

Article

A Novel Ensemble Weight-Assisted Yolov5-Based Deep Learning Technique for the Localization and Detection of Malaria Parasites

Sumit Paul ¹, Salil Batra ¹, Khalid Mohiuddin ² , Mohamed Nadhmi Miladi ², Divya Anand ^{1,3,*} 
and Osman A. Nasr ² 

¹ Department of Computer Science and Engineering, Lovely Professional University, Phagwara 144411, India

² Department of Management Information Systems, College of Business, King Khalid University, KSA, Abha 62529, Saudi Arabia

³ Higher Polytechnic School, Universidad Europea del Atlántico, C/Isabel Torres 21, 39011 Santander, Spain

* Correspondence: divyaanand.y@gmail.com

Abstract: The traditional way of diagnosing malaria takes time, as physicians have to check about 5000 cells to produce the final report. The accuracy of the final report also depends on the physician's expertise. In the event of a malaria epidemic, a shortage of qualified physicians can become a problem. In the manual method, the parasites are identified by visual identification; this technique can be automated with the use of new algorithms. There are numerous publicly available image datasets containing the intricate structure of parasites, and deep learning algorithms can recognize these complicated patterns in the images. This study aims to identify and localize malaria parasites in the photograph of blood cells using the YOLOv5 model. In this research, a publicly available malaria trophozoite dataset is utilized which contains 1182 data samples. YOLOv5, with the novel technique of weight ensemble and traditional transfer learning, is trained using this dataset, and the results were compared with the other object detection models—for instance, Faster RCNN, SSD net, and the hybrid model. It was observed that YOLOv5 with the ensemble weights yields better results in terms of precision, recall, and mAP values: 0.76, 0.78, and 0.79, respectively. The mAP score closer to 1 signifies a higher confidence in localizing the parasites. This study is the first implementation of ensemble YOLOv5 in the malaria parasite detection field. The proposed ensemble model can detect the presence of malaria parasites and localize them with bounding boxes better than previously used models.

Keywords: deep learning; ensemble learning; malaria detection; object detection



Citation: Paul, S.; Batra, S.; Mohiuddin, K.; Miladi, M.N.; Anand, D.; A. Nasr, O. A Novel Ensemble Weight-Assisted Yolov5-Based Deep Learning Technique for the Localization and Detection of Malaria Parasites. *Electronics* **2022**, *11*, 3999. <https://doi.org/10.3390/electronics11233999>

Academic Editor: Chunjie Zhang

Received: 18 October 2022

Accepted: 24 November 2022

Published: 2 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Malaria is a highly contagious and potent disease spread by the bite of an infected female mosquito. The malaria report from 2017 states that over 219 million people got infected with malaria, and 435,000 fatalities were reported worldwide. In 2019, there were an estimated 409,000 deaths from malaria [1]. Plasmodium parasites are responsible for malaria in humans [2]. *P. Falciparum*, *P. Vivax*, *P. Ovale*, and *P. Malaria* are the four Plasmodium parasites that can cause malaria in humans. *P. Vivax*, *P. Ovale*, and *P. Malaria* are less harmful variants, while *P. Falciparum* is the most dangerous variant. A patient suffering from *P. Falciparum* has a low chance of survival [3]. A patient suffering from malaria can be cured by treatment, but accurate results should be available in the early stage of malaria; a delay in the results can lead to major health consequences or even death. Currently, the benchmark for malaria detection using manual methods is the microscopic inspection of thick and thin blood smears [4]. Another method of detecting malaria is the RDT (Rapid Diagnosis Test), in which the blood samples of humans are tested against an antigen. This method is not very accurate, it tends to give false-positive results, and the

materials used for the manufacturing of the RDT kit are also harmful to the environment [5]. The accuracy of microscope-based diagnostic results mainly depends upon the slide-reading abilities of the physicians/doctors. The lack of qualified staff is hampering the effectiveness of the results of the microscope-based diagnostics in malaria-infested areas, especially in rural areas. Poor decision making will occur during the malaria eradication campaign when the results are not proper. The visual difference between a thin and thick blood smear is represented in Figure 1.

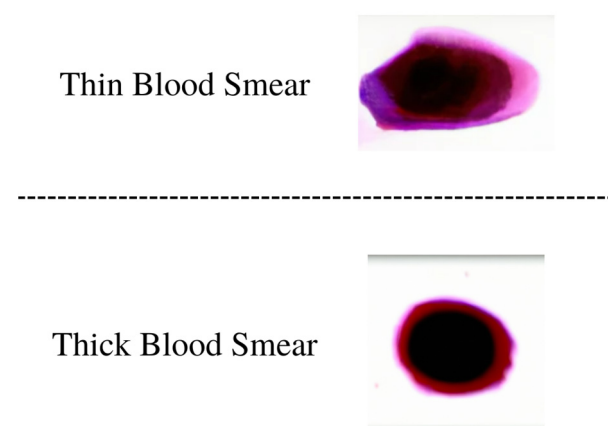


Figure 1. The glass slide with both thick and thin blood smears [6].

To overcome the constraints of manual diagnostic tests, a computer-assisted diagnostic system that is both rapid and accurate is necessary. Recent improvements in computer diagnostics, with the introduction of deep learning methods, have yielded promising results in various disciplines, and clinicians can use them to detect the malaria parasite. Recently, there has been an uprise in the field of deep learning because of the availability of data as well as computational power. If trained well, a deep learning model can differentiate between objects within a couple of seconds with the accuracy of a human expert. It is possible to use the deep learning models to detect the presence of malaria parasites in human blood and eliminate the time required to make the decision. These techniques will not replace human experts, but they can provide a second opinion on the case.

Object detection models can detect and mark boundary boxes to the objects in the image and videos in a couple of seconds. Several object detection models are available, including YOLO (You Only Look Once), FRCNN (Faster Region-based Convolutional Network), and custom-made models. The region proposal method, which predicts object bounds and objectness scores at each position, is used in the faster R-CNN. It saves a lot of computation time compared to the Selective Search algorithm. It works by extracting the fixed-length feature vector from each region with the help of the ROI pooling layer. One of the disadvantages of the Faster R-CNN or FRCNN is that all small batches are collected from a single image. There is a high probability that the samples are correlated, as they have similar features, which could make the network converge at a very slow rate [7].

YOLO uses an end-to-end neural network that predicts the class probability and boundary boxes all in a single inference. YOLO achieves better results as compared to other real-time object detection models. The FRCNN model uses a region proposal network; when a model uses this network, it must perform multiple iterations for a single image, but YOLO performs all predictions at a single iteration [8]. With each generation YOLO model getting much better, the advantage of YOLOv5 is that it is 88% smaller and 180% faster than the last-gen YOLOv4, which allows it to be deployed on handheld devices. One of the drawbacks of YOLOv5 is that it is still under development [9]. The purpose of this work is to look at the usefulness of object detection models for localizing malaria parasites in blood smear images obtained with a mobile phone camera. Researchers are not

investigating modern and very advanced deep learning-based object detection models for malaria parasite localization despite their success.

Contributions of This Study

1. Upon studying the available dataset, it was found that the dataset was not compatible with the YOLOv5 model, since YOLOv5 requires different formatting than the available datasets formatting. As a result, the pre-processing technique is utilized on the provided dataset to ensure that it complies with the YOLOv5 model.
2. Furthermore, variations in the data are integrated into the dataset by augmentation techniques—for instance, random rotation and random flipping of the image to avoid the overfitting of the model, because the model can easily memorize the dataset if the sample size is very small.
3. This is a novel technique that has not been used previously to ensemble YOLOv5 weights for the detection of malaria parasites in a blood smear.
4. The proposed ensemble technique is also tested against the transfer learning technique of YOLOv5.
5. After the completion of the study, comparisons are made with other base models based on the precision, recall, and mAP value.
6. The paper is divided as follows: Section 2 outlines the existing techniques for malaria parasite identification and localization. The implementation stages are described in Section 3. The results are discussed in Section 4, and the paper is concluded in Section 5.

2. Related Work

The object detection models work by taking a dataset that consists of the class label and coordinates of the boundary box, and it uses the internal layers and adjusts the weights to train the model on each epoch. It tries to reduce the generalization error based on the optimization function and loss function. The deep learning models tend to overfit if the patterns of training samples are similar [10].

The object detection model is the upgraded version of the image classification, as it performs image classification and localization of the classified objects. The evaluation metrics of the object detection models are also different compared to image classification. The image classification model is evaluated by taking the meaning of the classification error on all predicted labels. The object detection model is evaluated by precision and recall on the highly matching boundary box of the detected object. Mean Average Precision, or mAP, plays an important role in the evaluation of the object detection models. It compares the boundary boxes of the original dataset and the predicted one and provides a score. The model is considered much more accurate if the score is close to 1 and is considered worst if it is close to 0. The flow chart in Figure 2 depicts the traditional flow of the malaria localization of malaria with an object detection model.

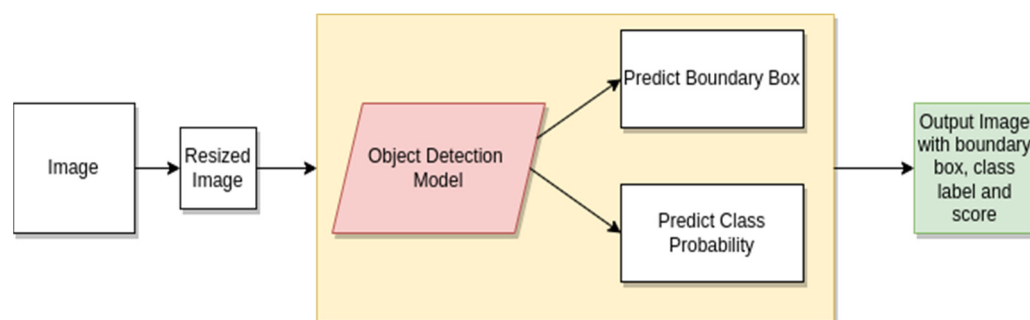


Figure 2. Traditional Flow Chart for Malaria Localization.

Even with the availability of a state-of-the-art object detection model, the models have not yet been applied in the domain of detecting and localizing malaria parasites.

There are two variants of blood smears utilized in manual or traditional malaria detection, and these variants are thick and thin blood smears. The authors in [11,12] have used the object detection method to localize the presence of parasites in thin blood smears. A thin blood smear is made of a single layer of blood. The parasites are visible in the thin blood smear, and the dataset with a thin blood smear is used to classify the presence of parasites, whereas the thick blood smear is made of several layers which can be used to identify the type of parasite as well. In recent studies, the authors had explored different ways of utilizing the YOLOv5 model for the detection of malaria, Zedda et al. [13] showcased the use of YOLOv5 and DarkNet-53 and achieved an accuracy of 95.2%. A. Shal and her colleagues [14] compared various object detection models including YOLOv4 and YOLOv5; they choose these algorithms because of two main factors: they are the most widely used and highly efficient.

Samson C. and his colleague's [15] study provides an example of an inexpensive alternative for malaria diagnostics; their technique is aimed at providing diagnostics at a low cost, which is not currently available. Their method has the potential to be utilized as a decision-making tool for diagnostic consistency in order to reduce the dependency on manual tests, reduce operator fatigue, and increase performance rates. To avoid bias, De Rong et al. [16] used synchronization methodologies to manage the period categorization of RBCs and fine-tuned labeling methodology during training. For microscopic and phenotypic testing, parasites planted on RBCs were smeared on glass slides, treated with methanol, and coated with a diluted Giemsa solution. They then developed a bespoke Deep Learning model to detect the malaria pathogen and extract its presence region. The finished model could generate segmentation masks.

Rose N. et al. [17] suggested a technique for detecting malaria parasitemia that quickly and correctly locates and predicts parasites and white blood cells. The authors have explored the use of Faster RCNN for localization tasks. They collected each image with other information—for instance, a microscope slide number, a stage micrometer grid reading, the smartphone zoom level used for the photo collection, the objective microscope size, and reagent staining. John A. [18] outlined how he used annotated photos to create a deep learning model, which he then used to test images for disease detection. The downsampling factor and patch size were determined by the type of pathogen to be identified in each case. Positive patches were taken in the annotation and centered on bounding boxes. Negative patches were randomly selected from random locations in each image that did not intersect with any annotated bounding boxes.

Koirala et al. [19] proposed an approach of consistent labeling for malaria detection on thick blood smear images taken from a microscope. The authors have utilized the YOLOv4 model to detect the parasite in the blood smear. Their study contains two different datasets, one is used for training and the other is used for validation. They converted the images with the Roboflow tool to make them compatible with the YOLO model. They have observed that the boundary boxes detected by the trained model have issues; some of them are bigger than expected and some only contain part of the pathogen. The improvement in the model was noticeable after following the method of centering the ground truth, bounding the boxes around parasites.

Ruskin R. et al. [20] utilized two different approaches: Level 1 and Level 2, respectively. In Level 1, they detect the infected cell by passing the image to the RPN or Region Proposal Network. The result of this network is then reshaped to the pooling layer of the RoI, which is used for the classification of the image inside the proposed region from the RPN, and it predicts the values for the boundary boxes. The feature maps are generated by passing the images to ResNet 50, and the result of this is fed to the FRCNN two-stage approach model.

The main problem in the field of object detection is the use of GPU power; it is always challenging to figure out ways to reduce the parameters of the model so that it can be trained efficiently on the hardware. Dong et al. [21] have utilized pathogen images from the University of Alabama. There were a total of 1,000,000 red blood cells, and only 0.2% were infected. Multiple datasets were generated by cropping the image with some morphological

operations. After encoding the images, the datasets were passed to the LSTM model for making the classification. The authors have successfully demonstrated that it is possible to use the compressed dataset in the input layer and skip the decompression part to save the computation power. For classification tasks, the computational power of the CNN models can be significantly reduced with the dilated CNN [22].

The brief summary of the related works is mentioned in Table 1. The study compares the proposed method with the works of Samson C. [15], Rose. N. [17], and John A. [18] because they worked on the same dataset that was used in this work and they have used widely used object detection models—YOLO, Hybrid, and FRCNN—to detect and localize the parasites in the blood smear.

Table 1. Summary of related works.

Author	Use of Prebuilt Object Detection Model	Limitations
Samson C. et al. [15]	✓	With small images, the mAP value of the YOLO model is very low.
Dr. Rong et al. [16]	✗	The number of samples in the dataset is very small.
Rose N. et al. [17]	✓	Poor annotations lead to high FP and FN results in testing.
John A. et al. [18]	✗	The CNN model was not able to extract minute details from the image, which leads to low accuracy.
Koirala et al. [19]	✓	The latest YOLO model is not explored.
Ruskin R. et al. [20]	✓	Variations related to automatic field testing are not considered

3. Methodology

3.1. Problem Statement

Malaria is a disease that can affect people of every age, and it can be cured, but for that, the patient needs to be diagnosed as early as possible. The traditional method of malaria diagnosis in a person can take days, and the accuracy of the result also depends upon the expertise of the physician or lab expert in finding and labeling malaria. One of the hard problems in the diagnosis is manually counting the infected cell. This method is used to identify the number of infected cells in the blood sample, and performing it manually is very time-consuming. With the use of advanced object detection algorithms, it is possible to automate the detection and localization of the parasites in a significantly less amount of time, and it also counts the number of detected cells. The previous works in this field of malaria detection and localization use traditional models, and those models provide less accuracy and are computationally expensive. Despite the success of the new object detection models on various datasets, these models have not been tested extensively for detecting malaria. Object detection models similar to YOLOv5 are faster and smaller compared to their previous generations, and they predict in only a single iteration over the image [23]. Therefore, there is a need to explore these models for the detection and localization of the model.

3.2. Dataset

For this study, widely used plasmodium parasite images were utilized, and this dataset is available in a public repository [18]. The dataset was collected with the use of a microscope and mobile device. The mobile device was placed on the eyepiece of the microscope using custom build equipment. There are 1182 RGB (Red Green Blue) microscopic images for a stained thick blood smear. The resolution of all images is 750×750 pixels. The annotations in the image are carried out by the lab experts. The dataset contains the *P. Falciparum* parasite in 948 images, and 234 healthy images do not have the parasite; the sample images of the dataset are represented in Figure 3. One of the problems with the

dataset is that it contains only the *P. Falciparum* parasite, which would make the proposed model detect only this type of parasite. The dataset is divided into training and validation in the test set. The evaluation of the model is performed on the image level because there is no information regarding the patient level in the dataset.

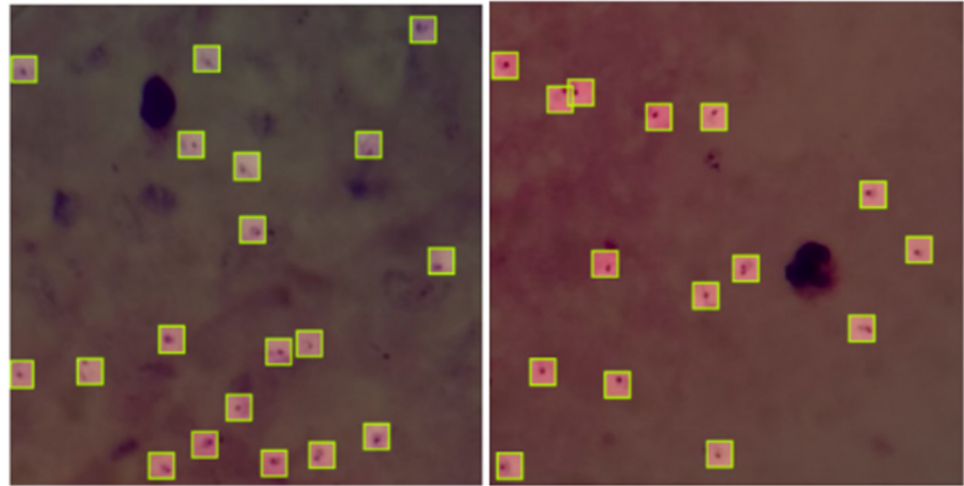


Figure 3. *P. falciparum* annotated dataset with highlighted malaria parasites and annotations.

3.3. Data Pre-Processing and Annotation

The data augmentation technique helps to create a robust model because it creates more variation in the dataset. During the training phase, the model does not memorize the data; hence, overfitting of the model is avoided [24]. In this research work, data augmentation techniques are utilized, which include resizing into 416×416 pixels and random horizontal and vertical flips. The flipping of the image is carried out to generate more variations in the data in order to make sure that the proposed model is not memorizing the input. The image can be flipped in two ways: horizontal and vertical. In order to flip the image, the following formulas are used:

- Vertical Flip: To flip an image vertically, the x coordinates of the image pixel need to be changed, and this can be accomplished using:

$$x(\text{new}) = (\text{width} - x(\text{old}) - 1) \quad (1)$$

- Horizontal Flip: While flipping an image horizontally, the y coordinates of the image pixel need to be changed, and this is implemented using:

$$y(\text{new}) = (\text{height} - y(\text{old}) + 1) \quad (2)$$

- During the image flip, the position of the bounding boxes also changes, and for implementation, the following formulas were used [25]:
- Bounding Box Vertical Flip: Vertical Flip forces the x coordinates to move to different locations. For bounding boxes, the new coordinate can be calculated using:

$$x(\text{new}) = \left(\frac{\text{width}}{2} \right) + \left(\frac{\text{width}}{2} - x(\text{old}) \right) \quad (3)$$

- Bounding Box Horizontal Flip: The Horizontal Flip deals with the y coordinate, and to manipulate the bounding boxes y coordinate, the following formula is utilized:

$$y(\text{new}) = \left(\frac{\text{height}}{2} \right) + \left(\frac{\text{height}}{2} - y(\text{old}) \right) \quad (4)$$

YOLOv5 uses a special type of dataset that includes information on the annotation in a specific way in the YAML file. Each row of the dataset should contain “class, x_center,

y_center, width, height” [26]. The snapshot of the dataset is presented in Table 2. The pre-processing and annotation of the dataset are carried out using the Roboflow tool [27], the pre-processing and augmentation information is mentioned in Table 3. Roboflow is utilized because it provides the direct download link for processed data, which is useful in case a study needs to be conducted on a remote environment of Google Colab or Kaggle, as there is no need to download and upload the data to the working runtime environment. To test the trained model, the unseen data need to be transferred to the model.

Table 2. Sample YOLOv5 annotation file contains the class, x_center, y_center, width, and height of the annotations.

Class	X_Center	Y_Center	Width	Height
0	0.7992788462	0.7896634615	0.05288461538	0.05288461538
0	0.6201923077	0.7524038462	0.05288461538	0.05288461538
0	0.4699519231	0.7331730769	0.05288461538	0.05288461538
0	0.7896634615	0.9302884615	0.05288461538	0.05288461538
0	0.9338942308	0.4086538462	0.05288461538	0.05288461538
0	0.2776442308	0.4927884615	0.05288461538	0.05288461538
0	0.9459134615	0.3137019231	0.05288461538	0.05288461538
0	0.9098557692	0.5637019231	0.05288461538	0.05288461538
0	0.5252403846	0.3822115385	0.05288461538	0.05288461538
0	0.1382211538	0.4543269231	0.05288461538	0.05288461538

Table 3. Data pre-processing information.

Type	Information
Pre-Processing	Auto Orient: Applied Resize: Stretch and Crop to 416×416
Augmentations	Outputs per training example: 3 Flip: Horizontal, Vertical

Therefore, the splitting of the dataset into train, test, and valid sets is carried out beforehand so that it can be ensured that no test data are being sent to the model during the training process.

The proposed Ensemble model is trained on the Kaggle platform. Kaggle offers free cloud-based Notebooks that may be used without any special hardware. It includes GPU acceleration (NVidia K80) and compatibility for all currently available Python packages.

3.4. Proposed Ensemble Technique

3.4.1. YOLOv5 Architecture

YOLOv5 extracts target information from an input image using CSPDarknet as a feature extraction network. In the detection phase, the input tensor gets divided into a number of grids. If the target’s center point falls within one of the grids, that grid is in charge of detecting the object. For each of the grids, there will be n anchors. Each frame has a $5+CS$ number of predicted values. The first five datapoints are used to regress the anchor’s center point location and the size of the anchor frame and determine if there is a target. CS is the number of classes [28]. Three main parts of the YOLOv5 model’s architecture are discussed below, and the graphical representation of the model is given in Figure 4.

- **Model Backbone:** Its end goal is to extract the key features from the provided input image. The Cross Stage Partial Network is used by YOLOv5. It outperforms several deep networks [29].

- **Model Neck:** It is used to detect and identify similar objects present in the image with different variations in size and scale. It constructs feature pyramids, which leads to a better performance in test datasets.
- **Model Head:** The final detection is carried out by the Model Head. The class probability, score, and boundary boxes are generated in this component.

Activation Function

The activation function is vital in object detection and deep learning models because it determines whether or not the neuron should fire. The activation functions Leaky ReLU [30] and Sigmoid [31] are used in YOLOv5 to balance recall and precision values. In the actual model, the Leaky ReLU is employed in the network's middle layers, and the Sigmoid is used in the prediction layer, which is the last layer of the model.

Loss Function

The loss function used by the YOLOv5 model is the BCE or Binary Cross-Entropy function. The YOLOv5 loss combines three different values: the confidence score, the probability of the class, and the boundary box. It also allows for YOLOv5 to be used in multi-label applications. The loss function can be represented by the formula:

$$loss = l_b + l_c + l_o \quad (5)$$

Here, b , c , and o are the bounding box regression function, classification loss function, and confidence loss function, respectively.

The boundary box loss function is defined as:

$$l_b = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^o b_j (2 - w_i \times h_i) \left[(x_i - \hat{x}_i^j)^2 + (y_i - \hat{y}_i^j)^2 + (w_i - \hat{w}_i^j)^2 + (h_i - \hat{h}_i^j)^2 \right] \quad (6)$$

The classification loss of YOLOv5 is expressed as:

$$l_c = \lambda_{class} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^o \sum_{C \in \text{classes}} p_i(C) \log(\hat{p}_i(C)) \quad (7)$$

Confidence loss in the model is expressed as:

$$l_o = \lambda_{no-o} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^{no-o} (C_i - \hat{C}_i)^2 + \lambda_o \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^o (C_i - \hat{C}_i)^2 \quad (8)$$

The above formula represents λ_{coord} as the coefficient of the position vector, λ_{class} as the category loss coefficient, \hat{x} , \hat{y} as the true central target coordinates, and \hat{h} , \hat{w} as the targets' height and width.

$I_{i,j}^o$ will be 1 if the target is inside (i, j) the anchor box; otherwise, the value is going to be 0. $p_i(C)$ represents the category probability of the class, and $\hat{p}_i(C)$ represents the categories' true value. The total number of categories CS is equal to the length of these two.

Optimization Function

Stochastic Gradient Descent (SGD) [32] or Adam [33] optimizers can be employed in YOLOv5 implementation. The default optimization function, however, is SGD for this study Adam optimizer is used. The parameter “—adam” has to be added in the command line argument during the training process.

Training YOLOv5 on the custom dataset is much easier compared to YOLO's previous versions, as the model automatically fits the anchors of the dataset without needing to

find it manually. The goals of YOLOv5 are the ease of use, exportability, speed, mAP, and memory requirement.

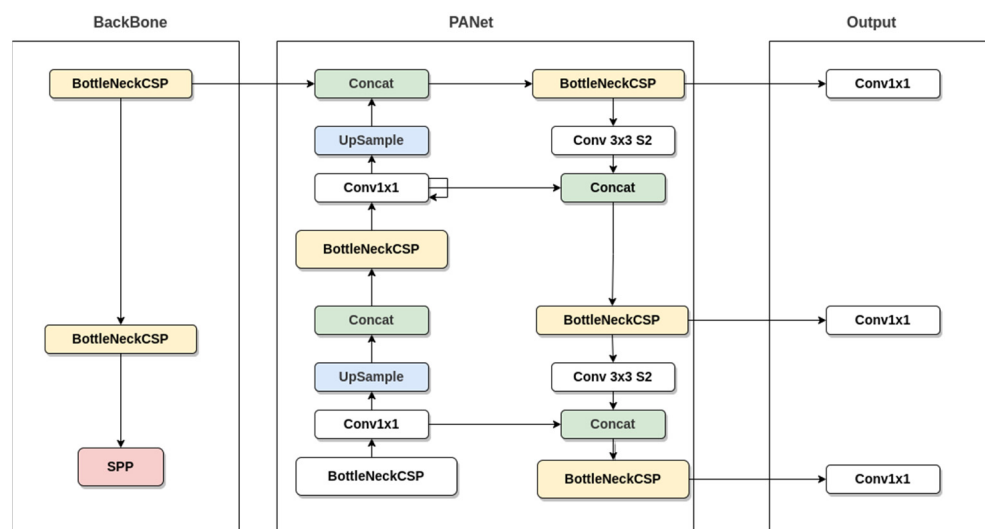


Figure 4. YOLOv5 architecture [29].

3.4.2. Parameter Changes in the YOLOv5 Model

Before training the model, some changes are required to obtain the desired results.

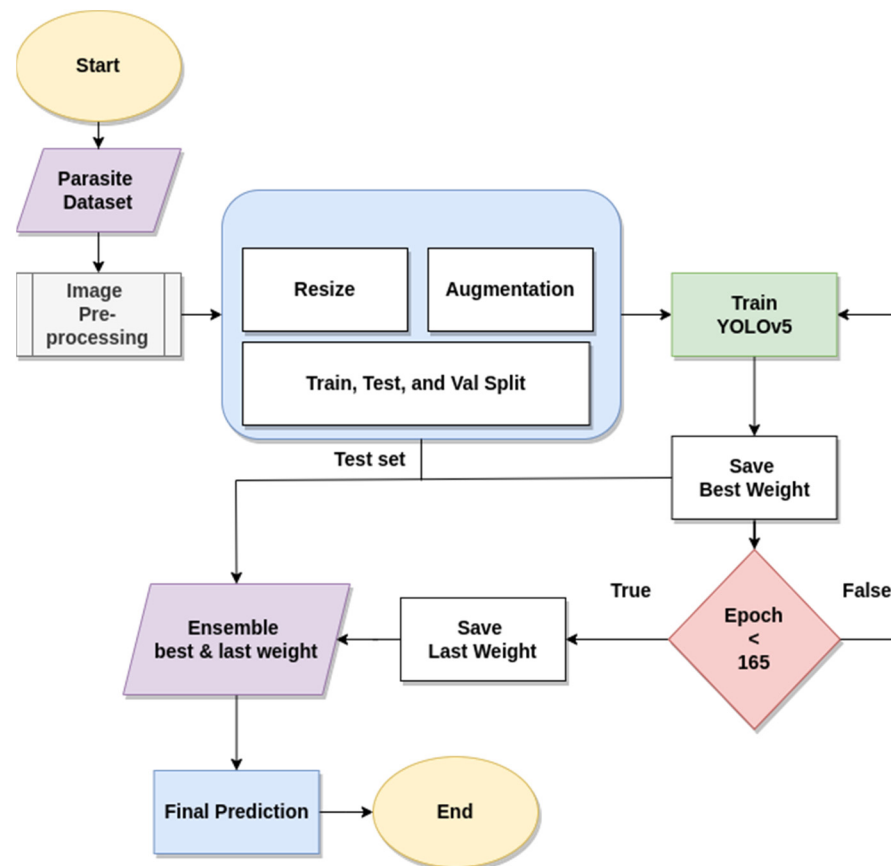
YOLOv5 requires a configuration file that contains the model layers' information. In this YAML configuration file, the first thing that requires a change is the `nc` variable, which is nothing but the number of classes in the dataset. In the used dataset, there is only one class; therefore, the value of `nc` will be 1.

In the same configuration file, different values have been tested for the model depth and width multiple. It is found that a depth multiple of 1.33 and a width multiple of 1.25 provide better results.

It is recommended to use the default bottleneck layers to obtain better results; the same backbone is used in this study. The proposed technique requires two weights for making the final prediction. The goal of employing ensemble models is to reduce forecast generalization errors. In the ensemble learning approach, the final models' prediction error decreased when model/weights were diverse and independent. The ensemble model is considered one single model because it works with multiple models but makes predictions as one unit. While predicting the unseen data, the model uses the wisdom of the multitude [34].

After the training process, two weights are obtained: the best weight and the last weight from YOLOv5. In this research work, it was observed that the best weight was missing some predictions, as it was providing some false-negative results, but it gave the highest accuracy overall, whereas the final/last weight was more generalized and maintains the results ratio. The idea behind the ensembling of the weights is that it finds both weights overlapping boxes and scores and shows them in the result; it also finds the missing predictions of one model, which increases the overall accuracy of the model.

Figure 5 provides brief information about the overall model implementation that is being used in this study. Image preprocessing is carried out with the help of resizing the image and dividing the overall dataset into the train, test, and validation sets. Random flipping of the image is performed to create new variations of the data, and this process is described in Figure 6. The test set is used only after the training of the model is completed. The represented [Save Best Weight] process in Figure 5 overwrites the existing best weight whenever a better weight is found, which provides a higher confidence than the previous best.pt weight file. The augmentation process flow chart is represented in Figure 6 and the algorithm for entire work is mentioned in Algorithm 1.

Algorithm 1: Localization using YOLOv5**Input:** Plasmodium parasite images.**Output:** Image with boundary boxes around the detected parasites.**Start****Get Dataset****Extract Zip File****Preprocess Image** **Change width and height of the image to 416×416** **Flip random image horizontally:** $y(\text{new}) = (\text{height} - y(\text{old}) + 1)$ **Flip random image vertically:** $x(\text{new}) = (\text{width} - x(\text{old}) - 1)$ **Download YOLOv5 model****Changes in yolo.yml** **nc: 1 #set number of classes to 1** **depth_multiple: 1.33** **width_multiple: 1.25****Changes in data.yml** **Set path of train_directory** **Set path of validation_directory****Set epoch to 0****While epoch < 165** **Train YOLOv5 #with new configurations** **Save Weights****End While loop****Ensemble best accuracy weight and last weight****Pass test set to the val.py****End****Figure 5.** Flow chart for Proposed Methodology.

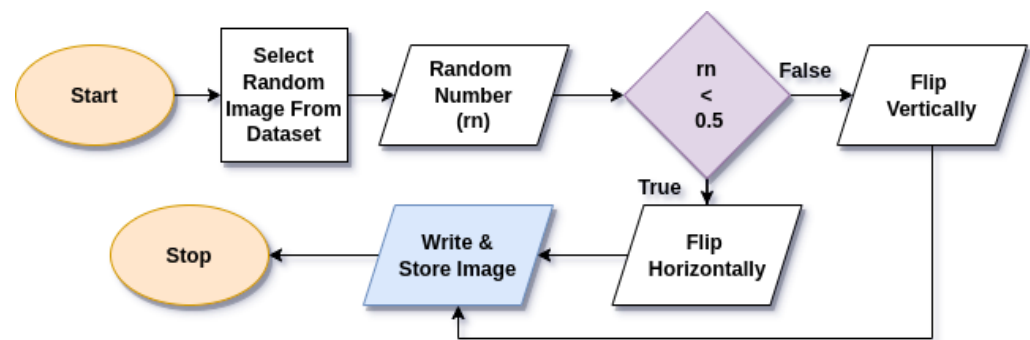


Figure 6. Flow chart for Image Augmentation.

4. Results and Discussions

4.1. Performance Metrics

To examine the performance of the model, $mAP@IoU = 0.5$, Precision, and Recall were utilized. IoU stands for Intersection over Union, and it is a basic metric used for comparing object detection models/systems [35]. In IoU, the relation between the test set or ground truth with the predicted result generated by the model is evaluated. It computes the intersection sets of these bounding boxes by dividing them by the union set. True Positive (TP) is obtained when the defined threshold value is smaller than the IoU value, and False Positive (FP) is obtained when the threshold is greater than the IoU value. Using the TP and FP values, Precision, Recall, and mAP are calculated using Equations (9)–(11) [36]. For calculating the mAP 0.5, the threshold value is taken as 0.5.

- Precision:

$$\frac{TP}{TP + FP} \quad (9)$$

- Recall:

$$\frac{TP}{TP + FN} \quad (10)$$

- mAP:

$$\frac{1}{N} \sum_{i=1}^N AP_i \quad (11)$$

In mAP, N is the number of queries and AP is the Average Precision.

The box loss, cls loss, precision, mAP, and recall obtained for the training and validation during the training phase are represented in Figure 7. After training the model over 160+ iterations, the test set is used to test the performance of the model. The dataset is split during pre-processing; therefore, there is no chance for the model to know about the test set. In the latest YOLOv5 file, there is no separate test.py file to test the model. Therefore, a YAML file was created that targets the location of the test set, and it is passed into the val.py file using the following arguments: “--data testdata.yaml”. Parameters for the transfer learning model are represented in Table 4.

Table 4. Transfer learning parameters used for training the YOLOv5 model.

Parameters	Values
Frozen Layers	9
Image shape	416
Batch	16
Epochs	30
Weights	yolov5x.pt
Name	yolov5x_tuned

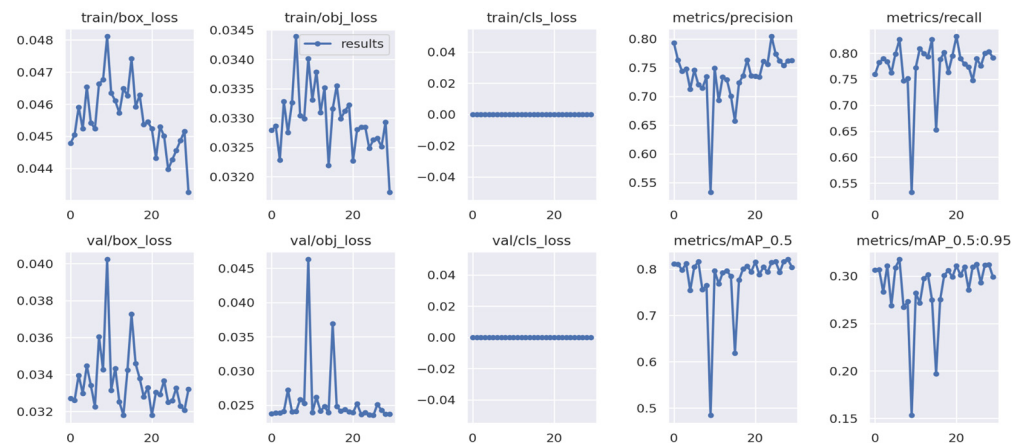


Figure 7. Results of YOLOv5 after 165 epochs.

The comparison of the performance of the last weight, best weight, transfer learning, and ensemble learning is carried out individually, and it was observed that ensemble learning is providing better results in terms of Precision, Recall, and mAP. After obtaining the final results, they were compared with previous works, and it was observed that the obtained results from the proposed model are better than the results of the previously used models. To test the ensemble technique performance, a separate YOLOv5 was trained using transfer learning by freezing the layers of the model and training them using the custom dataset, and the obtained results were compared with the ensemble model. Transfer learning is the best approach for training the model faster because it trains only the portion of the model; this speeds up the training process, but, in our test, it did not provide better results. We have also trained the LSTM [37], Bi LSTM [38], and GRU [39] models on the same dataset and used their results for comparison in Table 5, the graphical comparison of Precision, Recall and mAP@0.5 is represented in Figure 8. In this study, it was observed that the GRU model takes less time to train as compared to LSTM and Bi LSTM, but due to the less complex structure and the fewer gates of GRU, it does not perform well on the image dataset.

Table 5. Comparison of different methods based on precision, recall, and mAP values with the proposed methodology.

Methods	Precision	Recall	mAP@0.5	Shape
SW + CNN [18]	-	-	0.685	50 × 50
Modified YOLO [15]	-	-	0.76	224 × 224
Faster RCNN [17]	0.67	0.80	0.55	512 × 512
SSD Net [17]	0.76	0.50	0.62	512 × 512
Transfer Learning in YOLOv5	0.67	0.71	0.67	416 × 416
ResNet 50 + FRCNN (Trophozoite) [20]	0.73	0.85	0.74	Non-Consistent
LSTM [37]	0.65	0.68	0.71	416 × 416
Bi LSTM [38]	0.71	0.73	0.75	416 × 416
GRU [39]	0.67	0.70	0.69	416 × 416
Proposed Ensemble YOLOv5 Weights	0.76	0.78	0.79	416 × 416

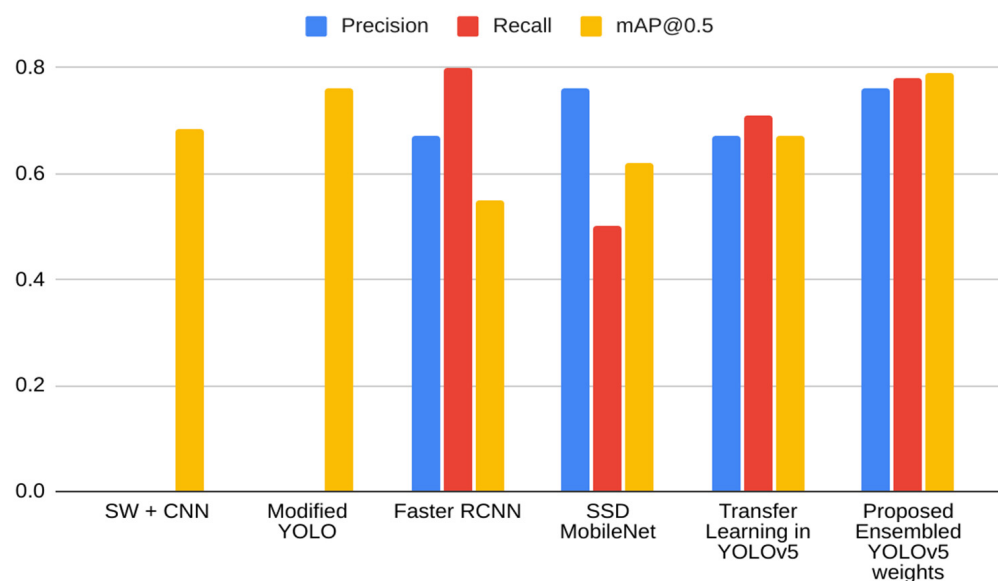


Figure 8. Result comparison of different object detection models.

The ground truth image in Figure 9 is the actual image with the annotation in the test set, and the predicted result image is generated by the model when the test image is passed without annotations. The actual result of the image has 11 infected cells, and the proposed model was able to detect 10 of them successfully and provided one false-positive result.

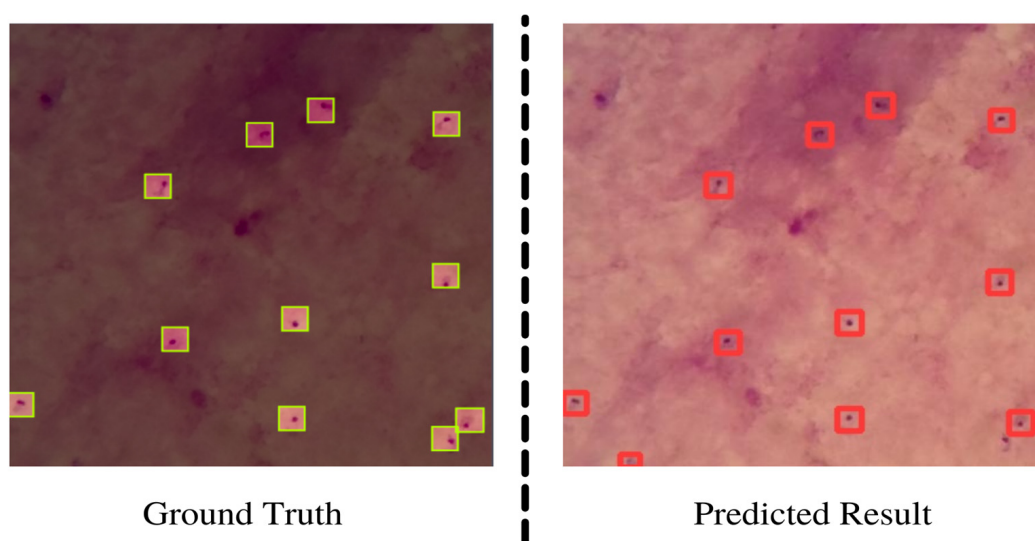


Figure 9. Comparison of Ground Truth and Predicted Result. The left image is taken from the dataset, and the right image is a prediction generated by the trained model. Green colored boxes are the labelled data and red color boxes represent the predicted data from final model.

4.2. Discussion

Even with the success of object detection models in the healthcare industry, the problem of detecting small particles similar to parasites in a blood cell with high accuracy is still not solved. YOLOv5 was named one of the most powerful object detection algorithms of the present time by Yan et al. [40]. In this study, it was found that it outperforms other object detection models—for example, Faster RCNN and SSD (Single Shot Detector) net and customized CNN-based algorithms. In this study, two variations of the YOLOv5 model were successfully implemented and tested—one with transfer learning and the other by ensembling the weights. It was observed that by using the ensemble technique,

better results were obtained, as averaging of the output produces a better prediction. The confidence of the model in detection can be further improved if a better dataset with a lot more variation in the data is available.

4.2.1. Strengths

The proposed model can detect and localize the parasites in the publicly available dataset with an mAP value of 0.79, which is higher compared to the other base object detection models mentioned in the previous works by the authors. The ensemble technique not only increased the overall performance of the model but also reduced some of the false-positive predictions from the final result that were being generated by the use of the single model. Some of the object detection models—for instance, FRCNN and SSD—struggle to find the small objects—for example, points and dots in the image. In the proposed study, it was observed that by changing some of the parameters of YOLOv5, it was able to identify and localize the small objects that were parasites in the dataset. The final model can work as a second opinion for physicians for the counting of the infected cells because, during the localization, the model is capable of keeping the count of the identified cells. If the foreign elements are present in a significant amount in the image or have similar visible patterns as the dataset, then this model can make false-positives. Hence, in future works, the model can be tested and optimized against the blood smear dataset of patients with other blood-related infections.

4.2.2. Limitations

The YOLOv5 model with an ensemble technique can help physicians to count and detect the infected cells in less time, but this model is also bound by the dataset and required processing power. Every deep learning model is bound by the example or training set provided to it; this research is also bound by the variations of the images and the annotations provided in the dataset. The dataset is small in size and does not contain detailed information that is required for the feature extraction process of the deep learning models in order to identify the objects with high confidence and adjust internal weights to make a robust model. Similar to other object detection models, this model is also highly dependent on python and other libraries that are required to make calculations and train the model. YOLOv5 runs on the GPU or Graphical Processing Unit, which is costly, and it can increase the cost requirement of the physical device that is required to run the model. The model can make false predictions if a significant number of foreign elements are present in the image or have similar visible patterns as the dataset.

5. Conclusions

The ensemble weight technique with the YOLOv5 model was successfully implemented and tested in this research work to detect and localize malaria parasites' presence in the blood smear images. This is a novel study, as there has been no previous research using ensemble YOLOv5 weights in the domain of malaria detection and localization. In this study, ensemble and transfer learning techniques were utilized to perform the task. In the future, other methods/techniques can also be explored. This study showcases the use of width and depth multipliers in the model training. By using the compatible values, the model can extract more complex features from the dataset. Furthermore, this work is bound by the limitation of the dataset, as the dataset includes only a single type of parasite, but in the real world, a patient can be affected by a different kind of parasite which will make the model less effective. In the future, the availability of a better dataset can help to build a better and more robust system for the detection of the malaria parasite. Currently, the model is trained on microscopic images. In future work, the model can be trained on the images directly taken from phone cameras. With the help of quantization methods, the sizes of models can be further reduced to deploy them on a variety of devices.

Author Contributions: The authors of this work contributed in the following ways: conceptualization and formal analysis, S.P. and S.B.; methodology, S.P. and D.A.; software, K.M. and O.A.N.; data curation, M.N.M.; resources, K.M. and M.N.M.; supervision, S.B.; writing—original draft, S.P.; writing—review and editing, S.P. and S.B. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through the Small Groups Project under grant number RGP.1/26/43.

Acknowledgments: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through the Small Groups Project under grant number RGP.1/26/43.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. World Health Organization World Malaria Report 2018; World Health Organization: Genève, Switzerland, 2019; ISBN 9789241565653.
2. White, N.J.; Ho, M. The pathophysiology of malaria. In *Advances in Parasitology*; Elsevier: Amsterdam, The Netherlands, 1992; Volume 31, pp. 83–173.
3. Kwiatkowski, D.; Sambou, I.; Twumasi, P.; Greenwood, B.; Hill, A.; Manogue, K.; Cerami, A.; Castracane, J.; Brewster, D. Tnf concentration in fatal cerebral, non-fatal cerebral, and uncomplicated Plasmodium falciparum malaria. *The Lancet* **1990**, *336*, 1201–1204. [CrossRef]
4. O'Meara, W.P.; Barcus, M.; Wongsrichanalai, C.; Muth, S.; Maguire, J.D.; Jordan, R.G.; Prescott, W.R.; McKenzie, F.E. Reader technique as a source of variability in determining malaria parasite density by microscopy. *Malar. J.* **2006**, *5*, 118. [CrossRef]
5. Talapko, J.; Škrlec, I.; Alebić, T.; Jukić, M.; Včev, A. Malaria: The Past and the Present. *Microorganisms* **2019**, *7*, 179. [CrossRef]
6. ZJan, Z.; Khan, A.; Sajjad, M.; Muhammad, K.; Rho, S.; Mehmood, I. A review on automated diagnosis of malaria parasite in microscopic blood smears images. *Multimed. Tools Appl.* **2017**, *77*, 9801–9826.
7. Faster R-CNN Explained for Object Detection Tasks. Available online: <https://blog.paperspace.com/faster-r-cnn-explained-object-detection/> (accessed on 27 January 2022).
8. YOLO: Real-Time Object Detection Explained. Available online: <https://www.v7labs.com/blog/yolo-object-detection> (accessed on 27 January 2022).
9. How to Use Yolo v5 Object Detection Algorithm for Custom Object Detection. Available online: <https://www.analyticsvidhya.com/blog/2021/12/how-to-use-yolo-v5-object-detection-algoritem-for-custom-object-detection-an-example-use-case/> (accessed on 24 January 2022).
10. Roy, S.S.; Goti, V.; Sood, A.; Roy, H.; Gavrilu, T.; Floroian, D.; Paraschiv, N.; Mohammadi-Ivatloo, B. L2 Regularized Deep Convolutional Neural Networks for Fire Detection. *J. Intell. Fuzzy Syst.* **2022**, *43*, 1799–1810. [CrossRef]
11. Hung, J.; Carpenter, A. Applying faster r-cnn for object detection on malaria images. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway Township, NJ, USA, 2017; pp. 808–813.
12. Pattanaik, P.; Swarnkar, T.; Sheet, D. Object detection technique for malaria parasite in thin blood smear images. In Proceedings of the 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Kansas City, MO, USA, 13–16 November 2017; IEEE: Piscataway Township, NJ, USA, 2017; pp. 2120–2123.
13. Zedda, L.; Loddo, A.; Di Ruberto, C. A Deep Learning Based Framework for Malaria Diagnosis on High Variation Data Set. In Proceedings of the International Conference on Image Analysis and Processing, Lecce, Italy, 23–27 May 2022; Springer: Cham, Switzerland; pp. 358–370.
14. Shal, A.; Gupta, R. A comparative study on malaria cell detection using computer vision. In Proceedings of the 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Virtual, 27–28 January 2022; IEEE: Piscataway Township, NJ, USA, 2022; pp. 548–552.
15. Chibuta, S.; Acar, A.C. Real-time Malaria Parasite Screening in Thick Blood Smears for Low-Resource Setting. *J. Digit. Imaging* **2020**, *33*, 763–775. [CrossRef] [PubMed]
16. Loh, D.R.; Yong, W.X.; Yapeter, J.; Subburaj, K.; Chandramohanadas, R. A deep learning approach to the screening of malaria infection: Automated and rapid cell counting, object detection and instance segmentation using Mask R-CNN. *Comput. Med. Imaging Graph.* **2021**, *88*, 101845. [CrossRef] [PubMed]
17. Nakasi, R.; Zawedde, A.; Mwebaze, E.; Tusubira, J.F.; Maiga, G. Localization of malaria parasites and white blood cells in thick blood smears. *arXiv* **2020**, arXiv:2012.01994. [CrossRef]
18. Quinn, J.A.; Nakasi, R.; Mugagga, P.K.; Byanyima, P.; Lubega, W.; Andama, A. Deep convolutional neural networks for microscopy- based point of care diagnostics. In Proceedings of the Machine Learning for Healthcare Conference, Los Angeles, CA, USA, 19–20 August 2016; pp. 271–281.

19. Koirala, A.; Jha, M.; Bodapati, S.; Mishra, A.; Chetty, G.; Sahu, P.K.; Mohanty, S.; Padhan, T.K.; Mattoo, J.; Hukkoo, A. Deep Learning for Real-Time Malaria Parasite Detection and Counting Using YOLO-mp. *IEEE Access* **2022**, *10*, 102157–102172. [CrossRef]
20. Manku, R.R.; Sharma, A.; Panchbhai, A. Malaria Detection and Classification. *arXiv* **2020**, arXiv:2011.14329. [CrossRef]
21. Dong, Y.; Pan, W.D. Image Classification in JPEG Compression Domain for Malaria Infection Detection. *J. Imaging* **2022**, *8*, 129. [CrossRef] [PubMed]
22. Roy, S.S.; Rodrigues, N.; Taguchi, Y.-H. Incremental Dilations Using CNN for Brain Tumor Classification. *Appl. Sci.* **2020**, *10*, 4915. [CrossRef]
23. Torch Hub Series #3: YOLOv5 and SSD—Models on Object Detection. Available online: <https://pyimagesearch.com/2022/01/03/torch-hub-series-3-yolov5-and-ssd-models-on-object-detection/> (accessed on 13 January 2022).
24. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
25. How Flip Augmentation Improves Model Performance. Available online: <https://blog.roboflow.com/how-flip-augmentation-improves-model-performance/> (accessed on 20 February 2022).
26. Yolo-v5 Object Detection on a Custom Dataset. Available online: <https://towardsai.net/p/computer-vision/yolo-v5-object-detection-on-a-custom-dataset> (accessed on 12 February 2022).
27. Give Your Software the Power to See Objects in Images and Video. Available online: <https://roboflow.com/> (accessed on 19 December 2021).
28. Xu, R.; Lin, H.; Lu, K.; Cao, L.; Liu, Y. A Forest Fire Detection System Based on Ensemble Learning. *Forests* **2021**, *12*, 217. [CrossRef]
29. Wang, C.-Y.; Liao, H.-Y.M.; Wu, Y.H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I.-H. Cspnet: A new backbone that can enhance learning capability of cnn. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
30. Xu, B.; Wang, N.; Chen, T.; Li, M. Empirical evaluation of rectified activations in convolutional network. *arXiv* **2015**, arXiv:1505.00853. [CrossRef]
31. Bergstra, J.; Bengio, Y. Quadratic features and deep architectures for chunking. In Proceedings of the Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Boulder, CO, USA, 31 May–5 June 2009; pp. 245–248.
32. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747. [CrossRef]
33. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980. [CrossRef]
34. Ganaie, M.; Hu, M.; Malik, A.; Tanveer, M.; Suganthan, P. Ensemble deep learning: A review. *Eng. Appl. Artif. Intell.* **2022**, *115*, 2104–22395. [CrossRef]
35. Padilla, R.; Netto, S.L.; da Silva, E.A.B. A Survey on Performance Metrics for Object-Detection Algorithms. In Proceedings of the 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Niterói, Brazil, 1–3 July 2020; IEEE: Piscataway Township, NJ, USA, 2020; pp. 237–242.
36. Raschka, S. An overview of general performance metrics of binary classifier systems. *arXiv* **2014**, arXiv:1410.5330. [CrossRef]
37. Yoma, N.B.; Wuth, J.; Pinto, A.; de Celis, N.; Celis, J.; Huenupan, F.; Fustos-Toribio, I.J. End-to-end LSTM based estimation of volcano event epicenter localization. *J. Volcanol. Geotherm. Res.* **2021**, *429*, 107615. [CrossRef]
38. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991. [CrossRef]
39. Subramanian, B.; Olimov, B.; Naik, S.M.; Kim, S.; Park, K.-H.; Kim, J. An integrated mediapipe-optimized GRU model for Indian sign language recognition. *Sci. Rep.* **2022**, *12*, 11964. [CrossRef] [PubMed]
40. Yan, B.; Fan, P.; Lei, X.; Liu, Z.; Yang, F. A Real-Time Apple Targets Detection Method for Picking Robot Based on Improved YOLOv5. *Remote Sens.* **2021**, *13*, 1619. [CrossRef]