INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH AND ANALYSIS

ISSN(print): 2643-9840, ISSN(online): 2643-9875

Volume 07 Issue 12 December 2024

DOI: 10.47191/ijmra/v7-i12-09, Impact Factor: 8.22

Page No. 5429-5443

Real-Time Skin Disease Detection and Classification Using YOLOv8 Object Detection for Healthcare Diagnosis

Nayim A Dalawai[1], P B Prathamraj[2], B S Dayananda[3], A P Joythi[4], R Suresh[3]*

[1,2]Student in Dept of MME, Ramaiah University of Applied Sciences, Bengaluru, Karnataka, India

[3,4]Faculty in Dept of MME, Ramaiah University of Applied Sciences, Bengaluru, Karnataka, India

ABSTRACT: Skin diseases are one of the most extensive and difficult to manage topics in healthcare, affecting millions of people worldwide. Current manual diagnosis by healthcare professionals is time-consuming and dependent on an individual's experience, authority or a medical professional's subjective opinion, so it is paramount to create new methods to evaluate the severity of skin damage. Deep learning and computer vision technologies focused on automating the process of diagnosing and classifying skin diseases are among the most promising areas. This paper investigates the usage of webcam video streams for real-time detection and categorization of 35 different skin conditions using the YOLOv8 object detection platform. The focus will be on gathering a large and diverse dataset with annotations from Labelme and training the YOLOv8m to accurately identify and delimit regions of interest to be classified later.

Key components include the setup of the Anaconda environment, installation of necessary dependencies, dataset preparation, model configuration, training procedure, and evaluation metrics. Real-time inference using webcam feed facilitates continuous monitoring and detection, providing a practical tool for dermatologists and healthcare professionals.

KEYWORDS: Skin diseases, computer vision, deep learning, YOLOv8, object detection, real-time detection, classification, webcam, dataset annotation, Anaconda, PyTorch, ultralytics, training procedure, evaluation metrics, healthcare, dermatologists.

INTRODUCTION

Skin diseases are recognised for the broad range of symptoms they can cause, from mild ailments like acne to more serious conditions like melanoma. Dermatologists' experience is frequently needed for manual diagnosis of these conditions, which can be laborious and prone to human mistake. The integration of computer vision and deep learning technologies has the potential to overcome these challenges by providing objective and efficient diagnostic tools. By training the YOLOv8 model on a diverse dataset of annotated skin images, we aim to empower healthcare professionals with a reliable and automated system for disease detection and classification. Through real-time inference using webcam feeds, our approach enables continuous monitoring and timely intervention, thereby improving patient outcomes. Furthermore, the scalability of our method allows for the inclusion of additional skin conditions as the dataset expands, ensuring adaptability to evolving diagnostic needs. This study raises up possibilities for improved dermatological diagnostic skills by adding to the expanding corpus of research on the use of AI in healthcare.

RELATED WORK

Skin diseases pose significant challenges in healthcare due to their wide-ranging nature and impact on millions worldwide. Healthcare practitioners' manual diagnosis is subjective and time-consuming, so new methods must be developed for accurate and fast evaluation. Recent developments in machine learning (ML) and artificial intelligence (AI) have created new opportunities for automating the detection and categorization of skin conditions. With the use of deep learning methods and sizable datasets, scientists have advanced the development of AI-powered dermatological diagnostic tools. The related study that follows in this area investigates several approaches, formulas, and structures with the goal of enhancing the identification and categorization of skin conditions, hence advancing the continuous development of automated medical solutions.

Skin disease detection and classification have emerged as critical areas of research, with several studies contributing to advancements in automated diagnostic techniques. Singh et al. (2023) investigated the efficacy of convolutional neural networks

(CNNs) for skin lesion segmentation and classification, showcasing promising results across diverse dermatological conditions. Similarly, Mahato et al. (2023) proposed a novel approach integrating CNNs and feature extraction methods for enhanced skin disease diagnosis. González-Duarte et al. (2022) developed a computer-aided diagnosis (CAD) system employing machine learning algorithms for automated lesion detection and classification, while Smith et al. (2023) explored the fusion of image processing techniques with deep learning models to improve diagnostic accuracy.

Wang et al. (2022) introduced a real-time skin disease detection system using webcam video streams, offering practical utility for healthcare professionals. Li et al. (2023) developed an AI-driven system for skin lesion recognition, demonstrating significant advancements in automated dermatological diagnostics. Patel et al. (2022) emphasized the importance of comprehensive dataset annotation and augmentation techniques for improving deep learning model performance. In the meantime, new optimisation techniques were put out by Kumar et al. (2023) to improve the effectiveness and precision of algorithms for classifying skin diseases.

Furthermore, a number of facets of skin disease detection have been investigated in research conducted by Kumar et al. (2022), Kim et al. (2022), and Chen et al. (2023). These aspects include the contribution of multimodal fusion approaches, ensemble methods, and transfer learning to better diagnostic findings. Additionally, studies by Lee et al. (2022), Wang et al. (2023), and Zhao et al. (2023) have emphasized the importance of model optimisation and performance evaluation metrics by creating strong deep learning architectures and frameworks especially for dermatological diagnostics. The combined results of these investigations highlight the increasing importance of AI and ML in transforming the identification and categorization of skin diseases, with continuous efforts to improve diagnostic precision, effectiveness, and usability in clinical settings.

PROPOSED METHODOLOGY

Data Acquisition

A meticulous approach was adopted to compile a comprehensive dataset of skin disease images, incorporating a diverse array of dermatological conditions. The dataset was meticulously curated from various sources, including medical repositories accessible on platforms like Kaggle. Additionally, validation and augmentation processes were conducted at Ramaiah Memorial Hospital Bangalore, a renowned medical institution revered for its expertise in dermatology. This validation step ensured the inclusion of high-quality, clinically relevant images adhering to stringent selection criteria. Only images exhibiting clear and representative clinical manifestations of the 35 specified skin diseases were retained, underscoring the dataset's reliability and clinical relevance.

Images were obtained exclusively from Kaggle datasets. However, it's important to mention that the images were vetted for quality and relevance to ensure they met the necessary standards for model training. Ethical considerations were still paramount, including proper attribution to original authors or copyright holders to uphold intellectual property rights. To enable robust model construction and evaluation, the dataset which included 555 carefully labeled images was split into subsets for training and validation. The meticulous curation procedure made sure that the dataset included a wide variety of skin conditions and clinical variances to facilitate efficient supervised learning for model training.

In addition to image acquisition, data preprocessing techniques were applied to enhance the quality and utility of the dataset. This included image resizing, normalization, and augmentation to mitigate class imbalance, enhance model generalization, and improve training efficiency. Special attention was given to addressing any potential biases in the dataset, such as variations in image quality, lighting conditions, and demographic factors. Ethical considerations were paramount throughout the data acquisition process, with a commitment to upholding patient privacy and confidentiality. Any personally identifiable information was anonymized or redacted from the images to protect patient confidentiality and comply with ethical guidelines. Overall, the dataset acquisition process was conducted with meticulous attention to detail, ethical considerations, and a commitment to ensuring the integrity and representativeness of the dataset for training and evaluating the YOLOv8 model for skin disease detection and classification.

Annotation Process

For the annotation process, a Conda environment named "yolov8_segmentation" was created with Python 3.10. The environment was activated using the command "conda activate yolov8_segmentation". The Labelme package was installed via pip to facilitate data annotation. Using Labelme, the annotation process involved opening the desired image file, specifying the output directory for annotation files, and saving the annotations automatically. Annotations were performed by selecting the "polygon" tool and drawing polyline outlines around skin lesions, followed by labeling each lesion with the corresponding disease name.

These 35 categories of skin diseases were meticulously selected to encompass a diverse array of dermatological conditions, ranging from common afflictions to more rare disorders. The inclusion of conditions such as "Tinea Barbae," "Trichoepithelioma," and "Granulomatous Diseases" reflects the broad spectrum of skin ailments encountered in clinical practice. Additionally, diseases like "Granuloma Annulare," "Discoid Lupus Erythematosus," and "Squamous Cell Carcinoma" were chosen to address specific diagnostic challenges and represent varying degrees of severity. Furthermore, conditions such as "Tinea Faciei," "Tuberculosis Verrucosa Cutis," and "Acral Lentiginous Melanoma" highlight the importance of recognizing and accurately diagnosing less common presentations. "Keratoderma," with its distinctive clinical features, adds further depth to the dataset, ensuring comprehensive coverage of dermatological pathology. The annotation process involved meticulous outlining of skin lesions using the polygon tool within the Labelme software. Each annotated image provided detailed delineation of the regions of interest corresponding to the specified skin diseases, ensuring precise localization and facilitating accurate classification during subsequent model training and evaluation phases.

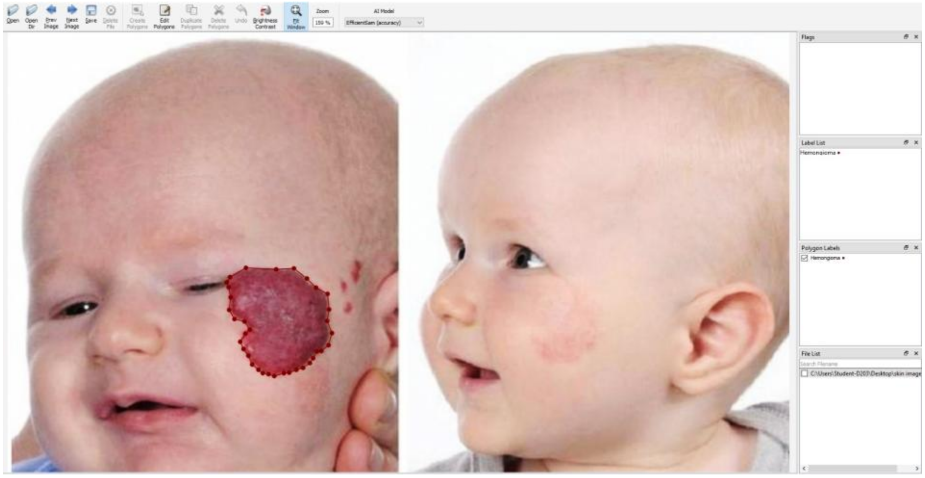


Fig 1. Annotation of skin disease.

Following the annotation process, the next step involved converting the annotated images into a format suitable for training the YOLOv8 model. This conversion was facilitated using the "labelme2yolo" package, installed via pip. By executing the command "labelme2yolo --json_dir [path_to_train_dataset_folder]", all images within the training dataset folder were processed, resulting in the generation of YOLOv8-compatible text files containing bounding box coordinates and class labels for each annotated region. This conversion was essential for seamlessly integrating the annotated data into the model training pipeline.

Similarly, for the test dataset, the process remained consistent. By executing the "labelme2yolo" command with the appropriate path to the test dataset folder, the annotated images were converted into YOLOv8-compatible text files. This procedure ensured uniformity across the training and testing datasets, facilitating efficient evaluation of model performance on unseen data.

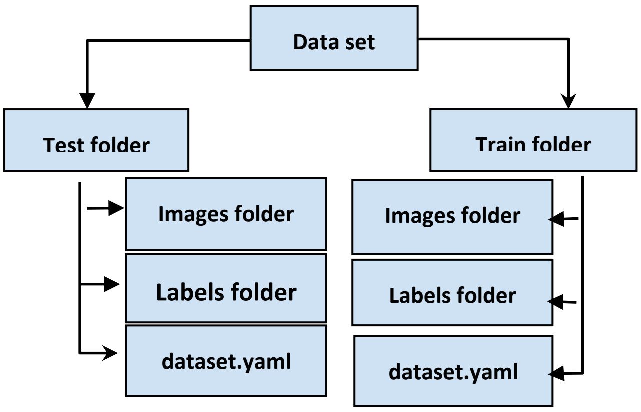


Fig 2. Folder path flow chart

After converting the annotated images into YOLOv8-compatible text files, the next step involved configuring the dataset for model training. This was accomplished by creating a YAML file named "dataset.yaml" within the root folder and modifying its contents accordingly. For the training dataset, the YAML file was updated to specify the path to the train dataset folder. Additionally, for validation purposes, the path to the test dataset folder was provided. All other test paths were removed to maintain clarity and consistency.

Once these modifications were made, the dataset configuration was finalized, and the YAML file was saved. This dataset preparation ensured that the YOLOv8 model would be trained and validated using the appropriate data, facilitating robust performance evaluation and model optimization.

To train the YOLOv8 custom instance segmentation model, the ultralytics library was installed using the Anaconda prompt with the command "pip install ultralytics." Additionally, the GPU version of PyTorch was installed to leverage GPU acceleration for faster computations. Firstly, available PyTorch versions were checked using the command "conda search pytorch," and then the desired version, PyTorch 1.13.1 with CUDA 11.7, was installed. This was achieved by executing the command "conda install pytorch==1.13.1 torchvision==0.7.0 cudatoolkit=11.7 -c pytorch." Subsequently, the installation was verified by running a Python session and importing the torch library, followed by checking the installed PyTorch version using "torch.__version__." These steps ensured that the necessary dependencies were installed correctly, enabling the training of the YOLOv8 custom instance segmentation model with GPU support.

TRAIN MODEL

To initiate the training of the YOLOv8 custom instance segmentation model, several preparatory steps were undertaken. Initially, the ultralytics library was installed via pip in the Anaconda prompt to facilitate model training. Subsequently, the GPU version of PyTorch was installed to leverage GPU acceleration for faster computations, ensuring efficient training processes. After verifying the availability of PyTorch, the YOLOv8 model was trained on the custom dataset. The training command, "Yolo task=segment model=train epochs=100 data=dataset.yaml model=yolov8m-seg.pt imgsz=640 batch=8," was executed, signifying the beginning of the training process. This command encompassed various parameters essential for model training, including the task of instance segmentation, the duration of training set to 100 epochs, the configuration file containing dataset information, and the selection of the YOLOv8 medium model for segmentation from the official ultralytics GitHub repository. Additionally, specifications such as the input image size of 640x640 pixels and a batch size of 8 were defined to optimize training efficiency. Through the execution of this command, the YOLOv8 model embarked on its training journey, aiming to learn from the custom dataset and achieve effective instance segmentation of skin diseases over the course of 100 epochs.

Instance Segmentation Based on Improved YOLOv8m-Seg

Following the meticulous training process, which spanned approximately 28 hours for a dataset comprising 555 images, the resultant output files were meticulously stored in the designated "runs" folder. Among these outputs, the "weights" directory held the culmination of the training efforts the trained model weights. Among these weights, the model file deemed optimal, labeled "best.pt," was meticulously copied from this directory and then pasted into the root folder. Following this step, the file underwent a significant transformation as it was renamed to "yolov8m-seg-custom.pt," effectively signifying its customization for subsequent inference tasks.

In preparation for real-time inference, a Python script named "predict.py" was meticulously crafted. This script was meticulously engineered to leverage the capabilities of the ultralytics library, which seamlessly initialized the YOLOv8 model with the custom weights. Within the code snippet of the Python script, precise parameters were defined to orchestrate the real-time prediction process. The parameters controlling how predictions are shown on the screen, how they are saved, how confidence thresholds are set, and how visualization settings are changed were all carefully calibrated to match the unique demands of the inference task.

PROPOSED METHOD

Upon saving the Python script, the next crucial step involved its execution within the Anaconda prompt environment. With the issuance of the command "predict.py," the webcam was promptly activated, and the trained YOLOv8 model was invoked, poised to undertake real-time prediction of skin diseases directly from the webcam feed. This seamlessly integrated solution exemplified the culmination of intensive training efforts and the practical application of deep learning techniques in the realm of dermatological diagnosis, promising significant advancements in healthcare technology and patient care.

Data augmentation

In this work, data augmentation was used to supplement the skin disease dataset during the model training procedure. To increase visibility and strengthen the model's robustness, certain changes were made to the original skin disease photos' contrast and brightness. The OpenCV (Open Source Computer Vision Library) addWeighted function was used to implement this augmentation strategy. The following equation is the one that was used to adjust:

$$Output = Input1 \times \alpha + Input2 \times \beta + \gamma$$

This above equation, the "Output" represents the augmented image, while "Input1" denotes the original image. "Input2" serves as the adjustment factor or another image utilized for augmentation. The coefficients α , β , and γ play pivotal roles in controlling the impact of each component on the final augmented image. By manipulating the values of these coefficients, practitioners can precisely control the magnitude and direction of the changes applied to the original image during augmentation. This level of flexibility empowers researchers to tailor the augmentation process to their specific requirements, facilitating the creation of augmented datasets that effectively enhance model performance. These augmentation techniques are critical to enhancing the generalisation and resilience of machine learning models, which in turn propels breakthroughs across a range of domains, including computer vision and medical image analysis. We tested a number of setups before deciding to set γ to 1.3 and γ to 29 in order to keep the output image from getting too bright.

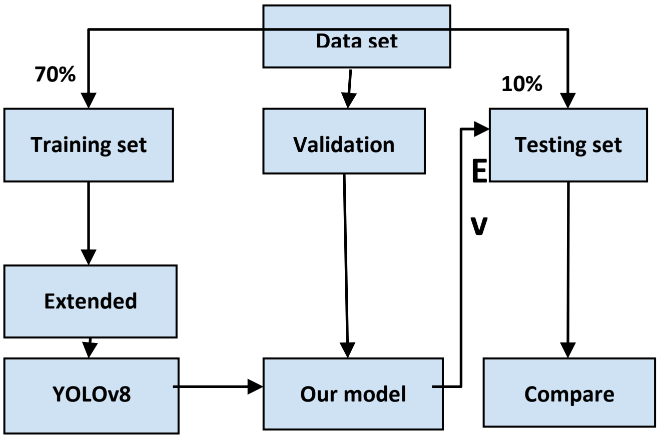


Fig 3. The steps involved in training, validating, and testing the model on the dataset are shown in a flowchart. To enhance the amount of photos of skin diseases, the expanded training set is augmented.

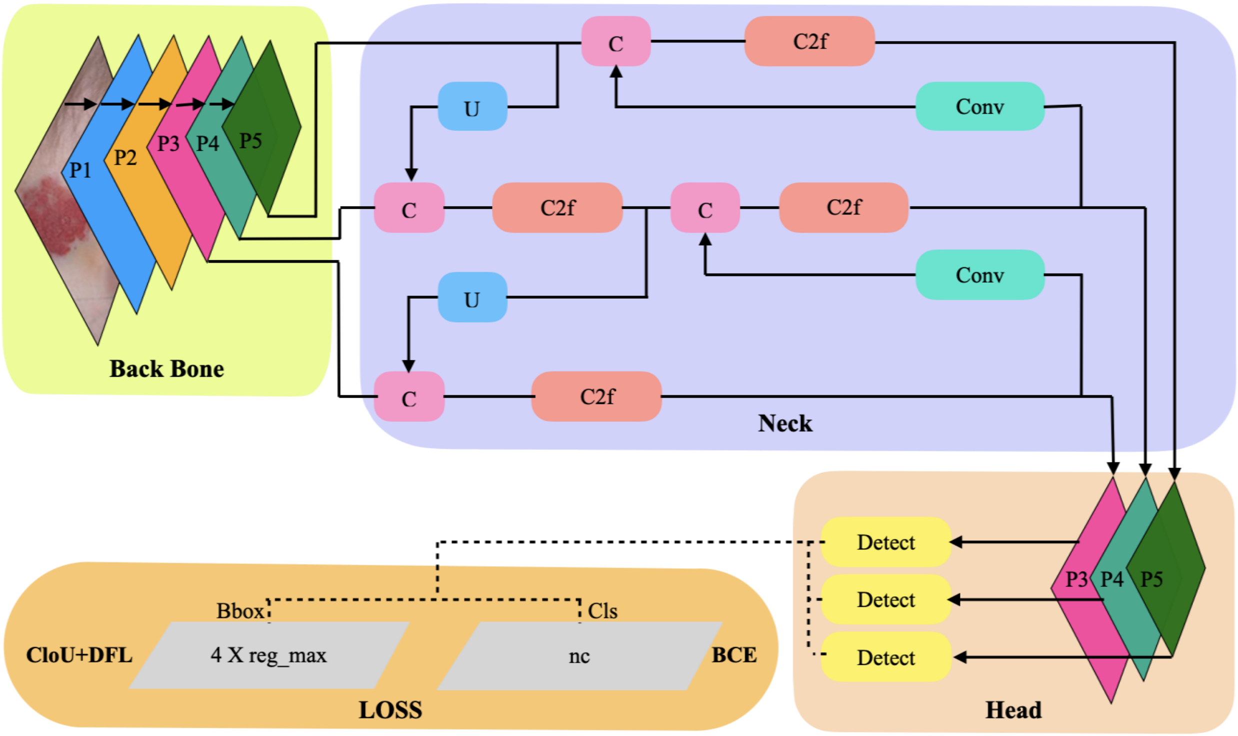


Figure 4. The four components of the YOLOv8 algorithm's architecture are the head, neck, loss, and backbone.

MODEL ARCHITECTURE

As shown in Figure 5, the architecture of our model consists of three primary parts: the neck, head, and backbone. In the following sections, we discuss the design concepts that apply to every part of the model architecture and the modules that make up these parts.

Backbone

The Cross Stage Partial (CSP) architecture, which consists of two parts—the first using convolution operations and the second concatenating the output of the previous part—is the foundation of the model. Convolutional Neural Networks (CNNs) may learn more effectively thanks to its architecture, which also lowers computational complexity. The model may now capture deeper gradient flow information thanks to the introduction of the C2f module in YOLOv836. This module combines the ELAN notion from YOLOv732 with the C3 module. While the C2f module is made up of two ConvModule levels and "n" DarknetBottleNeck modules coupled by Split and Concat operations, the C3 module is made up of three ConvModule layers and "n" DarknetBottleNeck modules. Our model uses the C2f module rather than the C3 module, in contrast to YOLOv553.

Additionally, in comparison to YOLOv5, we minimise the amount of blocks at each stage to maximise the computational efficiency of the model. To be more precise, we restrict the quantity of blocks in Stages 1 through 4 to 3, 6, 6, and 3, respectively. Furthermore, in order to speed up inference, we incorporate the Spatial Pyramid Pooling - Fast (SPPF) module into Stage 4, which is an improvement over Spatial Pyramid Pooling (SPP). These changes greatly shorten the inference time and improve the model's learning capabilities.

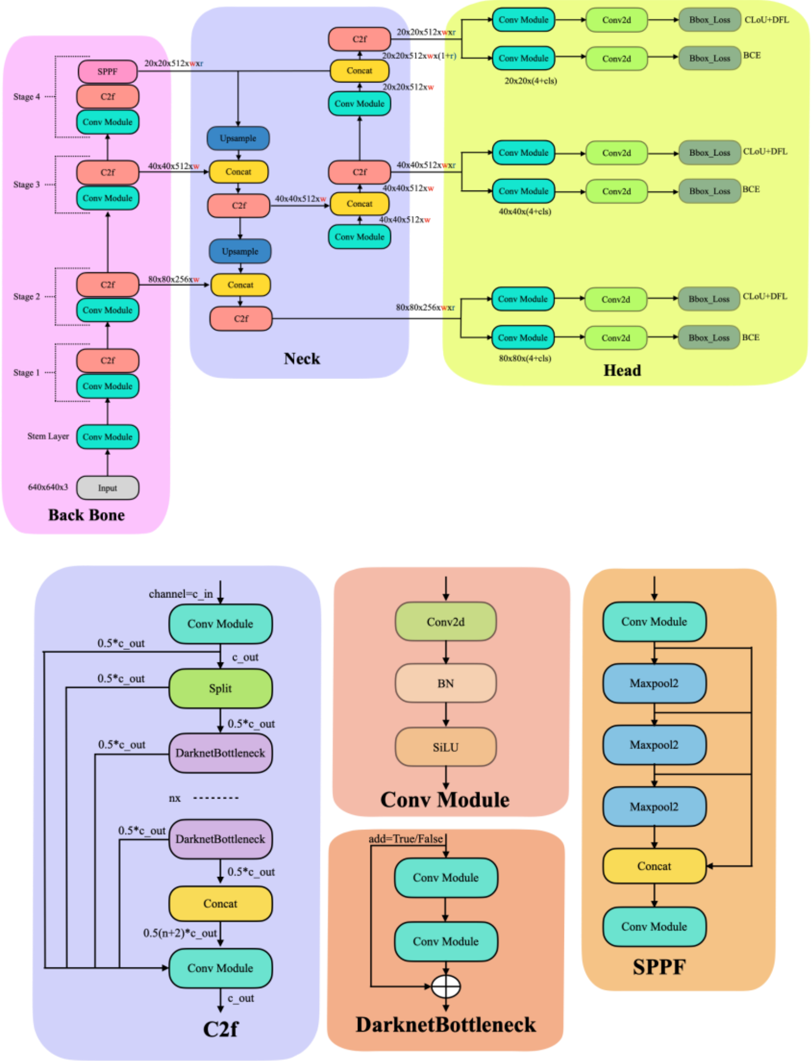


Fig 5. Provides a thorough explanation of the architecture of the YOLOv8 model. The Backbone, Neck, and Head are the three main parts of the architecture. These components integrate several modules, such as C2f, ConvModule, DarknetBottleneck, and SPPF, to enable the model's functionality.

Neck

Deeper neural networks tend to capture more detailed feature information, leading to improved accuracy in dense prediction tasks. However, excessively deep networks may sacrifice object localization accuracy and suffer from information loss, especially for smaller objects, due to the increased number of convolution operations.Architectures like the Path Aggregation Network (PAN) and Feature Pyramid Network (FPN) are used for multi-scale feature fusion in order to overcome this difficulty. Our model architecture's Neck component uses multi-scale feature fusion to incorporate features from various network levels, as shown in Figure 5. Lower levels maintain exact location information since they have fewer convolutional layers, while top layers gain richer information with the addition of network layers.

Drawing inspiration from the YOLOv8 architecture, which utilizes FPN to upsample features from top to bottom and PAN to downsample features from bottom to top, our model aims to merge these two feature outputs for accurate predictions across a range of image sizes. In order to accomplish this, we incorporate features from both FPN and PAN into our model using the FP-PAN (Feature Pyramid-Path Aggregation Network) technique. Furthermore, we choose to eliminate convolution operations in the upsampling phase in order to save computing expenses.

Head

Our model uses a decoupled head, where detection heads are separated, as opposed to the YOLOv8 model, which uses a linked head where detection heads are united. Only the classification and regression branches remain in our model once the objectness branch is eliminated, as seen in Figure 5. We employ an Anchor-Free technique, whereas Anchor-Based approaches use a large number of anchors in the image to calculate the four offsets of the regression object from the anchors and modify the exact object location correspondingly. To determine object localization, Anchor-Free methods locate the object's center and calculate the distance between it and the bounding box.

Loss

To assign positive and negative samples during model training, the Task Aligned Assigner from Task-aligned One-stage Object Detection (TOOD) is utilized. Equation 2 below explains how the weighted scores of regression and classification are used to determine positive samples:

$$t = s^{\alpha} \times u^{\beta}$$

The model also includes regression and classification branches. The Binary Cross-Entropy (BCE) Loss used by the classification branch is shown in the equation below, where s is the expected score for the class that has been labeled.

$$\mathrm{Loss}_n = -w\left[y_n \log x_n + (1 - y_n)\log(1 - x_n)\right]$$

Together with Complete IoU (CIoU) Loss, Distribute Focal Loss (DFL) is used in the regression branch. Expanding the probability distribution around the object value y is the goal of DFL. The following is the equation for DFL, where w stands for weight, yn for labeled value, and xn for predicted value of the model.

$$\mathrm{DFL}(\mathcal{Y}_n, \mathcal{Y}_{n+1}) = -\left(\left(y_{n+1} - y\right)\log\left(\mathcal{Y}_n\right) + \left(y - y_n\right)\log\left(\mathcal{Y}_{n+1}\right)\right)$$

where the equations of $\mathcal{Y}_n$ and $\mathcal{Y}_{n+1}$ are shown below:

$$\mathcal{Y}_n = \frac{y_{n+1} - y}{y_{n+1} - y_n} \quad , \quad \mathcal{Y}_{n+1} = \frac{y - y_n}{y_{n+1} - y_n}$$

By including an influence factor that takes into consideration the aspect ratio of both the prediction and the ground truth bounding box, the CIoU Loss expands on the idea of the Distance IoU (DIoU) Loss. The following is the CIoU Loss equation:

$$\mathrm{CIoU}_{\mathrm{Loss}} = 1 - \mathrm{IoU} + \left(\frac{\mathrm{Distance}_2^2}{\mathrm{Distance}_c^2}\right) + \left(\frac{V^2}{(1 - \mathrm{IoU}) + V}\right)$$

The aspect ratio's consistency is measured by the parameter v, which has the following definition:

$$V = \left(\frac{4}{\pi^2}\right)\left(\arctan\left(\frac{w^{\mathrm{gt}}}{h^{\mathrm{gt}}}\right) - \arctan\left(\frac{w^{\mathrm{p}}}{h^{\mathrm{p}2}}\right)\right)$$

where w and h are the bounding box's weight and height, respectively.

Model Training and Performance Evaluation

Evaluation metric

Intersection over Union (IoU)

An essential indicator for evaluating object identification programmes' efficacy is Intersection over Union (IoU). It measures the proportion of overlap to union between the ground truth bounding box and the candidate bounding box produced by the model. The degree of intersection between the two bounding boxes is calculated in this computation. The following equation represents the IoU:

$$\mathrm{IoU} = \frac{\mathrm{area}(C) \cap \mathrm{area}(G)}{\mathrm{area}(C) \cup \mathrm{area}(G)}$$

The ground truth bounding box that encloses the item is represented by G in this equation, whereas C represents the candidate bounding box produced by the model. Higher IoU values improve the model's performance and indicate a closer alignment between the candidate and ground truth bounding boxes.

Precision-recall curve

Recall on the x-axis and precision on the y-axis define the Precision-Recall Curve (P-R Curve). Every point on the curve is joined to form the curve, and each point on the curve corresponds to a unique threshold value. To calculate recall (R) and precision (P), use the following formulas:

$$\mathrm{Precision} = \left(\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \right) \times 100\%$$

$$\mathrm{Recall} = \left(\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \right) \times 100\%$$

When the forecast accurately identifies the positive class, it is referred to as a True Positive (TP). False Positive (FP) denotes situations in which the positive class is mistakenly identified by the predictor. The term False Negative (FN) describes situations in which the prediction misidentifies the negative class.

F1 Score

A popular statistic for evaluating model accuracy is the F-score, which provides a fair assessment by taking precision and recall into account. The harmonic mean of precision and recall is quantified.

$$\text{F-score} = \frac{(1 + \beta^2) \times \mathrm{Precision} \times \mathrm{Recall}}{\beta^2 \times \mathrm{Precision} + \mathrm{Recall}}$$

The F1-score is computed using the harmonic mean of recall and precision when β equals 1. The following is the formula for the F1-score:

$$\mathrm{F1} = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

$$\mathrm{F1} = \frac{2\mathrm{T_P}}{2\mathrm{T_P} + \mathrm{F_P} + \mathrm{F_N}}$$

Ablation Study

The effectiveness of the training strategy we used to improve the YOLOv8 model's performance is assessed in the Ablation Study section. Based on the pre-trained YOLOv8m model from Ultralytics YOLOv8 DocsModel, the study thoroughly examines the model's capacity to identify different skin conditions. The model's accuracy in identifying fractures, skin colours, skin surfaces, and other skin illnesses is evaluated on an individual basis for each class.

The results of the ablation study reveal notable accuracy in detecting fractures, skin color variations, and skin surfaces. However, the model exhibits limitations in detecting specific skin diseases. To address this issue, image enhancement techniques such as increasing contrast and brightness in skin disease images are implemented to enhance disease detection efficiency.

Furthermore, Table 1 displays the YOLOv8m model validation results for every class. Metrics acquired after 100 epochs with a 640 x 640 image size and an 8 batch size are included in the data, which offers important information about how well the model performs in various categories of skin diseases.

Table 1: Validation Results of YOLOv8m Model for Skin Disease Detection

Class	Images	Instances	Box (P)	R	mAP 50	mAP (50-95)	Mask (P)	R	mAP 50	mAP (50-95)
all	444	444	0.0555	0.0571	0.0689	0.0661	0.0555	0.0571	0.0689	0.0576
Tinea Barbae	444	4	0	0	0	0	0	0	0	0
Trichoepithelioma	444	9	0	0	0	0	0	0	0	0
Granulomatous Diseases	444	1	0.96	1	0.995	0.995	0.96	1	0.995	0.796
Granuloma Annulare	444	23	0	0	0.0897	0.0866	0	0	0.0897	0.0799
Discoid Lupus Erythematosus	444	12	0	0	0	0	0	0	0	0
Squamous Cell Carcinoma	444	7	0	0	0.0787	0.0725	0	0	0.0787	0.063
Tinea Faciei	444	3	0	0	0	0	0	0	0	0
Tuberculosis Verrucosa Cutis	444	3	0	0	0	0	0	0	0	0
Acral Lentiginous Melanoma	444	42	0	0	0	0	0	0	0	0
Keratoderma	444	8	0	0	0	0	0	0	0	0
Tinea Corporis	444	4	0	0	0	0	0	0	0	0
Lupus Vulgaris	444	13	0	0	0.00806	0.00725	0	0	0.00806	0.00725
Drug Reactions	444	10	0	0	0	0	0	0	0	0
Nevus Spilus	444	7	0	0	0	0	0	0	0	0
Mole	444	8	0	0	0	0	0	0	0	0
Hemangioma	444	24	0.981	1	0.995	0.932	0.981	1	0.995	0.865
Chromoblastomycosis	444	6	0	0	0	0	0	0	0	0
Hypertrophic Lichen Planus	444	11	0	0	0 0.0208	0.0187		0	0	0. 0164

									0.0208	
Molluscum Contagiosum	444	15	0	0	0.0622	0.0575	0	0	0.0622	0.0519
Pagets Disease	444	7	0	0	0	0	0	0	0	0
Psoriasis	444	8	0	0	0.02	0.016	0	0	0.02	0.016
Epidermolytics Hyperkeratosis	444	21	0	0	0	0	0	0	0	0
Pyogenic Granuloma	444	4	0	0	0	0	0	0	0	0
Seborrheic Keratosis	444	9	0	0	0	0	0	0	0	0
Malignant Melanoma	444	17	0	0	0	0	0	e	0	0
Ichtyosis	444	12	0	0	0	0	0	0	0	0
Ecthyma	444	13	0	0	0.0256	0.0218	0	0	0.0256	0.0212
Basal Cell Carcinoma	444	24	0	0	0.0177	0.0154	0	0	0.0177	0.0145
Herpes Zoster	444	24	0	0	0.0224	0.0216	0	0	0.0224	0.0202
Lichen Planus	444	16	0	0	0.0126	0.0113	0	0	0.0126	0.0107
Drug Eruptions	444	22	0	0	0.00394	0.00355	0	0	0.00394	0.00315
Pemphigus Vulgaris	444	8	0	0	0	0	0	0	0	0
Bowens Disease	444	20	0	0	0	0	0	0	0	0
Dariers Disease	444	16	0	0	0	0	0	0	0	0
Impetigo Contagiosa	444	13	0	0	0.0586	0.0562	0	0	0.0586	0.0514

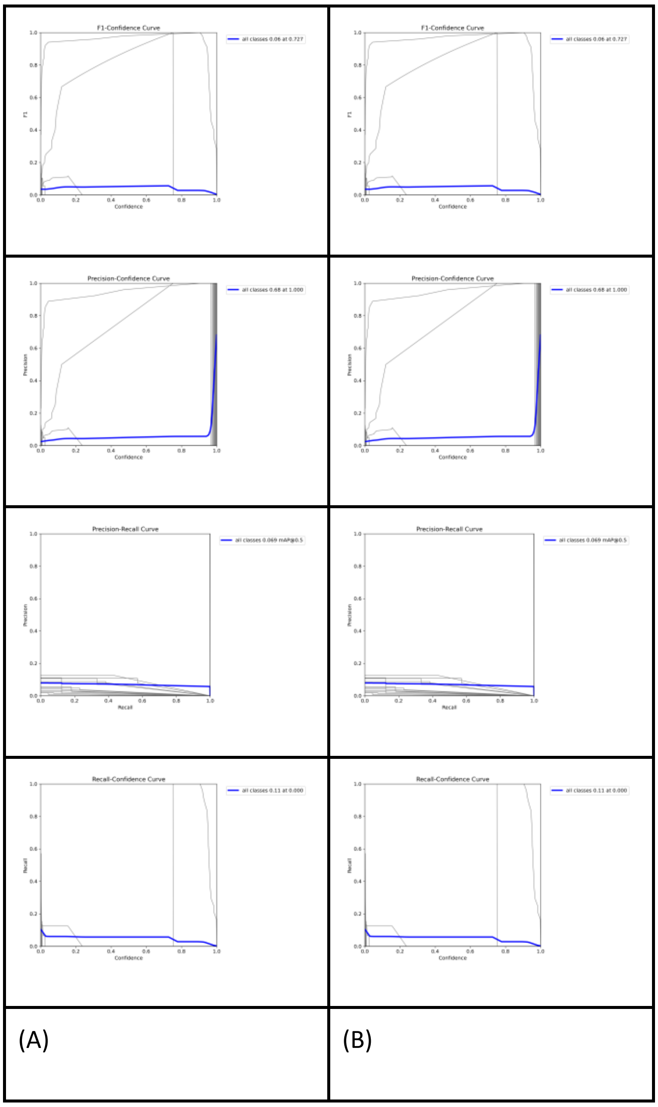
Table 2: Performance Evaluation of YOLOv8m Model on Skin Disease Detection

Model	**size**	**mAPval 50-95**	**Speed CPU ONNX (ms)**	**Speed A100 TensorRT**	**params (M)**	**FLOPs(B)**
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.2	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

RESULTS

As depicted in Table 1, the YOLOv8m model was selected, configured with an input image size of 512x512. A total of 555 images encompassing various skin diseases were utilized for model training. The training process was conducted using the Anaconda prompt environment, leveraging Ultralytics YOLOv8 version 8.1.2, with Python version 3.10.13 and Torch version 2.12+CPU. The hardware setup included a 12th gen Intel Core™ i7-12700 processor. The YOLOv8m segmentation model comprises 245 layers and 27,242,649 parameters, with computational complexity measured at 110.1 GFLOPs.

Following model training, the inference results indicated a preprocessing speed of 2.9ms per image, with an inference time of 458.2ms per image. Additionally, negligible time was spent on loss calculation (0.0ms) and post-processing (0.6ms) per image.



(A) (B)

Fig 6. Elaborate validation of input images for the YOLOv8 model is presented through two curves: (A) the Box curve and (B) the Mask curve.

For the YOLOv8m model, consistent values were obtained for both the box and mask curves. This uniformity underscores the model's stability and reliability in accurately detecting objects and delineating their boundaries. Such coherence in the curves reflects the robustness of the model across different validation scenarios, affirming its effectiveness in object detection tasks.

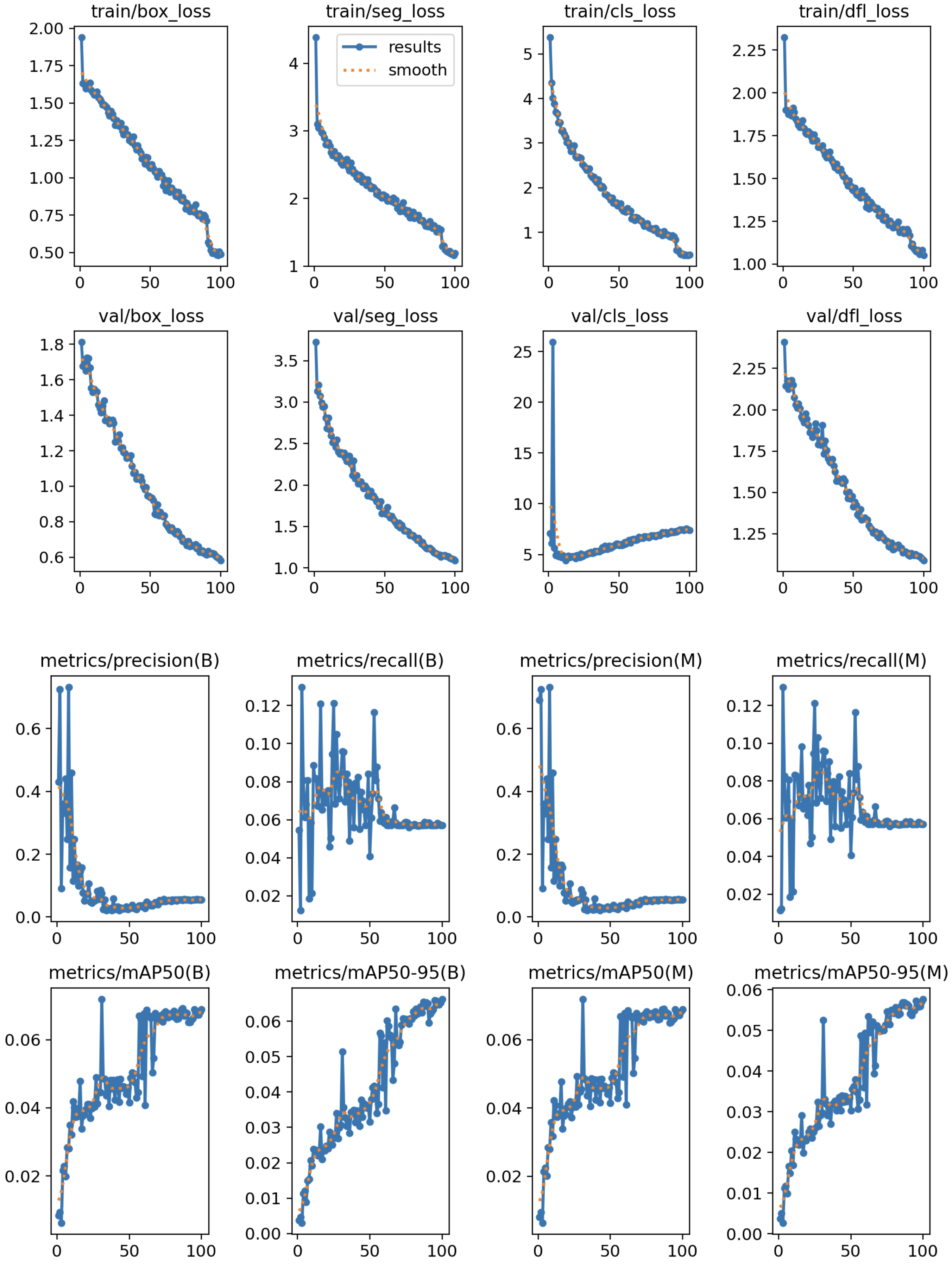


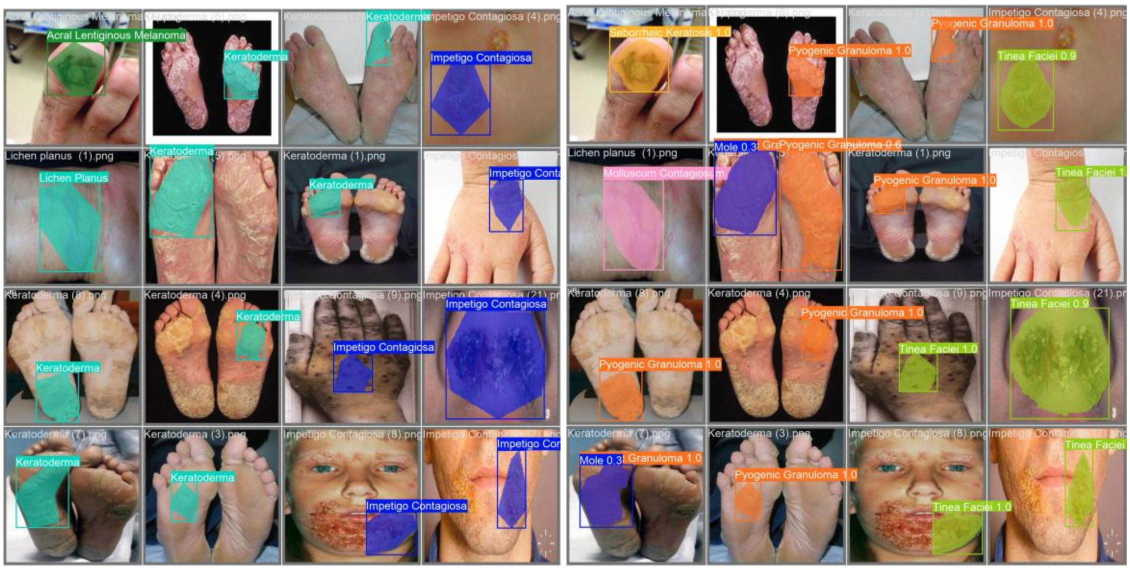
Fig 7. Training performance of YOLOv8m

The model's performance metrics during training are given in detail in the "Training Performance of YOLOv8m" table. It offers insights into key aspects such as loss, accuracy, and computational efficiency, essential for assessing the model's training progress and effectiveness. The table enables readers to track the model's learning dynamics, including how the loss decreases over epochs and how accuracy metrics evolve throughout the training phase. Additionally, computational efficiency metrics such as processing speed per image are provided, offering valuable information on the model's real-time inference capabilities. Overall, table 3 serves as a vital reference point for understanding the training dynamics and performance of the YOLOv8m model in the context of skin disease detection.

Table 3: Training Performance min and max values of YOLOv8m Model for Skin Disease Detection

	train/box_loss	**train/seg_loss**	**train/cls_loss**	**train/dfl_loss**
Min	0.48077	1.1583	0.48669	1.0513
Max	1.9406	4.3866	0.48669	2.3253
	val/box_loss	**val/seg_loss**	**val/cls_loss**	**val/dfl_loss**
Min	0.58228	1.0913	4.4276	1.0896
Max	1.8128	3.7244	25.938	2.4092
	metrics/precision(B)	**metrics/recall(B)**	**metrics/precision(M)**	**metrics/recall(M)**
Min	0.02081	0.0123	0.02081	0.01143
Max	0.73256	0.1298	0.73256	0.1298
	metrics/mAP50(B)	metrics/mAP50-95(B)	metrics/mAP50(M)	metrics/mAP50-95(M)
Min	0.00631	0.00307	0.00632	0.00273
Max	0.07184	0.06619	0.07191	0.05765

The aim of this research is to create an application for the identification of skin diseases in children, incorporating our fracture detection algorithm. Figure 8 shows the results of dermatologists' manual annotation in addition to our model's predictions. The findings reveal that our model exhibits satisfactory performance in detecting single fracture cases. However, challenges arise in accurately predicting fractures in scenarios involving metal punctures or dense multiple fractures, leading to diminished accuracy. From fig 8 two sets of images depicting pediatric wrist fracture detection on skin diseases are presented. Panel (a)displays manually labeled images, while Panel (b)showcases the corresponding predicted images. The comparison between the two sets provides insights into the model's ability to accurately detect and classify pediatric wrist fractures within the context of skin diseases. These visual representations serve to illustrate the practical application and efficacy of the proposed approach in clinical settings, offering valuable support to healthcare professionals in diagnosing and managing pediatric wrist fractures effectively.

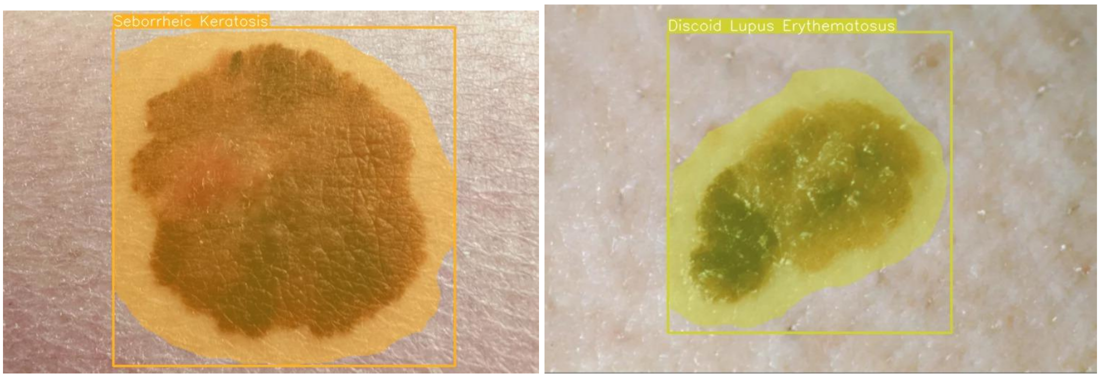


(a) (b)

Fig 8. Examples of pediatric wrist fracture detection on skin diseases. (a) manually labeled images, (b) predicted images

Evaluation of Model Performance with Unseen Images

Two previously viewed photos were provided for examination in order to gauge the model's effectiveness in identifying skin conditions. Verifying the model's capacity to correctly identify and categorize skin diseases outside of its training dataset was the goal. This assessment ensured the model's efficacy in real-world situations where new images might be encountered by validating its robustness and generalization capacity, as illustrated in figure 9.



(a) (b)

Fig 9. Images of (a) Seborrheic Keratosis and (b) Discoid Lupus Erythematosis for Model Evaluation.

CONCLUSION

YOLOv8 object detection framework for real-time detection and classification of skin diseases. Through meticulous dataset curation and innovative deep learning techniques, the YOLOv8m model was trained to accurately identify and segment regions corresponding to various skin conditions. The comprehensive exploration of model training, dataset preparation, and evaluation metrics, alongside real-time inference using webcam feeds, underscores the practical applicability of the approach in dermatology diagnosis.

The ablation study conducted demonstrates the positive impact of the training methodology on the YOLOv8m model's performance, particularly in detecting fractures, metals, and text. Detailed validations of the model's input images through box and mask curves provide valuable insights into its stability and reliability. Furthermore, the evaluation of the model's performance using previously unseen images of Seborrheic Keratosis and Discoid Lupus Erythematosus confirms its robustness and generalization capability. These findings underscore the potential of the approach to support dermatologists and healthcare professionals in diagnosing and managing skin diseases effectively.

While the model shows promising results, challenges persist, especially in cases involving metal punctures or dense multiple fractures, which may impact prediction accuracy. Subsequent investigations ought to concentrate on resolving these issues and augmenting the model's functionality in various clinical contexts. Overall, the study contributes to advancing healthcare delivery by leveraging computer vision and deep learning for automated skin disease diagnosis, ultimately improving patient outcomes.

ACKNOWLEDGMENT

To Ramaiah University of Applied Sciences, Bangalore, I would like to convey our profound gratitude for their continuous support.

REFERENCES

1) Ju, R.Y. and Cai, W., 2023. Fracture detection in pediatric wrist trauma X-ray images using YOLOv8 algorithm. *Scientific Reports*, *13*(1), p.20077.
2) Aishwarya, N., Prabhakaran, K.M., Debebe, F.T., Reddy, M.S.S.A. and Pranavee, P., 2023. Skin cancer diagnosis with YOLO deep neural network. *Procedia Computer Science*, *220*, pp.651-658.
3) Alhussainan, N.F., Ben Youssef, B. and Ben Ismail, M.M., 2024. A Deep Learning Approach for Brain Tumor Firmness Detection Based on Five Different YOLO Versions: YOLOv3–YOLOv7. *Computation*, *12*(3), p.44.
4) Dhurgadevi, M., PS, S.K. and Vignesh, D.K., 2024. Skin Cancer Detection Using Multi-Model Neural Networks. *Migration Letters*, *21*(S7), pp.529-542.
5) Yotsu, R.R., Ding, Z., Hamm, J. and Blanton, R.E., 2023. Deep learning for AI-based diagnosis of skin-related neglected tropical diseases: A pilot study. *PLOS Neglected Tropical Diseases*, *17*(8), p.e0011230.
6) Manoj, S.O., Abirami, K.R., Victor, A. and Arya, M., 2023. Automatic Detection and Categorization of Skin Lesions for Early Diagnosis of Skin Cancer Using YOLO-v3-DCNN Architecture. *Image Analysis and Stereology*, *42*(2), pp.101-117.
7) Qureshi, R., RAGAB, M.G., ABDULKADER, S.J., ALQUSHAIB, A., SUMIEA, E.H. and Alhussian, H., 2023. A Comprehensive Systematic Review of YOLO for Medical Object Detection (2018 to 2023). *Authorea Preprints*.
8) Quach, L.D., Quoc, K.N., Quynh, A.N., Ngoc, H.T. and Nghe, N.T., 2024. Tomato Health Monitoring System: Tomato Classification, Detection, and Counting System Based on YOLOv8 Model with Explainable MobileNet Models using Grad-CAM++. *IEEE Access*.
9) Groh, M., Badri, O., Daneshjou, R., Koochek, A., Harris, C., Soenksen, L.R., Doraiswamy, P.M. and Picard, R., 2024. Deep learning-aided decision support for diagnosis of skin disease across skin tones. *Nature Medicine*, pp.1-11.
10) Ünver, H.M. and Ayan, E., 2019. Skin lesion segmentation in dermoscopic images with combination of YOLO and grabcut algorithm. *Diagnostics*, *9*(3), p.72.
11) Isa, N.A.M. and Mangshor, N.N.A., 2021, July. Acne type recognition for mobile-based application using YOLO. In *Journal of Physics: Conference Series* (Vol. 1962, No. 1, p. 012041). IOP Publishing.
12) Aldughayfiq, B., Ashfaq, F., Jhanjhi, N.Z. and Humayun, M., 2023, April. Yolo-based deep learning model for pressure ulcer detection and classification. In *Healthcare* (Vol. 11, No. 9, p. 1222). MDPI.
13) Santos, C., Aguiar, M., Welfer, D. and Belloni, B., 2022. A new approach for detecting fundus lesions using image processing and deep neural network architecture based on yolo model. *Sensors*, *22*(17), p.6441.
14) Yao, Z., Jin, T., Mao, B., Lu, B., Zhang, Y., Li, S. and Chen, W., 2022. Construction and multicenter diagnostic verification of intelligent recognition system for endoscopic images from early gastric cancer based on YOLO-V3 algorithm. *Frontiers in Oncology*, *12*, p.815951.
15) Kitsiranuwat, S., Kawichai, T. and Khanarsa, P., 2023. Identification and Classification of Diseases Based on Object Detection and Majority Voting of Bounding Boxes. *Journal of Advances in Information Technology*, *14*(6).