JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 2, Juni 2024, Pp. 798-805

AUTOMATED ANAEMIA DETECTION FROM CONJUNCTIVA IMAGES : A MACHINE LEARNING APPROACH FOR ANDROID APPLICATION

Valentino Sas Henry*[1)], Yo'el Pieter Sumihar[2)], Febe Maedjaja[3)]
1. Immanuel Christian University, Yogyakarta, Indonesia
2. Immanuel Christian University, Yogyakarta, Indonesia
3. Immanuel Christian University, Yogyakarta, Indonesia

Article Info
Keywords: Anaemia Detection; Machine Learning; KNN; YoloV8

Article history:
Received 17 August 2018
Revised 15 February 2019
Accepted 4 April 2019
Available online 4 April 2019

DOI :
https://doi.org/10.29100/jipi.v9i2.5218

* Corresponding author.
Valentino Sas Henry
E-mail address:
valentino.s2042@student.ukrimuniversity.ac.id

ABSTRACT
Anaemia is a common global disorder condition in which the red blood cell count is lower than normal. Traditional diagnostic methods often prove costly, invasive, and inaccessible, leading to delays in treatment and severe consequences. This study explores non-invasive techniques leveraging smartphone technology for efficient anaemia detection. Some researcher investigated that eye's conjunctiva analysis can a viable alternative, considering its rich blood vessel network and sensitivity to anaemia indicators, independent of skin color. Utilizing smartphone cameras, the study establishes a robust correlation between the color of the conjunctiva and anaemia status, offering a promising avenue for non-invasive diagnosis. Employing a hybrid methodology, the study integrates You Only Look Once (YOLO) version 8 for efficient object detection, achieving a mean average precision of 96% in extracting Regions of Interest (ROI) from conjunctiva images. Subsequently, K-Nearest Neighbors (KNN) classification of features extracted from these ROI's attained an 83% accuracy rate in determining anaemia status. Furthermore, the study aims to extend its impact by developing an Android application using the Flutter framework. This application integrates the established YOLO and KNN approaches, enabling real-time anaemia detection through smartphone cameras. Such a tool holds the potential to facilitate early evaluations in resource-constrained regions, enabling timely diagnosis and intervention, thus mitigating the adverse effects of untreated anaemia.

.

I. INTRODUCTION

MILIONS of people worldwide suffer from anaemia, a disorder marked by a lack of red blood cells, which carries serious health risks [1]. Even though traditional invasive blood tests are effective, they can be costly, painful, time-consuming and difficult to get, especially for people living in low and middle income countries [2]. Delays in diagnosis and treatment may result in serious consequences, such as damage to organs and even death [3]. Many Studies have been conducted to solve this problem. These studies mentioned that there are some non-invasive and easier way of detecting anaemia range from the eye's conjunctiva, the color of fingernail, the hand palms and also from the tongue [4][5]. But among these, the most dominant and accurate is based on the conjunctiva analysis [6]. The conjunctiva, the mucous membrane covering the white of the eye, has rich blood vessel network and rich bilirubin, making it independent of skin color and can be used as an indicator of anaemia [7].

This study uses available smartphone technology with built-in camera to analyze the color, presenting an option around the cons of traditional methods [8][9]. Considering how widespread smart phones are in today's world, this approach could lead to universal accessibility. Previous research has established a strong correlation between the RGB color of the conjunctiva and anaemia status [10][11]. This study aims to develop a new way of detecting anaemia by combining You Only Look Once (YOLO) version 8 and K-Nearest Neighbors (KNN) methodologies within an Android application framework. YOLO version 8 is utilized for object detection due to its efficiency, fast process, and the ability to process images in real-time [12]. The version 8 has the most advance architecture and performance over previous iterations. This method will be employed to identify the Region of Interest (ROI) for the conjunctiva image. This choice is in line with the necessity for rapid detection, which is essential for effective intervention. KNN was utilized for its simplicity and effectiveness in recognizing patterns [13], then be used to classify the final result whether a person has anaemia or not by looking for patterns from extracted features of the ROI. KNN is a straightforward algorithm that classifies objects based on similarities to their neighbors in a feature space. KNN is well suited for anaemic detection by examining the features extracted from the eye conjunctiva ROI from YOLO and differentiate between anaemic and non-anaemic cases.

The dataset used in this study consists of images that are self-made, available through Google Search, and 710 conjunctiva images available online taken my medical professional from multiple selected hospitals in Ghana [14]. Furthermore, this study intends to expand its impact by creating an Android application with Flutter framework due its easy-to-learn widgets and suited for building small sized application [15]. Using smartphone cameras, this application will make advantage of the established YOLO and KNN approaches to deliver an intuitive user interface for real-rime anaemia detection. In resource-constrained areas, this kind of application might enable people to conduct early evaluations, enabling early diagnosis and immediate intervention [16].

The development of this application represents a significant step, as just a few studies have integrated smartphone technology with the latest version of YOLO and KNN to detect anaemia. This study broadens the literature by introducing how these methods implemented for practical approach that could enhance the accessibility of anaemia diagnosis.

II. METHOD

This study proposed four stages method for automated anaemia detection from conjunctiva images using a combination of YOLOv8 and KNN approaches. The process included data collection and preparation, model development, backend integration, and application development.

A. Data Collection and Preparation

There are two datasets collected in this study:

1) Dataset for YOLO model for Detecting Conjunctiva

The dataset included a wide range of eye images that showed conjunctiva and without. This dataset combined images that were taken with oneself and images that were downloaded from the internet. The self-generated images were taken in natural lighting with a smartphone's rear camera. In the meantime, Google's Image Search was used to find the online images when searching for terms like "eyes" or "eye's conjunctiva." By using data augmentation techniques, the problem of limited data which could lead to the model being over-fitting can be avoided [17][18]. Augmentation involved a series of transformations applied to the original dataset, including horizontal flipping, clockwise and counter-clockwise rotations of up to 30 degrees, and a final rotation of 180 degrees. After augmentation, the dataset increased to 480 images, which were divided into 240 images that showed conjunctiva and 240 images that did not. And for the segmentation process, 218 new images were added from the Roboflow dataset [19]. Dataset example can be seen on Figure 1 and Figure 2.

Fig. 1. Eye without conjunctiva

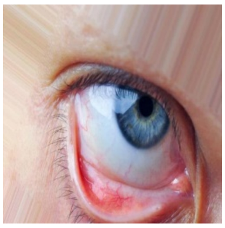

Fig. 2. Eye with conjunctiva

2) Dataset for KNN model for classifying Anaemia

The dataset for KNN model was collected from the Mendeley Dataset [14], consisting of 710 images of conjunctiva image that had been cropped leaving the ROI and divided into 2 classes with 424 images for anaemic person and 286 images for non-anaemic person. Every image also had been labelled on the image name based on the patient condition. The dataset was taken by laboratory personnel from 10 hospitals situated in Ghana. All images were captured in ambient natural light with a standard camera of 12MP. Additionally, to further enhance the dataset analysis, the light mean value was extracted from the CIE Lab color space for all images in the dataset. This process allowed to derive standard values indicating the brightness levels, distinguishing those images that is too bright or too dark. The mean brightness from all images ranged from 7.31 to 92.54 where value of 0 is the darkest value and 100 is the brightest. The variance of the brightness could be seen on Figure 3.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 2, Juni 2024, Pp. 798-805

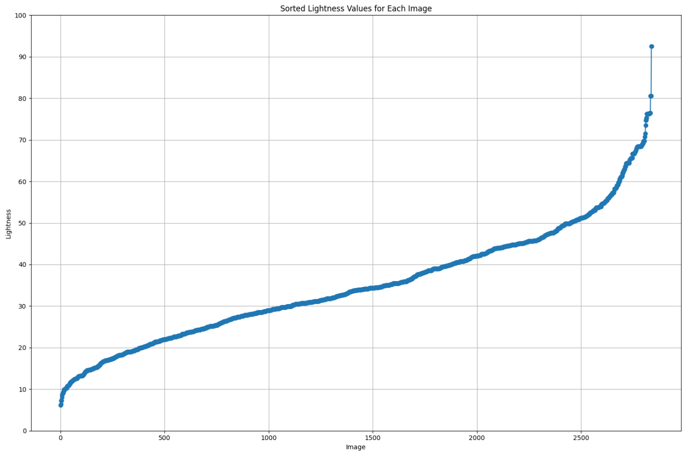

Fig. 3. The Brightness Graph of Each Image in The Dataset

All the images were augmented to avoid overfitting, resulting in 2840 images in total. Similar to the augmentation process implemented for the YOLO model, the augmentation strategy for this dataset remained consistent. The augmentation used horizontal flipping, clockwise and counter-clockwise rotations of 30 degrees, and 180 degrees rotation. An image example can be seen on Figure 4.


Fig. 4. Conjunctiva Image

B. Model Development

To achieve accurate anaemia detection from conjunctiva images, this study used a dual-phase methodology involving YOLOv8 and KNN.

1) YOLOv8 Development

The steps to extract final conjunctiva ROI within captured image involve detecting the eye, classifying whether the conjunctiva is present or not, and segmenting the conjunctiva to get the ROI. These crucial tasks were accomplished by employing YOLOv8, a robust deep learning model capable of performing object detection, object classification, and object segmentation [20]. The latest version of YOLO is used due to its ability to generate higher accuracy than its predecessor and easier-to-use package on Python. Pre-trained weights from YOLOv8 model were utilized, leveraging its powerful convolutional neural network (CNN) architecture. This architecture can extract intricate features from the image, including edges, textures, and color variants [21].

a. Eye Detection

Every picture was carefully labelled, with bounding boxes drawn precisely around the eye landmarks. The annotated dataset then was divided into training (70%), validating (20%), and testing (10%) to ensure robust model generalization [22]. Then YOLOv8 detection pre-trained weights were used to train the dataset for 50 epochs, creating the model to pinpoint the location of the eye. The metric used to determine how good the model to detect the eye was the Mean Average Precision (mAP) where it compared the labelled bounding box to the detected bounding box and returned the score. Higher the score was, more accurate the model would become.

b. Conjunctiva Classification

Images are tagged with two distinct classes: “has conjunctiva” and “no conjunctiva”. Similar to the phase of eye detection, the data was split into training, validating, and testing and trained for another 50 epochs training with YOLOv8 classification pre-trained model. The classification model performance was evaluated using several key metrics like the accuracy to assess the overall correctness of the conjunctiva classification model, precision that evaluated the accuracy of predictions specifically for the positive class (has_conjunctiva), recall or sensitivity that measured how good the model in identifying all actual “has_conjunctiva” samples among the total samples that labelled as “has_conjunctiva”, and the last one is F1-score as balanced evaluation between the recall and precision value.

c. Conjunctiva Segmentation

Only images with conjunctiva were used for segmentation training. First, every image was annotated using polygon shape tool to get the desired area around the conjunctiva. Next the data was split into three subsets and trained using 50 epochs of YOLOv8 segmentation pre-trained model. The final step was combining the segmentation model with cv2.contour library in order to crop the conjunctiva ROI. Object segmentation in YOLO used the same key metric with object detection to tell how good the model was by looking at the value of mAP.

2) KNN Development

Once the crucial model to extract conjunctiva with YOLOv8 is created, KNN takes center stage, as a way to detect the presence of anaemia. KNN operates by analyzing extracted features from the ROI. The *sklearn* library was used to implement the KNN. It will take k number of most similar neighbors (closest distance) and identified the similar class to get the final classification [23]. Two classes that will be used for KNN were “anaemia” and “non-anaemia”.

The dataset used in this development was the data sourced from the Ghanaian dataset [14]. Given the significant imbalance in the total data between anemic and non-anemic classes, solely relying on accuracy metric might be insufficient. Metrics like recall and precision was used to accurately assess the model's performance. Recall measures the ability to correctly identify anemic individuals among all actual anemic individuals, while precision evaluates the accuracy of anemia predictions, focusing on minimizing false positives. For each image, six key features were extracted: mean and standard deviation for each channel of the CIELAB color space (L, a, b). This color space could mimic how human perceive colors and ensures device independence to get consistency in analysis [24]. The neighborhood size (k) was determined by using k-fold-cross-validation to get a more balanced result [25]. By employing k-fold cross-validation, the impact of the specific data partitioning on model performance was mitigated. This technique provided a more robust evaluation, helping to ensure that the model's performance was representative across different subsets of the dataset [26][27].

C. Backend Integration

As mentioned before, this study aimed to develop a mobile android application for the user to have direct use to the anaemia detection model. In order to bridge the gap between the machine learning model and the application, Flask, a lightweight web framework and easy to use with python, was used. The backend was a combination of the YOLOv8 model and KNN model hosted on the Flask API server. The YOLOv8 model ensured that the conjunctiva was present and fall under the right category of brightness based on the used dataset. The model then extracted the conjunctiva from the submitted image. Then, the result of feature extraction was passed to the KNN model to get classified. All the test result such as color value and the final classification would be sent back to the user in JSON format.

D. Application Development and Testing

Flutter was used as the front-end mobile application that connected to the server where all the machine learning model are hosted. This framework was utilized due to its easy-to-use widget to create a wonderful design. The app allowed user to capture their conjunctiva using rear camera or front camera. The app displayed final result such as color value, the ROI image, and the final classification. To evaluate the performance of the application, a key performance indicator relied on the processing time from the moment the image was sent to the server until the response was received back by the application. The time taken for the image to be processed by the machine learning models hosted on the server was crucial in evaluating the application's responsiveness and efficiency.

E. System Architecture

The system architecture in Figure 5 shows the connection between all machine learning models up to the anaemia detection application by using Flask backend.

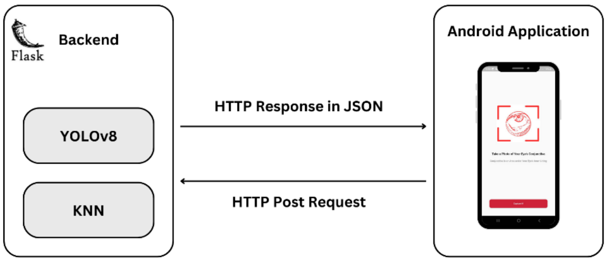

Fig. 5. System Architecture

III. Results and Discussion

A. Eye Detection Model

This model was trained, validated, and tested using 336, 96, 48 images respectively following 70%,20%,10% split data rule. The model was trained using 50 epochs and reached 99% mAP50. mAP is a metric to see how good the model is in object detection and object segmentation by calculating the difference between predicted Intersection over Union (IOU) to the actual IOU [28]. Out of the 48 eye images for testing (unseen data), the model was able to predict 48 eyes correctly (true positive), 3 incorrectly predicted as eye (false positive), 0 true negative data, and 0 false negative data. The total of correctly and incorrectly predicted data did not match to the initial testing data because the model can predict two or more objects in a single image. The training progress can be seen on Figure 6.

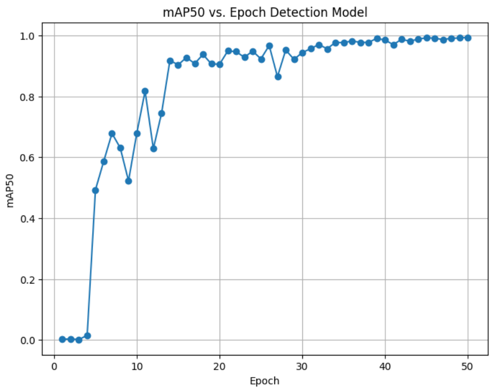

Fig. 6. Detection Model Training Progress

B. Conjunctiva Classification Model

The classification model was trained through 50 epochs. Even before the data was trained through all epochs, the model already reached the accuracy of 95% at epoch 18. There were 48 unseen data used to test the model. The data was divided into 24 data with conjunctiva and 24 without conjunctiva. From the 48-testing data, the model was able to predict 24 images with conjunctiva correctly (true positive), 22 images without conjunctiva correctly (true negative), 2 images with conjunctiva incorrectly (false positive), 0 images with no conjunctiva incorrectly (false negative). Utilizing this confusion matrix data, essential metrics were calculated and can accurately classify conjunctiva image with 100% recall, 92.31% precision, and 95.83% accuracy. Overall, the metrics suggested the effectiveness in classifying the conjunctiva by maintaining a low rate of false positives and false negatives. The training progress can be seen on Figure 7.

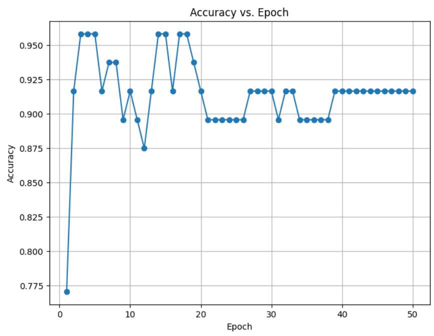

Fig. 7. Classification Model Training Progress

C. ROI Segmentation Model

The segmentation model was trained through 50 epochs process. There were 218 new images with conjunctiva added to the initial dataset. These images were taken from the roboflow site [19] totaling in 458 images. The segmentation model used 320 images for training, 91 images for validating, and 47 images for testing. The model reached mAP50 of 96% and out of 47 testing images, the model was able to locate the conjunctiva from 45 images correctly (true positive), 2 images as false negative and 5 images as false positive. The training progress can be seen on Figure 8.

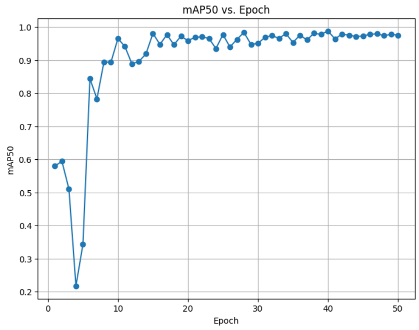

Fig. 8. Segmentation Model Training Progress

D. Anaemia Classifier Model

The dataset used for training the anaemia classifier totaling in 2840 images after augmentation consisted of 1144 conjunctiva images labelled as non-anaemia and 1696 conjunctiva images as anaemia. The dataset was divided with a 75% and 25% split into training and testing data respectively using *train_test_split()* function from sklearn library. K-fold-cross validation was used to find the best k for KNN. This study implemented 5- and 10-fold cross validation across k=1 to k=10. The result showed that the k with highest accuracy is k=1 with average accuracy more than 90%. But further research showed that model with k=1 while more accurate, it would lead to overfitting model because the model studied too well and would not see the underlying patterns from the data. To balance it, the higher k was needed. From Figure 9 we can see that k=5 was higher and still maintain the mean accuracy above 80% and the larger the k was, the more inaccurate the model became.

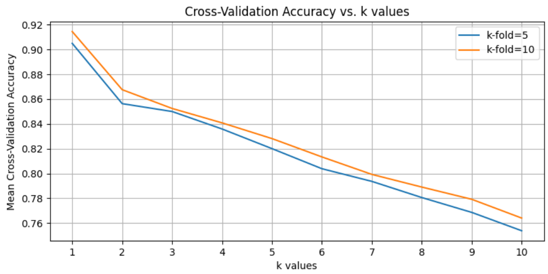

Fig. 9. 5 and 10-fold cross validation graph over k=1 to k=10

Out of 710 testing images, the classifier model was able to predict 200 anaemia images correctly (true positive), 390 non-anaemia images correctly (true negative), 74 anaemia images incorrectly (false positive), and 46 non-anaemia images incorrectly (false negative). The results obtained showed the model reached the accuracy of 83%, precision of 84.05 %, recall 89.44%, and F1-Score of 86.67%. As shown in Table 1, the comparison of the performance of KNN model in this study is pretty similar to another algorithm like the RestNet50 and Vit by Appiahene et al. [6] and even outperformed algorithms like Random Forest and Adaboost by Dimauro et al. [10].

TABLE I
SUMMARY OF RELATED WORKS PERFORMANCE AND METHOD USED

REFERENCE	ALGORITHM USED	PRECISION	RECALL	F1-Score
Proposed Study	KNN (K = 5)	84.05%	89.44%	86.67%
Appiahene et al. [6]	RestNet50	85.2%	83%	83.7%
Appiahene et al. [6]	ViT	83.3%	83.5%	83.33%
Dimauro et al. [10]	Random Forest	78%	73%	72.5%
Dimauro et al. [10]	Adaboost	83%	73%	74.7%

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 2, Juni 2024, Pp. 798-805

E. Mobile Application

The Flutter mobile application let users take pictures of their eyes from both the front and rear cameras of their phones. The application used an HTTP Post request to send the image, and it returned the results. In the case that the examination was successful, the ROI image, the classification (Anaemia or Non-Anaemia), and the conjunctiva color value would not be displayed. However, the user would not see the warning if there was an issue with the image, such as the absence of an eye or conjunctiva. The application took average of 5 seconds to process image when no eyes or conjunctiva was presented in the image and it took average of 7 seconds to process and get the classification result when the conjunctiva was presented. The application interface and application flowchart can be seen on Figure 10 and Figure 11.

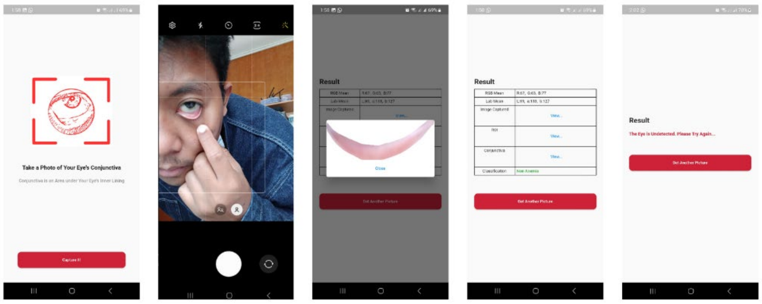

Fig. 10. Application Interface

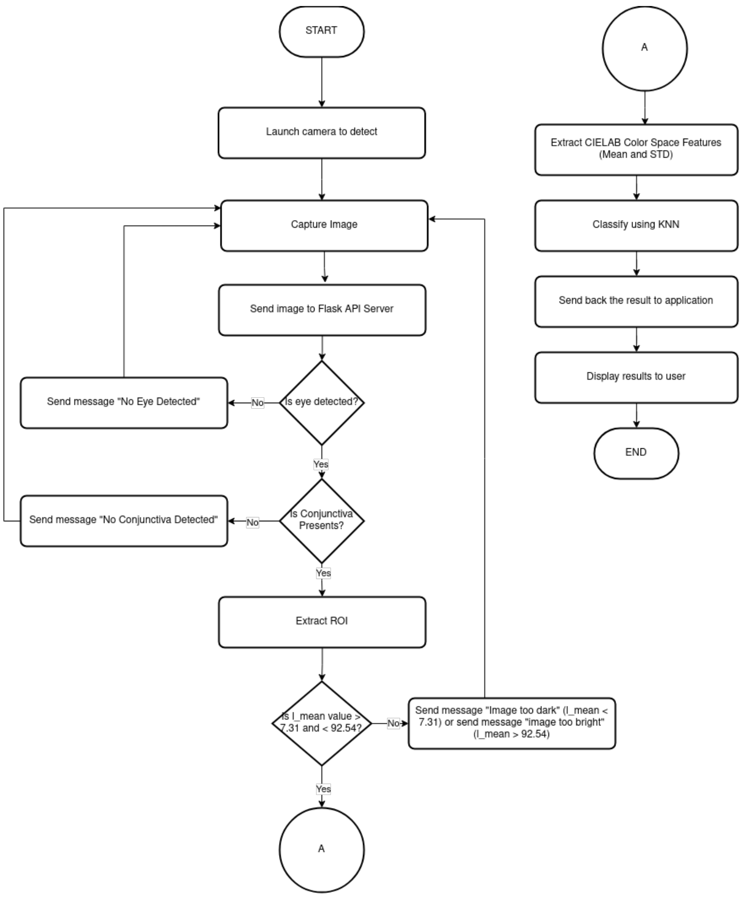

Fig. 11. Application Flowchart

IV. CONCLUSION

This study had developed an android application that utilized image processing and machine learning models to analyze conjunctiva image to predict anaemia. There were two machine learning framework used in this study, which were YOLO for conjunctiva extraction and KNN for anaemia classification. Our YOLO model was able to predict and locate the conjunctiva with mean average precision of 96% and the KNN with k=5 was able to predict anaemia with top accuracy of 83% and F1 score reached 86.67% with processing time of 5 to 7 seconds. The results indicated that the application is usable and might be used to replace the invasive method. Future research will focus on how to embed the machine learning model to the application itself to mitigate the slow internet connection problem.

REFERENCES

[1] World Health Organizatoin, "Anaemia." Accessed: Dec. 14, 2023. [Online]. Available: https://www.who.int/health-topics/anaemia
[2] J. P. Radich *et al.*, "Precision Medicine in Low- and Middle-Income Countries," *Annu. Rev. Pathol. Mech. Dis. 2022*, vol. 17, pp. 387–402, 2022, doi: 10.1146/annurev-pathol-042320.
[3] GBD 2021 Anaemia Collaborators, "Prevalence, years lived with disability, and trends in anaemia burden by severity and cause, 1990–2021: findings from the Global Burden of Disease Study 2021," *Lancet Haematol.*, vol. 10, no. 9, pp. e713–e734, Sep. 2023, doi: 10.1016/S2352-3026(23)00160-6.
[4] P. Appiahene *et al.*, "Application of ensemble models approach in anemia detection using images of the palpable palm," *Med. Nov. Technol. Devices*, vol. 20, Dec. 2023, doi: 10.1016/j.medntd.2023.100269.
[5] J. W. Asare, P. Appiahene, and E. T. Donkoh, "Detection of anaemia using medical images: A comparative study of machine learning algorithms – A systematic literature review," *Informatics in Medicine Unlocked*, vol. 40. Elsevier Ltd, Jan. 01, 2023. doi: 10.1016/j.imu.2023.101283.
[6] P. Appiahene, K. Chaturvedi, J. W. Asare, E. T. Donkoh, and M. Prasad, "CP-AnemiC: A conjunctival pallor dataset and benchmark for anemia detection in children," *Med. Nov. Technol. Devices*, vol. 18, Jun. 2023, doi: 10.1016/j.medntd.2023.100244.
[7] M. R. Rizvi, F. M. Alaskar, R. S. Albaradie, N. F. Rizvi, and K. Al-Abdulwahab, "A Novel Non-invasive Technique of Measuring Bilirubin Levels Using BiliCapture," *Oman Med. J.*, vol. 34, no. 1, pp. 26–33, Jan. 2019, doi: 10.5001/OMJ.2019.05.
[8] R. G. Mannino *et al.*, "Smartphone app for non-invasive detection of anemia using only patient-sourced photos," *Nat. Commun.*, vol. 9, no. 1, Dec. 2018, doi: 10.1038/s41467-018-07262-2.
[9] J. W. Asare, P. Appiahene, E. J. Arthur, S. Korankye, S. Afrifa, and E. T. Donkoh, "Detection of anemia using conjunctiva images: A smartphone application approach," *Med. Nov. Technol. Devices*, vol. 18, Jun. 2023, doi: 10.1016/j.medntd.2023.100237.
[10] G. Dimauro, M. G. Camporeale, A. Dipalma, A. Guarini, and R. Maglietta, "Anaemia detection based on sclera and blood vessel colour estimation," *Biomed. Signal Process. Control*, vol. 81, Mar. 2023, doi: 10.1016/j.bspc.2022.104489.
[11] W. Asare, P. Appiahene, E. T. Donkoh, and G. Dimauro, "Iron Deficiency Anemia Detection using Machine Learning Models: A Comparative Study of Fingernails, Palm and Conjunctiva of the Eye Images," 2023, doi: 10.22541/au.167570558.82410707/v1.
[12] M. Pulipalupula, S. Patlola, M. Nayaki, M. Yadlapati, J. Das, and B. R. Sanjeeva Reddy, "Object Detection using You only Look Once (YOLO) Algorithm in Convolution Neural Network (CNN)," in *2023 IEEE 8th International Conference for Convergence in Technology, I2CT 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/I2CT57861.2023.10126213.
[13] Y. Wang, Z. Pan, and J. Dong, "A new two-layer nearest neighbor selection method for kNN classifier," *Knowledge-Based Syst.*, vol. 235, p. 107604, Jan. 2022, doi: 10.1016/J.KNOSYS.2021.107604.
[14] Asare, Justice Williams; APPIAHENE, PETER; DONKOH, EMMANUEL, "CP-AnemiC (A Conjunctival Pallor) Dataset from Ghana," *Mendeley Data*, vol. V1, 2023, doi: 10.17632/m53vz6b7fx.1.
[15] Mohammed Kpannah Fahnbulleh and Xu SHUOBO, "Examining the Usage of Flutter to Design a Yatch in 3D," *Int. J. Res. Stud. Comput. Sci. Eng.*, vol. 8, no. 1, 2021, doi: 10.20431/2349-4859.0801001.
[16] I. J. Eliza, M. A. Urmi, M. T. T. Anan, M. T. H. Munim, F.-Z.-I. Galib, and A. B. M. A. Al Islam, "eDakterBari: A human-centered solution enabling online medical consultation and information dissemination for resource-constrained communities in Bangladesh," *Heliyon*, vol. 10, no. 1, p. e23100, Jan. 2024, doi: 10.1016/j.heliyon.2023.e23100.
[17] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0197-0.
[18] A. Mumuni and F. Mumuni, "Data augmentation: A comprehensive survey of modern approaches," *Array*, vol. 16, p. 100258, Dec. 2022, doi: 10.1016/J.ARRAY.2022.100258.
[19] Roboflow, "eye conjunctiva detector Image Dataset." Accessed: Dec. 19, 2023. [Online]. Available: https://universe.roboflow.com/eyeconjunctivadetector/eye-conjunctiva-detector
[20] encord, "YOLO models for Object Detection Explained [YOLOv8 Updated]." [Online]. Available: https://encord.com/blog/yolo-object-detection-guide/
[21] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimed. Tools Appl.*, vol. 82, no. 6, pp. 9243–9275, Mar. 2023, doi: 10.1007/s11042-022-13644-y.
[22] T. Kumar, A. Mileo, R. Brennan, and M. Bendechache, "Image Data Augmentation Approaches: A Comprehensive Survey and Future directions," Jan. 2023, [Online]. Available: http://arxiv.org/abs/2301.02830
[23] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, "Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction," *Sci. Rep.*, vol. 12, no. 1, pp. 1–11, 2022, doi: 10.1038/s41598-022-10358-x.
[24] A. Lee Kwan Yee *et al.*, "Preliminary analysis of rock mass weathering grade using image analysis of CIELAB color space with the validation of Schmidt hammer: A case study," *Phys. Chem. Earth, Parts A/B/C*, vol. 129, p. 103291, Feb. 2023, doi: 10.1016/J.PCE.2022.103291.
[25] I. K. Nti, O. Nyarko-Boateng, and J. Aning, "Performance of Machine Learning Algorithms with Different K Values in K-fold CrossValidation," *Int. J. Inf. Technol. Comput. Sci.*, vol. 13, no. 6, pp. 61–71, Dec. 2021, doi: 10.5815/ijitcs.2021.06.05.
[26] H. L. Vu, K. T. W. Ng, A. Richter, and C. An, "Analysis of input set characteristics and variances on k-fold cross validation for a Recurrent Neural Network model on waste disposal rate estimation," *J. Environ. Manage.*, vol. 311, p. 114869, Jun. 2022, doi: 10.1016/j.jenvman.2022.114869.
[27] M. Rafało, "Cross validation methods: Analysis based on diagnostics of thyroid cancer metastasis," *ICT Express*, vol. 8, no. 2, pp. 183–188, Jun. 2022, doi: 10.1016/j.icte.2021.05.001.
[28] R. Padilla, S. L. Netto, E. A. B. Da Silva, and S. L. Netto, "A Survey on Performance Metrics for Object-Detection Algorithms," 2020, doi: 10.1109/IWSSIP48289.2020.