NTU Journal of Engineering and Technology (ICSDT2025) special issue
DOI: https://doi.org/10.56286/ntujet.icsdt2025

IRAQI Academic Scientific Journals

P-ISSN: 2788-9971 E-ISSN: 2788-998X
NTU Journal of Engineering and Technology
Available online at: https://journals.ntu.edu.iq/index.php/NTU-JET/index

Leveraging Deep Learning for Anemia Detection Using Palm Images: Innovative Solutions for Sustainable Development

Saab Khalid Al bdrani[1,2], Fadwa Al Azzo[3]
[1]Technical Engineering College / Northern Technical University, Iraq
[2]Nineveh Health Directorate / Ministry of Health, Iraq,
[3]Technical Engineering College for computer and AI /Mosul / Northern Technical University, Iraq
saabkhalid1987@ntu.edu.iq , fadwaalezoo@ntu.edu.iq

Article Informations

Received: 15-03- 2025,
Accepted: 28-04-2025,
Published online: 01-03-2026

Corresponding author:
Name: Saab Khalid Al bdrani
Affiliation : Northern Technical University
Email: saabkhalid1987@ntu.edu.iq

Key Words:
anemia detection,
YOLOv11,
Sustainable Development.

A B S T R A C T

A wide variety of medical diagnoses depend on the study and testing of the dis-ease images. Anemia is one of the cases separated in the populations. Deep Learning (DL) is one of the subfields of artificial intelligence (AI) techniques applied in the healthcare system to diagnose diseases. In this work, YOLOv11 with its versions Nano (n), Small (s), Medium (m), Large (l), and Extra-large (x) produces deep-learning models to detect anemia based on the palm images. The models train on 4260 color images of children under 5 years labeled anemic and nonanemic. The models train using two different dataset splits, with two input image sizes, (64x64) and (128x128). The high training accuracy achieves at YOLOv11n 99.3% at the group two dataset with an input image size of 128 and YOLOv11n 98.9% at the group one dataset with an input image size of 128. All models are test, and the best test results obtains with the Yolov11n models, 98.9% and 99.5% of the two groups at input image size 128, which predicted correctly with a high percentage in a very short time. We produce these models to assess medical images, providing precise automated estimations and reducing diagnosis time and errors. Additionally, they help reduce death rates and promote early intervention to enhance the quality of life for patients.

1. Introduction

Anemia is a dangerous global public health problem that specifically affects young children, menstruating adolescent girls, pregnant women, and postpartum women [1]. Anemia is characterized by a reduction in the number of red blood cells, hemoglobin concentration, or abnormal structure of red blood cells, which results in a reduction in the oxygenation of tissues, leading to a variety of clinical symptoms and complica-tions [2,3].The WHO estimates that 40% of children 6–59 months of age, 37% of pregnant women, and 30% of women 15–49 years of age worldwide are anemic[4]. Anemia, particularly in children and expectant mothers, increases maternal and in-fant mortality risk. It impairs children's cognitive and physical development and low-ers adult productivity. Causes include dietary deficiencies, infections, inflammation, chronic illnesses, gynecological conditions, inherited red blood cell disorders, and defi-ciencies in folate and vitamins B12 [4]. Table 1 shows the human normal range of hemoglobin [5].

Table 1. Human normal range of hemoglobin

Age with Gender	Normal range
Adult men	14 - 18 g/dL
Adult women	12 - 16 g/dL
pregnant female	>11 g/dL
Newborn (male, female)	14 -24 g/dL
0 - 2 weeks	12 -20 g/dL
2 - 6 months	10 – 17 g/dL
6 months – 1 year & 1 - 6 years	9.5 – 14 g/dL
6 -18 years	10 - 15.5 g/dL

The complete blood count (CBC) test, which includes components like red blood cells (RBCs), white blood cells (WBCs), hemoglobin, hematocrit, and platelets, is typically the first test to be carried out for the diagnosis of anemia [6]. This approach is the gold standard for detecting anemia; it can be time-consuming, expensive, labor-intensive, and expose medical professionals to the risks of blood-transmissible illnesses [7]. In several circumstances, quick assessment of clinical indicators of anemia, such as lowering the eyelids and evaluating conjunctiva pallor, could offer a prompt indication of the patient's condition; however, these techniques are inaccurate due to a lack of instruments. The usage of computational systems could be quite beneficial in overcoming this obstacle [8]. Non-invasive techniques have gained popularity as an alternative to invasive methods for measuring hemoglobin since they don't require a blood sample, leave no scars, have a low risk of infection, allow for real-time patient monitoring, and are portable [9]. Actually, several factors have increased the use of artificial intelligence algorithms in significant scientific fields, including medicine, due to their remarkable ability to recognize faces and images and to comprehend patterns [10]. Machine learning and deep learning are crucial in disease classification and prediction, especially in computer-assisted medical imaging, aiding early detection, mitigating risk, and aiding physicians in making definitive diagnoses [11]. YOLO, or You Only Look Once, is a modern deep learning algorithm introduced in 2015 by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi for real-time object identification[12]. To obtain important insights, YOLO can be combined with computer vision techniques such as object identification, image classification, and instance segmentation in a variety of industries, from autonomous driving to medical imaging [13]. The Sustainable Development Goals (SDGs) of the United Nations place a strong emphasis on health and well-being, especially Goal 3, which seeks to guarantee healthy lives and advance well-being for people of all ages. These objectives are in line with efficient anemia detection by Improving Health Outcomes: Early detection based on deep learning can lead to timely treatment, reducing the morbidity and mortality associated with anemia. The enhancement of economic production is possible since anaemia can affect mental and physical functioning, which affects worker productivity and education. Improving health can lead to greater economic stability and growth for communities. Increasing Access to Healthcare: Deep learning algorithms can improve screening abilities in low-resource settings, increasing underserved populations' access to healthcare [4] .

2. Related Literature Review

Many studies have been produced for anemia detection, as follows:

In 2015 Carlo Barbieri et al, developed a machine learning model using feed-forward artificial neural networks (ANN) to predict hemoglobin levels in patients with secondary anemia in end-stage renal disease undergoing dialysis. Using feed-forward artificial neural networks (ANN) real data from 313 patients, the ANN model was implemented and has achieved accuracy of 93% [14] . In 2016 Shaun Collings et al, produced a non-invasive method for detecting anemia by quantifying conjunctiva pallor using digital photographs taken with a camera or smartphone. The study involved 106 participants, who were given 188 images analyzed using Image

software to calculate an erythema index (EI). The best results were reported for detecting anemia using the palpebral conjunctiva erythema index, with a sensitivity of 93% and specificity of 78% [15]. In 2017 Azwad Tamir et al, designed a noninvasive mechanism. Detection of Anemia uses a smartphone-based conjunctiva pallor analysis to detect anemia. With a 78.9% accuracy rate, the model reported 15 cases from 19 cases, based on digital photographs taken under suitable lighting conditions [16] . In 2018 R. G. Mannino et al , designed smartphone app has been to detect noninvasively anemia by analyzing fingernail bed photos. The app estimates hemoglobin levels based on color and metadata, with an accuracy of ±2.4 g/dL and a sensitivity of 97%. The dataset used consists of 337 subjects ages (1 to 60 years old) [17]. In 2019 Shubham Bauskar et al. proposed a noninvasive technique to detect anemia using eye conjunctiva images, extracting ROI and mean intensity values. Tested on 99 images, a machine learning algorithm predicted anemia with 93% accuracy [18]. In 2020, Laith Alzubaidi et al. proposed three deep learning models to classify red blood cells in microscopy images for sickle cell anemia diagnosis, achieving 99.54% accuracy on the erythrocyte IDB1 dataset and 98.87% accuracy on the collected dataset using a convolutional neural network [19]. In 2021 T. K. Yıldız et al , produced a study classified 12 anemia types using artificial neural networks, SVM, Naïve Bayes, and decision trees, with the highest accuracy (85.6%) achieved using decision trees[20].In 2022, Aixian Zhang et al. proposed a deep learning method to predict anemia using facial images from 316 videos of 217 patients, using a convolutional neural network. The model achieved 84.02% accuracy, outperforming senior doctors' assessments, making it suitable for emergency settings [21]. In 2023 S.Dhanasekaran et al , proposed a CNN All Net that outperformed DenseNet121 and Efficient Net B3 ,YOLO v5 detecting anemia from microscopic palm images on a dataset of 3,000 images, with CNN All Net achieving the highest accuracy of 96.8% [22] . In 2024, Nilesh Bhaskarrao Bahadure et al. developed a deep learning model, YOLOv6, classifying anemia from blood samples. The dataset consists of 551 blood samples collected from women in underprivileged areas, achieving an accuracy of 97.60% [23]. The studies that have been produced range from machine learning to deep learning. The studies have utilized a variety of data types, including CNN nets and mobile applications. This work focuses on using a new version of YOLO, YOLOv11, with its versions Nano (n), Small (s), Medium (m), Large (l), and Extra-large. The model's success for training and testing with different accuracy percentages on the PC software Visual Studio Code app.

3. YOLO

You Only Look Once (YOLO) is a cutting-edge, real-time object detection method introduced in 2015 by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Far-hadi. It is a modern deep learning technique employed across multiple domains [12] . YOLO is a convolutional neural network (CNN) specifically engineered for object detection and image segmentation tasks. In contrast to conventional object identifi-cation techniques that classify and localize independently, YOLO treats the issue as a unified regression problem. It divides an image into a grid and instantly predicts bounding boxes and class probabilities from the whole image in a single evaluation, which makes real-time applications faster and more efficient [24].

3.1 YOLO versions

YOLO was produced in several versions, from YOLOv1 to YOLOv11, the last version. Each version is different in input image size, with the internal structure shown in Table 2 [25] .

Table 2. YOLO versions from v1 to v11

Version	Maximum size of the Input image	Release date
Version 1 (v1)	(488 x 488)	2016
Version 2 (v2)	(416 x 416)	2017
Version 3 (v3)	(608 x 608)	2018
Version 4 (v4)	(608 x 608)	2020
Version 5 (v5)	(608 x 608)	2020
Version 6 (v6)	(640 x 640)	2022
Version 7 (v7)	(640 x 640)	2022
Version 8 (v8)	(640 x 640)	2023
Version 9 (v9)	(640 x 640)	2024
Version 10 (v10)	(640 x 640)	2024
Version 11 (v11)	(640 x 640)	2024

Each version has subversions Nano (n), Small (s), Medium (m), Large (l), and Extra-large (x). The YOLOv11 release in September of 2024, developed by the time of ultralytics, has task detection, pose estimation, segmentation, tracking, and classification [26].

3.2 YOLO classification

It is performed to obtain essential information and insights from medical images. The classification achieves improved accuracy and delivers useful information on the affected area by the diseases. The YOLOv11 version has been extracted, and the features depend on resizing the input image and its layers. Nano (n) and small (s) 112 layers, medium (m) 138 layers, large (l) and extra -large (x) 227 layers, training in the YOLO has three stages: input

stage, input image, features extraction stage, and prediction stage as shown in Fig. 1.

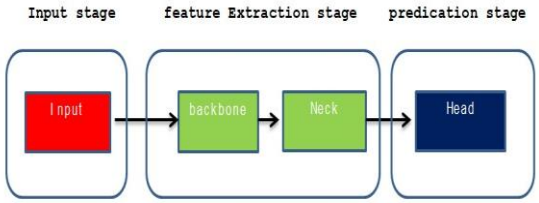

Fig. 1. YOLO process stages

3.2.1 Backbone: extraction of features

The backbone extracts key elements from the input image made up of several convolutional layers. YOLOv11's improved backbone architecture enhances feature extraction capabilities, leading to more accurate object detection. One notable innovation is the insertion of the C3k2 block into the backbone of YOLOv11. This block improves the Cross Stage Partial (CSP) bottleneck by using smaller 3x3 kernels for more efficient computing while preserving the model's ability to gather crucial image data. [28].

3.2.2 Neck fusion of features

The neck component prepares features from different stages of the backbone for the head by processing and combining them. The Spatial Pyramid Pooling - Fast (SPPF) module in the neck, which was included in YOLOv11, enhances the model's ability to handle objects at different scales by combining data from several regions of the image [29].

3.2.3 Head: predictions specific to the task

YOLOv11 uses the C2PSA technique in the head to improve the accuracy of de-tection and focus on important image areas, making final predictions for object clas-sification and localization [29].

3.3 Why YOLO?

YOLO (You Only Look Once) model is a contemporary choice because it provides many options, like object detection, classification, and other capabilities. It has exhibited remarkable efficiency in categorizing palm images with outstanding execution speed. YOLO executes detection and classification in a singular stage, rendering it more rapid than conventional models like Faster R-CNN. YOLO has significant computing efficiency, rendering it appropriate for implementation on low-power devices. This study updated the model for binary classification (anemic vs. non-anemic), showing that it can be used to evaluate palm features related to anemia. Because of these qualities, YOLO was chosen because it has the best balance of speed, efficiency, and accuracy, making it perfect for computer vision-based medical classification tasks.

4. Methodology

The proposed system has been using the dataset that is labeled into two classes, anemic and nonanemic, and converted it to the JPG type. We split the dataset into two groups based on percentages: group one comprises 70% training, 15% validation, and 15% testing, while group two consists of 80% training, 10% validation, and 10% testing. Each dataset group trained at two sizes of input images, 64x64 and 128x128, based on YOLOv11 versions n, s, m, l, and x. The high accuracy of training was 98.9% at group one dataset splitting when the input image size was 128 and 99.3% at group two dataset splitting when the input image size was 128 in the YOLOv11n. We tested all models using the dataset's test images. We illustrate the methodology in Fig. 2 below.

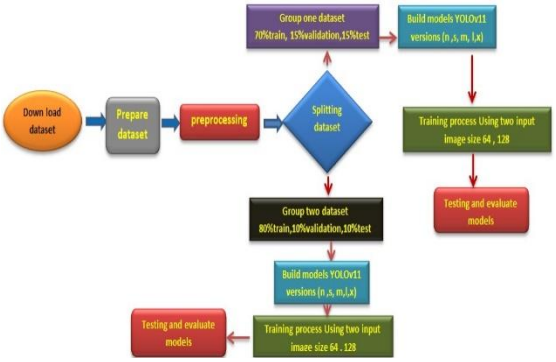

Fig. 2. Block diagram of methodology

4.1 Dataset

The dataset consists of 710 microscopic palm images of children under 5 years old, divided into two classes, anemic and non-anemic, which include 426 and 284 images, respectively. To create divergences of the original images, the dataset was augmented with 4260 labeled images for both classes. These divergences improve dataset diversity, model robustness, and generality. Fig. 3 shows samples of datasets for the anemic and non-anemic classes.

Fig. 3. (a) Anemic image (b) Non-anemic image [27].

4.2 Prepare the dataset

YOLOv11 classification deals with specific partitions for training, as Fig. 4 below illustrates.

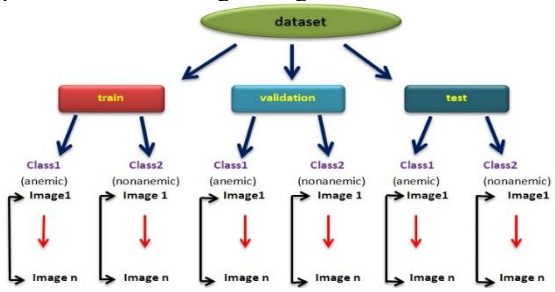

Fig. 4. Preparation of the dataset

4.3 Preprocessing

The dataset was converted from PNG to JPG, which led to easy processing using a program in Matlab 2022a.

4.4 Data splitting

The dataset is divided into two groups using a program in MATLAB 2022. The first group consists of 70% of the 2982 training images. 1810 anemic and 1172 non-anemic images, 15% validation 639 images, 386 anemic and 253 non-anemic images, and 15% of the 639 images. 366 anemic with 273 non-anemic images, the second group of data was split to 80% train, 3408 images, 2072 anemic and 1336 non-anemic images, 10% validation. 426 images, 243 anemic and 183 non-anemic images, and 10% test, 426 images, 35 anemic with 391 non-anemic images, as Fig. 5 below illustrates.

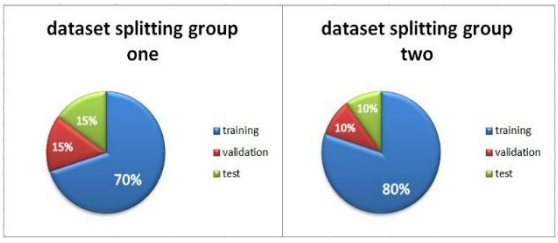

Fig. 5. The two groups of the dataset splitting percentage

4.5 build models

The Visual Studio Codes program used YOLOv11 to build deep learning models, which Python 3.12.7 then implemented

4.5.1 YOLO setup and start

Initiating the use of YOLO is expeditious and uncomplicated. The Ultralytics package can be installed via pip, allowing for rapid setup within minutes. This step is a funda-mental installation command [13] :

pip install ultralytics

YOLOv11 classification starts using the command [13] :

YOLO11 (version)-cls

4.6 Training the first group

YOLO v11with its versions n, m, s, l, and x were trained on a dataset split into 70% training, 15% validation, and 15% test, using two stages of input images sized 64 and 128 over 100 epochs.

4.7 Training the second group

The YOLO v11 with its versions n, m, s, l, and x were trained on a dataset that was split to 80% training, 10% validation, and 10% test with two stages of input images size 64 and 128 with 100 epochs.

4.8 Feature extraction

YOLOv11, a flexible deep learning model, excels not only in object recognition but also in image classification. After being trained for image classification, YOLOv11 assigns an image to a particular class by processing it through a sequence of layers intended to extract and analyze features. YOLOv11 gathers features during its classi-fication training.

4.8.1 The input layer

The input layer is the first step in the process, where the image is shrunk to a standard size appropriate for the model. This resizing guarantees consistency and alignment with the architecture of the network. [28].

4.8.2 Layer of global pooling

A global pooling layer reduces the spatial dimensions of the feature maps following feature extraction. By combining data, this layer creates a condensed representation that highlights the most important characteristics for classification.

4.8.3 Full connected Layer

Thereafter, the condensed characteristics are sent to fully linked layers, which help with decision-making by interpreting the features that were extracted. These layers associate the corresponding classes with the high-level features.

4.8.4 Output layer

The final output layer uses an activation function, such as softmax, to generate probability scores for each class. For the input image, the model predicts the class with the highest probability. YOLOv11's architecture improves image classification by gathering detailed information and complex patterns across different categories. This is done by making feature extraction and computation more efficient. In this work, the input image is set to a new size, either 128 or 64, meaning 128x128x3; the maximum size of YOLOv11 is 640x640; the original size of images varies between 126x240x3

and 138x126x3; and therefore, 128 has been chosen. Fig. 6 illustrates the classification process.

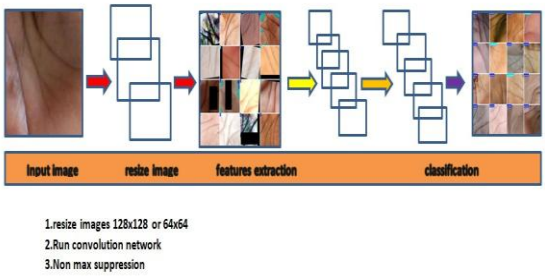

Fig. 6. Classification process

4.9 Testing the models

All YOLOv11 models have been trained and tested on test images of the dataset and evaluated the test results.

5 Results and Discussion

The work is done by PC that specification
Processor: Intel(R) Core (TM) i7-4600M CPU @ 2.90GHz 2.89 GHz
Installed RAM: 8.00 GB (7.61 GB usable)
System type: 64-bit operating system, x64-based processor
Edition: Windows 10 Pro
Version: 22H2

We have developed models for training and testing on two sets of datasets using Python 3.12.7 in the Visual Studio Code app. All models were run 100 times with 64- and 128-bit images as input, YOLO v11 versions n, m, l, and x, and the ultralytics library was used to classify features of diagnostic images.

5.1 Training results of group one datasets

YOLOv11 with its versions n, m, s, l, and x has been running with dataset splitting 70% training, 15% validation, and 15% test at 100 epochs. For an input size of 64×64, the maximum accuracy achieved was 96.2% (YOLO v11s), but the minimum was 92.3% (YOLO v11x). The highest accuracy attained for a 128x128 input size was 98.9% (YOLO v11n), while the lowest was 96.9% (YOLO v11x). Augmenting the input image size from 64 to 128 often enhanced accuracy, suggesting that the model gains from more intricate images. Reduced values signify enhanced learning throughout training. With a 64×64 input, the minimum loss recorded was 0.14449 (YOLO v11s), whereas the maximum was 0.30234 (YOLO v11x), indicating that YOLO v11s exhibited greater stability. With a 128×128 input, the minimum loss recorded was 0.07632 (YOLO v11n), while the maximum was 0.14368 (YOLO v11x), indicating that smaller models exhibited superior performance with bigger image dimensions. With a 64x64 input, the lowest validation loss recorded was 0.35729 for YOLO v11s and the highest was 0.40359 for YOLO v11x. This suggests that YOLO v11s was better at generalization on the validation dataset. With a 128×128 input, the minimum validation loss recorded was 0.32738 (YOLO v11n), whereas the maximum was 0.3527 (YOLO v11x), indicating the superior performance of YOLO v1n in validation. For optimal accuracy, YOLO v11n (128×128) is the superior choice. For optimal accuracy and speed, YOLO v11s (128×128) delivers commendable performance (97.7%) alongside minimal validation loss. Training results of this part are shown in Table **3**, Fig. 7 illustrates the training accuracy curves for the first dataset when the input image is 64.

5.2 Training results of group two datasets

YOLOv11 with its versions n, m, s, l, and x has been running with dataset splitting 80% training, 10% validation, and 10% test at 100 epochs. For an input size of 64×64, the maximum accuracy achieved was 98.8% (YOLO v11s), while the minimum was 93% (YOLO v11x). YOLO v11n, YOLO v11m, and YOLO v11l all attained accuracy exceeding 96%, signifying consistent performance. For an input size of 128×128, the maximum accuracy achieved was 99.3% (YOLO v11n), indicating exceptional performance. YOLO v11l and YOLO v11x achieved commendable performance, with accuracies of 99.1% and 98.4%, respectively, shown in Figure 8. Nonetheless, YOLO v11s decreased to 97.2%, which is inferior to its 64×64 accuracy. Increasing the input dimensions from 64 by 64 to 128 by 128 improved accuracy for most models, with the YOLO v11n model achieving the highest accuracy of 99.3%. Regarding Training Loss for an input size of 64×64, the minimum training loss recorded was 0.12873 (YOLO v11s), indicating effective learning. The peak training loss was 0.40498 (YOLO v11x), indicating instability. For an input size of 128×128, the minimum training loss recorded was 0.06362 (YOLO v11n), indicating optimal learning performance. YOLO v11x showed substantial improvement, achieving a score of 0.08116, surpassing its performance at 64×64. Observation: YOLO v11n exhibited the greatest advantage from a 128×128 input size, attaining the minimal training loss of 0.06362. Validation Loss (Val/Loss) For an input size of 64×64, the minimum validation loss recorded was 0.33128 (YOLO v11s), indicating robust generalization. The maximum was 0.42313 (YOLO v11x), indicating inadequate validation performance. For an input size of

Table 3. Training results of YOLOv11 versions of the group one dataset

Training results of group one dataset						
	Input image size (64 x64)			Input image size (128x128)		
YOLOv11 versions	Accuracy	Train / loss	Val /loss	Accuracy	Train / loss	Val/loss
n	95.6%	0.18154	0.36417	98.9 %	0.09102	0.32843
s	96.2 %	0.15999	0.35729	97.7 %	0.08354	0.34625
m	93.7 %	0.16314	0.38309	97.3 %	0.11413	0.34244
l	95.5 %	0.16816	0.36309	97.8 %	0.11992	0.34185
x	92.3 %	0.31744	0.40359	96.9 %	0.14938	0.35545

Table 4. Training results of YOLOv11 versions of the group two dataset

Training results of group two dataset						
	Input image size (64 x64)			Input image size (128x128)		
YOLOv11 versions	Accuracy	Train / loss	Val /loss	Accuracy	Train / loss	Val/loss
n	97.9 %	0.21559	0.3526	99.3 %	0.06362	0.32252
s	98.8 %	0.12873	0.33128	97.2 %	0.09842	0.34327
m	97.9 %	0.17572	0.3413	98.4 %	0.10653	0.33481
l	96.9 %	0.20822	0.35429	99.1 %	0.06597	0.32647
x	93 %	0.40498	0.42313	98.4 %	0.08106	0.33392

128×128, the minimum validation loss recorded was 0.32252 (YOLO v11n), hence affirming its dependability. The maximum was 0.34327 (YOLO v11s), indicating it had difficulties with generalization at elevated resolutions. YOLO version 11 consistently exhibited the lowest validation loss, indicating superior generalization compared to the other models, as shown in Figure 9. YOLO v11n (128×128) attained the best accuracy (99.3%) and the lowest validation loss (0.32252), establishing it as the preferred option. YOLO v11l (128×128) exhibited commendable performance with an accuracy of 99.0%, albeit with a little elevated loss. YOLO v11s (64×64) demonstrated exceptional accuracy (98.8%) and little validation loss (0.33128), rendering it an optimal choice for speed and efficiency. Nonetheless, at 128×128, its performance decreased to 97.2%, indicating potential scalability issues. YOLO v11x exhibited the poorest performance at 64×64, achieving 93% accuracy and a maximum validation loss of 0.42313. Nevertheless, it showed a big improvement at 128x128 (98.4% accuracy), as shown in Table 4, which suggests that bigger images are better. YOLO v11n (128×128) is the optimal selection for achieving the greatest accuracy and generalization. When speed and efficiency are prioritized, YOLO v11s (64×64) delivers robust accuracy (98.8%) at a little computational expense. I do not prefer YOLO v11x at (64×64) because it has high losses and reduced accuracy.

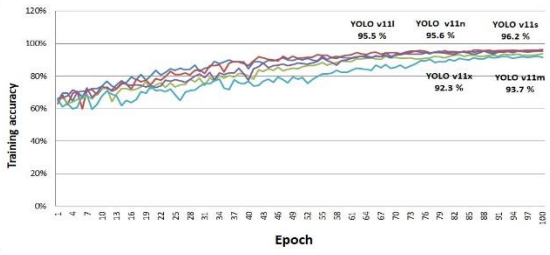

Fig. 7. Training accuracy models group one image size 64

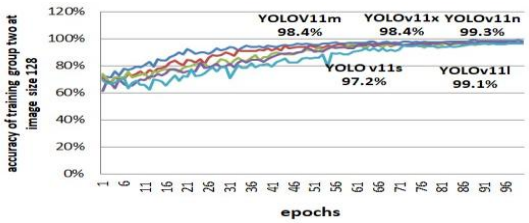

Fig. 8. Training accuracy models second dataset with image size 128

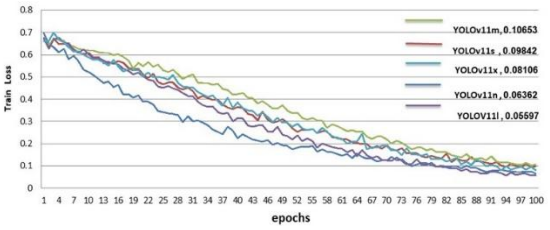

Fig. 9. Training loss of models at second dataset with image size 128.

5.3 Test results

We tested YOLOv11 versions using the dataset's test file, and the models displayed different results. The test image results in the models were different depending on a number of things, such as the number of datasets, how the datasets were split to give the model's features, the number of epochs, the PC's specs, and the type of YOLO versions used, including the version layer. As a result, the best test result we had at group one models exhibited enhanced performance with a 128×128 input size, resulting in improved accuracy and feature extraction. In the 64×64 configuration, YOLO v11s exhibits the greatest equilibrium, attaining an accuracy of 96.5%, whereas YOLO v11x demonstrates the least performance at 94.6%. For 128×128, YOLO v11n exhibits the highest accuracy at 98.9% with superior recall, succeeded by YOLO v11x at 98.1% accuracy. Optimal choice: YOLO v11n (128×128) for maximum accuracy. Optimal for minimal input: YOLO v11s (64×64) for equitable performance, as shown in
Table **6**. In the second group of models, increasing the input size to 128×128 enhances performance in all models except for YOLO v11s, which experiences a minor decline in accuracy. YOLO v11x exhibited suboptimal performance at 64×64 with an accuracy of 87.8% but showed substantial improvement at 128×128, with an accuracy of 97.8%. The optimal model for 64×64 resolution is YOLO v11s, succeeded by YOLO v11n. The optimal model at 128×128 is YOLO v11n (99.5% accuracy, 1.00 recall and precision). YOLO v11x necessitates larger images (128×128) for optimal performance. Utilize YOLO v11n (128×128) for

Table 5. Test results of group one dataset

Test Results of Dataset 70% training, 15% validation, 15% test

YOLO v11 Version	64×64 Accuracy	64×64 Precision	64×64 F1 Score	64×64 Recall	128×128 Accuracy	128×128 Precision	128×128 F1 Score	128×128 Recall
n	95.7%	0.96	0.94	0.93	98.9%	0.99	0.98	0.97
s	96.5%	0.96	0.96	0.95	97.9%	0.98	0.97	0.96
m	95.3%	0.95	0.94	0.93	97.8%	0.97	0.97	0.97
l	96.5%	0.98	0.96	0.93	97.0%	0.97	0.96	0.96
x	94.6%	0.95	0.94	0.92	98.1%	0.98	0.97	0.98

Table 6. Test results of group two dataset

Test Results of Dataset 80% training, 10% validation, 10% test

YOLOv11 Version	64×64 Accuracy	64×64 Precision	64×64 F1 Score	64×64 Recall	128×128 Accuracy	128×128 Precision	128×128 F1 Score	128×128 Recall
n	98.7%	1.00	0.99	0.97	99.5%	1.00	0.99	1.00
s	99.5%	1.00	0.99	0.99	98.2%	0.99	0.98	0.97
m	96.9%	0.97	0.97	0.96	98.7%	0.98	0.99	0.99
l	97.4%	0.98	0.97	0.96	98.7%	0.99	0.99	0.98
x	87.8%	0.90	0.86	0.82	97.8%	0.98	0.98	0.97

optimal accuracy; if limited to 64×64, YOLO v11s is the superior option, as shown in Table 5. The results found that YOLO v11x, a model with subpar performance at 64x64, showed a significant improvement at 128x128. YOLO v11n, a superior model, achieved 99.5% accuracy and 100% recall at 128x128 resolution. YOLO v11s also excelled at 64x64, with 99.5% accuracy. The models' superior performance is due to image quality and the percentage of the splitting dataset.

6. Conclusion

In this work, deep learning models successfully classify the anemia after training in the new version of YOLO models, YOLOv11, with its versions n, s, m, l, and x at JPG microscopic palm images labeled two classes: anemic and non-anemic. The models deal with two splitting datasets named the first group, which splits to 70% train, 30% validation, and test, and the second group, which splits to 80% train, 20% validation, and test. All models are tested with the test images of the dataset; the models have varying results depending on the features of the training image and the type of YOLO version used. High training accuracy has been obtained at YOLOv11n at two types of splitting datasets with a personal PC at the Visual Studio Code app. using the YOLOv11 deep learning model to produce a noninvasive model to detect the anemia with high accuracy of prediction, a very short time for testing, a low-cost using software on a personal PC, the ability to develop in the future for software, and the production of a hardware tool. Also, this technology can enhance early intervention, improve health outcomes, promote social justice, and boost economic output. This technology can also contribute to public health campaigns, gender equality, and the Sustainable Development Goals, ensuring healthy lives and overall well-being.

References

[1] D. Kinyoki, A. E. Osgood-Zimmerman, N. V. Bhattacharjee, N. J. Kassebaum, and S. I. Hay, "Anemia prevalence in women of reproductive age in low- and middle-income countries between 2000 and 2018," Nature Medicine, vol. 27, no. 10, pp. 1761–1782, 2021, doi: 10.1038/s41591-021-01498-0.

[2] M. D. Cappellini, K. M. Musallam, and A. T. Taher, "Iron deficiency anaemia revisited," Journal of Internal Medicine, vol. 287, no. 2, pp. 153–170, 2020, doi: 10.1111/joim.13004.

[3] T. Sonnweber, A. Pizzini, I. Tancevski, J. Löffler-Ragg, and G. Weiss, "Anaemia, iron homeostasis and pulmonary hypertension: A review," Internal and Emergency Medicine, vol. 15, pp. 573–585, 2020, doi: 10.1007/s11739-020-02288-1.

[4] World Health Organization, "Anaemia." Accessed: Nov. 12, 2024. [Online]. Available: https://www.who.int/health-topics/anaemia

[5] B. Y. Merritt, "Hemoglobin concentration (Hb)," Medscape. [Online]. Available: https://emedicine.medscape.com/article/2085614-overview

[6] K. Meena, D. K. Tayal, V. Gupta, and A. Fatima, "Using classification techniques for statistical analysis of anemia," Artificial Intelligence in Medicine, vol. 94, pp. 138–152, 2019, doi: 10.1016/j.artmed.2019.02.005.

[7] S. Haghi, R. Arjmand, and O. Safari, "The application of artificial intelligence in the diagnosis and management of anemia," Iranian Journal of Blood and Cancer, vol. 15, no. 3, pp. 84–92, 2023, doi: 10.61186/ijbc.15.3.84.

[8] A. Mitani et al., "Detection of anaemia from retinal fundus images via deep learning," Nature Biomedical Engineering, vol. 4, no. 1, pp. 18–27, 2020, doi: 10.1038/s41551-019-0487-z.

[9] R. A. Rochmanto, H. Zakaria, R. D. Alviana, and N. Shahib, "Non-invasive hemoglobin measurement for anemia diagnosis," in Proc. 4th Int. Conf. Electrical Engineering, Computer Science and Informatics (EECSI), 2017, pp. 1–5, doi: 10.1109/EECSI.2017.8239096.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[11] F. Abdulatif and A. Ali, "Review of classification medical images using ensemble learning," NTU Journal of Pure Sciences, vol. 3, no. 4, 2024, doi: 10.56286/ntujps.v3i4.

[12] J. Redmon, "You only look once: Unified, real-time object detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016. [Online]. Available: http://175.27.250.89:10000/media/attachment/2024/08/YOLOV1.pdf

[13] Ultralytics, "YOLO." [Online]. Available: https://www.ultralytics.com/

[14] C. Barbieri et al., "A new machine learning approach for predicting the response to anemia treatment in a large cohort of end stage renal disease patients undergoing dialysis," Computer Biology and Medicine, vol. 61, pp. 56–61, 2015, doi: 10.1016/j.compbiomed.2015.03.019.

[15] S. Collings, O. Thompson, E. Hirst, L. Goossens, A. George, and R. Weinkove, "Non-invasive detection of anaemia using digital photographs of the conjunctiva," PLoS One, vol. 11, no. 4, p. e0153286, 2016, doi: 10.1371/journal.pone.0153286.

[16] A. Tamir et al., "Detection of anemia from image of the anterior conjunctiva of the eye by image processing and thresholding," in Proc. IEEE Region 10 Humanitarian Technology Conf. (R10-HTC), 2017, pp. 697–701, doi: 10.1109/R10-HTC.2017.8289053.

[17] R. G. Mannino et al., "Smartphone app for non-invasive detection of anemia using only patient-sourced photos," Nature Communications, vol. 9, no. 1, p. 4924, 2018, doi: 10.1038/s41467-018-07262-2.

[18] S. Bauskar, P. Jain, and M. Gyanchandani, "A noninvasive computerized technique to detect anemia using images of eye conjunctiva," Pattern Recognition and Image Analysis, vol. 29, pp. 438–446, 2019, doi: 10.1134/S1054661819030027.

[19] L. Alzubaidi, M. A. Fadhel, O. Al-Shamma, J. Zhang, and Y. Duan, "Deep learning models for classification of red blood cells in microscopy images

to aid in sickle cell anemia diagnosis," Electronics, vol. 9, no. 3, p. 427, 2020, doi: 10.3390/electronics9030427.

[20] T. K. Yıldız, N. Yurtay, and B. Öneç, "Classifying anemia types using artificial learning methods," Engineering Science and Technology, an International Journal, vol. 24, no. 1, pp. 50–70, 2021, doi: 10.1016/j.jestch.2020.12.003.

[21] A. Zhang et al., "Prediction of anemia using facial images and deep learning technology in the emergency department," Frontiers in Public Health, vol. 10, p. 964385, 2022, doi: 10.3389/fpubh.2022.964385.

[22] S. Dhanasekaran and N. R. Shanker, "Anemia detection using a deep learning algorithm by palm images," International Journal of Recent Innovation Trends in Computing and Communication, vol. 11, no. 7s, 2023, doi: 10.17762/ijritcc.v11i7s.6979.

[23] N. B. Bahadure, R. Khomane, and A. Nittala, "Anemia detection and classification from blood samples using data analysis and deep learning," Automatika, vol. 65, no. 3, pp. 1163–1176, 2024, doi: 10.1080/00051144.2024.2352317.

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, doi: 10.1109/CVPR.2016.91.

[25] A. Yeerjiang et al., "YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in medical image detection," Journal of Artificial Intelligence Practice, vol. 7, no. 3, pp. 112–122, 2024, doi: 10.23977/jaip.2024.070314.

[26] Z. Shariff, "YOLOv11 is officially out! What you need to know!" Medium. [Online]. Available: https://medium.com/@zainshariff6506/yolov11-is-officially-out-what-you-need-to-know-6738c5d25be1

[27] J. W. Asare, P. Appiahene, and E. Donkoh, "Anemia detection using palpable palm image datasets from Ghana," 2023, doi: 10.17632/ccr8cm22vz.1.

[28] N. R. Sulake, "Data science blogathon," Analytics Vidhya. [Online]. Available: https://www.analyticsvidhya.com/blog/2024/10/yolov11-object-detection/

[29] M. H. R. Khanam, "YOLOv11: An overview of the key architectural enhancements," 2024. [Online]. Available: https://arxiv.org/abs/2410.1772519.6.266-273.