variations in slide preparation and imaging hardware.

## **5.4. LIMITATIONS OF THE STUDY**

---

- **Annotation Noise:** The dataset contains inherent noise, such as ambiguous cell boundaries and potential inconsistencies in labeling early-stage trophozoites. This noise creates a performance ceiling for supervised training and may explain some discrepancies between segmentation and detection performance.
- **Task Interference:** Multi-task synergy is not guaranteed. The heatmap localization task consistently interfered with other auxiliary tasks in higher-order combinations. Future work should explore architectures like cross-stitch networks or gradient surgery to mitigate this negative transfer.
- **Optimization Constraints:** Practical limits constrained our hyperparameter tuning. While we applied a consistent protocol, each configuration would likely benefit from an exhaustive search for optimal learning rates and loss weights. Additionally, Automatic Mixed Precision required careful implementation to prevent numerical instability.
- **Simplified Parasitemia Estimation:** We estimated parasitemia based on cell counts from single image fields. Clinical practice typically requires counting parasites against white blood cells over many fields to ensure statistical validity. A fully validated clinical tool would need to aggregate counts across multiple fields.

## Conclusion

---

This work investigated Multi-Task Transfer Learning (MTTL) as a solution for automated malaria diagnosis. We compared this approach against Single-Task Learning (STL) to determine if learning related tasks simultaneously could improve diagnostic reliability. Our findings provide clear evidence that MTTL, when combined with parameter-efficient fine-tuning, offers a robust and scalable framework for medical image analysis. This chapter summarizes our contributions, the principal findings, and the implications for future research.

## Summary of Contributions

The primary contribution of this work is the formulation and validation of a multi-task transfer learning framework specifically designed for malaria detection. The key contributions are:

- **A Mathematical Framework for MTTL:** We proposed a formal mathematical definition for MTTL that explicitly handles task interactions, loss aggregation, and regularization. This provides a reproducible basis for future research and ensures that performance gains are analytically interpretable.
- **System Design and Hybrid Tuning:** We designed a modular architecture that integrates detection, segmentation, and region-of-interest classification. By combining Low-Rank Adaptation (LoRA) with selective fine-tuning of deep layers, we introduced a computationally efficient strategy to adapt large pre-trained models without the cost of full retraining.
- **Empirical Validation of Task Synergy:** Through rigorous experimentation, we identified which auxiliary tasks provide beneficial inductive biases. We demonstrated that ROI Classification and Segmentation significantly enhance feature learning, while also highlighting cases where negative transfer can occur, such as with heatmap localization in certain configurations.

## Principal Findings

Our experiments yielded several critical insights into the behavior of deep learning models in diagnostic settings:

- 
- **MTTL Outperforms STL in Realistic Scenarios:** STL models proved to be excellent specialists but poor generalists. While the 1-Class STL detector achieved a high F1-score of 0.8182 on simple tasks, it collapsed when faced with the realistic complexity of multi-class detection. In contrast, MTTL models maintained robust performance, achieving a superior balance of sensitivity and specificity even as the diagnostic difficulty increased.
  - **Auxiliary Tasks act as Regularizers:** The addition of specific auxiliary tasks improved generalization. RoI Classification emerged as the most effective, delivering a relative gain of over 103% in the 3-class F1-score. Segmentation also provided strong structural cues, yielding a 90% improvement. However, these benefits are not automatic; in simple 1-class settings, auxiliary tasks sometimes reduced performance, indicating that task selection must match the problem complexity.
  - **Clinical Relevance:** The improved stability of the MTTL models translated directly into better clinical metrics. For parasitemia quantification, the 3-Class MTTL model reduced the mean absolute error from 3.30% (STL) to 1.08%. This 67% reduction in estimation error demonstrates that the theoretical advantages of our framework lead to tangible improvements in diagnostic utility.

## Future Directions

This work establishes a foundation for several avenues of future research:

- **Domain Expansion:** The MTTL framework is not limited to malaria. With careful task selection, it could be adapted to other pathologies, medical imaging domains and even other domains.
- **Data Diversity and Generalization:** To confirm true clinical robustness, future work must validate these models on multi-center datasets. This includes testing on different malaria species, such as *P. vivax*, varied staining protocols, and thick blood smears.
- **Advanced Multi-Task Architectures:** While our shared backbone approach proved effective, exploring advanced architectures like cross-stitch networks or gradient surgery could help mitigate the negative transfer observed with conflicting tasks.
- **Clinical Deployment:** Beyond algorithmic improvements, the next logical step is to evaluate this system in a real-world workflow. Assessing its impact in low-resource settings, where expert microscopists are scarce, would be the ultimate test of its value.

In conclusion, this study demonstrates that Multi-Task Transfer Learning is a powerful paradigm for automated malaria diagnosis. By learning to perform complementary tasks simultaneously, the model develops richer representations that are more accurate, robust, and clinically useful than those learned in isolation. This work offers a clear path toward scalable, reliable AI systems capable of supporting global health initiatives.

---

## References

---

- Abdurahman, Fante, Aliy** (2021). "Malaria parasite detection in thick blood smear microscopic images using modified YOLOV3 and YOLOV4 models". In: *BMC bioinformatics* 22, pp. 1–17. DOI: [10.1186/s12859-021-04036-4](https://doi.org/10.1186/s12859-021-04036-4).
- Abraham, Khan** (2019). "A Novel Focal Tversky Loss Function with Improved Attention U-Net for Lesion Segmentation". In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI)*. IEEE, pp. 683–687. DOI: [10.1109/ISBI.2019.8759329](https://doi.org/10.1109/ISBI.2019.8759329).
- Alonso-Ramírez** (2022). "Classifying Parasitized and Uninfected Malaria Red Blood Cells Using Convolutional-Recurrent Neural Networks". In: *IEEE Access* 10, pp. 97348–97359. DOI: [10.1109/ACCESS.2022.3206266](https://doi.org/10.1109/ACCESS.2022.3206266).
- Alzubaidi** (2021). "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions". In: *Journal of big Data* 8, pp. 1–74.
- Banerjee** (2022). "Deep convolutional neural network (falcon) and transfer learning-based approach to detect malarial parasite". In: *Multimedia Tools and Applications* 81.10, pp. 13237–13251. DOI: [10.1007/s11042-021-10946-5](https://doi.org/10.1007/s11042-021-10946-5).
- Bejnordi** (2017). "Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer". In: *JAMA* 318.22, pp. 2199–2210. DOI: [10.1001/jama.2017.14585](https://doi.org/10.1001/jama.2017.14585).
- Bibin, Nair, Punitha** (2017). "Malaria parasite detection from peripheral blood smear images using deep belief networks". In: *IEEE Access*. Vol. 5, pp. 9099–9108. DOI: [10.1109/ACCESS.2017.2707460](https://doi.org/10.1109/ACCESS.2017.2707460).
- Boit, Patil** (2024). "An Efficient Deep Learning Approach for Malaria Parasite Detection in Microscopic Images". In: *Diagnostics (Basel)* 14.23, p. 2738. DOI: [10.3390/diagnostics14232738](https://doi.org/10.3390/diagnostics14232738).
- Caruana** (1997). "Multitask Learning". In: *Machine Learning* 28.1, pp. 41–75. DOI: [10.1023/A:1007379606734](https://doi.org/10.1023/A:1007379606734).
- Chandani, Jaison, Pandeya, Sharma** (2025). "Investigating the Benefits of Multi-Task Transfer Learning for Medical Image Segmentation". In: *Proceedings of the 5th International Conference on Data Science, Machine Learning and Applications; Volume 2*. Ed. by Amit Kumar, Vinit Kumar Gunjan, Sabrina Senatore, and Yu-Chen Hu. Singapore: Springer Nature Singapore, pp. 487–493. ISBN: 978-981-97-8043-3.
- Chawla, Bowyer, Hall, Kegelmeyer** (2002). "SMOTE: Synthetic Minority Over-Sampling Technique". In: *Journal of Artificial Intelligence Research* 16, pp. 321–357. DOI: [10.1613/jair.953](https://doi.org/10.1613/jair.953). URL: <https://www.jair.org/index.php/jair/article/view/10302>.

- 
- Chen, Badrinarayanan, Lee, Rabinovich** (2018). "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks". In: *International conference on machine learning*. PMLR, pp. 794–803. URL: <https://proceedings.mlr.press/v80/chen18a>.
- Dalal, Triggs** (2005). "Histograms of Oriented Gradients for Human Detection". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1, pp. 886–893. DOI: [10.1109/CVPR.2005.177](https://doi.org/10.1109/CVPR.2005.177).
- Davis, Goadrich** (2006). "The Relationship Between Precision-Recall and ROC Curves". In: *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*, pp. 233–240. DOI: [10.1145/1143844.1143874](https://doi.org/10.1145/1143844.1143874). URL: <https://doi.org/10.1145/1143844.1143874>.
- Deng** (2009). "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. DOI: [10.1109/CVPR.2009.5206848](https://doi.org/10.1109/CVPR.2009.5206848).
- Dheer, Kothari, Zaidi, Singh** (2025). "Exploiting Multi-Task Transfer Learning to Improve Medical Image Segmentation Accuracy". In: *Proceedings of the 5th International Conference on Data Science, Machine Learning and Applications; Volume 2*. Ed. by Amit Kumar, Vinit Kumar Gunjan, Sabrina Senatore, and Yu-Chen Hu. Singapore: Springer Nature Singapore, pp. 348–353. ISBN: 978-981-97-8043-3. DOI: [10.1007/978-981-97-8043-3\\_37](https://doi.org/10.1007/978-981-97-8043-3_37).
- Dice** (1945). "Measures of the Amount of Ecologic Association Between Species". In: *Ecology* 26.3, pp. 297–302. DOI: [10.2307/1932409](https://doi.org/10.2307/1932409). URL: <https://doi.org/10.2307/1932409>.
- Dosovitskiy** (2020). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *arXiv preprint arXiv:2010.11929*. arXiv: [2010.11929](https://arxiv.org/abs/2010.11929). URL: <https://arxiv.org/abs/2010.11929>.
- Esteva** (2017). "Dermatologist-level classification of skin cancer with deep neural networks". In: *Nature* 542.7639, pp. 115–118. DOI: [10.1038/nature21056](https://doi.org/10.1038/nature21056).
- Everingham, Van Gool, Williams, Winn, Zisserman** (2010). "The PASCAL Visual Object Classes (VOC) Challenge". In: *International Journal of Computer Vision* 88.2, pp. 303–338. DOI: [10.1007/s11263-009-0275-4](https://doi.org/10.1007/s11263-009-0275-4). URL: <https://doi.org/10.1007/s11263-009-0275-4>.
- Eyong** (2022). "Plasmodium falciparum histidine-rich protein 2 and 3 gene deletion in the Mount Cameroon region". In: *IJID Regions* 3, pp. 300–307. ISSN: 2772-7076. DOI: [10.1016/j.ijregi.2022.05.006](https://doi.org/10.1016/j.ijregi.2022.05.006). URL: <https://www.sciencedirect.com/science/article/pii/S2772707622000716>.
- Fawcett** (2006). "An Introduction to ROC Analysis". In: *Pattern Recognition Letters* 27.8, pp. 861–874. DOI: [10.1016/j.patrec.2005.10.010](https://doi.org/10.1016/j.patrec.2005.10.010). URL: <https://doi.org/10.1016/j.patrec.2005.10.010>.
- Gatton** (2020). "Impact of Plasmodium falciparum gene deletions on malaria rapid diagnostic test performance". In: *Malaria Journal* 19. DOI: [10.1186/s12936-020-03460-w](https://doi.org/10.1186/s12936-020-03460-w). URL: <https://doi.org/10.1186/s12936-020-03460-w>.
- Gulshan** (2016). "Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs". In: *JAMA* 316.22, pp. 2402–2410. DOI: [10.1001/jama.2016.17216](https://doi.org/10.1001/jama.2016.17216).

- 
- He, Zhang, Ren, Sun** (2016a). "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. DOI: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- (2016b). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.
- Hoerl, Kennard** (1970). "Ridge Regression: Biased Estimation for Nonorthogonal Problems". In: *Technometrics* 12.1, pp. 55–67. DOI: [10.1080/00401706.1970.10488634](https://doi.org/10.1080/00401706.1970.10488634). URL: <https://doi.org/10.1080/00401706.1970.10488634>.
- Houlsby** (2019). "Parameter-Efficient Transfer Learning for NLP". In: *Proceedings of the 36th International Conference on Machine Learning (ICML)*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research, pp. 2790–2799. DOI: [10.48550/arXiv.1902.00751](https://doi.org/10.48550/arXiv.1902.00751).
- Hoyos-Villegas, Hoyos-Quintero** (2024). "Supporting Malaria Diagnosis Using Deep Learning and Data Augmentation". In: *Diagnostics* 14.7, p. 690. DOI: [10.3390/diagnostics14070690](https://doi.org/10.3390/diagnostics14070690).
- Hu** (2021). "LoRA: Low-Rank Adaptation of Large Language Models". In: *International Conference on Learning Representations (ICLR) 2022*. Preprint arXiv:2106.09685. DOI: [10.48550/arXiv.2106.09685](https://doi.org/10.48550/arXiv.2106.09685). URL: <https://arxiv.org/abs/2106.09685>.
- Huang, Liu, Van Der Maaten, Weinberger** (2017). "Densely Connected Convolutional Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708. DOI: [10.1109/CVPR.2017.243](https://doi.org/10.1109/CVPR.2017.243).
- Huang** (2022). "MTL-ABS3Net: Atlas-Based Semi-Supervised Organ Segmentation Network With Multi-Task Learning for Medical Images". In: *IEEE Journal of Biomedical and Health Informatics* 26.8, pp. 3988–3998. DOI: [10.1109/JBHI.2022.3153406](https://doi.org/10.1109/JBHI.2022.3153406).
- Ioffe, Szegedy** (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research. PMLR, pp. 448–456. DOI: [10.48550/arXiv.1502.03167](https://doi.org/10.48550/arXiv.1502.03167). URL: <https://proceedings.mlr.press/v37/ioffe15.html>.
- Islam, Vibashan, Lim, Ren** (2021). "St-mtl: Spatio-temporal multitask learning model to predict scanpath while tracking instruments in robotic surgery". In: *Medical Image Analysis* 69, p. 101968. DOI: [10.1016/j.media.2020.101837](https://doi.org/10.1016/j.media.2020.101837).
- Jacobs** (2014). "Harmonization of malaria rapid diagnostic tests: best practices in labelling including instructions for use". In: *Malar J* 13, p. 505. DOI: [10.1186/1475-2875-13-505](https://doi.org/10.1186/1475-2875-13-505).
- Jameela, Athota, Singh, Gunjan, Kahali** (June 2022). "Deep Learning and Transfer Learning for Malaria Detection". In: *Computers in Intelligence and Neuroscience* 2022. PMCID: PMC9259269; PMID: 35814548, p. 2221728. DOI: [10.1155/2022/2221728](https://doi.org/10.1155/2022/2221728).
- Kassim** (2021). "Clustering-Based Dual Deep Learning Architecture for Detecting Red Blood Cells in Malaria Diagnostic Smears". In: *IEEE Journal of Biomedical and Health Informatics* 25.5, pp. 1735–1746. DOI: [10.1109/JBHI.2020.3034863](https://doi.org/10.1109/JBHI.2020.3034863).
- Kendall, Gal, Cipolla** (2018). "Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics". In: *Proceedings of the IEEE Conference on Computer Vision*

---

*and Pattern Recognition (CVPR)*. IEEE, pp. 7482–7491. DOI: [10.1109/CVPR.2018.00781](https://doi.org/10.1109/CVPR.2018.00781). URL: <https://doi.org/10.1109/CVPR.2018.00781>.

**Khan, Shah, Hassan, Junaid, Sardar** (2023). “Intelligent Systems for early malaria disease detection in patient cells using transfer learning approaches”. In: *2023 4th International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. IEEE, pp. 1–6. DOI: [10.1109/iCoMET57998.2023.10099260](https://doi.org/10.1109/iCoMET57998.2023.10099260).

**Kim, Purdie, McIntosh** (2023). “Cross-Task Attention Network: Improving Multi-task Learning for Medical Imaging Applications”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023 Workshops*. Ed. by M. Emre Celebi et al. Cham: Springer Nature Switzerland, pp. 119–128. ISBN: 978-3-031-47401-9. DOI: [10.1007/978-3-031-47401-9\\_12](https://doi.org/10.1007/978-3-031-47401-9_12).

**Kouw, Loog** (2021). “A review of domain adaptation without target labels”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.3, pp. 766–785. DOI: [10.1109/TPAMI.2019.2945942](https://doi.org/10.1109/TPAMI.2019.2945942).

**Krizhevsky, Sutskever, Hinton** (2012). “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems 25 (NeurIPS 2012)*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Curran Associates, Inc., pp. 1097–1105. URL: <https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html>.

**Lab** (2023). *Lacuna Malaria Datasets*. Version 5.0. DOI: [10.7910/DVN/VEADSE](https://doi.org/10.7910/DVN/VEADSE). URL: <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/VEADSE>.

**LeCun, Bengio, Hinton** (2015). “Deep learning”. In: *Nature* 521.7553, pp. 436–444. DOI: [10.1038/nature14539](https://doi.org/10.1038/nature14539).

**LeCun, Bottou, Bengio, Haffner** (1998). “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324. DOI: [10.1109/5.726791](https://doi.org/10.1109/5.726791).

**Li, Ma** (2022). “Residual attention learning network and SVM for malaria parasite detection”. In: *Multimedia Tools and Applications* 81.8, pp. 10935–10960. DOI: [10.1007/s11042-022-12373-6](https://doi.org/10.1007/s11042-022-12373-6).

**Li, Yang, Wu** (2023). “Multi-task learning based histologic subtype classification of non-small cell lung cancer”. In: *La Radiologia Medica* 128.8, pp. 1560–1570. DOI: [10.1007/s11547-023-01632-5](https://doi.org/10.1007/s11547-023-01632-5).

**Lin, Dollár** (2017). “Feature Pyramid Networks for Object Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2117–2125. DOI: [10.1109/CVPR.2017.106](https://doi.org/10.1109/CVPR.2017.106).

**Lin, Goyal, Girshick, He, Dollár** (2017). “Focal Loss for Dense Object Detection”. In: *CoRR* abs/1708.02002. DOI: [10.48550/arXiv.1708.02002](https://doi.org/10.48550/arXiv.1708.02002). arXiv: [1708.02002](https://arxiv.org/abs/1708.02002). URL: [http://arxiv.org/abs/1708.02002](https://arxiv.org/abs/1708.02002).

**Litjens** (2017). “A survey on deep learning in medical image analysis”. In: *Medical Image Analysis* 42, pp. 60–88. DOI: [10.1016/j.media.2017.07.005](https://doi.org/10.1016/j.media.2017.07.005).

**Loddo, Di Ruberto, Kocher, Prod’Hom** (2019). “MP-IDB: The Malaria Parasite Image Database for Image Processing and Analysis”. In: *Processing and Analysis of Biomedical Information*. Ed. by Natasha Lepore, Jorge Brieva, Eduardo Romero, Daniel Racoceanu,

- 
- and Leo Joskowicz. Cham: Springer International Publishing, pp. 57–65. ISBN: 978-3-030-13835-6. URL: <https://github.com/andreoloddo/MP-IDB-The-Malaria-Parasite-Image-Database-for-Image-Processing-and-Analysis>.
- Loshchilov, Hutter** (2019). “Decoupled Weight Decay Regularization”. In: *International Conference on Learning Representations (ICLR)*. arXiv:1711.05101. URL: <https://arxiv.org/abs/1711.05101>.
- Lowe** (2004). “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60.2, pp. 91–110. DOI: [10.1023/B:VISI.0000029664.99615.94](https://doi.org/10.1023/B:VISI.0000029664.99615.94).
- Maturana, Oliveira, Nadal, Bilalli, Serrat** (2022). “Advances and challenges in automated malaria diagnosis using digital microscopy imaging with machine learning tools: a review”. In: *Frontiers in Microbiology* 13. DOI: [10.3389/fmicb.2022.1006659](https://doi.org/10.3389/fmicb.2022.1006659). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9705958/>.
- Medicine (NLM)** (2018). *Malaria Dataset*. <https://lhncbc.nlm.nih.gov/LHC-research/LHC-projects/image-processing/malaria-datasheet.html>. URL: <https://lhncbc.nlm.nih.gov/LHC-downloads/downloads.html#malaria-datasets>.
- Missoup** (2025). “Malaria infection prevalence and diagnostic performance of the Abbott Bioline Malaria Ag P.f/Pan rapid test in rural populations of Central Cameroon”. In: *Scientific Reports* 15, p. 27930. DOI: [10.1038/s41598-025-13688-8](https://doi.org/10.1038/s41598-025-13688-8). URL: <https://www.nature.com/articles/s41598-025-13688-8>.
- Mmileng, Whata, Olusanya, Mhlongo** (2025). “Application of ConvNeXt with transfer learning and data augmentation for malaria parasite detection in resource-limited settings using microscopic images”. In: *PLoS One* 20.6, e0313734. DOI: [10.1371/journal.pone.0313734](https://doi.org/10.1371/journal.pone.0313734).
- Nasra, Gupta, Kumar, Aluvala** (2025). “Automated Malaria Detection Using Deep Learning and the Lacuna Dataset”. In: *2025 3rd International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, pp. 156–161. DOI: [10.1109/InCACCT65424.2025.11011462](https://doi.org/10.1109/InCACCT65424.2025.11011462).
- Nayak, Nayak, Vimal, Arora, Sinha** (2022). “An ensemble artificial intelligence-enabled MIoT for automated diagnosis of malaria parasite”. In: *Expert Systems* 39.4, e12906. DOI: [10.1111/exsy.12906](https://doi.org/10.1111/exsy.12906).
- Ngalame, Watching, Kibu, Zeukóo, Nsagha** (2025). “Assessment of the diagnostic performance of the SD Bioline Malaria antigen test for the diagnosis of malaria in the Tombel health district, Southwest region of Cameroon”. In: *PLoS ONE* 20.3, e0298992. DOI: [10.1371/journal.pone.0298992](https://doi.org/10.1371/journal.pone.0298992). URL: <https://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0298992>.
- Ohdar, Nigam** (2024). “Multi-Species Classification of Malaria Parasites using Ensemble Transfer Learning Technique”. In: *2024 International Conference on Computing, Sciences and Communications (ICCSC)*, pp. 1–7. DOI: [10.1109/ICCSC62048.2024.10830422](https://doi.org/10.1109/ICCSC62048.2024.10830422).
- Ojala, Pietikäinen, Mäenpää** (2002). “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.7, pp. 971–987. DOI: [10.1109/TPAMI.2002.1017623](https://doi.org/10.1109/TPAMI.2002.1017623).

- 
- Omaer Faruq Goni** (2023). "Diagnosis of Malaria Using Double Hidden Layer Extreme Learning Machine Algorithm With CNN Feature Extraction and Parasite Inflator". In: *IEEE Access* 11, pp. 4117–4130. DOI: [10.1109/ACCESS.2023.3234279](https://doi.org/10.1109/ACCESS.2023.3234279).
- Pan, Yang** (2010). "A Survey on Transfer Learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22.10, pp. 1345–1359. DOI: [10.1109/TKDE.2009.191](https://doi.org/10.1109/TKDE.2009.191).
- Pizer** (1987). "Adaptive histogram equalization and its variations". In: *Computer Vision, Graphics, and Image Processing* 39.3, pp. 355–368. DOI: [10.1016/S0734-189X\(87\)80186-X](https://doi.org/10.1016/S0734-189X(87)80186-X).
- Poostchi, Silamut, Maude, Jaeger, Thoma** (2018). "Image analysis and machine learning for detecting malaria". In: *Translational Research* 194, pp. 36–55. DOI: [10.1016/j.trsl.2017.12.004](https://doi.org/10.1016/j.trsl.2017.12.004). URL: <https://pubmed.ncbi.nlm.nih.gov/29360430/>.
- Prosser** (2021). "Plasmodium falciparum Histidine-Rich Protein 2 and 3 Gene Deletions in Strains from Nigeria, Sudan, and South Sudan". In: *Emerging Infectious Diseases* 27.2, pp. 471–479. DOI: [10.3201/eid2702.191410](https://doi.org/10.3201/eid2702.191410). URL: [https://wwwnc.cdc.gov/eid/article/27/2/19-1410\\_article](https://wwwnc.cdc.gov/eid/article/27/2/19-1410_article).
- Raffel** (2020). "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *Journal of machine learning research* 21.140, pp. 1–67. URL: <https://arxiv.org/abs/1910.10683v4>.
- Rajaraman** (2018). "Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images". In: *PeerJ*. Dataset available at: <https://lhncbc.nlm.nih.gov/LHC-research/LHC-projects/image-processing/malaria-datasheet.html>. DOI: [10.7717/peerj.4568](https://doi.org/10.7717/peerj.4568). URL: <https://lhncbc.nlm.nih.gov/LHC-downloads/downloads.html#malaria-datasets>.
- Rajpurkar** (2017). "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning". In: *arXiv preprint arXiv:1711.05225*. arXiv: [1711.05225](https://arxiv.org/abs/1711.05225). URL: <https://arxiv.org/abs/1711.05225>.
- Ramos** (2024). "A transfer learning approach to identify Plasmodium in microscopic images". In: *PLOS Computational Biology* 20.8, e1012327. DOI: [10.1371/journal.pcbi.1012327](https://doi.org/10.1371/journal.pcbi.1012327).
- Ramos-Briceño, Flammia-D'Aleo, Fernández-López, Carrión-Nessi, Forero-Peña** (2025). "Deep learning-based malaria parasite detection: convolutional neural networks model for accurate species identification of Plasmodium falciparum and Plasmodium vivax". In: *Scientific Reports* 15.1, p. 3746. DOI: [10.1038/s41598-025-87979-5](https://doi.org/10.1038/s41598-025-87979-5).
- Ren, He, Girshick, Sun** (2017). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6, pp. 1137–1149. DOI: [10.1109/TPAMI.2016.2577031](https://doi.org/10.1109/TPAMI.2016.2577031).
- Ronneberger, Fischer, Brox** (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Cham: Springer International Publishing, pp. 234–241. DOI: [10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28). URL: <http://lmb.informatik.uni-freiburg.de/people/ronneber/u-net>.
- Ruder** (2017). "An Overview of Multi-Task Learning in Deep Neural Networks". In: *arXiv preprint arXiv:1706.05098*. arXiv: [1706.05098](https://arxiv.org/abs/1706.05098). URL: <https://arxiv.org/abs/1706.05098>.

- 
- Sachs, Commission on Macroeconomics and Health** (2001). *Macroeconomics and Health: Investing in Health for Economic Development*. Tech. rep. Report of the Commission on Macroeconomics and Health. Geneva: World Health Organization. URL: <https://apps.who.int/iris/handle/10665/42435>.
- Schoenpflug, Lafarge, Frei, Koelzer** (2023). "Multi-task learning for tissue segmentation and tumor detection in colorectal cancer histology slides". In: *arXiv preprint*. DOI: [10.48550/arXiv.2304.03101](https://doi.org/10.48550/arXiv.2304.03101).
- Siłka, Wieczorek, Siłka, Woźniak** (2023). "Malaria Detection Using Advanced Deep Learning Architecture". In: *Sensors* 23.3. ISSN: 1424-8220. DOI: [10.3390/s23031501](https://doi.org/10.3390/s23031501). URL: <https://www.mdpi.com/1424-8220/23/3/1501>.
- Simonyan, Zisserman** (2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *arXiv preprint arXiv:1409.1556*. arXiv: [1409.1556](https://arxiv.org/abs/1409.1556). URL: <https://arxiv.org/abs/1409.1556>.
- Sora-Cardenas** (2025). "Image-Based Detection and Classification of Malaria Parasites and Leukocytes with Quality Assessment of Romanowsky-Stained Blood Smears". In: *Sensors* 25.2, p. 390. DOI: [10.3390/s25020390](https://doi.org/10.3390/s25020390).
- Srivastava, Hinton, Krizhevsky, Sutskever, Salakhutdinov** (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *J. Mach. Learn. Res.* 15.1, pp. 1929–1958. DOI: [10.5555/2627435.2670313](https://doi.org/10.5555/2627435.2670313). URL: <http://jmlr.org/papers/v15/srivastava14a.html>.
- Sukumarran** (June 2023). "An automated malaria cells detection from thin blood smear images using deep learning". eng. In: *Tropical Biomedicine* 40.2, pp. 208–219. ISSN: 2521-9855. DOI: [10.47665/tb.40.2.013](https://doi.org/10.47665/tb.40.2.013).
- Szegedy, Liu** (2015a). "Going Deeper with Convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9. DOI: [10.1109/CVPR.2015.7298594](https://doi.org/10.1109/CVPR.2015.7298594).
- (2015b). "Going Deeper with Convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9. DOI: [10.1109/CVPR.2015.7298594](https://doi.org/10.1109/CVPR.2015.7298594).
- Szegedy, Vanhoucke, Ioffe, Shlens, Wojna** (2016). "Rethinking the Inception Architecture for Computer Vision". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826. DOI: [10.1109/CVPR.2016.308](https://doi.org/10.1109/CVPR.2016.308). URL: <https://doi.org/10.1109/CVPR.2016.308>.
- Tan, Le** (2019). "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *Proceedings of the 36th International Conference on Machine Learning (ICML)*. Vol. 97. PMLR, pp. 6105–6114.
- Tibshirani** (1996). "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1, pp. 267–288. DOI: [10.1111/j.2517-6161.1996.tb02080.x](https://doi.org/10.1111/j.2517-6161.1996.tb02080.x). URL: <https://doi.org/10.1111/j.2517-6161.1996.tb02080.x>.
- Ufuktepe** (2021). "Deep Learning-Based Cell Detection and Extraction in Thin Blood Smears for Malaria Diagnosis". In: *2021 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pp. 1–6. DOI: [10.1109/AIPR52630.2021.9762109](https://doi.org/10.1109/AIPR52630.2021.9762109).

- 
- Wang, Dai, Póczos, Carbonell** (2019). "Characterizing and Avoiding Negative Transfer". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 11293–11302. DOI: [10.1109/CVPR.2019.01155](https://doi.org/10.1109/CVPR.2019.01155).
- Wongsrichanalai, Barcus, Muth, Sutamihardja, Wernsdorfer** (2007). "A review of malaria diagnostic tools: microscopy and rapid diagnostic test (RDT)". In: *Defining and Defeating the Intolerable Burden of Malaria III: Progress and Perspectives*. Ed. by J. G. Breman, M. S. Alilio, and N. J. White. American Society of Tropical Medicine and Hygiene, pp. 119–127. URL: <https://www.ncbi.nlm.nih.gov/books/NBK1695/>.
- World Health Organization** (2016). *Malaria microscopy quality assurance manual – Version 2*. Geneva: World Health Organization. ISBN: 9789241549394. URL: <https://wkc.who.int/resources/publications/i/item/9789241549394>.
- (2023). *World malaria report 2024*. Geneva: World Health Organization. URL: <https://www.who.int/teams/global-malaria-programme/reports/world-malaria-report-2024>.
- Yadav, Kumar, Kannagi, Sahoo** (2024). "Exploring Multi-Task Learning for Transfer Learning Based Active Learning in Medical Image Segmentation". In: *Proceedings of the 5th International Conference on Data Science, Machine Learning and Applications (ICDSMLA)*. Springer, pp. 559–565. DOI: [10.1007/978-981-97-8043-3\\_88](https://doi.org/10.1007/978-981-97-8043-3_88).
- Yosinski, Clune, Bengio, Lipson** (2014). "How transferable are features in deep neural networks?" In: *Advances in Neural Information Processing Systems 27 (NIPS 2014)*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. Curran Associates, Inc., pp. 3320–3328. URL: <https://proceedings.neurips.cc/paper/2014/file/375c71349b295fbe2dcdca9206f20a06-Paper.pdf>.
- Yu** (2020). "Gradient Surgery for Multi-Task Learning". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33, pp. 5824–5836. DOI: [10.48550/arXiv.2001.06782](https://doi.org/10.48550/arXiv.2001.06782). URL: <https://proceedings.neurips.cc/paper/2020/file/3fe78a8acf5fda99de95303940a2420c-Paper.pdf>.
- Zhang, Xu** (2019). "Multiview Transfer Learning and Multitask Learning". In: *Multiview Machine Learning*. Ed. by Guest Editor(s). Springer, pp. 85–104. DOI: [10.1007/978-981-13-3029-2\\_7](https://doi.org/10.1007/978-981-13-3029-2_7).
- Zhuang** (2020). "A comprehensive survey on transfer learning". In: *Proceedings of the IEEE* 109.1, pp. 43–76. DOI: [10.1109/JPROC.2020.3004555](https://doi.org/10.1109/JPROC.2020.3004555).

---

## Experimental Configurations

---

This appendix details the configurations for our Single-Task Learning (STL) and Multi-Task Transfer Learning (MTTL) models.

### A.1 Training and Optimization Strategy

We applied a **Hybrid tuning strategy** across all models. The backbone remained frozen except for the final residual block (layer4), which was fine-tuned at a low learning rate ( $1 \times 10^{-6}$ ) to refine high-level features. Task-specific heads and LoRA adapters were trained at a higher rate ( $1 \times 10^{-3}$ ).

Optimization utilized **AdamW** with a weight decay of  $1 \times 10^{-4}$  and **Automatic Mixed Precision (AMP)**, following a linear warmup and cosine annealing schedule. For model selection, STL training stopped when the primary validation metric plateaued. For MTTL, we used a **composite score** of weighted task metrics to guide early stopping. We implemented a dual checkpointing system: the backbone was saved based on the best composite score, while individual heads were saved whenever they achieved their specific peak performance.

### A.2 Model Configurations

Table A.1 lists the best STL configurations. Table A.2 outlines the consistent setup used for all MTTL experiments, while Table A.3 provides the specific schedules and weighting schemes for each combination.

Table A.1: Single-Task Learning (STL) Champion Model Configurations.

Task	Classes	Batch Size	Epochs	LR Head	LR Backbone	Warmup
Det	1	16	150	$1 \times 10^{-3}$	$1 \times 10^{-6}$	15 epochs
Det	2	16	150	$1 \times 10^{-3}$	$1 \times 10^{-6}$	15 epochs
Det	3	16	200	$1 \times 10^{-3}$	$1 \times 10^{-6}$	20 epochs
RoI Classif	2	16	150	$1 \times 10^{-3}$	$1 \times 10^{-6}$	15 epochs
RoI Classif	3	16	200	$1 \times 10^{-3}$	$1 \times 10^{-6}$	20 epochs
Seg	–	16	200	$1 \times 10^{-3}$	$1 \times 10^{-6}$	20 epochs
Loc	–	16	200	$1 \times 10^{-3}$	$1 \times 10^{-6}$	15 epochs

Table A.2: Multi-Task Transfer Learning (MTTL) Configurations.

Parameter	Value
Batch Size	8 (with 2 gradient accumulation steps)
Optimizer	AdamW
Learning Rate (Heads & Adapters)	$1 \times 10^{-3}$
Learning Rate (Backbone)	$1 \times 10^{-6}$
Scheduler	Cosine
<i>Task Weighting Schemes (Primary Task Weights)</i>	
Det + Seg	Det: 0.7, Seg: 0.3
Det + Loc	Det: 0.7, Loc: 0.3
Det + RoI (2- or 3-Class)	Det: 0.7, RoI: 0.3
Det + Seg + Loc	Det: 0.6, Seg: 0.2, Loc: 0.2
Det + Seg + RoI	Det: 0.6, Seg: 0.2, RoI: 0.2
Det + Loc + RoI	Det: 0.6, Loc: 0.2, RoI: 0.2
Det + Seg + Loc + RoI	Det: 0.5, Seg: 0.2, Loc: 0.1, RoI: 0.2

Table A.3: Detailed training configurations for MTTL task combinations.

Tasks	Classes	Epc,W	Patience	Primary Task Weights
<i>Combinations with 1-Class Detection</i>				
Det + Seg	Det: 1	250, 25	30	Det: 0.7, Seg: 0.3
Det + Loc	Det: 1	250, 25	30	Det: 0.7, Loc: 0.3
Det + Seg + Loc	Det: 1	300, 30	30	Det: 0.6, Seg: 0.2, Loc: 0.2
<i>Combinations with 2-Class Detection / RoI</i>				
Det + Seg	Det: 2	325, 30	30	Det: 0.7, Seg: 0.3
Det + Loc	Det: 2	325, 30	30	Det: 0.7, Loc: 0.3
Det + RoI	Det: 2, RoI: 2	325, 30	30	Det: 0.7, RoI: 0.3
Det + Seg + Loc	Det: 2	375, 30	35	Det: 0.6, Seg: 0.2, Loc: 0.2
Det + Seg + RoI	Det: 2, RoI: 2	375, 30	35	Det: 0.6, Seg: 0.2, RoI: 0.2
Det + Loc + RoI	Det: 2, RoI: 2	375, 30	35	Det: 0.6, Loc: 0.2, RoI: 0.2
Det + Seg + Loc + RoI	Det: 2, RoI: 2	400, 40	40	Det: 0.5, Seg: 0.2, Loc: 0.1, RoI: 0.2
<i>Combinations with 3-Class Detection / RoI</i>				
Det + Seg	Det: 3	375, 30	40	Det: 0.7, Seg: 0.3
Det + Loc	Det: 3	375, 30	40	Det: 0.7, Loc: 0.3
Det + RoI	Det: 3, RoI: 3	375, 30	40	Det: 0.7, RoI: 0.3
Det + Seg + Loc	Det: 3	400, 40	40	Det: 0.6, Seg: 0.2, Loc: 0.2
Det + Seg + RoI	Det: 3, RoI: 3	400, 40	40	Det: 0.6, Seg: 0.2, RoI: 0.2
Det + Loc + RoI	Det: 3, RoI: 3	400, 30	40	Det: 0.6, Loc: 0.2, RoI: 0.2
Det + Seg + Loc + RoI	Det: 3, RoI: 3	400, 40	40	Det: 0.5, Seg: 0.2, Loc: 0.1, RoI: 0.2

---

## Performance Leaderboards for All Tasks

---

This appendix summarizes the final performance of all STL and MTTL models. For each primary task, we compare the top STL baseline with the relevant MTTL configurations, for a direct assessment of performance and task synergy. Models are ranked by the most relevant metric F1(Inf) for detection, macro F1 for RoI classification, and Dice for segmentation and heatmap localization.

### B.1 Detection Performance and Synergy

Detection was evaluated in 1-, 2-, and 3-class settings. The tables report MTTL performance relative to the strongest STL baseline, with  $\Delta$  F1(Inf) highlighting the impact of multi-task learning.

Table B.1: MTTL results for **1-Class Detection**.

<b>Task(s)</b>	<b>Paradigm</b>	<b>mAP@50</b>	<b>F1(Inf)</b>	<b>R(Inf)</b>	<b>P(Inf)</b>
<i>Det(1) Baseline</i>	STL	<b>0.8260</b>	<b>0.8182</b>	<b>0.8218</b>	<b>0.8146</b>
Det(1)+Seg	MTTL	0.7885	0.7654 (-6.5%)	0.6953	0.8514
Det(1)+Loc	MTTL	0.7367	0.7285 (-11.0%)	0.7098	0.7483
Det(1)+Seg+Loc	MTTL	0.7709	0.7497 (-8.4%)	0.7388	0.7609

Table B.2: MTTL results for **2-Class Detection**.

<b>Task(s)</b>	<b>Paradigm</b>	<b>mAP@50</b>	<b>F1(Inf)</b>	<b>F1-Macro</b>
<i>Det(2) Baseline</i>	STL	0.5834	0.5990	0.9274
Det(2)+RoI(2)	MTTL	<b>0.6452</b>	<b>0.7810 (+30.4%)</b>	<b>0.9411</b>
Det(2)+Seg	MTTL	0.6281	0.7464 (+24.6%)	0.9330
Det(2)+Loc	MTTL	0.6398	0.7415 (+23.8%)	0.9282
Det(2)+Seg+RoI(2)	MTTL	0.6113	0.6976 (+16.5%)	0.9256
Det(2)+Seg+Loc	MTTL	0.6025	0.6120 (+2.2%)	0.9025
Det(2)+RoI(2)+Loc	MTTL	0.5958	0.6065 (+1.3%)	0.8958
Det(2)+Seg+RoI(2)+Loc	MTTL	0.5872	0.6030 (+0.7%)	0.8897

Table B.3: MTTL results for 3-Class Detection.

Task(s)	Paradigm	mAP@50	F1(Inf)	F1-Macro
<i>Det(3) Baseline</i>	STL	0.3341	0.3791	0.8977
Det(3)+RoI(3)	MTTL	<b>0.7402</b>	<b>0.7710 (+103.4%)</b>	<b>0.9409</b>
Det(3)+Seg	MTTL	0.7013	0.7230 (+90.7%)	0.9351
Det(3)+Loc	MTTL	0.6994	0.7034 (+85.6%)	0.9195
Det(3)+Seg+RoI(3)	MTTL	0.7089	0.6662 (+75.7%)	0.9289
Det(3)+Loc+RoI(3)	MTTL	0.7289	0.6722 (+77.3%)	0.9375
Det(3)+Seg+Loc	MTTL	0.6913	0.5468 (+44.2%)	0.9362
Det(3)+Seg+RoI(3)+Loc	MTTL	0.7324	0.7360 (+94.2%)	0.9364

## B.2 Auxiliary Task Leaderboards

We now present the leaderboards for the auxiliary tasks to evaluate how they perform as standalone specialists (STL) versus when they are part of a larger multi-task system (MTTL).

Table B.4: Segmentation Performance Leaderboard (Ranked by Dice).

Task(s)	Paradigm	Dice	IoU
Seg	STL	<b>0.9656</b>	<b>0.9468</b>
Det(2)+Seg	MTTL	0.9575	0.9332
Det(3)+Seg	MTTL	0.9564	0.9327
Det(3)+Seg+RoI(3)+Loc	MTTL	0.9518	0.9293
Det(2)+Seg+RoI(2)	MTTL	0.9504	0.9264
Det(3)+Seg+Loc	MTTL	0.9502	0.9276

Table B.5: Heatmap Localization Performance Leaderboard (Ranked by Dice).

Task(s)	Paradigm	Dice	Pearson	MAE
Loc	STL	<b>0.5422</b>	<b>0.7866</b>	<b>0.9719</b>
Det(3)+Loc	MTTL	0.5108	0.7680	0.9900
Det(1)+Seg+Loc	MTTL	0.5081	0.7636	0.9893
Det(3)+Loc+RoI(3)	MTTL	0.5007	0.7571	0.9891
Det(1)+Loc	MTTL	0.4975	0.7541	0.9898
Det(2)+Loc	MTTL	0.0557	0.1698	0.5930
Det(3)+Seg+Loc+RoI(3)	MTTL	0.0358	0.1565	0.3221

Table B.6: RoI Classification Performance Leaderboard (Ranked by F1-Macro).

Task(s)	Paradigm	Classes	Accuracy	F1-Macro
<b>2-Class Classification</b>				
RoI(2)	STL	2	<b>0.9859</b>	<b>0.9029</b>
Det(2)+RoI(2)	MTTL	2	0.9832	0.9024
Det(2)+Seg+RoI(2)	MTTL	2	0.9626	0.8278
<b>3-Class Classification</b>				
Det(3)+RoI(3)	MTTL	3	<b>0.9802</b>	<b>0.9103</b>
Det(3)+Seg+RoI(3)+Loc	MTTL	3	0.9709	0.8732
Det(3)+Loc+RoI(3)	MTTL	3	0.9470	0.8326
RoI(3)	STL	3	0.9373	0.8047
Det(3)+Seg+RoI(3)	MTTL	3	0.9296	0.7618

**Observations:** The synergy of MTTL is highly dependent on the complexity of the primary task, in our case object detection. For simple 1-class detection, all auxiliary tasks introduce a performance penalty (ranging from  $-6.5\%$  to  $-11.0\%$ ). However, as the task shifts to the more diagnostically realistic 2-class and 3-class problems, MTTL demonstrates its power. For 2-class detection, RoI Classification provides the most significant boost (+30.4%), followed by Segmentation (+24.6%) and Localization (+23.8%). For 3-class detection, RoI Classification becomes the most effective auxiliary task, yielding a remarkable +103.4% improvement in  $F1(\text{Inf})$  over the weak STL baseline, with Segmentation also providing substantial gains (+90.7%) and Localization contributing +85.6%. This highlights that as the primary task becomes more challenging, the regularization and shared representations learned from auxiliary tasks become increasingly beneficial.

For the auxiliary tasks themselves (Segmentation, Heatmap Localization, and RoI Classification), the specialist STL models consistently outperform their MTTL counterparts on their respective primary metrics. Segmentation achieves Dice = 0.9656 (STL) versus 0.9575 (best MTTL), Heatmap Localization reaches Dice = 0.5422 (STL) versus 0.5108 (best MTTL), while RoI Classification shows a notable exception: for 3-class problems, MTTL ( $F1 = 0.9103$ ) surpasses STL ( $F1 = 0.8047$ ) by +13.1%, demonstrating that the shared representations in multi-task learning can benefit auxiliary tasks when the primary task provides complementary supervision. For 2-class RoI Classification, STL and MTTL perform nearly identically (0.9029 versus 0.9024). This pattern reveals a key trade-off: while auxiliary tasks can significantly boost primary task performance (especially in complex scenarios like 3-class detection), they typically do so at the cost of their own specialist performance, as the shared backbone must find a compromise representation serving all tasks simultaneously.

## Qualitative Examples

This appendix provides qualitative examples to visually complement the quantitative leaderboards in Appendix B. We compare the outputs of the best Single-Task (STL) and Multi-Task (MTTL) models against the ground truth for each core task. These visualizations offer tangible insights into the practical strengths and weaknesses of each learning paradigm.

### C.1 Auxiliary Segmentation and Localization

We begin by visualizing performance on the pixel- and region-level auxiliary tasks, as their ability to provide spatial priors is critical to the success of MTTL.

For segmentation, Figure C.1 compares the predicted cell masks from the best STL and MTTL models against the ground truth. While both models produce high-quality outputs, the specialist STL model's mask is visibly cleaner and aligns more precisely with the ground truth boundaries. This visual evidence supports the quantitative findings that specialist models achieve peak performance on their designated task, as the shared backbone in MTTL must find a representation that is a compromise across all tasks.

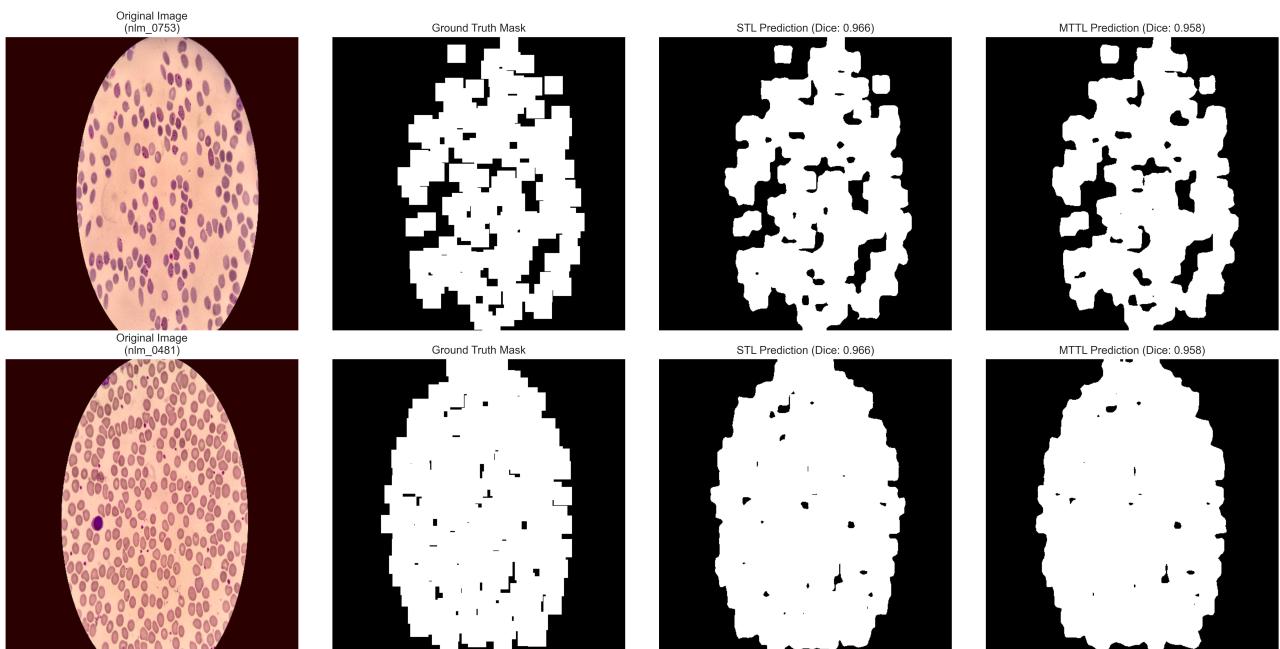


Figure C.1: Qualitative comparison for Cell Segmentation.

Both paradigms locate infections (Figures C.2, C.3), but the STL maps are sharper. This reflects the trade-off where auxiliary tasks sacrifice their own precision to regularize the primary detector.

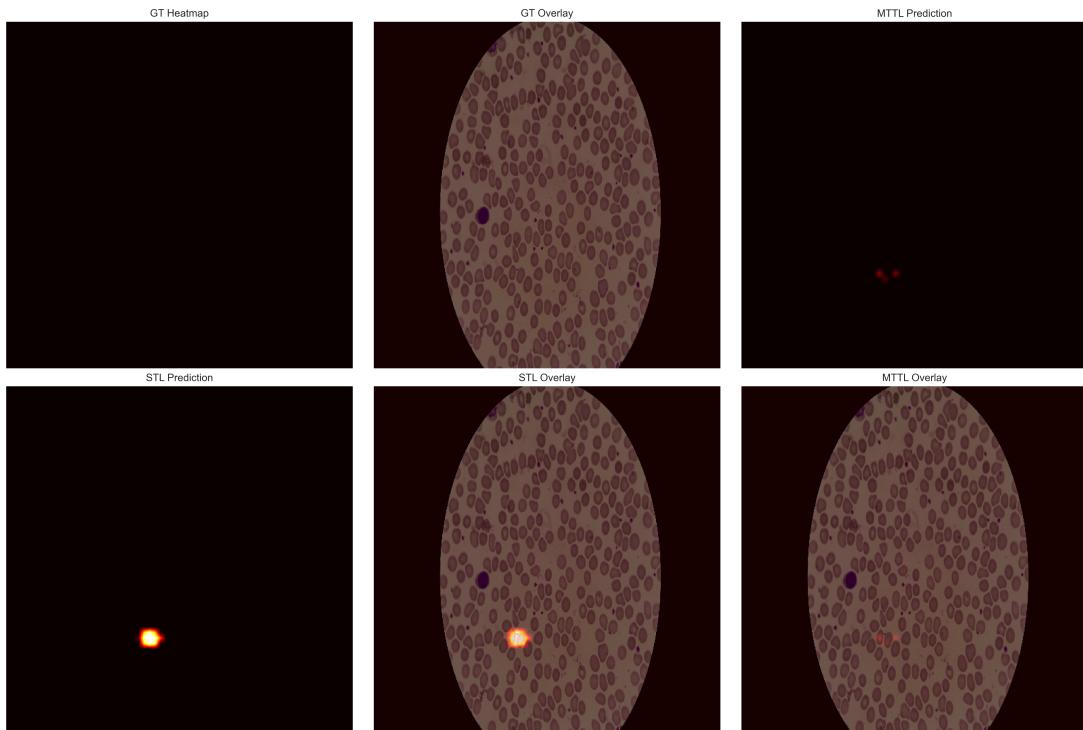


Figure C.2: Qualitative comparison for Heatmap Localization.

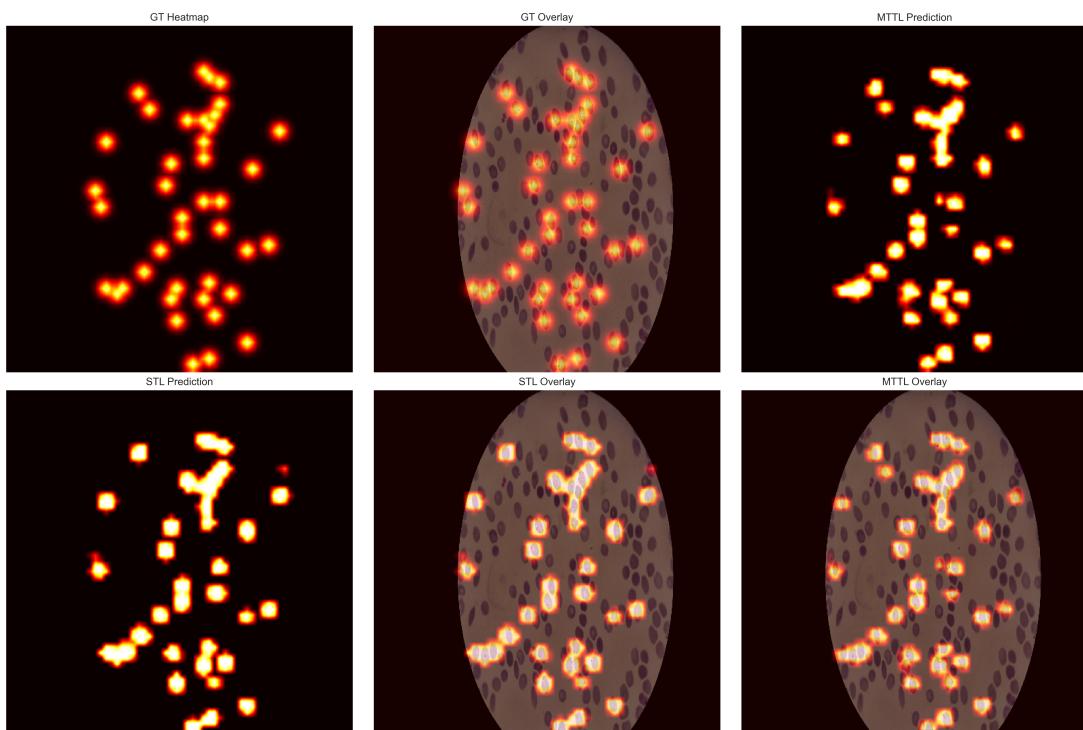


Figure C.3: Qualitative comparison for Heatmap Localization.

## C.2 Auxiliary ROI Classification

To analyze the classification capabilities at a granular level, we perform inference on a curated set of ground truth cell patches. Figure C.4 shows the cropped cell image alongside the predicted probability distributions from the STL and MTTL models for a 3-class task. Both models demonstrate high accuracy.

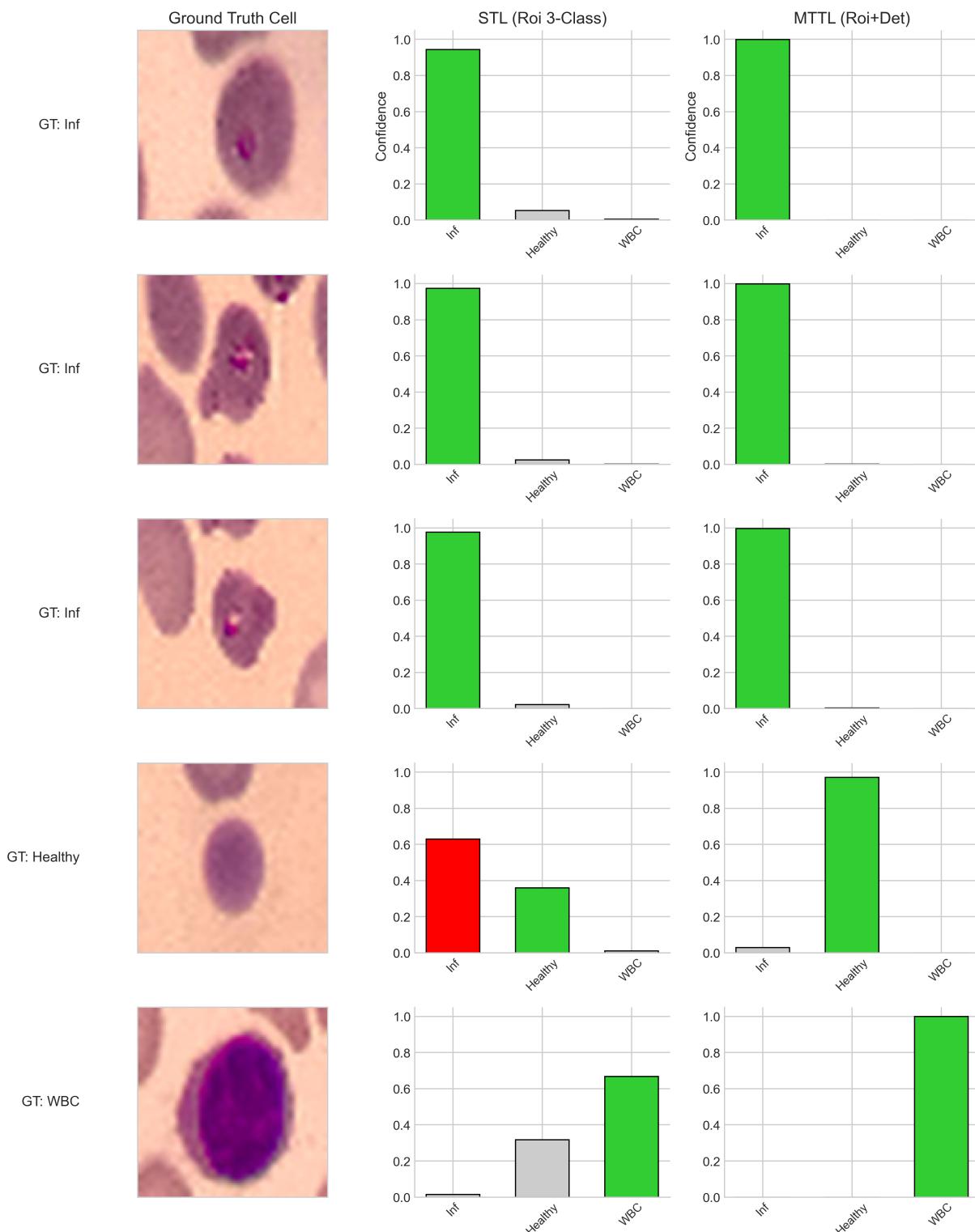


Figure C.4: Qualitative comparison for 3-Class ROI Classification.

## C.3 Detection Performance

Finally, we visualize the performance on the primary detection task.

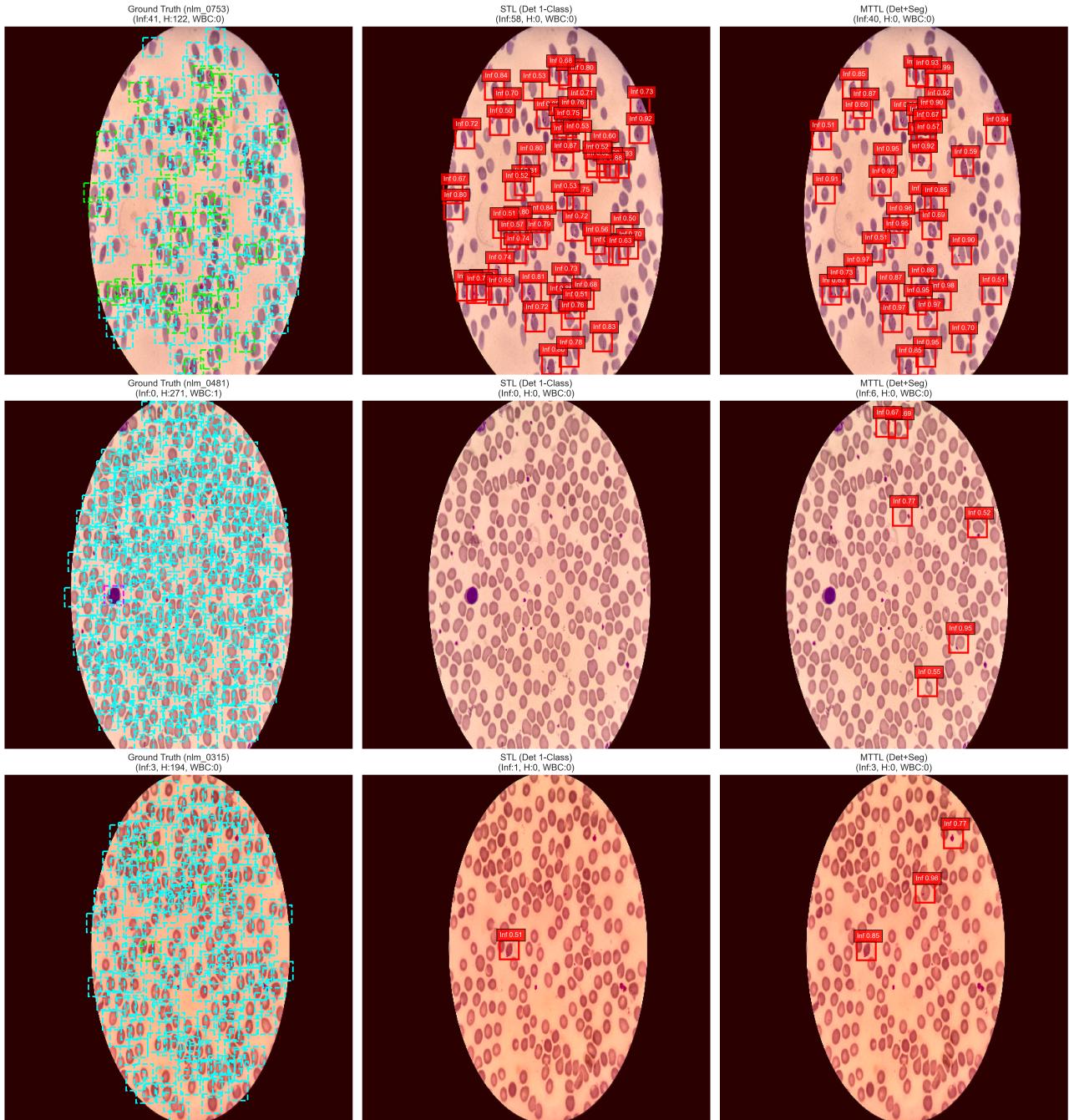


Figure C.5: Qualitative comparison for 1-Class Detection.

- |                   |                  |              |
|-------------------|------------------|--------------|
| — Infected (GT)   | — Healthy (GT)   | — WBC (GT)   |
| — Infected (Pred) | — Healthy (Pred) | — WBC (Pred) |

In the simple 1-class setting (Figure C.5), the specialist STL model demonstrates excellent precision and recall. The MTTL model, however, produces several false negatives, visually confirming the negative synergy observed in the quantitative results when auxiliary tasks are applied to a simple primary objective.

As task complexity increases to the diagnostically realistic 3-class problem (Figure C.7), the

limitations of the STL approach become stark. The specialist model struggles to generalize, missing numerous infected cells. The MTTL model, in contrast, leverages the regularizing effects of its auxiliary tasks to maintain robust performance, accurately identifying most cells from all three classes. This figure provides the core visual evidence for our work MTTL is a superior paradigm for developing comprehensive and effective diagnostic models in complex settings.

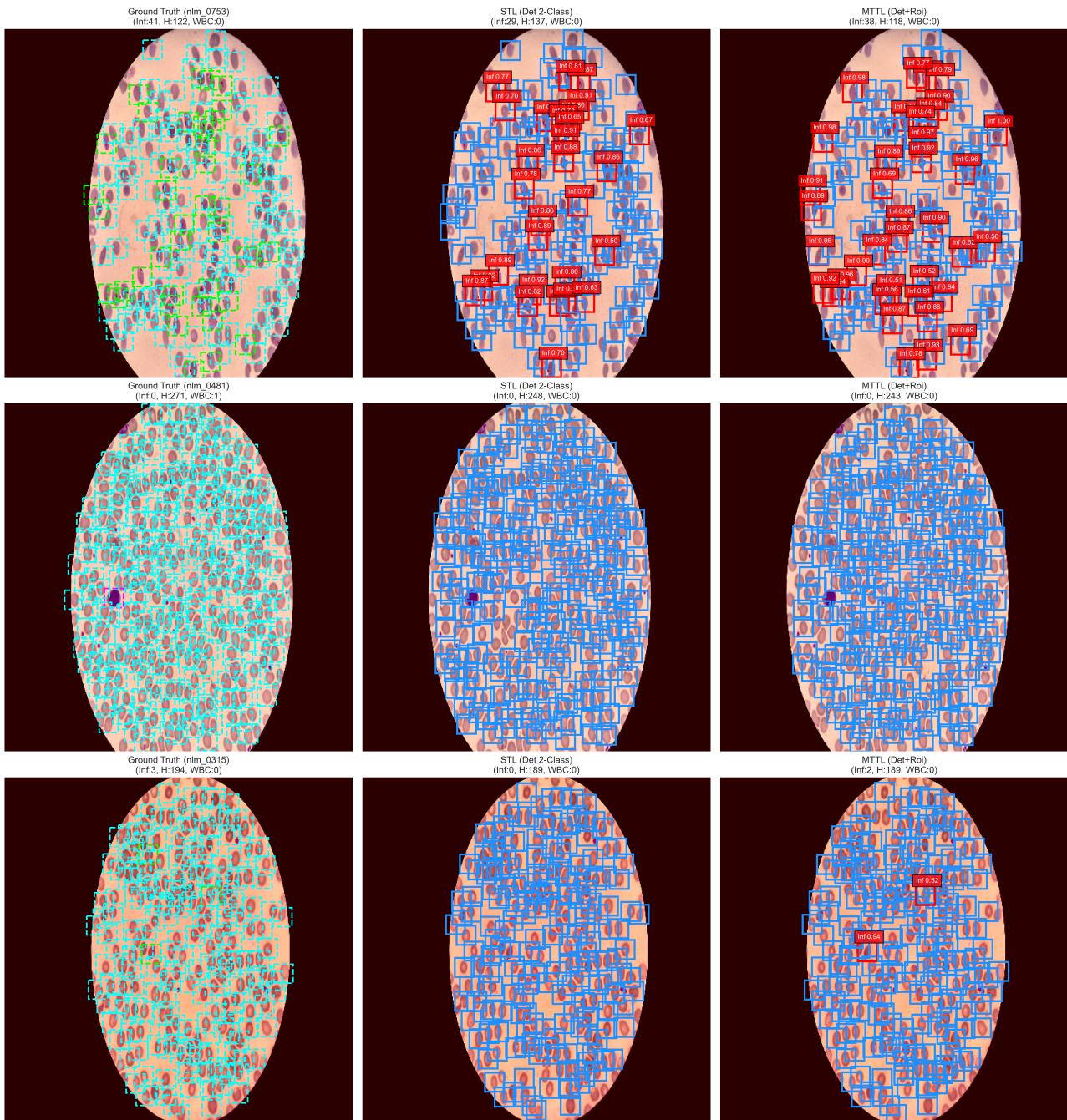


Figure C.6: Qualitative comparison for 2-Class Detection.

... Infected (GT)	... Healthy (GT)	... WBC (GT)
— Infected (Pred)	— Healthy (Pred)	— WBC (Pred)

In the most challenging 3-class problem (Figure C.7), the limitations of the STL approach become stark. The specialist model struggles to generalize, missing numerous infected cells.

The MTTL model, in contrast, leverages the regularizing effects of its auxiliary tasks to maintain robust performance, accurately identifying most cells from all three classes. This figure provides the core visual evidence for our work: MTTL is a superior paradigm for developing comprehensive and effective diagnostic models in complex settings.

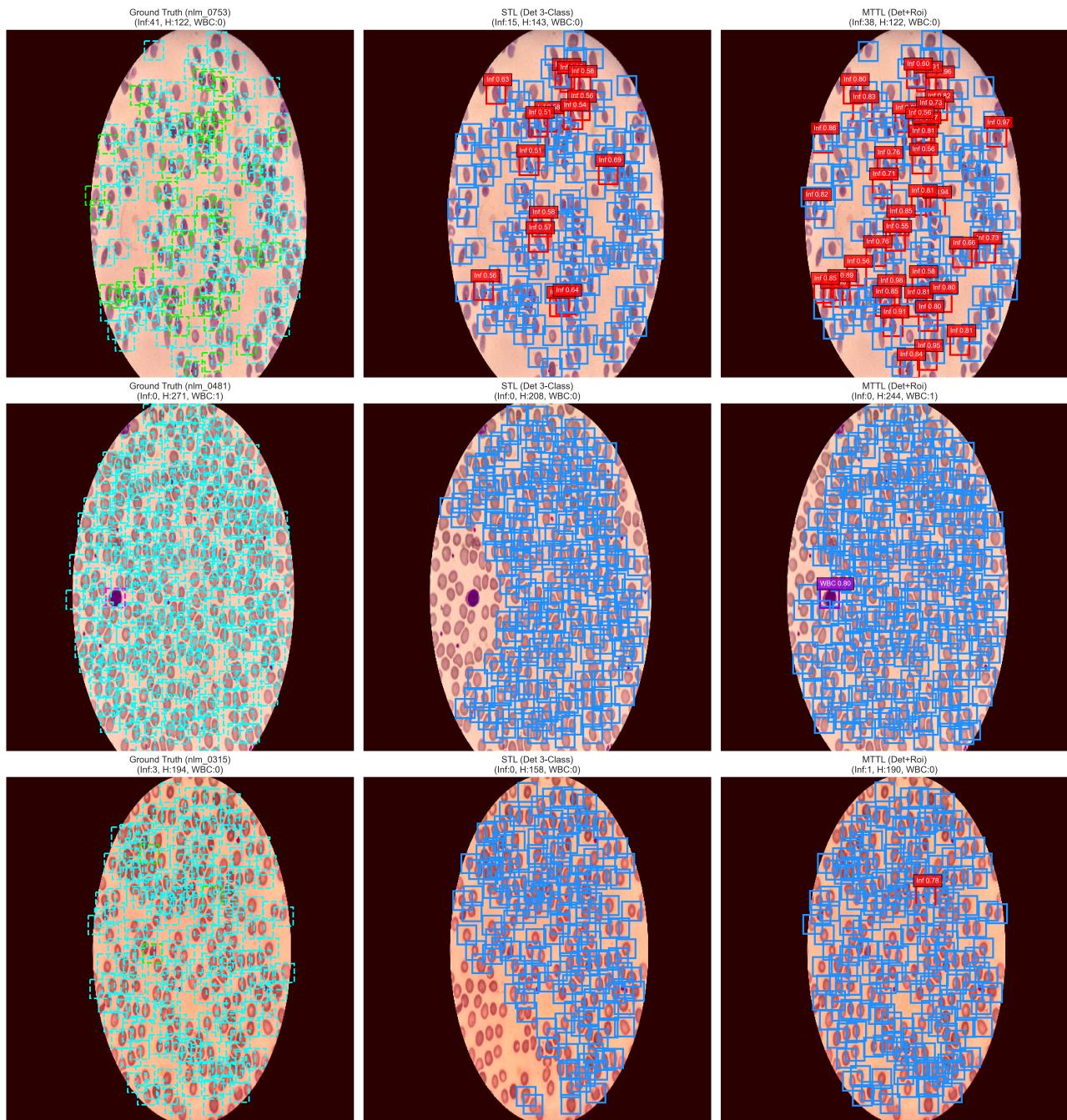


Figure C.7: Qualitative comparison for 3-Class Detection.

- |  |  |  |
|--|--|--|
| <span style="color: green;">---</span> Infected (GT)<br><span style="color: red;">---</span> Infected (Pred) | <span style="color: blue;">---</span> Healthy (GT)<br><span style="color: blue;">---</span> Healthy (Pred) | <span style="color: purple;">---</span> WBC (GT)<br><span style="color: purple;">---</span> WBC (Pred) |
|--|--|--|