Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research
Echahid Hamma Lakhdar University of El Oued

Faculty of Exact Sciences
Department of Computer Sciences

Ref : .....

Thesis Presented with a view to obtaining the
Master Academic degree in computer Science
Option: Artificial Intelligence and Data Science

Entitled:

Malaria Disease Detection via Deep Learning Methods Derived from Blood Cell Images

Presented by:
MEDAOUI Ahmed
KEHILI Anouar

Supervisor:
Tarablesse Settou

Publicly defended on :**26/05/2025**

In front of the Jury committee composed of:

GHERBI Kaddour	University of El-Oued	President
ZAOUI Sayah	University of El-Oued	Discussed

Dedication

From Medaoui:

To my father Medaoui Boubaker…
To the one who left so suddenly, without warning…
You left in a moment when I needed you the most.
You were gone, and I stood alone, broken…
No hand to hold, no voice to calm me, no smile to carry me through.

Dad,
You were not just my father.
You were my friend, my joy.
We used to laugh, watch football matches,
You supported one team, I supported another — we argued, we laughed…
Those were the best days of my life.

Even now, I still can't believe you're gone.
Even when I kissed your forehead for the last time…
I hoped you would wake up and smile again.
My tears wished they could bring you back…
Because you weren't just my dad — you were everything.

I've never felt pain like this.
I froze the day you left, but I stood tall for our family.
Because I am your son… and I had to be strong.

I dedicate this graduation to you, Dad.
To your heart that taught me strength,
To your smile that still lives in my memory.

Dad…
Your son didn't fail you. He kept his promise.

Acknowledgments

From both authors:

We thank Allah Almighty who gave us strength and patience —
the patience to complete this work.

We send our sincere thanks, deep appreciation, and respect
to our supervisor, Mrs. Tarablesse Settou,
for accepting to guide us, and for the valuable advice and direction
she gave us during this project.
Without her help, this work would not have been possible.

We also sincerely thank the members of the jury
for giving us the honor of evaluating our work.

Our thanks also go to everyone who supported us —
especially the teachers from the Computer Science Department
at Hamma Lakhdar University in El-Oued,
for their help during the training period.

We also thank our families, the Kehili and Medaoui families,
our dear friends,
and everyone who contributed — closely or from afar —
to the development of this work.

Abstract

Malaria remains a significant global health challenge, claiming hundreds of thousands of lives each year, thereby highlighting the urgent need for rapid and accurate diagnostic methods. Traditional diagnostic techniques, such as microscopy and rapid diagnostic tests, are often labor-intensive, time-consuming, and prone to human error.

Although many machine learning-based studies have aimed to achieve automated detection of malaria-infected cells, their effectiveness is limited by challenges such as variability in microscopic red blood cell images, including variations in cell shape, density, and staining properties, as well as ambiguity in distinguishing between certain cell classes.

Recently, Deep learning-based methods have demonstrated as powerful tools for automatic feature extraction and detection in various image recognition tasks, including malaria diagnosis.

To this end, this study proposes a deep learning-based approach for the automatic classification of malaria-infected and uninfected red blood cells using labeled high-resolution microscopic images. The main contribution is the application of EfficientNetB2 through transfer learning, which is known for its optimized performance-to-complexity ratio. This model is employed to extract discriminative features from red blood cell images, enabling robust classification of parasitized and uninfected cells. The model is fine-tuned on a publicly available dataset from the National Institutes of Health (NIH), which includes 27,558 microscopic images of red blood cells. Experimental results demonstrate that the EfficientNetB2-based model significantly improves performance, achieving an accuracy of 99.35%, and outperforms both traditional methods and most previous studies

Keywords: Malaria diagnosis, Microscopic red blood cell images, Automated Malaria classification, Parasitized and uninfected cells, Deep learning, Convolutional Neural Networks, EfficientNetB2, Transfer learning.

ملخص

لا تزال الملاريا تمثل تحديًا كبيرًا للصحة العالمية، حيث تحصد أرواح مئات الآلاف من الأشخاص سنويًا، مما يبرز الحاجة الملحة إلى وسائل تشخيص سريعة ودقيقة. إن طرق التشخيص التقليدية، مثل الفحص المجهري والاختبارات التشخيصية السريعة، غالبًا ما تكون مرهقة وتتطلب وقتًا طويلًا، كما أنها عرضة لأخطاء بشرية.

وعلى الرغم من أن العديد من الدراسات المعتمدة على تعلم الآلة تهدف إلى الكشف التلقائي عن الخلايا المصابة بالملاريا، إلا أن فعاليتها لا تزال محدودة بسبب تحديات مثل التباين في صور الخلايا الدموية تحت المجهر، بما في ذلك اختلافات في شكل الخلايا وكثافتها وخصائص الصبغة، بالإضافة إلى صعوبة التمييز بين بعض أنواع الخلايا.

ومؤخرًا، أثبتت الأساليب المعتمدة على التعلم العميق فعاليتها كأدوات قوية لاستخلاص السمات تلقائيًا والكشف عنها في مهام التعرف على الصور، بما في ذلك تشخيص الملاريا.

وبناءً على ذلك، تقترح هذه الدراسة نهجًا قائمًا على التعلم العميق لتصنيف الخلايا الدموية المصابة بالملاريا وغير المصابة بشكل تلقائي، باستخدام صور مجهرية عالية الدقة وموسومة. وتتمثل المساهمة الرئيسية في استخدام نموذج EfficientNetB2 من خلال التعلم بالنقل (Transfer Learning)، المعروف بأنه يتميّز بتحقيق أداء عالٍ مع الحفاظ على درجة منخفضة من التعقيد. يُستخدم هذا النموذج لاستخلاص سمات تمييزية من صور الخلايا الدموية، مما يمكّن من تصنيف دقيق وموثوق للخلايا المصابة وغير المصابة. وقد تم تدريب النموذج وتحسينه باستخدام مجموعة بيانات متاحة علنًا من المعاهد الوطنية للصحة (NIH)، والتي تحتوي على 27,558 صورة مجهرية لخلايا دم حمراء. وتُظهر النتائج التجريبية أن النموذج القائم على EfficientNetB2 يُحسّن الأداء بشكل ملحوظ، محققًا دقة تبلغ 99.35 % متفوقًا بذلك على الطرق التقليدية ومعظم الدراسات السابقة.

الكلمات المفتاحية: تشخيص الملاريا، صور خلايا الدم المجهرية، التصنيف الآلي للملاريا، الخلايا المصابة وغير المصابة، التعلم العميق، الشبكات العصبية التلافيفية، EfficientNetB2 ، التعلم بالنقل.

Résumé

Le paludisme demeure un défi majeur pour la santé mondiale, causant chaque année la mort de centaines de milliers de personnes, ce qui souligne l'urgence de développer des méthodes de diagnostic rapides et précises. Les techniques de diagnostic traditionnelles, telles que la microscopie et les tests de diagnostic rapide, sont souvent laborieuses, longues et sujettes à des erreurs humaines.

Bien que de nombreuses études basées sur l'apprentissage automatique aient tenté de détecter automatiquement les cellules infectées par le paludisme, leur efficacité reste limitée en raison de divers défis, notamment la variabilité des images microscopiques des cellules sanguines : différences de forme, de densité, de propriétés de coloration, ainsi que l'ambiguïté dans la distinction entre certaines classes cellulaires.

Récemment, les méthodes basées sur l'apprentissage profond se sont révélées être des outils puissants pour l'extraction automatique de caractéristiques et la détection dans diverses tâches de reconnaissance d'images, y compris le diagnostic du paludisme.

Dans ce contexte, cette étude propose une approche basée sur l'apprentissage profond pour la classification automatique des cellules sanguines infectées et non infectées par le paludisme, en utilisant des images microscopiques haute résolution annotées. La principale contribution réside dans l'application du modèle EfficientNetB2 à travers l'apprentissage par transfert (Transfer Learning), reconnu pour son excellent rapport performance-complexité. Ce modèle est utilisé pour extraire des caractéristiques discriminantes à partir des images cellulaires, permettant une classification robuste des cellules parasitées et non parasitées. Le modèle est affiné à l'aide d'un jeu de données public fourni par les National Institutes of Health (NIH), contenant 27 558 images microscopiques de globules rouges. Les résultats expérimentaux montrent que le modèle basé sur EfficientNetB2 améliore considérablement les performances, atteignant une précision de 99,35 %, surpassant ainsi les méthodes traditionnelles et la majorité des études précédentes.

Mots clés : Diagnostic du paludisme, Images de cellules sanguines microscopiques, Classification automatisée du paludisme, Cellules parasitées et non infectées, Apprentissage profond, Réseaux de neurones convolutifs, EfficientNetB2, Apprentissage par transfert.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
EDA	Exploratory Data Analysis
SVM	Support Vector Machine
KNN	K- Nearest Neighbors
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
ANN	Artificial Neural Network
WHO	World Health Organization
DBSCAN	Density-Based Clustering
PCA	Principal Component Analysis
t-SNE	t-distributed Stochastic Neighbor Embedding
LLE	Locally Linear Embedding
MDPs	Markov decision processes
MCTS	Monte Carlo Tree Search
SGD	Stochastic Gradient Descent
NLM	National Library of Medicine
DNN	Deep Neural Networks
ReLU	Rectified Linear Unit
CL	Convolution Layer
FCN	Fully Connected Layer
AUC	Area Under the ROC Curve

General Introduction

Malaria disease is one of the deadliest mosquito-borne diseases, caused by Plasmodium parasites transmitted through the bite of an infected female Anopheles mosquito. It remains one of the most serious and widespread infectious diseases, posing a significant global health challenge, particularly in tropical and subtropical regions. According to reports from the World Health Organization, bites from infected mosquitoes cause around 400,000 deaths every year, along with millions of new infections.

The detection of Plasmodium parasites in blood samples is typically performed through microscopic examination. Thick blood smears are used for initial screening and parasite detection, while thin smears assist in identifying the specific parasite species. This diagnostic process requires trained personnel, specialized equipment, and well-equipped clinical laboratories. As a result, traditional diagnostic methods are often labor-intensive, time-consuming, and prone to human error, underscoring the urgent need for more rapid and accurate diagnostic solutions.

In response to these limitations, many studies have explored automated approaches for detecting malaria-infected cells. Previous research has demonstrated that artificial intelligence techniques—particularly machine learning and deep learning—can significantly enhance diagnostic accuracy. Nonetheless, the task of detecting infected cells remains challenging due to various factors, including differences in cell shape, density, and staining properties, as well as ambiguities in distinguishing certain cell types. Despite these challenges, recent studies have shown that deep learning can significantly enhance diagnostic accuracy, particularly through the use of Convolutional Neural Networks (CNNs), which have demonstrated exceptional performance in a variety of domains, including image recognition, natural language processing, and medical diagnostics.

To address these challenges, this study proposes a deep learning-based approach for the automatic classification of malaria-infected and uninfected red blood cells using labeled high-resolution microscopic images. Specifically, the EfficientNetB2 architecture is applied through transfer learning and fine-tuned on a publicly available dataset provided by the National Institutes of Health (NIH), which includes 27,558 microscopic images of red blood cells.

The rest of the thesis is organized as follows:

Chapter 1 presents a brief background of malaria, including its types and symptoms, and examines both traditional and modern diagnostic methods currently in use. Further-

more, it introduces the fundamentals of machine learning, explores various techniques, and discusses their application in the detection and diagnosis of malaria.

Chapter 2 provides a comprehensive overview of deep learning, and its techniques employed throughout this work.

Chapter 3 describes the proposed deep learning-based approach for the automatic classification of malaria-infected and uninfected red blood cells. It details the methodology, implementation, and experimental results.

Finally, the conclusion summarizes the key findings of this study and outlines open challenges for future research.

Chapter 1

MACHINE LEARNING FOR MALARIA DETECTION

1 Introduction

Malaria remains a significant global health challenge, threatening millions of lives worldwide. Early diagnosis is crucial for reducing the spread of the disease and enhancing patient outcomes [1]. This chapter provides an overview of malaria, including its types and symptoms, and examines both traditional and modern diagnostic methods currently in use. Furthermore, it introduces the fundamentals of machine learning, explores various techniques, and discusses their application in the detection and diagnosis of malaria.

2 What is Malaria?

2.1 Definition

Malaria is a significant public health issue and an infectious disease transmitted through the bites of infected female mosquitoes, which introduce microscopic parasites into the bloodstream and liver. The disease is particularly prevalent in tropical and subtropical regions. According to the World Health Organization (WHO) 2015 report, the Plasmodium parasite is responsible for approximately 405,000 fatalities annually. Studies indicate that the majority of malaria cases occur in Africa (85%), Southeast Asia (71%), and the Eastern Mediterranean (71%). Furthermore, malaria is a zoonotic disease, capable of being transmitted not only among humans but also between humans and animals [1].

2.2 Symptoms of malaria

Malaria is an acute febrile illness. In non-immune individuals, symptoms typically appear 10 to 15 days after an infective mosquito bite [1]. Initial symptoms may include:

- Headache and sweating
- Fever and fatigue
- Muscle discomfort
- Cough and difficulty breathing
- Nausea and vomiting

In severe cases, complications such as kidney failure, seizures, or coma may occur.

2.3 Types of malaria

Malaria can be classified into five types based on the species of Plasmodium parasites that cause the infection [2]:

- **Plasmodium falciparum**: Plasmodium falciparum is the most prevalent species in Africa and is a leading cause of malaria-related mortality worldwide. It multiplies rapidly in the human body, leading to severe complications such as vascular blockage and significant blood loss. Clinical manifestations include muscle pain, headache, fever accompanied by diarrhea and nausea, anemia, and seizures. The fever typically follows a 48–72-hour cycle.
- **Plasmodium vivax**: Plasmodium vivax is mainly found in Asia and South America and is the second most common and dangerous type of malaria. Unlike other types, it can remain dormant in the liver, with symptoms potentially appearing months after an infection or mosquito bite. Key symptoms include high fever, chills, and paroxysmal fever, which typically lasts 42 to 56 hours.
- **Plasmodium ovale**: Plasmodium ovale malaria is rare but can be serious. Its symptoms include paroxysmal fever occurring every 42 to 56 hours, headache, muscle pain, diarrhea, and abdominal discomfort. Timely treatment is crucial, and close monitoring is required to prevent the progression of symptoms.

- **Plasmodium malariae**: Plasmodium malariae is the fourth type of malaria, with relatively few reported cases. Most individuals recover following appropriate medical treatment. Symptoms include fever, headache, and seizures, though they are typically milder and may be due to inadequate treatment. Like Plasmodium ovale, symptoms can persist for months or even years.
- **Plasmodium knowlesi :** Plasmodium knowlesi is a type of malaria that affects only animals, especially primates, and is transmitted through blood transfusions, injections, organ transplants, or mosquito bites. It is important to ensure that pet toys and equipment are clean and safe.

2.4 Diagnostic Methods for Malaria

2.4.1 Traditional Malaria Diagnosis

The presence or absence of Plasmodium parasites in blood samples is detected through microscopic examination. Thick blood smears are used for screening and parasite detection, while thin smears help identify the species. This process requires trained experts, technicians, and a clinical laboratory, as well as appropriate testing materials [3].

2.4.2 Modern Malaria Diagnosis

Many studies have focused on developing accurate automated methods for detecting malaria-infected cells. However, this infected cell detection task remains challenging due to several factors, including variations in cell shape, density, and staining color, as well as ambiguity in distinguishing certain cell classes. Despite these challenges, previous studies have demonstrated that malaria diagnosis using artificial intelligence (AI) techniques, including machine learning (ML) and deep learning (DL), can significantly enhance diagnostic accuracy. These approaches involve the analysis of microscopic images of red blood cells, where image complexity varies depending on the Plasmodium species responsible for the infection. ML and DL models are trained on datasets containing both infected and healthy samples, enabling them to learn and recognize patterns indicative of malaria. As a result, these models offer high diagnostic accuracy and rapid processing, which can assist clinicians in making timely and informed decisions [4].

3 Machine learning

Machine learning (ML), a branch of artificial intelligence (AI), involves the development and application of algorithms that enable systems to learn from and analyze data without explicit programming. This approach supports the extraction of patterns and the construction of accurate predictive models. Key functions of ML include classification, regression, clustering, recommendation, and prediction. These functions allow systems to adapt to evolving data, optimize operations, support data-driven decision-making, and generate personalized outputs based on user preferences. Collectively, they enhance a system's ability to interact efficiently with new data [5].

3.1 Machine Learning Workflow

ML is mainly data-driven, relying on inputs (features) and outputs (labels) to train and evaluate predictive models. During the training phase, the model learns patterns from the input data, while its performance is assessed during the testing phase to evaluate its ability to make accurate predictions. Developing a machine learning model involves several key steps through which the data is processed, as illustrated in Figure 1.1.

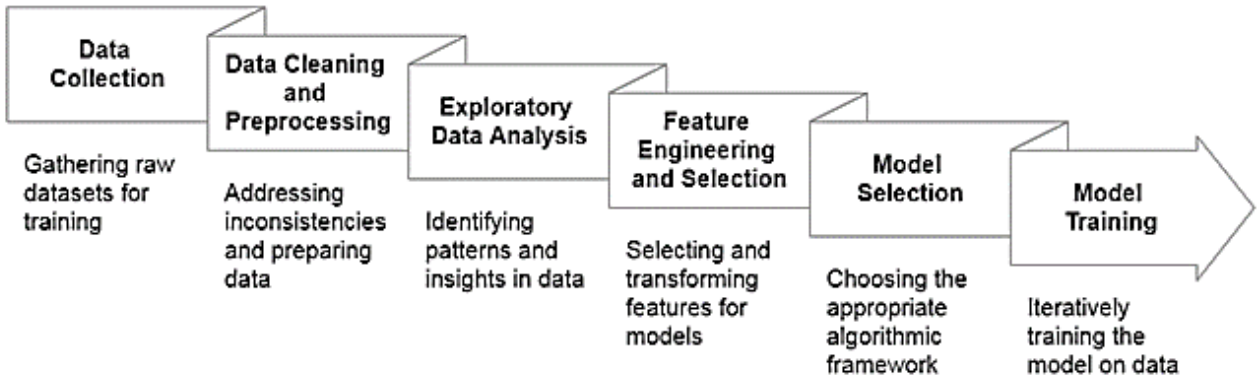


Figure 1.1: Machine Learning Workflow

3.1.1 Data Collection

Data collection is a systematic process of gathering raw datasets for training models. The diversity and quality of the collected data significantly impact the model's generalization and robustness. A well-organized approach ensures accurate training and evaluation, and its usability in real-world scenarios [6].

3.1.2 Data Cleaning and Preprocessing

In machine learning training, data cleaning and preprocessing are critical steps to ensure the quality of the training data. Data cleaning addresses issues such as corrupted images, inconsistencies, missing values, and outliers, all of which impact model accuracy and performance. Preprocessing involves tasks such as categorical variables, resizing images, normalizing pixel values, standardizes formats, augmenting the dataset through techniques like rotation and flipping, and encoding class labels. These steps collectively produce a well-organized dataset that is suitable for model training. The main objective is to transform raw data into meaningful formats suitable for training and analysis, ensuring the model is trained on high-quality, reliable data [6].

3.1.3 Exploratory Data Analysis (EDA)

Exploratory data analysis (EDA) is a statistical technique used to identify hidden features in data and discover relationships. It provides valuable insights into data trends, structure, and potential challenges, helping in making informed decisions. EDA uses visualizations to provide a clear and understandable statistical summary, which aids in model selection and feature engineering [6].

3.1.4 Feature Engineering and Selection

Feature engineering and selection is a transformational process that involves selecting relevant features for model prediction, to enhance efficiency, and to improve parameters. It requires an understanding of the problem and domain expertise to ensure meaningful contributions to model prediction. The process involves transforming existing features, identifying important influential features, or creating new features, using domain knowledge, and improving accuracy [6].

3.1.5 Model Selection

Model selection is critical to effective machine learning models, as it defines the algorithmic framework for prediction. This includes matching the dataset and problem characteristics, considering decision and complexity factors such as scalability, performance, and interpretability, and experimenting to find the best solution to the problem [6].

3.1.6 Model Training

Model training is an iterative process in which the model is exposed to historical data to learn relationships and dependencies within the data. During this process, the model adjusts its internal parameters to enhance its predictive capabilities. The objective is for the model to generalize well to novel or previously unobserved data, thereby providing reliable predictions in real-world scenarios [6].

3.1.7 Model Evaluation and Tuning

Model evaluation involves running rigorous tests on test or validation datasets to assess the accuracy of the model on new, unseen data. It provides insight into the strengths and weaknesses of the model, and if it fails, iterative tuning is necessary to improve the prediction accuracy. To measure accuracy, there are metrics such as precision, accuracy, F1 score, and recall [6].

4 Types of machine learning

Machine learning includes a diverse set of algorithms developed to solve various types of problems. It is generally categorized into four main types: supervised learning, unsupervised learning, reinforcement learning, and semi-supervised learning [5].

4.1 Supervised learning

Supervised learning is a subfield of machine learning that uses labeled datasets to train algorithms for pattern recognition and outcome prediction. Each training instance consists of a set of input features (X) paired with their corresponding target output values (y). This approach allows the model to learn by identifying correlations between input features and expected target outputs [7]. As illustrated in Figure 1.2, supervised learning algorithms are generally classified into two main categories : classification and regression [8], as summarized in Table 1.1.

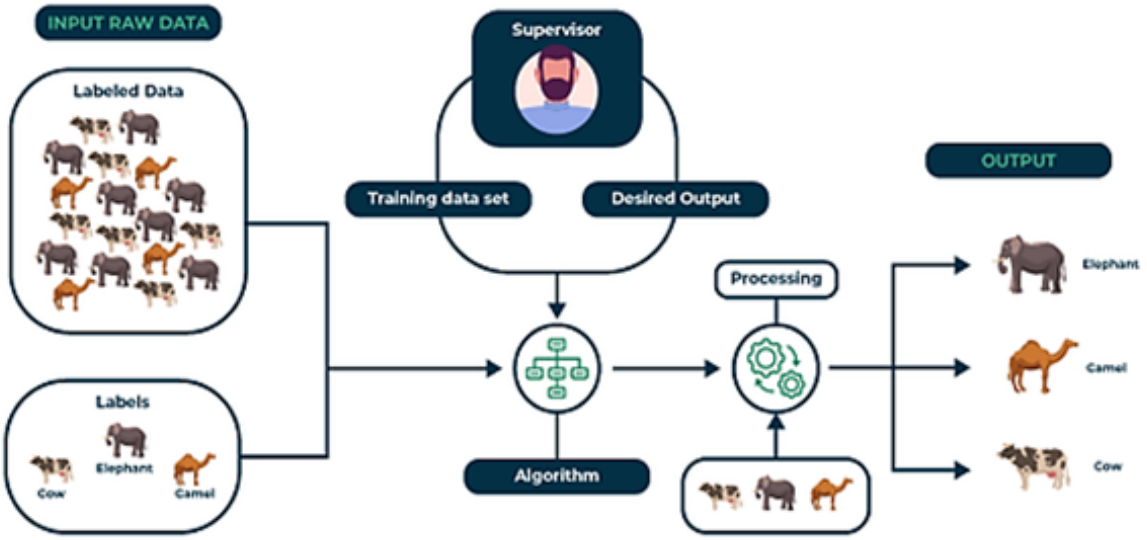


Figure 1.2: Supervised learning [7]

Table 1.1: Difference between Classification and Regression [7]

	Classification	**Regression**
Problem Type	Determine the category to which the item belongs	Expect a continuous numeric value
Output Type	Specific categories or labels (e.g., ”yes” or ”no”)	Continuous numeric values (e.g., 50.5, 102.3)
Example of Output	”Regular message” or ”Spam”	House price forecast of $250,000
Practical Examples	Image classification, email filtering, disease diagnosis	Predicting real estate prices, forecasting temperatures, estimating salaries
Evaluation Metrics	Accuracy, F1-score, Confusion Matrix	MAE, MSE, R^2 score

4.1.1 Supervised Learning Algorithms

Different algorithms are specifically designed to solve different problems. Below we present the most important common algorithms in supervised learning.

- **Linear Regression** Linear regression is a simple and widely used supervised learning algorithm that predicts continuous output values [7].

- **Logistic regression** Logistic regression is a supervised machine learning algorithm for binary classification, predicting class probability using a sigmoid function and a 0.5 threshold value [7].
- **Decision Trees** Decision trees are tree-like structures used to model decisions and their potential consequences. A decision within this structure is represented by an internal node, while a leaf node represents a possible outcome [7].
- **Random Forests** Random forests are an ensemble of multiple decision trees that are trained on different subsets of data and input features, and the sub-predictions are eventually combined to create a final prediction [7].
- **Support Vector Machine (SVM)** A support vector machine (SVM) is an algorithm that creates a subplane in the n-dimensional space into classes to classify new data points, using extreme cases called support vectors [7].
- **K- Nearest Neighbors (KNN)** K-Nearest Neighbors (KNN) is a machine learning algorithm that uses k training examples to predict the class based on the predominant class or mean value of those neighbors [7].
- **Gradient Boosting** Gradient Boosting is an iterative process that combines weak learners like decision trees to create a strong model by correcting previous errors [7].
- **Naive Bayes Algorithm** Naive Bayes Algorithm is a supervised machine learning technique that relies on applying Bayes' theorem with the "naive" assumption that features are independent of each other given their class classification [7]. The following table shows the most important differences between supervised machine learning algorithms.

Table 1.2: Summary of Supervised Machine Learning Algorithms

Algorithm	Regression or Classification	Purpose	Method	Use Cases
Linear Regression	Regression	Predict continuous output values	Linear equation minimizing sum of squares of residuals	Predicting continuous values
Logistic Regression	Classification	Predict binary output variable	Logistic function transforming linear relationship	Binary classification tasks
Decision Trees	Both	Model decisions and outcomes	Tree-like structure with decisions and outcomes	Classification and regression tasks
Random Forests	Both	Improve classification and regression accuracy	Combining multiple decision trees	Reducing overfitting, improving prediction accuracy
SVM (Support Vector Machine)	Both	Create hyperplane for classification or predict continuous values	Maximizing margin between classes or predicting continuous values	Classification and regression tasks
KNN (K-Nearest Neighbors)	Both	Predict class or value based on k closest neighbors	Finding k closest neighbors and predicting based on majority or average	Classification and regression tasks, sensitive to noisy data
Gradient Boosting	Both	Combine weak learners to create strong model	Iteratively correcting errors with new model	Classification and regression tasks to improve prediction accuracy
Naive Bayes	Classification	Predict class based on feature independence assumption	Bayes' theorem with feature independence assumption	Text classification, spam filtering, sentiment analysis, medical diagnosis

4.2 Unsupervised learning

Unsupervised learning is also a subfield of machine learning that uses unlabeled data to identify relationships and patterns without human intervention. It uses input parameter values and determines patterns independently, in contrast to supervised learning, which classifies data based on predefined outcomes, as illustrated in Figure 1.3 [8].

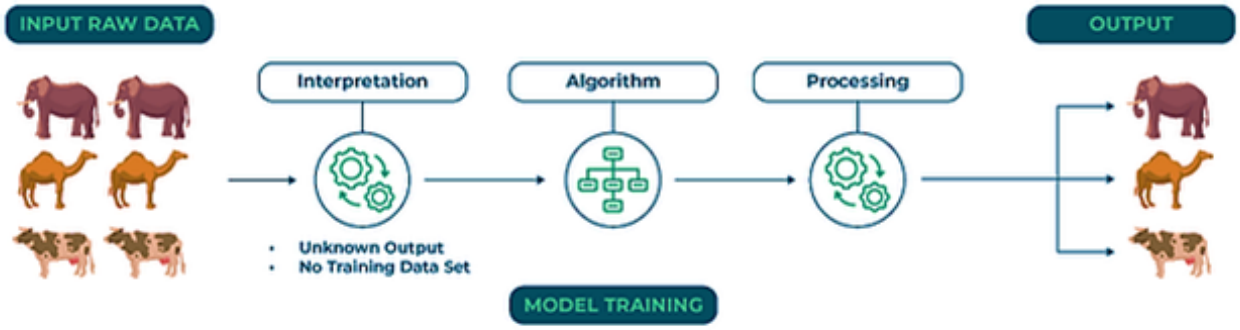


Figure 1.3: Unsupervised learning [8]

Figure 1.3 illustrates raw data for animals such as cows, elephants and camels, which is processed by an unsupervised learning algorithm. The algorithm uses techniques such as dimensionality reduction, anomaly detection and clustering to identify structures and patterns, with the output displaying the results grouped by species [10]. Unsupervised learning is typically categorized into three main types: association rule mining, dimensionality reduction, and clustering [8].

- **Clustering** in unsupervised machine learning is the process of grouping unclassified data into groups based on differences or similarities. Clustering methods are divided based on the methods used. These methods include distribution-based, data-center-based, connectivity-based, and densitybased methods. The objective is to discover relationships and identify patterns [8].
- **Association Rule** Association rule extraction is a technique used in unsupervised machine learning, also known as association rule extraction. This technique is used to find relationships in large data sets and identify patterns based on the co-occurrence and frequency of occurrence of items in the data set. For example, in shopping cart analysis, if a customer buys milk, they might also buy butter, bread, or eggs [8].

- **Dimensionality Reduction** To simplify datasets, dimensionality reduction is used by reducing the number of features while retaining the most important information, which can improve the performance of machine learning algorithms and data visualization [8].

4.2.1 Unsupervised Learning Algorithms

- **K-means Clustering** K-means Clustering is an unsupervised machine learning algorithm that groups unlabeled data into different clusters based on similarity. The algorithm randomly selects the center points and assigns each data point to the closest cluster, updating the weight points until they stop changing, as shown in Figure 1.4 [8].

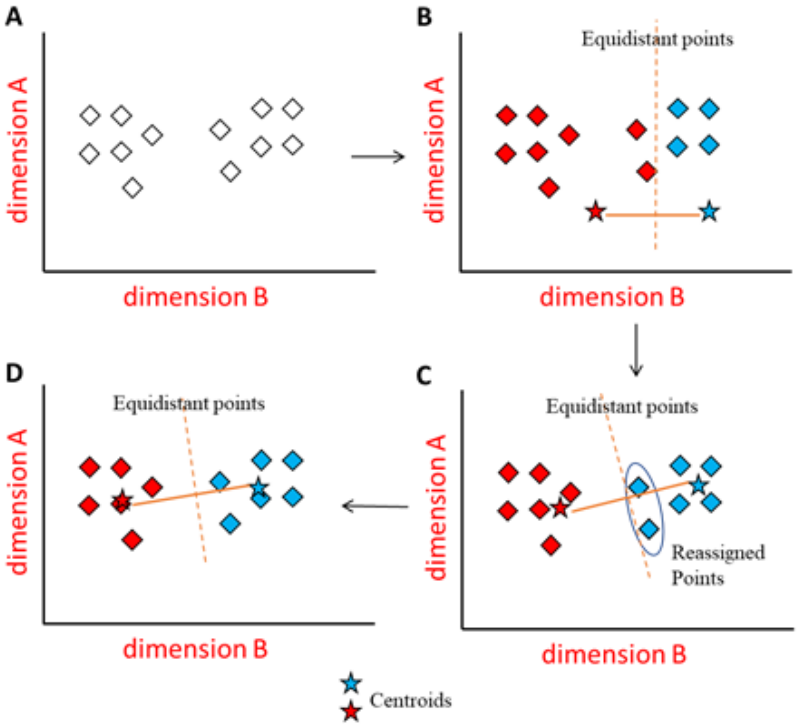


Figure 1.4: K-means clustering algorithm [9]

- **Density-Based Clustering (DBSCAN)** DBSCAN is a density-based clustering algorithm that searches for clusters in dense regions and treats scattered points as noise so that it identifies dense clusters separated by regions of lower density and can handle clusters of arbitrary shapes and noise points [8].

- **Apriori Algorithm** The Apriori algorithm is a basic data mining method that searches for patterns and generates association rules by exploring sets of duplicate items step

by step. In large data sets, it has the ability to identify relationships between items, for example, market basket analysis [8].

- **Principal Component Analysis (PCA)** Principal Component Analysis (PCA): Reduces dimensionality by selecting and extracting important features by converting data into uncorrelated principal components, which allows for avoiding overfitting, speeding up the computation process, and higher accuracy by preserving important features [8].

- **t-distributed Stochastic Neighbor Embedding (t-SNE)** T-distributed stochastic neighbor embedding (t-SNE) is a nonlinear dimensionality reduction technique that simplifies complex data sets by combining correlated and similar features. The technique preserves local pattern and structure, improving efficiency and analysis. It is used to visualize high-dimensional data in 2D or 3D spaces [8].

- **Locally Linear Embedding (LLE)** LLE is an unsupervised approach that transforms its original high-dimensional space into a lowdimensional representation while preserving its geometric properties. It creates a graph of nearest neighbors, optimizes the weight values for each data point, and computes a lower-dimensional representation using the eigenvectors from the weight matrix [8].

4.3 Semi-Supervised Learning

To train the model, semi-supervised learning uses a set of labeled and unlabeled data. Based on the input variables, the output variables are accurately predicted. As shown in Figure 1.5, this is especially useful when the classification process is time-consuming or expensive [10]. To train models and understand the relationship between data points, semi-supervised learning relies on certain assumptions about the data and unlabeled data, which improves on SSL techniques but may affect performance if inconsistent data sets are added implicitly or explicitly. Semi-supervised learning algorithms work to satisfy at least one of the following assumptions to train models [10]:

- **Cluster assumption** The cluster assumption states that data can be divided into separate groups such that data points that belong to the same group will also belong to the same class. It is a generalization of other assumptions. The clustering of data points depends on the concept of similarity used [10].

- **Smoothness assumption** The smoothness assumption, also known as the continuity assumption, assumes that points that are closest to each other in the input space should have the same output label [10].
- **Manifold assumption** The manifold assumption states that the input space consists of lower-dimensional manifolds with data points sharing the same label, allowing the use of distances and densities defined on the manifold [10].

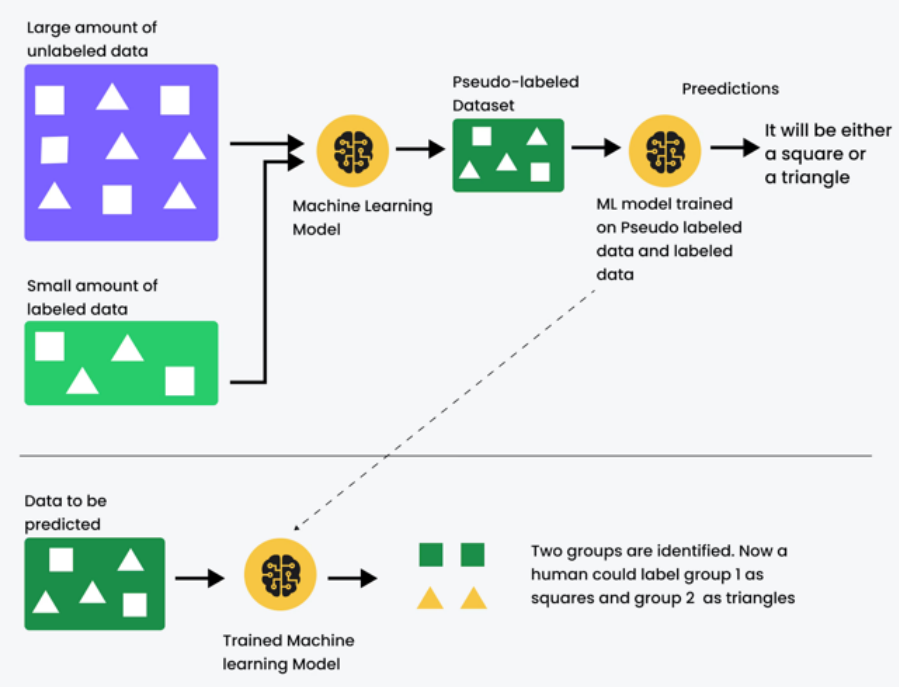


Figure 1.5: Semi-Supervised Learning [11]

4.3.1 Semi-supervised learning techniques

- **Self-training** In self-training, both labeled and unlabeled data are used to train the model. A small amount of labeled data is used to train a base model, then pseudo-labels are applied to make predictions for the unlabeled database. The more confident model predictions are combined with the labeled dataset to train an improved model. More pseudo-labels can be added as needed, and the process can be repeated multiple times [12].
- **Co-training** Co-training trains two classifiers based on two views of the data, independent of class. It is effective for web content classification tasks, where each view has different features. The process involves training a separate classifier for each

view using a small amount of labeled data, then adding a larger set of unlabeled data to receive pseudo-labels. The classifiers co-train using the highest confidence level, updating each other as needed. The final step involves combining the predictions to create a single classification result [12].

- **graph-based labeling** Used to represent classified and unclassified data in the form of graphs, this method is useful in personalization and recommendation systems, where customers can predict interests based on information about other customers. By calculating the paths from points to a given node, the algorithm can identify points that belong to different categories. This method can be applied to social media communications, where there are likely to be similar interests [12].

4.4 Reinforcement learning

Reinforcement learning is a branch of machine learning that differs from supervised learning in that it does not rely on pre-defined answers. It is ideal for decision-making tasks because it allows agents to learn through trial and error and receive feedback through rewards or punishments [13], as shown in Figure 1.6.

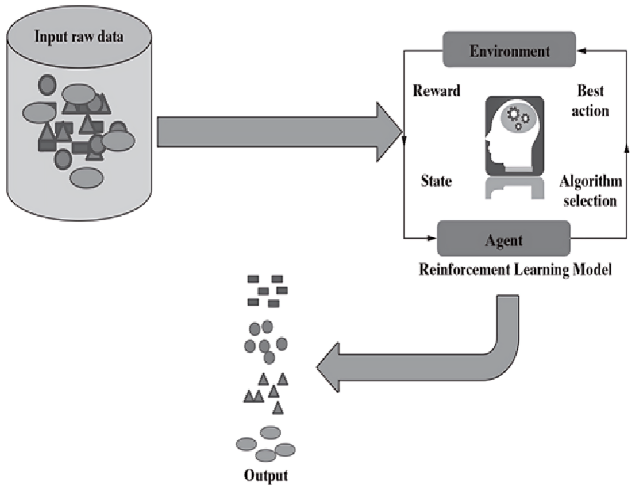


Figure 1.6: Reinforcement Learning Process [14]

Reinforcement learning methods can be classified into model-free and model-based methods, each of which has unique interactions with the environment, where agents learn from their environment and receive feedback [13].

- **Policy-based Methods** Policy is a strategy of linking cases to actions so that based

on the current case the next action is determined and policies are continually modified to maximize rewards [13].

- **Value-Based Methods** The expected return from each action is estimated and the highest value is chosen by learning the value of the cases or actions [13].
- **Model of the Environment** These methods use a representation of the environment to aid in planning and allow agents to predict future states and rewards by simulating expected outcomes [13].

4.4.1 Reinforcement Learning Algorithms

- **Markov decision processes (MDPs)**: MDPs originated in the 1950s and are a stochastic decision-making process for a dynamic system by modeling scenarios in which outcomes are either under the control of the decision maker or random. MDPs are classified into four types based on factors such as actions, available states, and frequency of decision-making [13].
- **Bellman equation**: The Bellman equation was introduced by Richard Bellman in the 1950s, and by evaluating future states and rewards it allows agents to make decisions in complex situations. The Bellman equation is crucial to understanding how agents interact with their environment. It also allows the calculation of the value of different states and actions by breaking complex problems down into smaller steps and directing agents to achieve the greatest possible rewards [13].
- **Value iteration algorithm**: It is a fundamental technique in dynamic programming and reinforcement learning in which it calculates the expected utility of each state using neighboring states until the benefits are close enough, ensuring convergence and accuracy under certain conditions [13].
- **Monte Carlo Tree Search (MCTS)**: Combining Monte Carlo strategies and tree-based search techniques, MCTS simulates stochastic performance until a final state is reached and progressive search is a balance between exploration and exploitation, choosing moves based on high win rates and unexplored or lesser moves [13].
- **Q-Learning**: Q-learning is an algorithm that determines which actions lead to rewards or punishments. The iterative model uses trial and error with key components including

states, loops, rewards, Q values, actions, and agents. The iterative process involves exploring the environment and continually updating the model based on the exploration [13].

5 Applications of Machine Learning

5.1 Enhancing Decision-Making

By analyzing massive amounts of data, machine learning algorithms enhance the decision-making process, allowing for the identification of trends that humans cannot perceive and the discovery of patterns, leading to better results, increased efficiency and high accuracy in predictive analytics such as finance to predict the stock market [15].

5.2 Improving Efficiency and Automation

Machine learning enhances productivity by allowing employees to focus on more strategic tasks by improving efficiency and automation by reducing human error and automating repetitive tasks [15].

5.3 Personalizing User Experiences

By enhancing engagement and providing personalized recommendations, machine learning algorithms improve user experience and, by analyzing consumer data, help in personalized marketing, leading to customer satisfaction and loyalty [15].

5.4 Advancing Healthcare

By analyzing medical images and patient data, machine learning has revolutionized healthcare by enhancing disease detection, predicting disease prevalence, accelerating drug discovery, and reducing costs [15].

5.5 Enhancing Security

By analyzing transaction patterns and user behavior in real time, machine learning enables enhanced security by detecting fraud and cybersecurity, reducing financial losses, enabling

faster responses, and identifying potential security breaches [15].

5.6 Transforming Industries

By analyzing and processing data from sensors in real time and making real-time decisions, machine learning has revolutionized industries such as self-driving vehicles and smart cities, as well as enhancing resource management, transportation efficiency, reducing energy consumption, and improving traffic flow [15].

5.7 Facilitating Scientific Research

By analyzing data and facilitating simulations, machine learning enables researchers to create accurate models and discover new insights in fields as diverse as environmental science, physics, and chemistry [15].

6 Machine Learning of diagnosing Malaria

Traditional diagnostic techniques, such as microscopy and rapid diagnostic tests, can be labor-intensive, time-consuming, and prone to human error. In recent years, the application of machine learning techniques has improved the speed and accuracy of malaria detection and classification.

Das et al. (2013) [16] proposed a machine learning-based system for the automated classification of malaria-infected erythrocytes using light microscopic images of Giemsa-stained peripheral blood smears. Their methodology involved image acquisition, illumination correction, noise reduction, erythrocyte segmentation using marker-controlled watershed transformation, followed by feature extraction, feature selection, and classification. A total of 96 features describing shape, size, and texture of erythrocytes were extracted, with 94 identified as statistically significant in distinguishing six infection classes. For classification, a feature selection-cum-classification framework was developed by integrating F-statistics with Bayesian learning and Support Vector Machine (SVM) techniques. The Bayesian classifier achieved the highest accuracy of 84% using the top 19 selected features, while the SVM achieved an accuracy of 83.5% using 9 significant features.

Rashke Jahan and Shahzad Alam (2023) [17] presented a hybrid machine learning framework to enhance the classification performance of malaria-infected red blood cells using the

NIH Malaria dataset. They proposed methodology involved sequential stages including image acquisition, preprocessing, segmentation, feature extraction, and classification. Five individual classifiers—Stochastic Gradient Descent (SGD), Decision Tree, Logistic Regression, XGBoost, and Random Forest—were initially trained to detect parasitized and uninfected cells. To overcome the limitations of individual classifiers and reduce misclassification, the authors introduced two hybrid combinations using a majority voting technique. The first hybrid model combined SGD, Logistic Regression, and Decision Tree, achieving an accuracy of 95.64%, compared to their individual accuracies of 90.63%, 92.23%, and 93.43%, respectively. The second hybrid model, which combined SGD, XGBoost, and Random Forest, further improved performance, reaching 96.22% accuracy, surpassing the individual scores of 90.63%, 95.86%, and 96.11%. These results confirm that hybrid ensemble methods significantly enhance diagnostic accuracy and robustness, marking a promising advancement over traditional machine learning approaches in medical image analysis.

Shashikiran S. and Sunitha H. D. (2024) [18] proposed an enhanced machine learning framework for the automatic identification of malaria-infected red blood cells. Utilizing the publicly available NIH Malaria dataset comprising 27,558 microscopic images equally distributed between infected and uninfected cells, the study explored three experimental scenarios based on the Support Vector Machine (SVM) classifier. In the first scenario, a standalone SVM model was employed, achieving an accuracy of 84% in binary classification. The second scenario integrated SVM with the t-Distributed Stochastic Neighbor Embedding (t-SNE) technique to improve feature space visualization, resulting in enhanced classification performance with an accuracy of 86.23%. In the third scenario, Principal Component Analysis (PCA) was used alongside SVM for dimensionality reduction, leading to reduced computational complexity and further improvement in detection accuracy, surpassing 88%. The results highlight the effectiveness of combining dimensionality reduction and visualization techniques with SVM in improving malaria diagnosis accuracy from microscopic images.

Table 1.3: Summary of Related Works on Malaria Diagnosis Using Machine Learning

Author(s)	Year	Technique (Models)	Parameters (Features)	Dataset (Samples)	Accuracy (%)
Das et al. [16]	2013	Bayesian Network	Feature extraction (shape, size, texture), feature selection using F-statistic, dimensionality reduction	Microscopy Images	84.00
		SVM			83.50
Jahan and Alam [17]	2023	SGD, LR, DT	Texture-based, statistical, shape-based features	NIH Malaria Dataset (27,558 images)	95.64
		SGD, RF, XGBoost			96.22
Shashikiran S. et al. [18]	2024	SVM	Grayscale conversion, brightness adjustment, flattening, normalization, PCA/t-SNE for dimensionality reduction	NIH Malaria Dataset (27,558 images)	84.00
		SVM + t-SNE			86.23
		SVM + PCA			88.27

7 Conclusion

This chapter provides a comprehensive overview of malaria, its various types and symptoms, as well as the evolution of diagnostic approaches. It also examines machine learning (ML), detailing its methods, techniques, and significance across diverse sectors. Furthermore, the chapter discusses how machine learning has significantly transformed malaria diagnosis, leading to improved health outcomes. Although ML methods can achieve satisfactory performance to some extent, their effectiveness is often limited by various factors such as high variability in the characteristics of microscopic red blood cell images, including variations in cell shape, density, and staining properties, as well as ambiguity in distinguishing between certain cell classes.To address these limitations, more advanced techniques are necessary and required. The next chapter will explore deep learning and investigate its potential to overcome these challenges, offering more accurate and efficient solutions for malaria detection and the analysis of complex medical imagery.

Chapter 2

DEEP LEARNING TECHNIQUES

1 Introduction

Malaria diagnosis is a critical step in reducing its spread and improving treatment outcomes. Although machine learning techniques have provided promising results in this field, they still face certain limitations, as previously discussed. These challenges highlight the need for more advanced and adaptable approaches. In this context, deep learning, particularly deep neural networks, has emerged as a powerful tool for automatically extracting complex patterns from medical images. Among these techniques, Convolutional Neural Networks (CNNs) have demonstrated outstanding performance in a variety of image analysis tasks. This chapter provides a comprehensive overview of deep learning and its techniques. It starts by introducing deep learning and its relationship with machine learning, then the fundamental concepts of artificial neural networks, which serve as the core component of deep learning architectures. Subsequently, various types of deep learning methods are presented. Finally, the chapter discusses the application of DL in the detection and diagnosis of malaria.

2 Deep Learning

Deep Learning (DL) is a prominent subfield within the broader domains of machine learning (ML) and artificial intelligence (AI), which focuses on using artificial neural networks with many layers. In recent years, DL has gained considerable attention due to its outstanding performance across a wide range of complex tasks. It encompasses a set of methods that automatically learn hierarchical representations of data through deep neural architectures, particularly excelling in classification problems. The main objective of DL is to extract

increasingly abstract and meaningful features at higher levels of representation, which enables models to effectively capture and distinguish underlying patterns in data. Although DL is a subset of ML, its distinctive advantage lies in the depth of representation learning, which allows it to outperform traditional approaches in domains such as image recognition, speech processing, and medical diagnostics [19].

2.1 Understanding the Relationship Between Machine Learning and Deep Learning

Though Deep Learning (DL) is a subfield of Machine Learning (ML), the two approaches differ significantly in methodology, capabilities, and application scope. ML relies heavily on manual feature engineering, where domain expertise is required to extract relevant input features before training. In contrast, DL models automatically learn and extract features from raw data through hierarchical layers of abstraction [19].

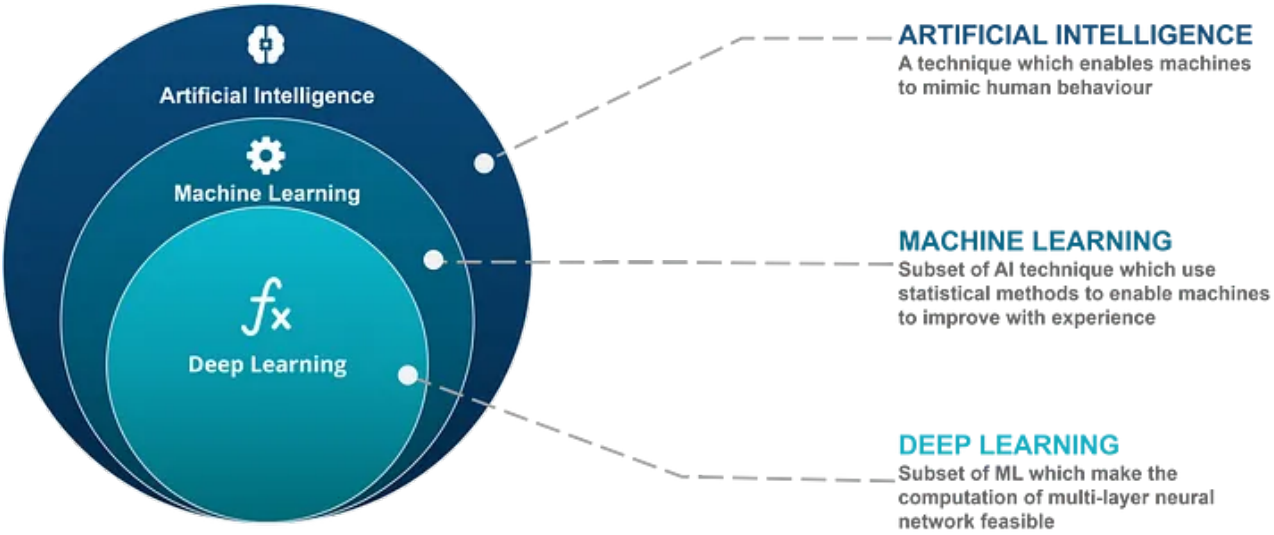


Figure 2.1: Relationship between AI, ML and DL [20]

Another prominent distinction lies in algorithmic complexity and data requirements. ML algorithms such as decision trees, support vector machines, and k-nearest neighbors perform well on structured, tabular data and relatively smaller datasets. DL models, however, excel when dealing with large-scale, high-dimensional data such as images or audio, This capability is primarily achieved through advanced neural network architectures, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) [19].

From a computational perspective, ML models are generally lightweight and faster to

train, making them suitable for scenarios with limited hardware resources. DL models, on the other hand, require significant computational power and benefit greatly from GPU acceleration due to their deeper and more complex network structures. In terms of performance, DL has demonstrated higher accuracy in many domains, particularly in image recognition, natural language processing, and medical diagnostics. Nevertheless, ML remains more interpretable and is often preferred in applications where explainability and model transparency are essential [19].

3 Neural Networks

Artificial Neural Networks (ANNs) are a foundational element of artificial intelligence (AI) and machine learning, particularly in terms of their architecture and design. Inspired by the biological neural networks found in the human brain, ANNs aim to replicate the processes of human learning and information processing by modeling the structure and function of biological neurons. These models are capable of learning complex relationships within data and making accurate predictions.Neural networks have demonstrated significant effectiveness in a variety of applications, including image and speech recognition, medical diagnosis, and natural language processing. This section introduces the fundamental components of a neural network—namely, neurons, layers, weights, and activation functions—providing the foundational knowledge necessary to understand more advanced deep learning architectures [21].

3.1 Basic Concepts of Neural Networks

The essential building blocks of artificial neural networks (ANNs) define their structure and functionality. A brief overview of the main components is provided below.

3.1.1 Layers

ANNs are structured as sequential layers of interconnected neurons, with each layer playing a distinct role in the processing of data. These layers collectively contribute to the network's ability to learn from input data and generalize to new, unseen instances [19]. Typically, a neural network is composed of three primary types of layers:

- **Input Layer**: The input layer serves as the entry point of the neural network, receiving the features or attributes that represent each data sample, for example, pixel intensities in image data. Each feature is assigned to a node (or neuron), which then passes its value to the subsequent layer, typically the first hidden layer. In general, the input layer is responsible for preparing the data for processing by the subsequent layers, contributing to the final predictions made by the output layer [22].

- **Hidden Layers**: Hidden layers play a pivotal role in Artifcial neural networks (ANNs), serving as intermediate stages for processing data through an interconnected network of neurons.These layers progressively extract features, building on each other to uncover deeper patterns in the data. Determining the number of hidden layers and the number of neurons within them is a fundamental design step, tailored to the specific requirements of each task. These layers are sometimes referred to as dense layers and are considered the core component that enables the network to learn and produce accurate results [23].

- **Output Layer**: The output layer is responsible for generating the final predictions of the neural network, based on the information processed by the previous layers. The network's architecture and the nature of the task determine the type of output, which can be continuous (as in regression problems), binary, multi-class, or ordinal. The structure of this output depends on the activation function used in the neurons of the output layer [23]

.

3.1.2 Weights

Weights represent the strength of the connections between neurons within an artificial neural network. A lower weight indicates that the data passing through this connection has minimal impact on the final predictions. In contrast, a significant positive or negative weight modifies the information received by subsequent layers, which may impact predictions. This approach is similar to how brain cells communicate, with connections growing or decreasing as they acquire experience. As connections define specific brain regions activated or deactivated in response to processed information [19].

3.1.3 Biases

They are associated with each neuron and act as an offset or threshold, allowing neurons to activate even when the weighted sum of inputs is insufficient. This adaptability enhances the network's ability to learn and make accurate predictions. The bias can be a significant factor, especially when some features are missing or have a value of zero [19].

3.1.4 Forward Propagation

It is the first step where the neural network starts working with input data. The input layer receives the data, then hidden layers process it by applying activation functions. As the data moves forward through the layers, each part helps shape the final result. The output layer finally gives the prediction or classification. This process is important because it helps the network learn by adjusting weights and biases during training [24].

3.1.5 Activation Functions

ANNs rely on activation functions, which are the fundamental mechanisms within each neuron, to control the transformation of inputs into outputs. These functions act as calculators that process the combined influence of incoming neuron signals (weighted sum). After processing, activation functions determine whether the resulting value is significant enough to be passed on to the next layer. This thresholding process allows the ANN to focus on relevant information and introduces non-linearity, a critical factor in solving complex problems. Activation functions act as decision gates, determining which signals are strong enough to influence the network's overall prediction [19][23].

- **Sigmoid Function (Logistic Function):** It converts different inputs into probabilities ranging from 0 to 1. Its S-shaped curve introduces critical non-linearity for tackling complex tasks. The popularity of this function stems from its alignment with probabilistic models and binary classifications. However, vanishing gradients—where error signals diminish during training—limit its effectiveness in deeper ANN. Therefore, alternative activation functions may be necessary for deeper architectures. This function is represented as [19][23] :

$$A(x) = \frac{1}{1 + e^{-x}}$$

- **Tanh Function:** It offers a compelling alternative to the sigmoid function in ANNs

and DNNs. Both share a sigmoidal output curve, but Tanh's key advantage lies in its broader output range (-1 to 1) compared to sigmoid's 0 to 1. This seemingly minor difference translates to a significant benefit: Tanh is less susceptible to the vanishing gradient problem, a hurdle that hinders learning in deep networks. While not entirely immune, Tanh offers a clear advantage over sigmoid, potentially leading to faster learning [19][23]. As shown in the equation below, the Tanh function is represented as:

$$A(x) = \frac{2}{1 + e^{-2x}} - 1$$

- **ReLU Function (Rectified Linear Unit):** has emerged as a dominant choice in artificial neural networks (ANNs)and deep neural networks (DNNs) due to its efficiency and ability to overcome limitations present in previous activation functions. Unlike sigmoid and Tanh functions with their sigmoidal curves, ReLU operates according to a segmented linear rule. For negative input values, ReLU outputs zero. However, for positive input values, ReLU maintains a linear relationship, essentially acting like a standard identity function [19][23]:

$$g(x) = \mathsf{max}(0, x)$$

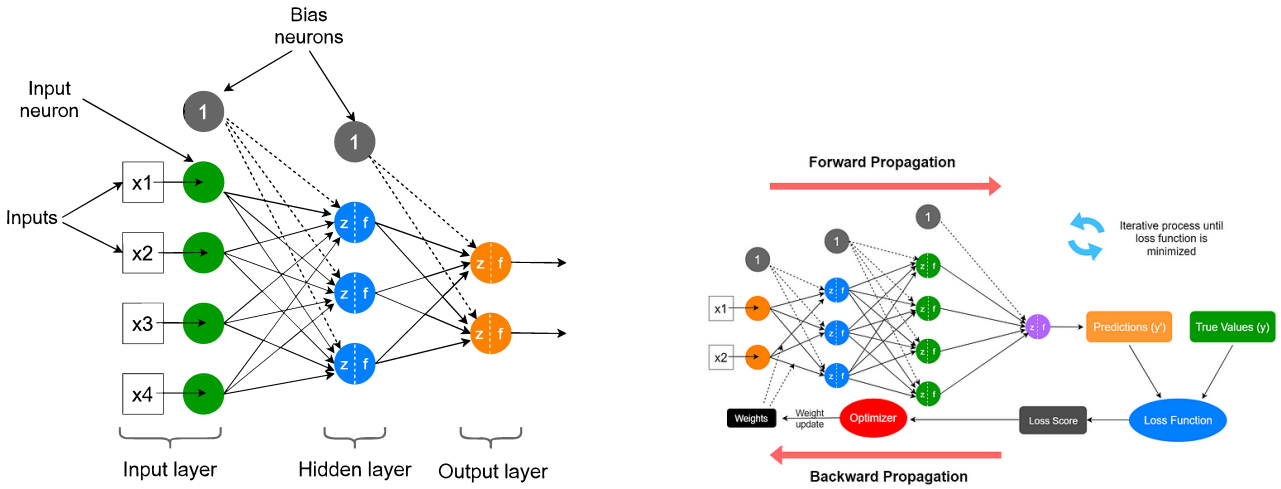
- **Softmax Function:** Similar to the sigmoid function, the softmax function manages categorical outcomes in multinomial labeling systems. It converts the model's outputs into a probability distribution, enabling more nuanced predictions [19][23]. It can be defined as:

$$f(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}, \quad j = 1, \ldots, K$$

- **Loss Functions** The loss function (LF) serves as a quality check, regularly assessing the difference between the network's predictions and actual values. The LF calculates a specific error metric for each data point, providing valuable feedback on network performance. Iterative optimization procedures like gradient descent use this data to improve the network's internal parameters (weights and biases). Consider the LF to be a teacher who regularly corrects the network's errors, allowing it to change and improve its predictions over time [19].

3.1.6 Backpropagation

It is the second key step where the neural network improves its performance. After the prediction is made in the forward pass, the network compares it with the actual output to calculate the error. Backpropagation then works backward through the layers, using this error to compute gradients. These gradients help adjust the weights and biases using optimization algorithms like gradient descent. This step is essential for minimizing error and making the network more accurate over time [25].



(a) (b)

Figure 2.2: Artificial Neural Network Components: (a) Basic NN Layers [26] , (b) An overview of the neural network training process [27].

4 Deep Learning Methods

Deep learning is fundamentally built upon deep neural networks (DNNs), which are a specialized form of artificial neural networks (ANNs) characterized by multiple hidden layers. Since their initial conception in the 1940s, ANNs have evolved significantly, comprising interconnected layers of perceptrons that process data through input (image data), hidden (information processing), and output (classifications) stages. The number of hidden layers and neurons in these networks helps them understand complicated patterns in data. During training, ANNs adjust the strength of connections between neurons based on prediction errors, gradually learning to distinguish between classes. Deep ANNs, in particular, ex-

cel at automatic feature extraction. This ability is especially useful in image classification tasks, where layered processing captures both low- and high-level visual features. While deep models typically demand large datasets and significant computational power, recent advancements in hardware (GPUs) and training techniques (e.g., unsupervised pre-training) have greatly improved their efficiency and scalability. Building on this foundation [19], The following section will explore some of the common types of deep neural networks:

4.1 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are a type of deep learning architecture that emerged in the 1980s, with foundational contributions by Rumelhart et al. (1986) [28], Werbos (1988) [29], and Elman (1990) [30]. They are specifically designed for modeling sequential data. Inspired by the structure of feedforward networks and the way the human brain processes information over time, RNNs incorporate a loop within their hidden layers, allowing them to retain information from previous inputs — a form of memory. This capability is essential for tasks such as language translation and detecting long-range dependencies in sequences (Figure 2.3).

In addition to their memory capabilities, RNNs offer advantages such as parameter efficiency by sharing weights across time steps, and compatibility with convolutional layers for tasks involving sequential data with spatial characteristics, such as image captioning. Despite facing training challenges like the vanishing gradient problem, RNNs remain a potent tool for applications that depend on sequence understanding [31].

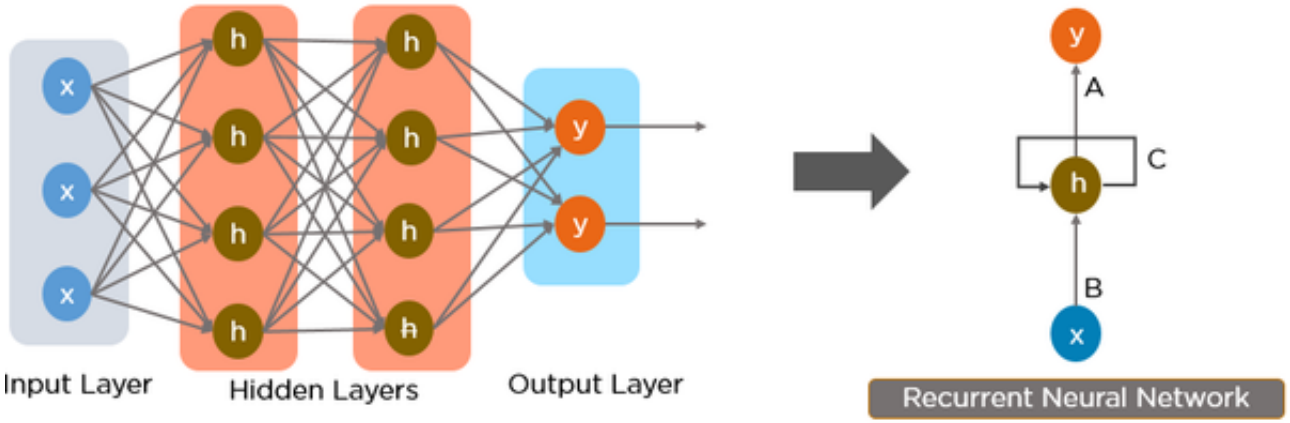


Figure 2.3: Recurrent neural network (RNN) architecture [32]

4.2 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are also a type of deep learning architecture inspired by the animal visual cortex, renowned for their success in image recognition and computer vision tasks. They have a unique architecture with neurons arranged in three dimensions, allowing them to process spatial relationships and reduce complexity. CNNs extract features from input data through convolution operations, using weight sharing to improve efficiency. They can directly work with raw images, simplifying tasks like image classification and object detection (see Figure 2.4). However, their layered architecture makes design and maintenance more challenging, and training is computationally expensive and slower [31].

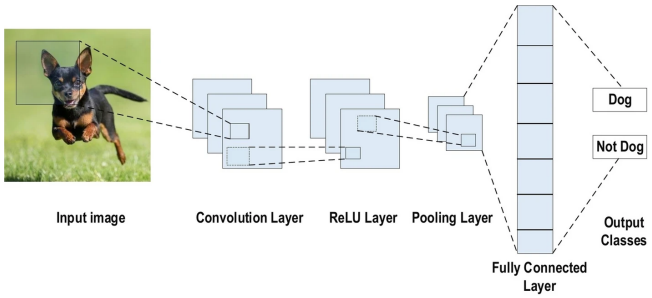


Figure 2.4: Example of CNN Architecture [33]

4.2.1 Basic Concepts of CNN

CNNs consist of multiple essential layers that collaboratively extract and process features from input data. These layers include the convolution layer (feature extractor), pooling layer (dimensionality reduction), fully connected layer (decision making), and dropout layer (regularization), each serving a unique function in the network's architecture. Here is a brief overview of these essential components:

- **Convolution Layer:** The Convolution Layer (CL) is the main component of any convolutional neural networks , following the input layer, and is where most computations occur. Multiple filters (feature detectors), small in height and width but spanning the entire image depth, scan across the input image. By convolving the filters with the input, the CL generates feature maps that highlight the presence and location of specific patterns within the image. These filters function as pattern detectors, identifying specific features within the image [34].

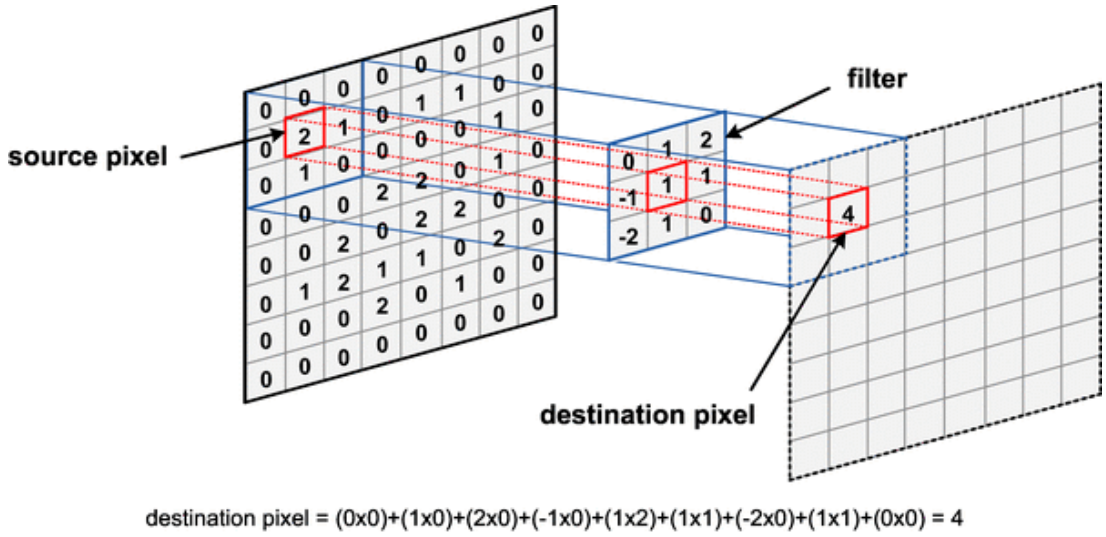


Figure 2.5: Diagram illustrating the convolution operation [35]

- **Pooling Layer:** situated between convolution layers, plays a crucial role in reducing the spatial resolution of the feature maps. There are four main types of pooling layers as illustrated in Figure 2.6: Max pooling, which consistently chooses the most significant value in the region; Average pooling, which computes the mean within the region; Global max pooling; and Global average pooling [19]. They effectively downsample the image by selecting the maximum or average value within defined pooling regions.

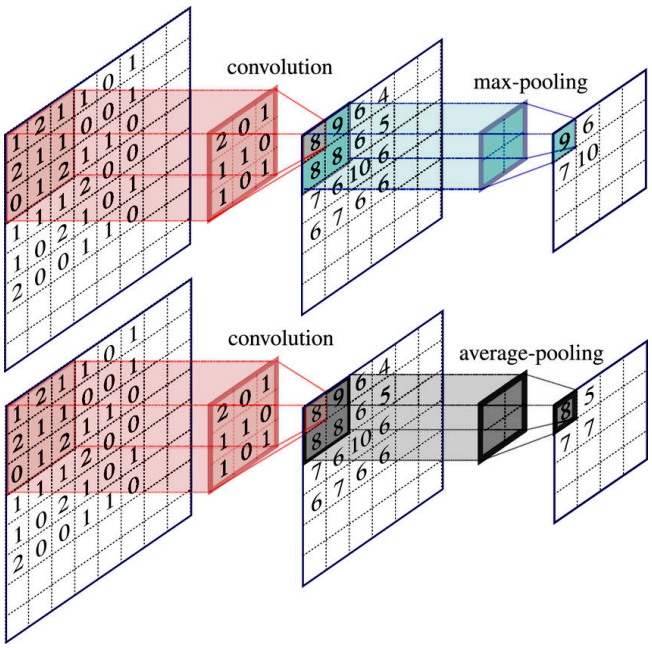


Figure 2.6: Example illustrating the pooling layer types [36]

- **Fully Connected Layer (FCN):** fully connected layer constitutes the final stage of a

CNN, following the convolution and pooling layers. It plays a crucial role in the classification process, combining information from previous layers to make predictions about the input image's category. This layer outputs a probability distribution over the possible classes using the softmax activation function, which ensures that the sum of all output values equals one. As the FCN requires a one-dimensional input vector, the three-dimensional feature maps from previous layers are first flattened. Each neuron in FCN is connected to each neuron in the previous layer, and the number of output nodes typically corresponding to the number of target classes [37].

- **Dropout Layer:** represents a common technique in CNNs to enhance model generalization and reduce overfitting. During training, approximately 50% of neurons are randomly deactivated, disrupting overly reliant connections and encouraging the network to learn more robust and diverse features. As a result, the model generalizes more effectively to unseen data, leading to improved performance on both validation and test sets [37].

4.2.2 CNN Architectures

CNN architectures have evolved significantly, introducing various structural innovations that enhance performance across a wide range of computer vision tasks. Each architecture is designed with specific goals in mind—whether to improve accuracy, reduce computational cost, or enable deployment in real-time and resource-constrained environments. In the following section, we highlight some of the most prominent CNN architectures that have shaped the field and are widely used as foundational models in deep learning applications.

- **LeNet-5:** It devised by Yann LeCun in the 1990s, LeNet is one of the earliest successful applications of Convolutional Neural Networks (CNNs). Among these, the most notable is the LeNet-5 architecture, developed in 1998 to recognize handwritten digits for postal code identification in mail services. LeNet-5, a 7-layer CNN model, was adopted by several banks to identify handwritten digits on scanned checks, using grayscale input images. It receives a grayscale image as input and aims to recognize digit patterns. The model employs 5×5 convolutional filters with a stride of 1, followed by pooling layers for downsampling. The resulting features are then passed through fully connected (FC) layers—one with 120 nodes, followed by another with 84 nodes. Nonlinear activation functions such as Sigmoid or Tanh are used. The output layer contains 10 nodes

representing the digit classes from 0 to 9. LeNet-5 was trained on the MNIST dataset, which includes 60,000 training samples. Figure 2.7 illustrates the overall architecture of LeNet-5 [38].

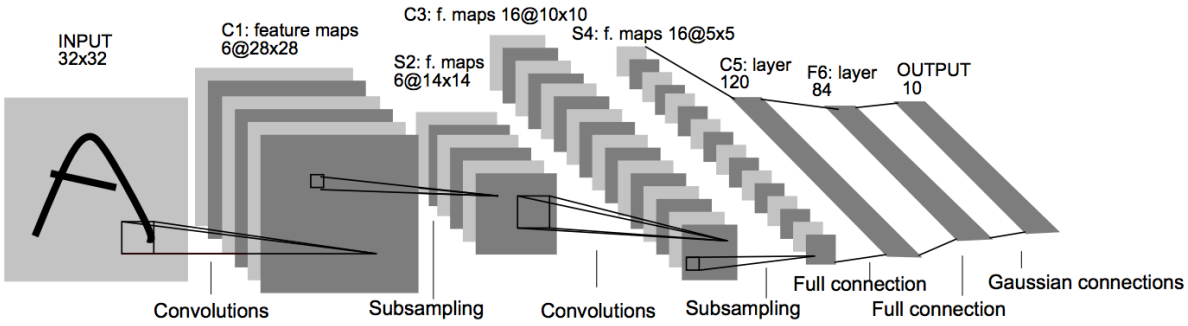


Figure 2.7: LeNet architecture [38]

- **AlexNet:** It developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, AlexNet is a deep and groundbreaking CNN architecture that won the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC-2012). The model consists of five convolutional layers, followed by pooling layers and three fully connected (FC) layers, as illustrated in Figure 2.8. It begins with an input image and applies 96 filters with a stride of 4 in the first convolutional layer, reducing the spatial dimensions. The following pooling layer performs 3×3 max pooling with a stride of 2. The network continues through additional layers and eventually reaches an FC layer with 9216 nodes, followed by two more FC layers with 4096 nodes each. The final output layer uses the Softmax function to classify across 1000 categories. AlexNet contains approximately 60 million parameters [38]. Its success relies on several practical techniques, such as the use of Rectified Linear Units (ReLU) for non-linearity, data augmentation, and dropout layers. Specifically, ReLU significantly accelerates the training process. Data augmentation improves classification performance and reduces overfitting by generating more training samples—e.g., by cropping small patches and flipping them horizontally. Dropout layers are applied within the fully connected layers to further mitigate overfitting. Overall, the success of AlexNet popularized the application of deep CNNs in visual recognition tasks, establishing it as a foundational architecture in modern deep learning [39].

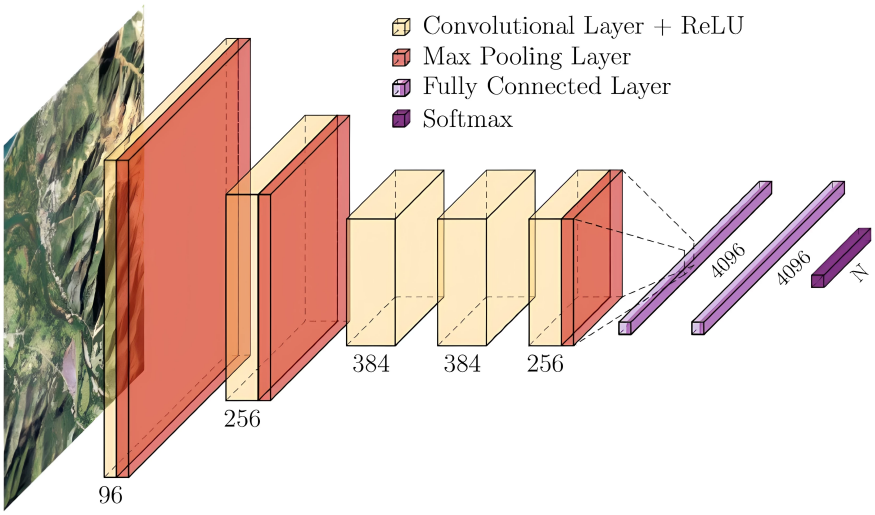


Figure 2.8: AlexNet architecture [40]

- **VGGNet:** It devised by Simonyan and Zisserman from the University of Oxford in 2014, VGGNet represents a significant advancement in CNN architecture. Available in configurations with 16 and 19 layers, the model utilizes 3×3 filters with a stride and padding of size one, alongside 2×2 max pooling with a stride of 2. This strategic design enhances network depth while maintaining a manageable number of parameters. Although the spatial dimensions decrease due to max pooling, the number of filters increases progressively in deeper layers, ensuring robust feature extraction. VGGNet-19 follows the same architectural design as VGGNet-16, with additional convolutional layers that contribute to a deeper network and comparable performance. VGGNet-16 contains approximately 138 million parameters. Despite this large number, the architecture maintains a commendable uniformity, with a consistent filter count increment. Its architectural simplicity (Figure 2.9), paired with small-sized filters, contributes to VGGNet's widespread use and effectiveness in image classification tasks [22].

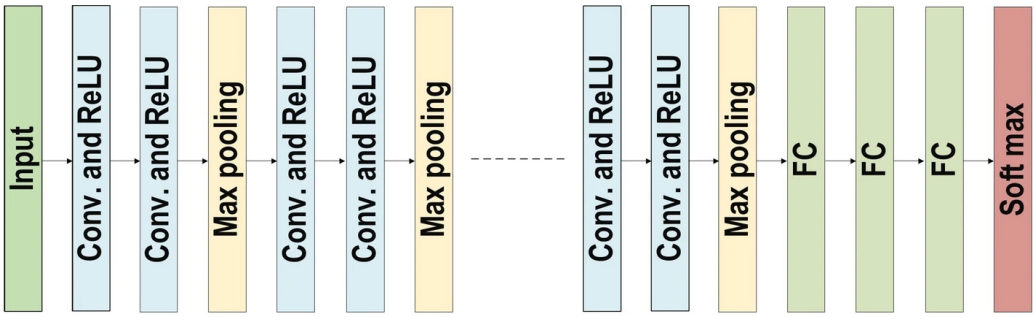


Figure 2.9: VGGNet architecture [33]

- **ResNet:** Introduced in 2015 by He et al. from Microsoft Research, ResNet marked a major breakthrough in deep learning by winning the ImageNet competition. It challenged the belief that deeper networks necessarily lead to overfitting due to vanishing gradients or excessive parameters. The key innovation in ResNet is the use of residual blocks, which connect layers through shortcut connections.These blocks cleverly connect layers within the network, facilitating the ow of gradients during training and addressing the vanishing gradient problem prevalent in deep CNNs (Figure 2.10). By learning residual functions with respect to the input, ResNet enables efficient optimization and achieves higher accuracy even with increased network depth [22].

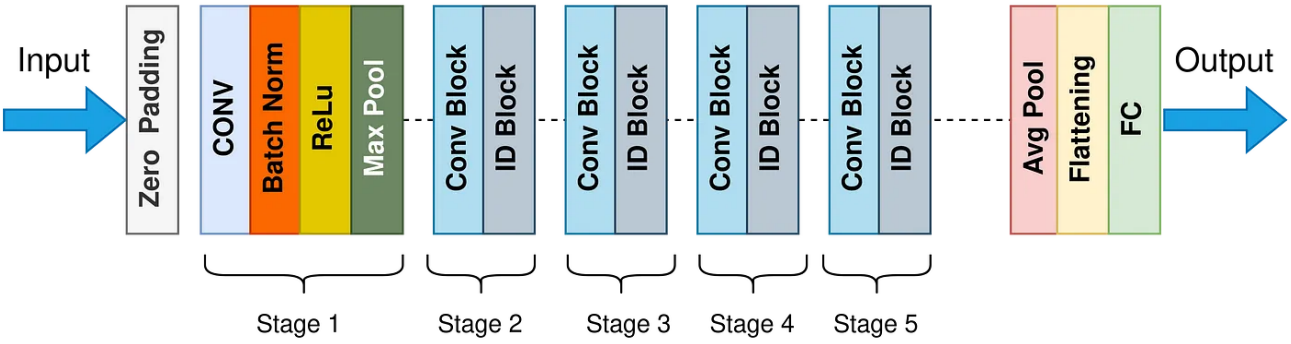


Figure 2.10: ResNet architecture [41]

- **EfficientNet:** Devised by Mingxing Tan and Quoc V. Le from Google AI in 2019, EfficientNet represents a major innovation in CNN architecture through the introduction of compound scaling. Unlike traditional scaling methods that independently scale network dimensions such as depth, width, or input resolution, EfficientNet employs a uniform and principled scaling approach. This method simultaneously balances all three dimensions, enabling the model to achieve state-of-the-art accuracy with significantly fewer parameters and Floating Point Operations (FLOPs) [42].

  EfficientNet is designed to address the growing computational cost of building deeper and wider CNNs. For context, Google previously developed the GPipe framework, which demonstrated that increasing model size can lead to significant improvements in accuracy on benchmarks such as ImageNet. Using GPipe, they trained a convolutional neural network that scaled up from 6.8 million to 557 million parameters. Although this massive increase in depth and width improved accuracy, it also led to high resource

consumption, making them impractical for real-time or resource-constrained environments such as mobile or edge devices [42].

The baseline model, EfficientNet-B0, is optimized through neural architecture search and serves as the foundation for the family of models EfficientNet-B1 to B7. Each subsequent variant is systematically scaled in depth, width, and input resolution according to the compound scaling method. Architecturally, EfficientNet-B0 begins with a standard 3×3 convolutional layer, followed by a series of MBConv blocks (Mobile Inverted Bottleneck Convolutions) inspired by MobileNetV2. These blocks incorporate depthwise separable convolutions with varying kernel sizes (e.g., 3×3 and 5×5), enabling efficient feature extraction. As the network deepens, it progressively reduces spatial dimensions while increasing feature richness, culminating in a 7×7×320 feature map that is passed to the classification head. Despite their lightweight nature, these models outperform larger traditional architectures in both speed and accuracy [42][43][44].

Its scalable efficiency (Figure 2.11) and adaptability to various hardware constraints have made EfficientNet a preferred choice for real-world computer vision applications. The architecture family, summarized in Table 2.1, presents detailed performance metrics for each EfficientNet variant (B0 to B7), including the number of parameters, FLOPs, and top-1/top-5 accuracy on the ImageNet benchmark. This comparison highlights the model's scalability and its ability to balance accuracy with computational efficiency across a wide range of configurations, making it an ideal candidate for modern vision tasks demanding both precision and resource awareness.

Table 2.1: EfficientNet Performance Results on ImageNet [42]

Model	**Top-1 Acc.**	**Top-5 Acc.**	**Params**	**FLOPs**
EfficientNet-B0	76.3%	93.2%	5.3M	0.39B
EfficientNet-B1	78.8%	94.4%	7.8M	0.70B
EfficientNet-B2	79.8%	94.9%	9.2M	1.0B
EfficientNet-B3	81.1%	95.5%	12M	1.8B
EfficientNet-B4	82.6%	96.3%	19M	4.2B
EfficientNet-B5	83.3%	96.7%	30M	9.9B
EfficientNet-B6	84.0%	96.9%	43M	19B
EfficientNet-B7	84.4%	97.1%	66M	37B

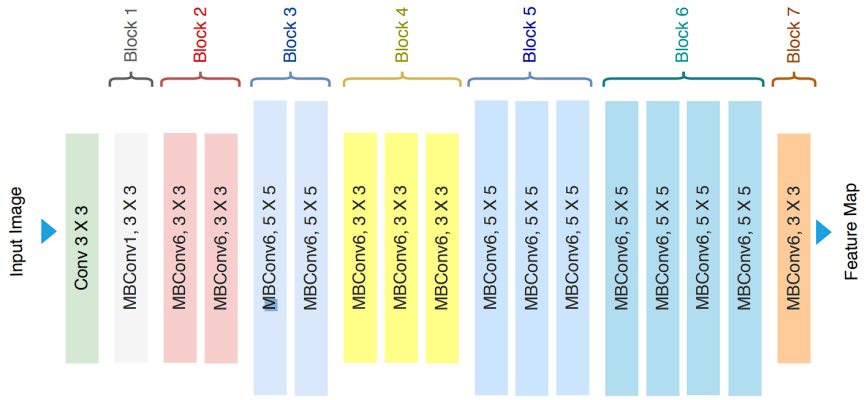


Figure 2.11: EfficientnetB0 Model Architecture [43]

4.2.3 Transfer Learning

Transfer learning encompasses a range of strategies to reuse knowledge gained from one task in a new, often related, task. One common approach is to utilize pre-trained models—originally trained on large-scale datasets like ImageNet—as a foundation for solving novel problems, especially when labeled data is scarce. Among the most widely used techniques are feature extraction, where the pre-trained network is used as a fixed feature extractor, and fine-tuning, where some or all layers of the pre-trained model are further trained for the new task. Typically, this involves replacing the final classification layer to match the target classes and retraining it on the new dataset (Figure 2.12). These approaches allow for a significant reduction in training time and required data volume. Transfer learning has proven effective across various domains, such as image classification, natural language processing, and medical diagnosis, by leveraging learned representations from one domain to boost performance in another [31][37][45].

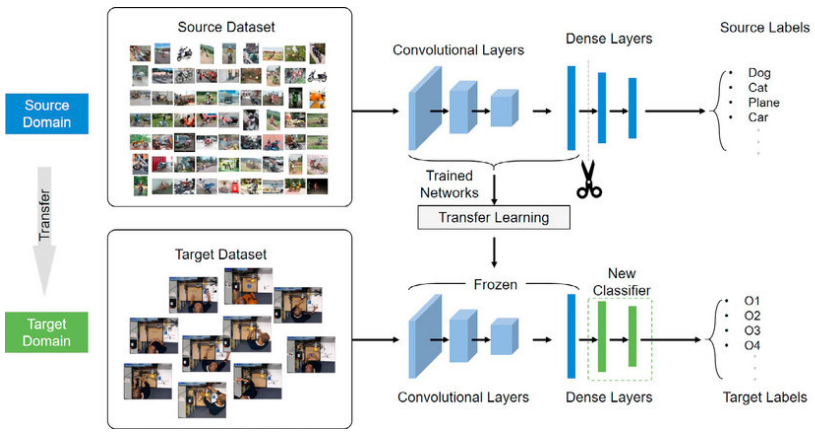


Figure 2.12: The architecture of transfer learning [46]

5 Deep Learning for Malaria Diagnosis

Deep learning techniques have significantly improved malaria detection, offering enhanced accuracy and efficiency compared to traditional diagnostic methods. The following researches support this assertion:

In a study by Barath Narayanan et al. (2019) [47], several deep learning architectures were evaluated for malaria parasite detection in cell images. The NIH malaria dataset (27,558 images) was randomly split 80/20 for training and testing, respectively. Images were resampled (50×50) and preprocessed using color constancy. A fast CNN architecture was proposed, consisting of convolutional layers (3×3 filters), batch normalization, ReLU activation, and 2×2 max-pooling. Training was conducted using gradient descent with momentum (batch size of 64, initial learning rate of 0.0001). Furthermore, transfer learning models based on AlexNet, ResNet, VGG-16, and DenseNet were explored. The proposed ensemble method (Average of All) achieved an overall accuracy of 96.7% and an AUC of 0.994 on the test set.

In a study by Muhammad Umer et al. (2020) [48], a stacked CNN architecture was proposed for malaria parasite detection in thin blood smear images. The NIH malaria dataset (27,558 images) was split 70/30 for training and testing. The architecture included 5 convolutional layers with varying kernel sizes (2×2, 3×3, 4×4), 2 max-pooling layers (2×2), 1 average pooling layer (3×3), 4 dense layers, and 8 dropout layers (20%), with ReLU activation. The model was optimized using the Adam optimizer (batch size of 32) over 13 epochs with binary cross-entropy loss. The results are shown in the table 2.2.

In a study by Varun Magotra and Mukesh Kumar Rohil (2022) [49], a lightweight convolutional neural network (CNN) was proposed for malaria diagnosis using thin blood smear images. The NIH malaria dataset (27,558 images) was used with three different training/testing splits (30/70, 50/50, and 70/30). The proposed architecture consisted of 6 convolutional layers (with increasing filters from 16 to 128), each followed by ReLU activation and max-pooling layers, along with 3 dropout layers and 3 batch normalization layers to reduce overfitting and improve convergence. The flattened features were passed through two dense layers before a final sigmoid-activated output neuron. The model was trained using the Adam optimizer with a learning rate of 0.001, and binary cross-entropy loss. Data augmentation techniques such as rotation, zoom, shift, and horizontal flipping were applied. The proposed model

achieved an accuracy of up to 96.23% when using the 70/30 training/testing split, outperforming transfer learning models based on VGG-19 and Inception-v3.

In a study by Muhammad Mujahid et al. (2024) [4], a deep learning model based on EfficientNet-B2 was proposed for malaria detection using red blood cell smear images. The balanced dataset (27,558 images) was split 80/20 for training and testing. The model employed dropout, ReLU activation, and was trained with a learning rate of 0.0001, batch size of 32, and 15 epochs. the model achieved 97.57% accuracy, 0.9921 AUC, 0.9755 F1-score, and 0.9862 precision, with a testing loss of 0.0995.

Sorio Boit and Rajvardhan Patil (2024) [50] conducted a comprehensive study to develop an efficient and scalable deep learning model for malaria parasite detection in microscopic red blood cell images, with a particular focus on enhancing classification performance while maintaining low computational complexity. Their proposed model, named EDRI (Efficient-Dense-Residual-Inception), was designed by integrating multiple advanced architectures, including EfficientNetB2 as the backbone, DenseNet blocks for improved feature reuse, Residual blocks to ensure stable training of deep networks, and Inception modules to capture multi-scale features across varied spatial resolutions. The model architecture omits the fully connected layers of EfficientNetB2 to allow seamless integration with custom-designed modules and applies Global Average Pooling followed by a Dropout layer to reduce overfitting. The final classification layer uses a sigmoid activation function to enable binary classification between parasitized and uninfected cells. The EDRI model was trained on a balanced dataset and optimized for performance in resource-constrained settings. These architectural innovations enabled the model to generalize effectively across diverse imaging conditions, capturing complex spatial patterns and achieving state-of-the-art classification results.

Table 2.2: Summary of Related Works on Malaria Diagnosis Using Deep Learning

Author(s)	Year	Technique (Models)	Parameters (Features)	Dataset (Samples)	Data Split	Accuracy (%)
Barath Narayanan et al. [47].	2019	Fast CNN (Proposed)	3×3 Conv, BatchNorm, ReLU, 2×2 MaxPooling, Gradient Descent with Momentum (batch=64, lr=0.0001).	NIH Malaria Dataset (27,558 images)	80/20	96.0
		AlexNet				96.4
		ResNet				96.0
		VGG-16				96.5
		DenseNet				96.6
Muhammad Umer et al. [48].	2020	Stacked CNN-1 layer	5 Conv (2×2, 3×3, 4×4), 2 MaxPooling (2×2), 1 AvgPooling (3×3), 8 Dropout (20%), 4 Dense, ReLU, Adam (batch=32, 13 epochs), Binary Cross-Entropy	NIH Malaria Dataset (27,558 images)	70/30	50.145
		Stacked CNN-3 layers				61.412
		Stacked CNN-5 layers				99.879
		Stacked CNN-5 with 5-fold)				99.964
Varun Magotra et al. [49].	2022	Proposed Lightweight CNN	6 Conv (16–128 filters), M-Pooling, 3 Dropout, 3 BatchNorm, 2 Dense, ReLU, Adam (lr=0.001)	NIH Malaria Dataset (27,558 images)	70/30	96.36
		VGG-19				93.34
		Inception-v3				91.06
Muhammad Mujahid et al [4].	2024	EfficientNet-B2	3 dense layers, 4 BN layers. ReLU/Sigmoid, dropout. LR = 0.0001, BS = 32. 15 epochs, Adam optimizer.	NIH Malaria Dataset (27,558 images)	80/20	97.57
					90/10	97.50
Sorio Boit et al [50].	2024	EDRI (EfficientNetB2 + DenseNet + ResNet + Inception)	EfficientNetB2 backbone. Dense/Residual/Inception blocks. GAP, Dropout, Dense + Sigmoid. Multi-scale feature extraction.	NIH Malaria Dataset (27,558 images)	80/10/10	97.68

6 Conclusion

This chapter concludes that deep learning methods have significantly enhanced malaria detection, with numerous studies demonstrating their effectiveness across various scenarios and overcoming the limitations of traditional machine learning techniques. These methods not only improve diagnostic accuracy but also accelerate treatment processes. In the next chapter, we present our proposed approach and evaluate its effectiveness for malaria disease detection from red blood cell images.

Chapter 3

PROPOSED METHOD FOR MALARIA DISEASE CLASSIFICATION

1 Introduction

The previous chapters provided background information on malaria and its diagnostic approaches, encompassing both traditional and modern diagnostic methods. Additionally, they offered a comprehensive overview of machine learning and deep learning techniques, highlighting their applications in malaria detection and diagnosis. This chapter presents the proposed deep learning-based approach for the automatic classification of malaria-infected and uninfected red blood cells, along with its validation results. It outlines the methodology, implementation, and experimental outcomes. First, an overview of the system architecture is provided, followed by a detailed description of each phase. Finally, the experimental design is presented, and the corresponding results are discussed.

2 Methodology

The overall schema of the proposed method for the automatic classification of malaria-infected and uninfected red blood cell images follows a multi-step methodology, as illustrated in Figure 3.1. The process begins with the preparation of an available dataset of microscopic red blood cell images, which includes both malaria-infected and uninfected cells. This is followed by a data preprocessing step, including resizing, normalization, and data augmentation to enhance model performance and robustness. The dataset is then divided into training, validation, and testing subsets to ensure fair evaluation and overcome the problem of overfitting. The main step of the methodology involves the proposal and implementation of a deep

learning model for malaria classification. Finally, the model is trained and evaluated using standard metrics to assess its ability to correctly classify both infected and uninfected cells.

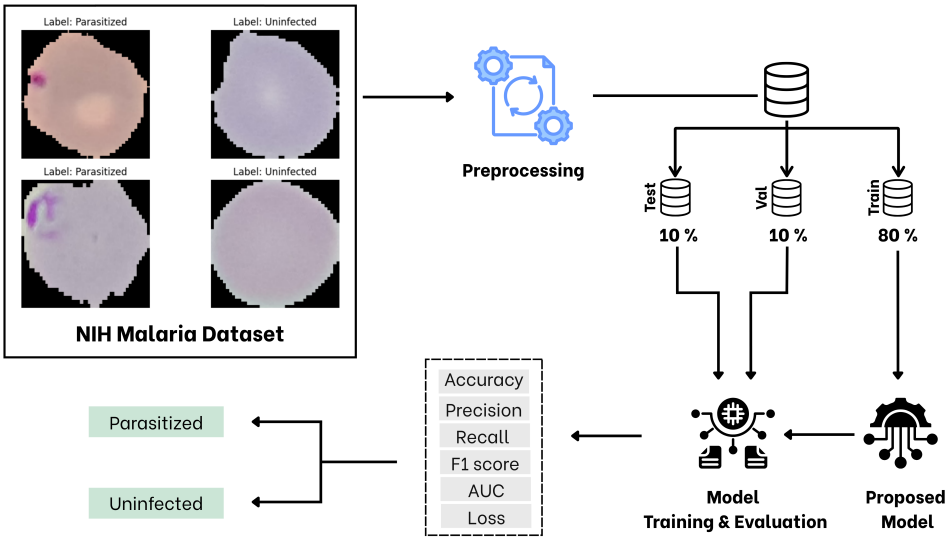


Figure 3.1: Work flow of proposed methodology

2.1 Dataset Description

The dataset used in this study was obtained from the publicly available malaria cell image dataset provided by the National Library of Medicine (NLM) at the National Institutes of Health (NIH) [51]. The NIH is a U.S.-based biomedical research agency located in Bethesda, Maryland. It operates under the U.S. Department of Health and Human Services and is one of the world's foremost medical research centers. The dataset contains 27,558 labeled cell images, equally divided into 13,779 parasitized and 13,779 uninfected samples. These images were collected from blood smear slides prepared from 150 infected patients and 50 uninfected individuals. Figure 3.2 illustrates examples of both parasitized and uninfected red blood cells.

2.2 Data Preprocessing and Splitting

Preprocessing is a crucial step in deep learning-based image classification. The dataset used in this study contains a balanced number of Parasitized and uninfected cell images. Since the samples have different sizes, all images were resized to a fixed size of 224×224 pixels

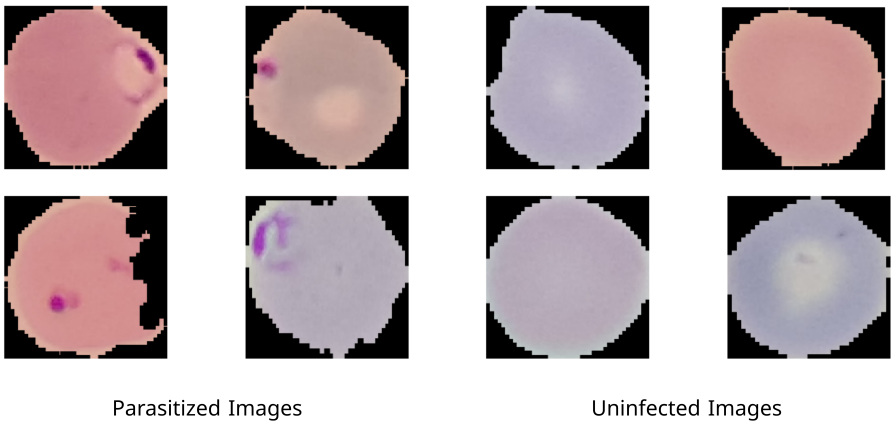


Figure 3.2: Sample images of parasitized and uninfected red blood cells from the Malaria Cell Image Dataset.

to achieve the input requirements of the EfficientNetB2 model. Furthermore, the dataset images were augmented using common techniques such as rotation, zooming, shifting, and horizontal flipping to enhance the model's generalization capability. In order to collect meaningful results, the dataset was randomly and independently divided into three subsets: 80% for training, 10% for validation, and 10% for testing, as shown in Table 3.3. Parasitized images and uninfected images were labeled as class ”0” and class ”1,” respectively.

After preprocessing, the next important step was to split the dataset. The images were divided into three subsets: 80% for training, 10% for validation, and 10% for testing. The splitting was performed separately for each class to maintain class balance across all subsets. This ensures fair evaluation and robust model performance. The class labels were assigned as 0 for Parasitized and 1 for Uninfected. Table 3.3 presents the distribution of parasitized and uninfected images after splitting the dataset into training, validation, and testing sets.

Dataset	**Parasitized cells**	**Uninfected cells**	**Total images**
Training images	11,023	11,023	22,046
Validation images	1,378	1,378	2,756
Testing images	1,378	1,378	2,756
Total images	13,779	13,779	27,558

Table 3.3: Distribution of parasitized and uninfected images after data splitting.

2.3 Overview of the Proposed Methodology

Deep learning models have demonstrated exceptional capabilities in extracting complex patterns from visual data, making them highly suitable for medical image classification tasks. In this study, a deep convolutional neural network (CNN) based on the EfficientNet-B2 architecture is applied to automatically classify red blood cell images as either parasitized or uninfected. While deep learning Network typically require large datasets and significant computational power, their efficiency can be enhanced through optimized architectural choices, transfer learning, and regularization techniques.

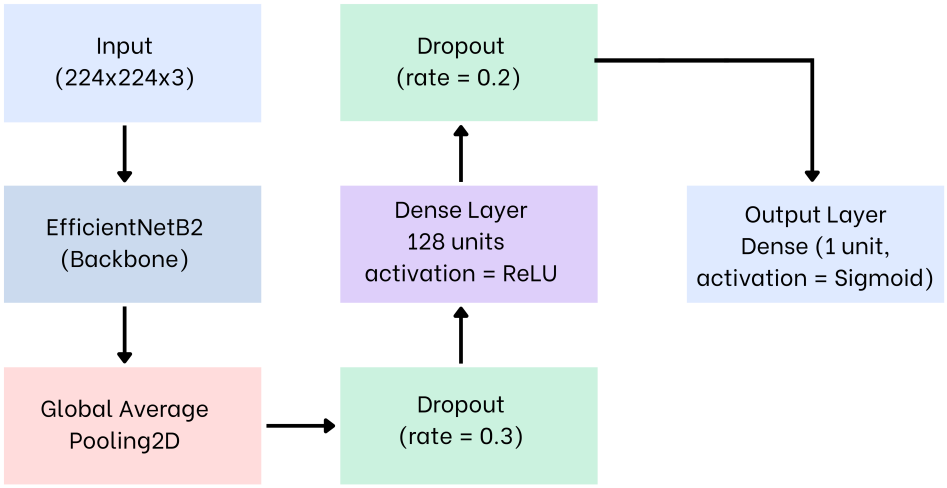


Figure 3.3: the architecture of the proposed model

Thus, the EfficientNet-B2 network, which was pre-trained on the large-scale ImageNet dataset, was fine-tuned using the NIH malaria dataset. Figure 3.3 illustrates the architecture of the proposed methodology. First, the pre-trained EfficientNet-B2 model was loaded, and its base layers were frozen during the initial training phase to retain previously learned features. To adapt the model to the binary classification task, several layers were added on top of the base model. These included a GlobalAveragePooling2D layer to reduce spatial dimensions, followed by a Dropout layer with a rate of 0.3 to prevent overfitting. Next, a Dense layer with 128 units and ReLU activation was added to learn high-level representations, followed by another Dropout layer with a rate of 0.2. Finally, a Dense output layer with a single unit and a sigmoid activation function was used to produce the probability score for binary classification. This configuration enables the model to efficiently extract and classify features that are essential for detecting malaria-infected cells.

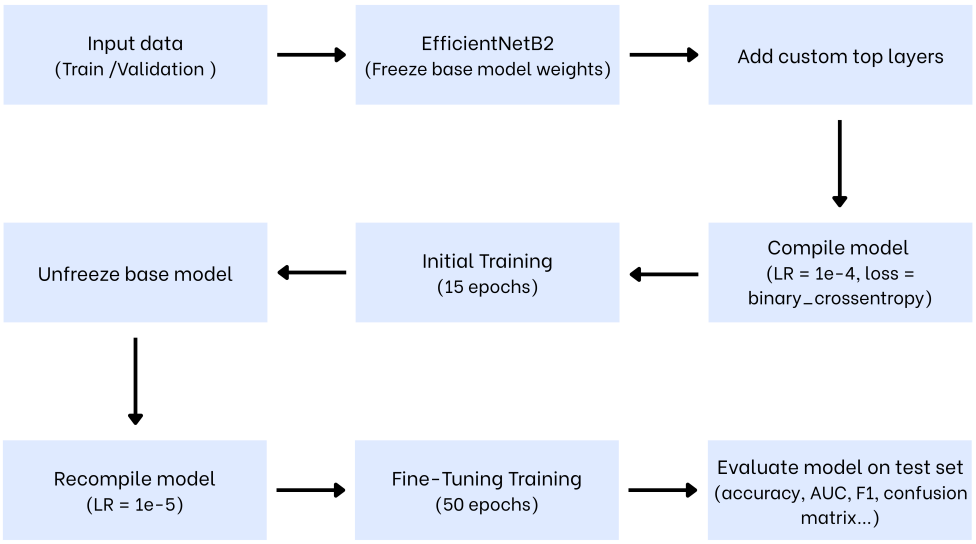


Figure 3.4: Training and Evaluation Pipeline

The training step consists of two main phases (as shown in Figure 3.4). In the first phase, the base model was entirely frozen to train only the newly added classification layers. In the second phase, the entire model was unfrozen to allow fine-tuning of both the base and custom layers.

The model was trained using mini-batches of size 32. During the initial phase, the model was compiled with the Adam optimizer (learning rate = 1×10^{-4}) and trained for 15 epochs. Afterward, the base model was unfrozen and recompiled with a reduced learning rate (1×10^{-5}), followed by 50 epochs of fine-tuning.

Table 3.4: The number of layers, types, output-shape and its parameters for the proposed model. (Parameters: 7,949,050. Trainable parameters: 180,481. Non-trainable parameters: 7,768,569)

No.	Layer Type	Output Shape	Parameters
1	EfficientNetB2	(None, 7, 7, 1408)	7,768,569
2	GlobalAveragePooling2D	(None, 1408)	0
3	Dropout (rate = 0.3)	(None, 1408)	0
4	Dense (128, ReLU)	(None, 128)	180,352
5	Dropout (rate = 0.2)	(None, 128)	0
6	Dense (1, Sigmoid)	(None, 1)	129

To overcome the problem of overfitting, early stopping was applied over a period of 7 epochs, restoring the best weights based on validation AUC. Additionally, the learning rate

was automatically reduced by a factor of 0.3 when the validation AUC plateaued for 3 consecutive epochs.

The final model was evaluated on the test set using key performance metrics including accuracy, precision, recall, F1-score, and AUC. Moreover, visualization tools such as the confusion matrix and ROC curve were utilized to further interpret the model's performance.

3 Results and Discussion

3.1 Experimental Environment

The experiments were conducted on a remote Linux-based machine accessed via SSH. The system was running Ubuntu with a symmetric multiprocessing (SMP) supported kernel (version #107-Ubuntu SMP), enabling efficient parallel processing. The hardware environment included an NVIDIA A5000 GPU and 125 GB of RAM, providing substantial computational capacity for deep learning tasks. Additionally, some experiments were executed using Google Colab, leveraging cloud-based GPU acceleration. The software environment consisted of Python 3.8, TensorFlow 2.x, and supporting libraries such as Keras, NumPy, and Pandas.

3.2 Performance Evaluation Metrics

To evaluate the performance of the proposed malaria classification model, multiple widely used metrics were employed. These metrics provide a holistic assessment of the model's ability to distinguish between parasitized and uninfected cells.

- **Accuracy:** Accuracy measures the overall correctness of the model. It is computed as shown below in Equation 3.1 [52].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

- **Precision:** Represents the proportion of correctly predicted positive samples among all predicted positives and is defined as Equation 3.2 [52].

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.2}$$

- **Recall (Sensitivity):** Indicates the proportion of actual positive samples that were correctly identified, which are calculated as shown below in Equation 3.3 [52].

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.3}$$

- **F1-Score:** The F1 score is the harmonic mean of precision and recall, providing a balance between the two in cases of class imbalance, It is computed as shown below in Equation 3.4 [52].

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

- **AUC (Area Under the ROC Curve):** Reflects the model's ability to distinguish between the two classes across all classification thresholds. A higher AUC indicates better performance [52].

- **Confusion Matrix:** A tabular visualization showing the distribution of predicted and actual labels, helping to identify types of misclassifications such as false positives and false negatives [52].

  Where:

  - TP (True Positive): correctly predicted parasitized images
  - TN (True Negative): correctly predicted uninfected images
  - FP (False Positive): uninfected images incorrectly predicted as parasitized
  - FN (False Negative): parasitized images incorrectly predicted as uninfected

3.3 Evaluation Results

This section illustrates three sets of experiments:

First Set of Experiments – Performance Evaluation of the Main Contribution for Malaria Disease Classification: this set evaluates the effectiveness and accuracy of the proposed method in classifying malaria-infected cells.

Second Set of Experiments – Comparative Analysis of Different Models for Malaria Classification: this set compares several deep learning models to assess their performance in classifying malaria-infected cells.

Final Set of Experiments – Comparison of the Proposed Approach with Related Work: this set contrasts the proposed method with existing approaches in the literature to demonstrate its improvements and contextualize its contribution within the field.

3.3.1 First Set of Experiments – Performance Evaluation of the Main Contribution for Malaria Disease Classification:

The proposed model was evaluated on the test set of the NIH Malaria dataset, achieving an overall accuracy of 99.35%. This high level of accuracy indicates that the model correctly classified the vast majority of the blood smear images, effectively distinguishing between parasitized and uninfected red blood cells (RBCs). The model's precision of 98.99% reflects its ability to minimize false positives, ensuring that most cells predicted as parasitized were indeed infected. Similarly, the recall of 99.71% demonstrates the model's effectiveness in identifying actual parasitized cells, thereby reducing the number of false negatives.

The F1 score of 99.35%, being the harmonic mean of precision and recall, highlights a strong balance between the two metrics. This balance is particularly critical in medical diagnostics, where both false positives and false negatives can lead to serious consequences. Furthermore, the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) of 99.98% reinforces the model's exceptional discriminative ability across various classification thresholds. An AUC score close to 1.0 signifies near-perfect performance in distinguishing between the two classes.

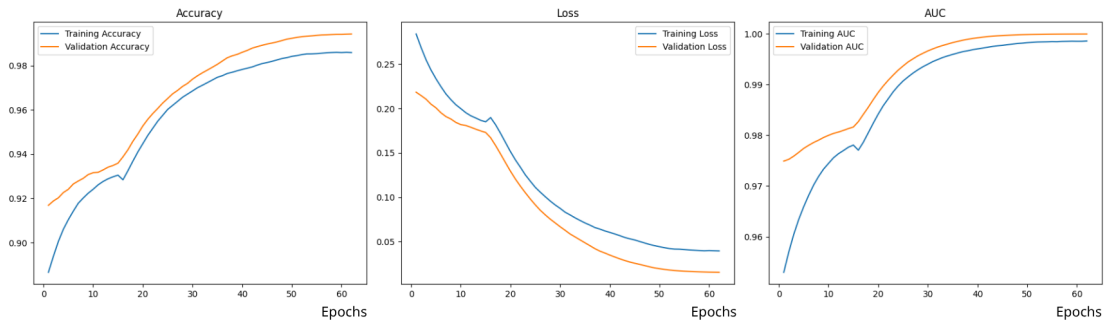


Figure 3.5: Model Performance Curves (Accuracy, Loss, AUC)

Furthermore, the model achieved a final loss of 0.0194, indicating not only high accuracy but also strong confidence and low uncertainty in its probabilistic outputs. The classification report confirms consistently high scores for both classes, with precision,

recall, and F1-score values of approximately 0.99 for Parasitized and Uninfected class.

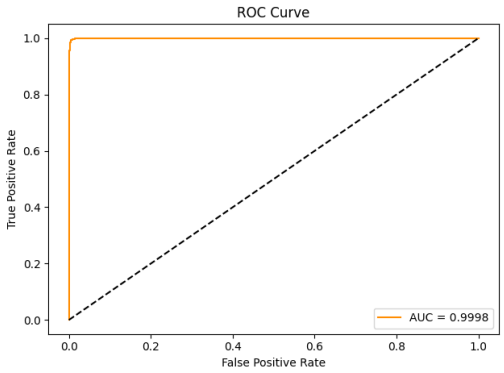


Figure 3.6: Receiver Operating Characteristic (ROC) Curve

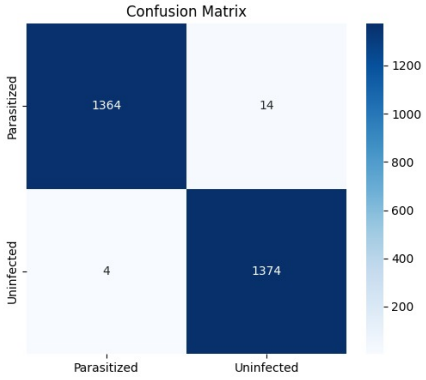


Figure 3.7: Confusion Matrix

Figure 3.7 illustrates the confusion matrix of the proposed model on the test set. The model successfully classified 1364 parasitized and 1374 uninfected samples correctly, with only 14 false positives (uninfected classified as parasitized) and 4 false negatives (parasitized classified as uninfected). From these results, two key findings can be observed. First, the proposed methodology demonstrates strong generalization capabilities and effectively derives powerful feature representations, enabling it to overcome challenges commonly associated with microscopic red blood cell images, as previously discussed. Second, the model achieved significantly better classification performance, outperforming both traditional methods and prior studies.

3.3.2 Second Set of Experiments – Comparative Analysis of Different Models for Malaria Classification:

This subset of experiments applied three widely used pre-trained deep learning models, namely VGG-16, MobileNetV2, and EfficientNet-B2, as previously introduced. The first model, VGG-16, was employed as a feature extractor; the extracted feature vectors were then fed into a supervised classifier (SVM) to obtain the classification results. In contrast, the other two models, MobileNetV2 and EfficientNet-B2, were fine-tuned using the same NIH Malaria dataset to perform end-to-end classification.

- **Hybrid Approach: VGG-16 with SVM:**

Here, in this experiment, the VGG-16 convolutional neural network was employed as a feature extractor for red blood cell images by removing all fully connected layers, thereby focusing exclusively on the last convolutional layers to capture spatial features. All input images were first resized to 224×224 pixels to match the input requirements of VGG-16. Then the output from the final convolutional block was flattened, resulting in a 25,088-dimensional feature vector representing each image. Then, Principal Component Analysis (PCA) reduced the feature dimensionality to the top 100 components to enhance efficiency and reduce overfitting. This reduced feature was fed into a supervised classifier (SVM) with a radial basis function (RBF) kernel to obtain the classification results. This approach achieved strong performance, with 97.44% accuracy, 96.57% precision, 98.35% recall, and a 97.45% F1-score, demonstrating the effectiveness of integrating deep feature extraction with classical classifiers for malaria detection.

- **Transfer Learning with MobileNetV2:**

In this experiment, the MobileNetV2 architecture was fine-tuned to develop a binary classification model for detecting malaria-infected cells. As previously explained with EfficientNet-B2, the convolutional layers of the base model were initially frozen to retain the pre-trained weights, and additional classification layers were added on top. These layers included a GlobalAveragePooling2D layer, followed by a Dropout layer with a rate of 0.3, a Dense layer with 128 units and ReLU activation, another Dropout layer with a rate of 0.2, and a final Dense layer with sigmoid activation to produce binary output. The training process consisted of two phases: an initial phase with the base model frozen, and a fine-tuning phase in which all

layers were unfrozen and retrained using a lower learning rate. The model was compiled with the Adam optimizer (with an initial learning rate of 1×10^{-4}, reduced to 1×10^{-5} during fine-tuning), and binary cross-entropy was used as the loss function. Accuracy and AUC were employed as the primary evaluation metrics. To enhance training efficiency and prevent overfitting, EarlyStopping was applied to halt training when the validation AUC plateaued, and ReduceLROnPlateau was used to dynamically lower the learning rate based on performance. The fine-tuned MobileNetV2 model demonstrated strong performance on the validation set, achieving an AUC of 0.9972, an F1-score of 97.62%, an accuracy of 97.60%, a precision of 96.38%, and a recall of 98.90

- **Transfer Learning with EfficientNetB2 (First scenario):**

In this experiment, the available dataset of microscopic red blood cell images was prepared and divided as previously described. To enhance model generalization and overcome overfitting, various data augmentation techniques were applied during training. These included random rotations up to 30°, vertical and horizontal shifts of up to 25%, random zooming and cropping, vertical and horizontal flipping, and random color shifting. To address class imbalance between parasitized and uninfected samples, class weights were automatically computed and incorporated during training. The EfficientNetB2 architecture, pre-trained on ImageNet, was fine-tuned for the binary classification task. The top classification layers of the base model were removed, and the following custom layers were appended:

The base model EfficientNetB2, pretrained on ImageNet, was used as the feature extractor, with the top classification layers removed. The following custom layers were added:

- GlobalAveragePooling2D
- BatchNormalization, Dropout (rate = 0.4), Dense (512 units, ReLU)
- BatchNormalization, Dropout (rate = 0.3), Dense (128 units, ReLU)
- Dense (1 unit, Sigmoid)

The training process was conducted in two phases. In the first phase, the base model was entirely frozen to train only the newly added classification layers. In the second phase, the entire model was unfrozen to allow fine-tuning of both the base and custom layers. The Adam optimizer was employed with a learning rate of 1×10^{-4}, and training was carried out for up to 100 epochs.

Final evaluation results showed a test loss of 0.0530, an accuracy of 98.00%, a precision of 97.61%, and a recall of 98.93%, indicating the strong effectiveness of the fine-tuned EfficientNetB2 model in detecting malaria-infected cells.

Table 3.5 summarizes the quantitative results of all conducted experiments. As illustrated, the proposed EfficientNetB2-based model outperformed the other approaches across most evaluation metrics, particularly in terms of accuracy and AUC. These results demonstrate the model's strong generalization capability and emphasize its potential for reliable and effective automated classification of malaria-infected cell images.

Table 3.5: Comparison of classification performance across different models.

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**AUC**
MobileNetV2 + PCA + SVM	94.38%	92.73%	96.30%	94.48%	98.40%
VGG16 + FC1 + SVM	95.10%	95.13%	95.07%	95.10%	98.65%
ResNet50 + PCA + SVM	95.17%	94.37%	96.08%	95.22%	98.81%
VGG16 + FC1 + PCA + SVM	95.86%	95.08%	96.73%	95.90%	98.80%
EfficientNetB2 + SVM	96.37%	95.71%	97.10%	96.40%	–
VGG16 + PCA + SVM	97.44%	96.57%	98.35%	97.45%	99.68%
MobileNetV2 (Transfer Learning)	97.60%	96.38%	98.90%	97.62%	99.72%
EfficientNetB2 (Transfer Learning)	98.00%	97.61%	98.93%	–	99.82%
Proposed Model	99.35%	98.99%	99.71%	99.35%	99.98%

3.3.3 Final Set of Experiments – Comparison of the Proposed Approach with Related Work:

This set of experiments compares the proposed method with existing approaches from the literature to highlight its improvements and contextualize its contribution within the field of malaria-infected cell image classification. After establishing the strong performance of our proposed method, we proceed to compare it with other techniques reported in the literature. To ensure a fair and objective comparison, it is crucial that the experimental conditions remain consistent across studies. All compared approaches should be validated and tested on the same dataset or on datasets with similar characteristics, thereby maintaining the reliability and relevance of the performance evaluation.

Table 3.6 presents a summary of classification performance metrics—including accuracy, precision, recall, F1-score, and AUC—as reported in various related studies. Barath

Narayanan et al. [47] implemented a basic CNN architecture, achieving 96% accuracy; however, other key metrics such as precision, recall, and F1-score were not reported, limiting the ability to evaluate overall performance. Muhammad Umer et al. [48] proposed a stacked CNN-5 architecture combined with 5-fold cross-validation, reporting near-perfect results with 99.96% accuracy, 100% precision, and a 99.96% F1-score. While impressive, these results may be optimistic due to the gap of detailed reproducibility information.

Varun Magotra et al. [49] developed a custom CNN model, achieving 96.36% accuracy with acceptable precision (92%) and high recall (97%), resulting in an F1-score of 94%. Muhammad Mujahid et al. [4] transferd learning using EfficientNet-B2, achieving 97.57% accuracy, 96.59% precision, and a strong AUC of 99.21%, indicating solid generalization. Sorio Boit et al. [50] introduced a hybrid ensemble model (EDRI), combining EfficientNetV2, DenseNet, ResNet, and Inception architectures. This ensemble achieved 97.68% accuracy, 98.88% precision, and an AUC of 99.76%. However, the model's complexity and computational demands could limit its practical deployment in resource-constrained environments.

In contrast, the proposed fine-tuned EfficientNetB2 model outperformed existing approaches across nearly all metrics, achieving 99.35% accuracy, 98.99% precision, 99.71% recall, 99.35% F1-score, and a remarkable AUC of 99.98%. These results demonstrate a strong balance between sensitivity and specificity, with minimal misclassifications. As we can see, our proposal is the most generalized and provides the best performance in all cases. Specifically, our approach is robust and the most generalized across to various factors, including differences in cell shape, density, and staining properties, as well as ambiguities in distinguishing certain cell types.

Table 3.6: Performance metrics of deep learning models for malaria detection using the NIH Malaria dataset

Reference	Method	No. of Images	Accuracy	Precision	Recall	F1 Score	AUC
Barath Narayanan et al. [47]	CNN	27,558	96%	–	–	–	99.1%
Muhammad Umer et al. [48]	Stacked CNN-5 with 5-fold	27,558	99.964%	100%	99.928%	99.964%	–
Varun Magotra et al. [49]	Custom CNN	27,558	96.36%	92%	97%	94%	–
Muhammad Mujahid et al. [4]	EfficientNet-B2	27,558	97.57%	96.59%	98.62%	97.55%	99.21%
Sorio Boit et al. [50]	EDRI	27,558	97.68%	98.88%	96.44%	97.65%	99.76%
Proposed Model	EfficientNetB2 (TL + fine-tuning)	27,558	99.35%	98.99%	99.71%	99.35%	99.98%

4 Conclusion

In this chapter, we presented our contribution to malaria disease classification using a publicly available dataset provided by the National Institutes of Health (NIH), which contains 27,558 microscopic images of red blood cells. We began by describing the overall architecture of the proposed system, followed by a detailed explanation of each of its individual phases. Additionally, we reported the experimental results and their validation to demonstrate the effectiveness of the proposed approach. Our main set of experiments focused on evaluating the performance gains achieved by fine-tuning the EfficientNet-B2 network using the NIH malaria dataset. Subsequently, we conducted three additional sets of experiments aimed at comparative analysis. These included evaluations against other popular deep learning models, as well as comparisons with existing methods reported in the literature, to further validate the robustness and superiority of the proposed approach.

The results demonstrated that the proposed methodology achieves strong generalization capabilities, effectively addressing the challenges commonly associated with microscopic red blood cell image classification, as previously discussed. Furthermore, the model outperformed both traditional machine learning methods and previously published deep learning approaches.

General conclusion

conclusion, we applied an EfficientNetB2 model utilizing transfer learning and fine-tuning techniques to classify red blood cell images for accurate malaria diagnosis. The model was trained on the publicly available malaria dataset provided by the National Institutes of Health (NIH). It improved performance, achieving an accuracy of 99.35%, an F1-score of 0.9935, a precision of 98.99%, and a recall of 99.71%. These results validate the effectiveness of the proposed model in distinguishing between infected and uninfected cells.

The strong performance highlights the potential of the model to support medical diagnosis, particularly in low-resource and remote settings. By reducing diagnostic errors and minimizing the need for extensive human intervention, this approach can contribute to the development of intelligent diagnostic tools that assist healthcare professionals in making faster and more accurate clinical decisions. The model's efficiency and reliability suggest that it can serve as valuable aid in healthcare centers, improving early detection and enhancing the overall quality of care. This work represents an important step toward the development of smart systems for automated disease diagnosis.

For future work, we aim to further enhance and optimize the model to improve classification accuracy for malaria-infected cell images. Additionally, we plan to evaluate the generalizability of the proposed methodology in the diagnosis of other diseases. We are also interested in exploring the performance of alternative CNN architectures for malaria classification to identify models that offer better trade-offs between accuracy, efficiency, and deployment feasibility.

Bibliography

[1] World Health Organization. *Malaria*. https://www.who.int/health-topics/malaria. Accessed: February 4, 2025. 2023.

[2] BIMC Hospital Bali. *Types of Malaria and Their Symptoms, the Same or Different?* Accessed: February 4, 2025. 2023. URL: https://bimcbali.com/health-e-update/5-types-of-malaria-and-their-symptoms-the-same-or-different.html.

[3] Casey W. Pirnstill and Gerard L. Coté. "Malaria Diagnosis Using a Mobile Phone Polarized Microscope." In: *Scientific Reports* 5 (2015), p. 13. DOI: 10.1038/srep13368. URL: https://www.nature.com/articles/srep13368.

[4] Muhammad Mujahid et al. "Efficient deep learning-based approach for malaria detection using red blood cell smears." In: *Scientific Reports* (2024). Accessed: April 14, 2025. DOI: 10.1038/s41598-024-63831-0. URL: https://www.nature.com/articles/s41598-024-63831-0.

[5] GeeksforGeeks. *Machine Learning Tutorial*. https://www.geeksforgeeks.org/machine-learning/?ref=home-articlecards. Accessed: February 3, 2025. 2024.

[6] GeeksforGeeks. *Machine Learning Lifecycle*. https://www.geeksforgeeks.org/machine-learning-lifecycle/. Accessed: February 5, 2025. 2025.

[7] GeeksforGeeks. *Supervised Machine Learning*. https://www.geeksforgeeks.org/supervised-machine-learning/. Accessed: February 5, 2025. 2025.

[8] GeeksforGeeks. *What is Unsupervised Learning?* https://www.geeksforgeeks.org/unsupervised-learning/. Accessed: February 5, 2025. 2025.

[9] Andrew Ochoa. *K-means clustering made simple*. Accessed: February 9, 2025. 2020. URL: https://www.blopig.com/blog/2020/07/k-means-clustering-made-simple/.

[10] Dave Bergmann. *Semi-Supervised Learning*. https://www.ibm.com/think/topics/semi-supervised-learning. Accessed: February 5, 2025. 2024.

[11] Roman Panarin. *Semi-Supervised Learning Explained: Techniques and Real-World Applications*. Accessed: February 8, 2025. Mad Devs. 2024. URL: https://maddevs.io/blog/semi-supervised-learning-explained/.

[12] AltexSoft. *Semi-Supervised Learning: Combining the Best of Supervised and Unsupervised Learning*. https://www.altexsoft.com/blog/semi-supervised-learning/. Accessed: February 5, 2025. 2024.

[13] GeeksforGeeks. *What is Reinforcement Learning?* https://www.geeksforgeeks.org/what-is-reinforcement-learning/. Accessed: February 5, 2025. 2023.

[14] Subramanian Chandramouli, Saikat Dutt, and Amit Kumar Das. *Machine Learning*. Pearson Education India, 2018. ISBN: 9789389588132.

[15] GeeksforGeeks. *Why Machine Learning is Important?* https://www.geeksforgeeks.org/why-ml-is-important/?ref=lbp. Accessed: February 5, 2025. 2023.

[16] Dev Kumar Das et al. "Machine learning approach for automated screening of malaria parasite using light microscopic images." In: *Micron* 45 (2013). Accessed: February 8, 2025, pp. 97–106. ISSN: 0968-4328. DOI: 10.1016/j.micron.2012.11.002.

[17] Rashke Jahan and Shahzad Alam. "Improving Classification Accuracy Using Hybrid Machine Learning Algorithms on Malaria Dataset." In: *Proceedings of the 4th International Electronic Conference on Applied Sciences*. Vol. 56. Eng. Proc. 1. Accessed: February 8, 2025. MDPI, 2023, p. 232. DOI: 10.3390/ASEC2023-15924. URL: https://www.mdpi.com/2673-4591/56/1/232.

[18] S. Shashikiran and H. D. Sunitha. "Malaria Cell Identification Using Improved Machine Learning and Modified Deep Learning Architecture." In: *Indonesian Journal of Electrical Engineering and Computer Science* 34.3 (June 2024). Accessed: February 8, 2025, pp. 2078–2086. ISSN: 2502-4752. DOI: 10.11591/ijeecs.v34.i3.pp2078-2086. URL: http://ijeecs.iaescore.com.

[19] John Paul Mueller and Luca Massaron. *Deep Learning For Dummies*. Accessed: February 15, 2025. New Jersey: John Wiley & Sons, 2019, p. 368. ISBN: 9781119543046.

[20] Sahiti Kappagantula. “Artificial Intelligence vs Machine Learning vs Deep Learning.” In: *Medium* (2018). Accessed: February 8, 2025. URL: https://medium.com/edureka/ai-vs-machine-learning-vs-deep-learning-1725e8b30b2e.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Accessed: February 10, 2025. MIT Press, 2016.

[22] R. Shanmugamani and S.M. Moore. *Deep Learning for Computer Vision: Expert Techniques to Train Advanced Neural Networks Using TensorFlow and Keras*. Accessed: February 22, 2025. Packt Publishing, 2018. ISBN: 9781788295628. URL: https://books.google.dz/books?id=dgdOswEACAAJ.

[23] Osval Antonio Montesinos López, Abelardo Montesinos López, and Jose Crossa. “Fundamentals of Artificial Neural Networks and Deep Learning.” In: *Multivariate Statistical Machine Learning Methods for Genomic Prediction*. Accessed: February 22, 2025. Cham: Springer International Publishing, 2022, pp. 379–425. ISBN: 978-3-030-89010-0. DOI: 10.1007/978-3-030-89010-0_10. URL: https://doi.org/10.1007/978-3-030-89010-0_10.

[24] GeeksforGeeks. *What is Forward Propagation in Neural Networks?* Accessed: February 28, 2025. 2023. URL: https://www.geeksforgeeks.org/what-is-forward-propagation-in-neural-networks/.

[25] GeeksforGeeks. *Backpropagation in Neural Network*. Accessed: February 28, 2025. 2023. URL: https://www.geeksforgeeks.org/backpropagation-in-neural-network/.

[26] Saqib Imran et al. “Artistic Style Recognition: Combining Deep and Shallow Neural Networks for Painting Classification.” In: *Mathematics* 11.22 (2023). Accessed: February 25, 2025. ISSN: 2227-7390. DOI: 10.3390/math11224564. URL: https://www.mdpi.com/2227-7390/11/22/4564.

[27] Rukshan Pramoditha. *Overview of a neural network’s learning process*. https://medium.com/data-science-365/overview-of-a-neural-networks-learning-process-61690a502fa. Accessed: February 26, 2025. Feb. 2022.

[28] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Learning representations by back-propagating errors.” In: *Nature* 323.6088 (1986). Accessed: May 2, 2025, pp. 533–536. DOI: 10.1038/323533a0.

[29] Paul J. Werbos. “Generalization of backpropagation with application to a recurrent gas market model.” Accessed: May 3, 2025. Ph.D. thesis. Harvard University, 1988.

[30] Jeffrey L. Elman. “Finding structure in time.” In: *Cognitive Science* 14.2 (1990). Accessed: May 5, 2025, pp. 179–211. DOI: 10.1207/s15516709cog1402_1.

[31] A. K. Tyagi and A. Abraham. *Recurrent Neural Networks: Concepts and Applications*. Accessed: March 2, 2025. CRC Press, 2022. ISBN: 9781003307822. URL: https://books.google.dz/books?id=Wy42zwEACAAJ.

[32] Aj Cheng. *Different Between CNN & RNN [Quote]*. Accessed: March 13, 2025. 2020. URL: https://medium.com/@Aj.Cheng/different-between-cnn-rnn-quote-7c224795db58.

[33] Laith Alzubaidi et al. “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions.” In: *Journal of Big Data* 8.1 (2021). Accessed: March 2, 2025, pp. 1–74.

[34] Sajja Tulasi Krishna and Hemantha Kumar Kalluri. “Deep Learning and Transfer Learning Approaches for Image Classification.” In: *International Journal of Recent Technology and Engineering (IJRTE)* 7.5S4 (2019). Accessed: March 8, 2025, pp. 427–432.

[35] Mehmet Kose and Turgay Yildirim. *The Convolution Operator in a CNN*. https://www.researchgate.net/figure/The-convolution-operator-in-a-CNN_fig3_323927498. Accessed: March 9, 2025. 2018.

[36] Moulay A. Akhloufi et al. *Operations carried out by the convolution and pooling layers. A max-pooling layer picks the most prominent features from the feature map.* https://www.researchgate.net/figure/Operations-carried-out-by-the-convolution-and-pooling-layers-A-max-pooling-layer-picks_fig2_354036236. Accessed: March 9, 2025. 2021.

[37] S. Bhattacharyya et al. *Deep Learning: Research and Applications*. De Gruyter Frontiers in Computational Intelligence. Accessed: March 13, 2025. De Gruyter, 2020. ISBN: 9783110670905. URL: https://books.google.dz/books?id=yEj2DwAAQBAJ.

[38] Jeremy Jordan. *Common architectures in convolutional neural networks*. https://www.jeremyjordan.me/convnet-architectures/. Accessed: March 20, 2025.

[39] F. Hu et al. “Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery.” In: *Remote Sensing* 7.11 (2015). Accessed: March 20, 2025, pp. 14680–14707. DOI: 10.3390/rs71114680.

[40] Guillaume Rousset et al. “Assessment of Deep Learning Techniques for Land Use Land Cover Classification in Southern New Caledonia.” In: *Remote Sensing* 13.12 (2021). Accessed: March 21, 2025, p. 2257. DOI: 10.3390/rs13122257. URL: https://www.researchgate.net/publication/352264347_Assessment_of_Deep_Learning_Techniques_for_Land_Use_Land_Cover_Classification_in_Southern_New_Caledonia.

[41] “Comparison of CNN’s Architecture GoogleNet, AlexNet, VGG-16, Lenet-5, ResNet-50 in Arabic Handwriting Pattern Recognition.” In: *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control* 4.2 (2023). Accessed: March 18, 2025. ISSN: 2503-2259. DOI: 10.22219/kinetik.v8i2.1667. URL: https://kinetik.umm.ac.id/index.php/kinetik/article/view/1667.

[42] Mingxing Tan and Quoc V. Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.” In: *Proceedings of the 36th International Conference on Machine Learning (ICML)*. Accessed: March 23, 2025. PMLR, 2019, pp. 6105–6114. URL: https://proceedings.mlr.press/v97/tan19a.html.

[43] Tashin Ahmed and Noor Hossain Nuri Sabab. “Classification and Understanding of Cloud Structures via Satellite Images with EfficientUNet.” In: *arXiv preprint arXiv:2009.12931* (2021). Accessed: March 22, 2025. URL: https://arxiv.org/abs/2009.12931.

[44] Nico Klingler. *EfficientNet: Optimizing Deep Learning Efficiency*. Accessed: March 29, 2025. Mar. 2023. URL: https://viso.ai/deep-learning/efficientnet/.

[45] Stuart J. Miller et al. “Multi-Modal Classification Using Images and Text.” In: *SMU Data Science Review* 3.3 (2020). Accessed: March 15, 2025. URL: https://scholar.smu.edu/datasciencereview/vol3/iss3/6.

[46] Wenjin Tao et al. “Real-Time Assembly Operation Recognition with Fog Computing and Transfer Learning for Human-Centered Intelligent Manufacturing.” In: *Procedia Manufacturing* 48 (Jan. 2020). Accessed: March 13, 2025, pp. 926–931. DOI: 10.1016/j.promfg.2020.05.131.

[47] Barath Narayanan, Redha Ali, and Russell C. Hardie. “Performance Analysis of Machine Learning and Deep Learning Architectures for Malaria Detection on Cell Images.” In: *Electrical and Computer Engineering Faculty Publications, University of Dayton* (2019). Accessed: April 26, 2025, pp. 1–9.

[48] Muhammad Umer et al. “A Novel Stacked CNN for Malarial Parasite Detection in Thin Blood Smear Images.” In: *IEEE Access* 8 (2020). Accessed: April 18, 2025, pp. 97300–97312. DOI: 10.1109/ACCESS.2020.2994810.

[49] Varun Magotra and Mukesh Kumar Rohil. “Malaria Diagnosis Using a Lightweight Deep Convolutional Neural Network.” In: *Journal of Healthcare Engineering* 2022 (2022). Accessed: April 21, 2025, pp. 1–10. DOI: 10.1155/2022/4040292.

[50] Sorio Boit and Rajvardhan Patil. “An Efficient Deep Learning Approach for Malaria Parasite Detection in Microscopic Images.” In: *Diagnostics* 14.4 (2024). Accessed: April 28, 2025, pp. 1–18. DOI: https://doi.org/10.3390/diagnostics14232738. URL: https://www.mdpi.com/2075-4418/14/23/2738.

[51] Lister Hill National Center for Biomedical Communications. *Malaria Cell Image Dataset*. https://lhncbc.nlm.nih.gov/LHC-research/LHC-projects/image-processing/malaria-project.html. Accessed: May 7, 2025. 2016.

[52] Emma Oye, Rhoda Oyin, and Richmore Zion. “Performance Evaluation Metrics.” In: *ResearchGate* (Dec. 2024). Accessed: May 7, 2025.