LONTAR KOMPUTER VOL. 16, NO. 2 AUGUST 2025 **p-ISSN 2088-1541**
DOI : 10.24843/LKJTI.2025.v16.i2.p06 **e-ISSN 2541-5832**
Accredited Sinta 2 by RISTEKDIKTI Decree No. 158/E/KPT/2021

Classification of Acne Severity Using K-Nearest Neighbor (KNN) and Random Forest Methods

Gloria Flourin Maitimu[a1], Putu Harry Gunawan[a2], Muhammad Ilyas[b3]
[a]School of Computing, Telkom University, Bandung, Indonesia
[b]Mathematics Department, Kuwait College of Science and Technology, Kuwait City, Kuwait
[1]glorymaitimu@student.telkomuniversity.ac.id
[2]phngunawan@telkomuniversity.ac.id (Corresponding author)
[3]m.ilyas@kcst.edu.kw

Abstract

The development of machine learning technology, especially in dermatology, offers excellent opportunities for classifying and diagnosing skin conditions such as acne. This study aims to apply and compare two machine learning methods, K-Nearest Neighbors (KNN) and Random Forest methods, to classify acne severity into three levels: mild, moderate, and severe. The acne density and average confidence features were extracted from facial images using the YOLOv8 model based on acne bounding boxes. While the KNN model achieves 95% accuracy, the Random Forest model reaches 97%, indicating superior performance with excellent precision, recall, and F1-score values. These results suggest that the integration of YOLOv8-based feature extraction with the Random Forest classifier offers a promising and effective approach for automated acne severity classification in dermatological applications.

Keywords: *Acne, Machine Learning, K-Nearest Neighbors, Random Forest, YOLOv8*

1. Introduction

Machine learning technology is developing rapidly in the field of healthcare, particularly in dermatology. Previous studies [1,2] have employed various machine learning algorithms, including traditional methods such as Support Vector Machines (SVM) and more modern approaches like Convolutional Neural Networks (CNN), to detect, classify, and identify skin conditions, yielding reasonably accurate results.

Acne, or in medical terms called Acne Vulgaris, is one of the most common skin problems in humans, both men and women, especially in adolescence and young adulthood. The Indonesian Cosmetic Dermatology Study Group, or PERDOSKI, stated in 2017 that acne ranked third in the number of visitors to the Department of Dermatology and Venereology in clinics and skin hospitals. The highest prevalence rate occurs in women aged 14-17 with a percentage of 83-85% and in men aged 16-19 with a percentage of 95-100% [3,4,5]. Acne is caused by blockage of the oil glands due to excessive oil (sebum) production [6]. Acne not only affects physical appearance but also contributes to psychological distress, such as low self-esteem, depression, and anxiety [7]. Acne can be cured with medical help. Determining the severity of acne correctly and accurately is very important to help the medical team treat it appropriately.

Traditional machine learning techniques, such as K-Nearest Neighbors (KNN) and Random Forest, have been widely used to classify and diagnose various skin diseases, including acne, cherry angioma, melanoma, and psoriasis. These approaches often rely on automatic segmentation methods such as GrabCut to isolate skin lesions. However, previous studies reported limited classification accuracy—67.1% for KNN and 84.2% for Random Forest—and were generally unable to determine the severity level of skin conditions [8]. In contrast, recent studies have investigated object detection models such as YOLO (You Only Look Once), which have demonstrated promising results in acne detection through image analysis. For instance, Sankar et al. [9] utilized YOLOv8 in combination with StyleGAN2 to detect acne lesions on

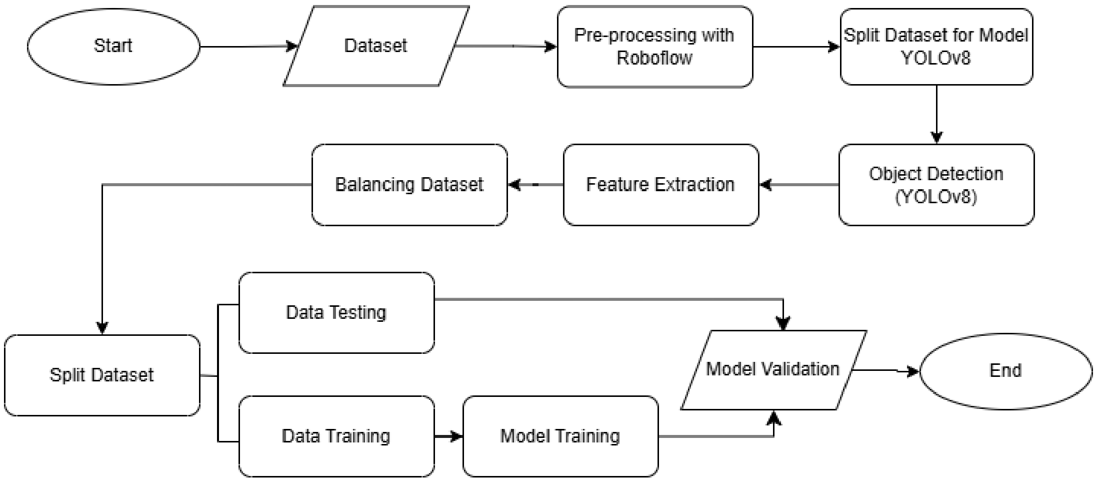

Figure 1. Research Flowchart

synthetic facial images, achieving a mean average precision (mAP) of 73.6%. This shows the potential of YOLOv8 in handling detection tasks even under data augmentation settings.

Unlike prior research that focused solely on lesion counts or object detection, this study introduces a novel approach by extracting acne density and average confidence from YOLOv8 detection outputs. These two features are then used as input variables for traditional machine learning classifiers (KNN and Random Forest). This dual-feature design provides a compact yet informative representation of acne severity and, to the best of our knowledge, has not been extensively explored in prior dermatological classification studies—particularly in the context of combining object detection outputs with classical machine learning methods.

Based on this foundation, this study proposes two machine learning methods—K-Nearest Neighbors (KNN) and Random Forest—to classify acne severity into three categories: mild, moderate, and severe. YOLOv8 is used for initial acne detection, while the classification is performed using the extracted features. The combination of YOLOv8's detection capability and the classification strength of KNN and Random Forest is expected to yield an effective and accurate acne severity classification system.

2. Research Methods

2.1. Research Structure

The research methodology followed a systematic pipeline beginning with dataset acquisition and preparation. The novelty of this study lies in the transformation of detection outputs (bounding boxes from YOLOv8) into structured numerical features—acne density and average confidence—which are then used as inputs for further classification. This intermediate representation bridges the gap between deep object detection and interpretable machine learning, providing a lightweight and effective solution for acne severity classification.

Acne images were obtained from the Kaggle platform and preprocessed using the Roboflow tool, involving three key steps: (1) manual annotation of acne lesions, (2) standardized image resizing, and (3) data augmentation to enhance dataset diversity and generalization.

For model development, we implemented a three-phase data partitioning strategy within the YOLOv8 framework:

1. Training data (60%): Used for model parameter optimization.
2. Validation data (20%): Monitored training progress and prevented overfitting.
3. Testing data (20%): Provided final performance evaluation.

After object detection using YOLOv8, two critical features were extracted: acne density and average confidence score. These features were then reformatted into a structured dataset, which served as the input for comparative classification models (KNN and Random Forest). Finally, both

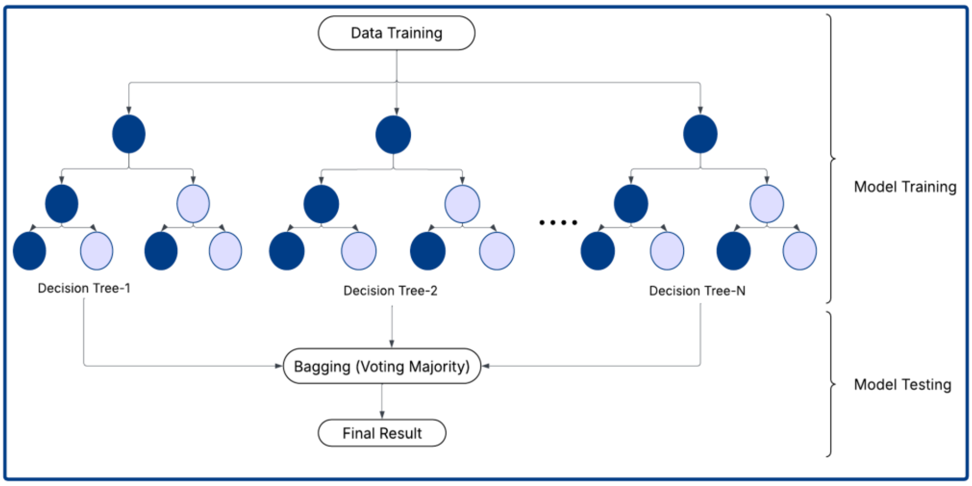

Figure 2. Random Forest

classification models were evaluated comprehensively using multiple performance metrics. Figure 1 illustrates the complete research workflow.

2.2. K-Nearest Neighbors

K-Nearest Neighbors, commonly abbreviated as KNN, is a classic machine learning algorithm used to solve classification problems. It is a type of supervised learning algorithm, meaning it classifies new data based on its ‘K’ closest neighbors, where the most frequently occurring class among these neighbors becomes the predicted label [10]. Finding a new instance's "K" nearest neighbors in the training dataset is the basic idea behind KNN. The distance between instances is typically calculated using the Euclidean distance formula [11, 12].

The KNN algorithm is also included in the nonparametric learning model. The advantage of KNN lies in its nonlinearity, as this algorithm can produce non-linear decision lines and is very flexible. This model is also known as a simple algorithm that is straightforward to implement and understand and has proven to be very effective for solving classification problems. But it has a high computational cost because it must first calculate the distance between x and all other data points and determine the K value [13, 14].

2.3. Random Forest

Random Forest is another classical machine learning algorithm that is widely used for both classification and regression tasks. This method is well-regarded for its high prediction accuracy, ease of implementation, and robust performance across various types of datasets. One of its key strengths lies in its ability to effectively handle high-dimensional data, making it a compelling alternative to other traditional algorithms [15]. The Random Forest technique operates by constructing an ensemble of decision trees. For each tree in the ensemble, a random subset of data points is drawn from the training dataset (typically using bootstrapping), and a random subset of features is selected at each node to determine the best split. This randomized process reduces the correlation between individual trees, thereby decreasing the risk of overfitting and enhancing the overall accuracy of the model [16, 17].

Figure 2 illustrates the general workflow of a Random Forest algorithm. During the training phase, the model builds multiple decision trees using randomly selected subsets of the training data. Each decision tree is constructed with randomly chosen features, allowing for diverse decision boundaries. In the testing phase, when the model receives new input data, each tree generates a prediction. These individual predictions are then aggregated through a majority voting process—commonly known as bagging—to determine the final output label. The final prediction corresponds to the class most frequently predicted by the ensemble of trees

Some of the advantages of random forest as an algorithm for combining multiple classifiers are that it is able to maximize the performance of the classification system as a whole by combining

p-ISSN 2088-1541
e-ISSN 2541-5832

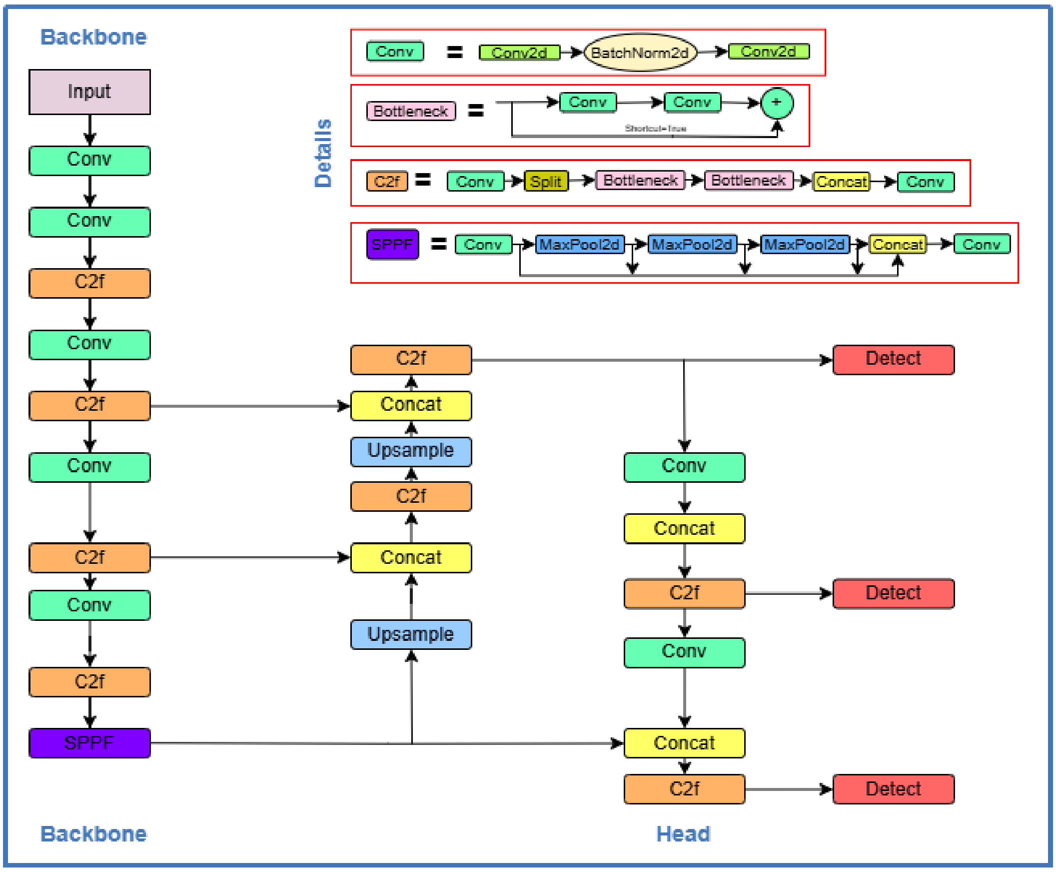

Figure 3. The Architecture of YOLOv8

the capabilities of several less robust classifiers so that the results are more optimal than a single classifier. In addition, this model is also effective for extensive data with complex parameters and supports parallel processing (multiple Random Forests). However, random forests also have the disadvantage that the performance of the algorithm will decrease when faced with data with many unbalanced feature dimensions because such data has irrelevant or redundant features [18].

2.4. YOLOv8

YOLOv8, introduced in 2020, is the eighth version of the YOLO (*You Only Look Once*) object detection algorithm. This version offers several improvements over its predecessors, including multi-scale prediction capabilities, an enhanced anchor system, and a more efficient backbone network [19]. One of its main advantages lies in its ability to achieve high detection performance while maintaining a compact model size, making it easier to deploy and integrate [19, 20]. YOLOv8 was specifically designed with computational efficiency in mind, allowing it to perform object detection at high frame rates, making it highly suitable for real-time applications [21].

The YOLOv8 architecture shown in Figure 3 consists of three main components, namely the backbone, neck, and head, which work in concert to process the input image in the object detection task. In this study, YOLOv8 is used to detect acne areas in facial images as it has a good balance between computational speed and detection accuracy. Despite the advent of newer object detection models, YOLOv8 remains a relevant choice due to its real-time detection capability, lightweight architecture, and ease of implementation, especially in resource-constrained environments. This study leverages YOLOv8's output—acne density and average confidence—without modifying its original architecture as input features for the next classification stage using traditional machine learning algorithms such as K-Nearest Neighbors and Random Forest.

Figure 4. Examples of Acne Image Annotations from the Dataset

3. Result and Discussion

3.1. Dataset

This research uses a dataset in the form of a set of acne images obtained from the Kaggle website. It consists of 2154 images with various levels of severity and different image sizes and rotations. After collection, the dataset is labelled or annotated as a bounding box on the acne image to define the object to be detected. This labelling helps ensure that each acne object in the image is correctly marked and given the appropriate class.

Figure 4 presents examples of annotated images from the dataset. Each bounding box highlights the area of acne detected on the facial region, categorized based on severity levels. These samples illustrate the variations in skin appearance and labeling outcomes across different individuals. The process of labeling and sharing data is done using the Roboflow platform, so that the dataset is divided into three categories, namely training data, testing data and validation data with each proportion of 60% for training data and 20% for each testing data and validation data. Labelling is divided into three severity levels: mild levels marked with 'Mild', moderate severity levels marked with 'Moderate' and severe severity levels marked with 'Severe' as in Table 1. After labelling, the dataset is exported into YOLOv8 format so that the model can perform the training process to detect acne objects in the image.

Table 1. Dataset

Class Acne	Data Splitting			Total Data
	Training	Testing	Validation	
Mild	318	172	153	643
Moderate	533	168	217	918
Severe	597	180	184	961

However, the dataset does not include demographic annotations such as skin tone, ethnicity, or age group of the individuals. Additionally, while the images vary in size, orientation, and lighting conditions, there is no standardized distribution of environmental backgrounds. These limitations may affect the model's ability to generalize to diverse clinical populations. Therefore, this study acknowledges the lack of demographic diversity in the dataset as a factor that may limit the applicability of the results in real-world dermatological settings.

3.2. YOLOv8 Implementation

In this study, the YOLOv8 model plays the role of detecting acne objects that provide information about the level of confidence and the location of acne in the image. After labelling the Roboflow application, the dataset will be used to train the YOLOv8 model for 100 epochs with an image size of 420x420 pixels. Training the YOLOv8 model enables accurate acne detection in facial images. After the training process is complete, the model with the best performance will be validated using validation data. To further validate the effectiveness of the YOLOv8 model, visual detection results with bounding boxes are presented in Figure 5, and performance metrics are detailed in Table 2.

Table 2. YOLOv8 Performance Matrics

Class	Images	Instance	(P)	(R)	mAP50	mAP50-95
All	546	680	0.539	0.68	0.576	0.34
Mild	130	174	0.462	0.508	0.416	0.333
Moderate	231	262	0.518	0.722	0.545	0.403
Severe	218	244	0.638	0.809	0.766	0.583

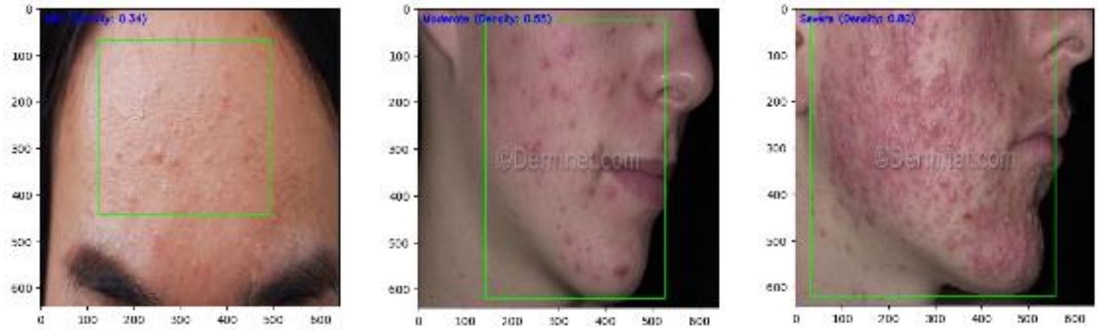

Figure 5. Acne Detection Using YOLOv8

Table 2 shows the metric results of the YOLOv8 model by providing the precision, recall, and mAP metric values used to measure the performance and accuracy of the model in detecting acne. The Precision (P) column indicates the precision value for all classes, quantifying the ratio of correct positive detections to the total detections made by the model. The Recall (R) column is the recall value for all classes, evaluating the model's efficacy in object detection; specifically, it indicates the proportion of actual pimples in the image that the model successfully identifies. And, mAP50 (Mean Average Precision at IoU=0.50) column is a more comprehensive combined metric, calculating the average precision across acne classes (mild, moderate, severe) with an overlap threshold (Intersection over Union/IoU) of 50%. The table shows that the model successfully detects acne, with fairly good performance at moderate and severe severity but low metrics at mild severity.

Although the YOLOv8 model has been trained to detect acne based on severity (mild, moderate, severe), its detection results are limited to per-object predictions in the form of bounding boxes and confidence values, without providing an aggregated assessment at the overall image level. Therefore, the detection labels from YOLOv8 are not used directly as the final output of the classification. As an alternative, this study extracted numerical features - namely acne density and average confidence - from all detected acne objects, which were then used as inputs for further classification processes using traditional machine learning algorithms such as K-Nearest Neighbors (KNN) and Random Forest. Although the evaluation results of YOLOv8 in this study showed moderate performance, with a precision value of 0.539 and mAP50 of 0.576, this model was still chosen because of its lightweight architecture, real-time inference capability, and ease of integration with two-stage classification systems. The utilization of YOLOv8 as an efficient feature extraction tool makes it a great choice to support a more structured and thorough acne severity classification.

3.3. Feature Extraction

The main features used from the YOLOv8 model detection results in this study are acne density and the average confidence level of the model for each acne detection. Acne density is the result of dividing the total acne area by the face image's total area. Meanwhile, the average confidence is obtained by summing all acne confidence levels and dividing them by the number of acne detected. There are also labels used to divide the severity of acne into three categories, namely mild severity, moderate severity, and severe severity, which is determined based on the density value as follows :

- **Mild** : Density < 0.4
- **Moderate**: 0.4 ≤ Density < 0.7

- **Severe**: Density ≥ 0.7

After both features are extracted, the extraction results are compiled into a numerical dataset that is used as input for the classification model. In contrast, the acne severity label must use an encoding technique to be converted into a numeric format to be processed by the machine learning model, namely KNN and Random Forest. Before training the two models, the data is then divided into training data and testing data with a proportion of 80% and 20%, which is helpful for the model to learn input mapping effectively and produce quality output [22]. Furthermore, when the training data experiences an imbalance in the number of samples, the model will find it difficult to detect the minority class, so the accuracy in predicting objects will be low [23]. One of the oversampling methods used to balance the number of samples between classes is the Synthetic Minority Oversampling Technique method or usually abbreviated as SMOTE. This method synthesizes new samples from the minority class so that the dataset becomes balanced [24]. Table 3 shows that the class 'Mild' is a minority class in the data. Therefore, the SMOTE method is used to balance the training data in this study.

Table 3. Class Distribution Before and After SMOTE

Class Acne	Before Smote	After Smote
Mild	85	133
Moderate	116	133
Severe	133	133

While SMOTE is often associated with the risk of overfitting-especially in image-based data-its application in this study is limited to low-dimensional numerical features generated from YOLOv8 detection, rather than high-dimensional raw image data. This makes the use of SMOTE more appropriate in this context as it reduces noise and maintains feature integrity. Moreover, by applying SMOTE to predefined statistical features such as pimple density and average confidence, the oversampling process can improve the class representation without compromising the reliability and interpretability of the dataset. Therefore, SMOTE is considered a suitable and effective technique in addressing class imbalance in this study.

3.4. Models Implementation

The classification models used in this research are K-Nearest Neighbors (KNN) and Random Forest. After balancing the dataset using the SMOTE method, the model performs a training process to predict the severity of acne based on the features that have been extracted. KNN and Random Forest models perform the training process using oversampled training data with parameters that have been selected to optimize the performance of each model in classifying acne severity. The parameters used in the model training process are summarized in Table 4. Both models are trained to learn the patterns and characteristics of objects well so that the KNN and Random Forest models can classify acne based on its severity.

Table 4. The Parameters Selected for Model Training

Model	Parameters	Comment
KNN	n_neighbors = 3	The number of nearest neighbors
Random Forest	n_estimators = 100	The number of decision trees to be built in the ensemble.
	random_state = 42	Ensure consistent results by setting the random generator seed.
	max_depth = 10	Limiting the maximum depth of each tree to avoid overfitting.

The use of the parameter value K = 3 in the KNN model and 100 decision trees in the Random Forest model has a significant impact on the characteristics and performance of the model. In the KNN model, if the K value is increased, the model can be more stable, but it can also cause

 p-ISSN 2088-1541
e-ISSN 2541-5832

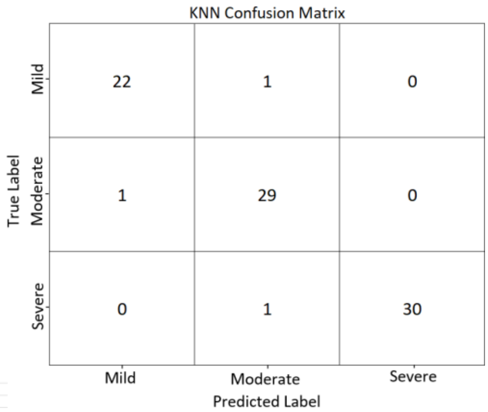

Figure 6. KNN Performance Metrics

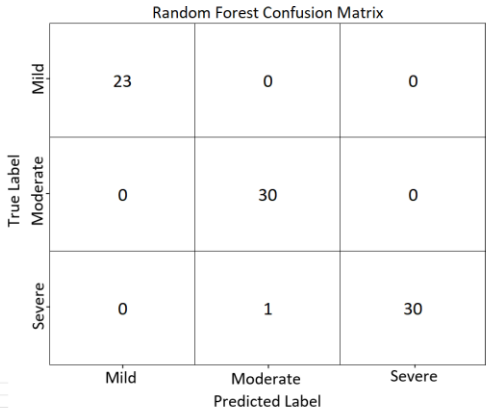

Figure 7. RF Performance Metrics

underfitting, and the ability to capture important local patterns is reduced. Conversely, if the K value is reduced, the model will be sensitive and cause overfitting. For the Random Forest model, if you increase the number of trees, the accuracy and stability of the model will increase, but the computational cost will be greater. Conversely, reducing the number of trees will speed up the training and prediction process, but the performance of the model will decrease.

3.5. Models Performance

After the models are built and trained, the KNN and RF models will be evaluated to ensure that both models can classify acne severity well, so that the performance between the two models can be compared. This research conducts an evaluation process using a confusion matrix, which consists of four main elements, namely [25, 26]:

- True Positive (TP) refers to instances where the model correctly predicts the positive class, meaning both the predicted label and the actual class label are positive. This indicates the model successfully identified a relevant or target case.
- True Negative (TN) occurs when the model correctly predicts the negative class, where both the predicted outcome and the ground truth are negative. This demonstrates the model’s ability to accurately recognize non-target cases.
- False Positive (FP) is recorded when the model incorrectly predicts a positive outcome for a sample that actually belongs to the negative class. Such errors can lead to false alarms or unwarranted actions in critical applications.
- False Negative (FN) arises when the model fails to detect a positive case, predicting it as negative instead. This type of error is particularly concerning in scenarios where missing a positive case carries serious consequences, such as disease detection or security alerts.

These four main elements are used to calculate the accuracy, precision, recall, and F1 score metrics of each model. The formulas for each metric can be found in several references, for example [21] and [27, 28, 29, 30].

Figure 6 is a Confusion Matrix image for the KNN model showing the prediction results for three acne classes, namely ’Mild’, ’Moderate’ and ’Severe’. Of the total 23 samples of the ’Mild’ class, there is 1 sample that is predicted incorrectly and classified as the ’Moderate’ class. For the ’Moderate’ class, there is also 1 sample is predicted incorrectly and classifies the ’Mild’ class out of 30 samples. The ’Severe’ class contains 31 samples, and 30 samples are predicted correctly, while 1 sample is incorrectly classified as ’Moderate’. Compared to Figure 5, the confusion matrix of the Random Forest model, the image shows very accurate prediction results with only one small error in the ’Severe’ class. The ’Mild’ and ’Moderate’ classes are predicted correctly without error, with each class containing 23 and 30 samples. In the ’Severe’ class, there are 31 samples, and only 1 sample is incorrectly classified as ’Moderate’. This shows that the Random Forest

LONTAR KOMPUTER VOL. 16, NO. 2 AUGUST 2025 **p-ISSN 2088-1541**
DOI : 10.24843/LKJTI.2025.v16.i2.p06 **e-ISSN 2541-5832**
Accredited Sinta 2 by RISTEKDIKTI Decree No. 158/E/KPT/2021

model is more consistent in correctly classifying each class, especially in the ’Mild’ and ’Moderate’ classes.

Table 5. KNN Evaluation

Class Acne	Precision	Recall	F1-Score	Accuracy
Mild	0.96	0.96	0.96	0.94
Moderate	0.94	0.97	0.95	0.95
Severe	0.99	0.97	0.98	0.96

Table 5 shows the results of the evaluation of the KNN model, which results in the values of the four main metrics of each acne class. In the precision metric, the ’Severe’ class has the highest value of 0.99. The ’Mild’ and ’Moderate’ classes also have high precision values, 0.96 and 0.94, respectively, meaning that the model accurately predicts these classes without many false positives. For the recall matrix, the KNN model shows the best values in the ’Moderate’ and ’Severe’ classes, which are 0.97 and the ’Mild’ class, which shows a value of 0.96. This indicates that the KNN model is quite good at recognising the three acne classes. Furthermore, in the F1-Score value, which is a harmonisation between the recall and precision values, the highest value occurs in the ’Severe’ class, which is 0.98, and the ’Mild’ and ’Moderate’ classes, respectively, are 0.96 and 0.95. The high F1-score value in the KNN model shows a good balance between accuracy and sensitivity. Each acne class has a fairly high accuracy value, namely 0.94 for the ’Mild’ class, 0.95 for the ’Moderate’ class and 0.96 for the ’Severe’ class.

Table 6. Random Forest Evaluation

Class Acne	Precision	Recall	F1-Score	Accuracy
Mild	0.98	0.98	0.98	0.96
Moderate	0.97	0.96	0.98	0.98
Severe	0.99	0.97	0.98	0.99

Furthermore, in Table 6, the Random Forest model evaluation results show superior performance compared to the previous KNN model evaluation results. This is indicated by a fairly high value in all evaluation metrics. The precision value for the three acne classes ranges from 0.97 to 0.99, with the highest value in the ’Severe’ class reaching 0.99. Furthermore, the recall value in the Random Forest model is also very good, with the highest value in the ’Mild’ class of 0.98 and each of the ’Moderate’ and ’Severe’ classes of 0.96 and 0.97. The F1-score value for all classes is at 0.98, indicating an optimal balance between precision (accuracy of positive predictions) and recall (ability to find all relevant cases) for this model. In terms of accuracy, random forest has a high accuracy value of 0.96 for the ’Mild’ class, 0.98 for the ’Moderate’ class and 0.99 for the ’Severe’ class.

Table 7. Comparison of Model Accuracy

Method	Avg Accuracy
K-Neirest Neighbors	95%
Random Forest	97%

Table 7 compares the average accuracy values between the KNN and Random Forest models. The KNN model obtained an average accuracy of 95%, while the Random Forest model had a better average accuracy of 97%. This shows that the Random Forest model successfully classifies the dataset to classify acne severity more accurately and efficiently.

In addition to classification accuracy, the computational trade-offs between both algorithms reveals distinct performance characteristics. During the training process, the KNN model exhibits low computational demand, making it suitable for smaller datasets. However, it suffers from scalability limitations at the prediction stage due to the need to compute distances for each query. In contrast, the Random Forest model requires higher computational resources during training,

LONTAR KOMPUTER VOL. 16, NO. 2 AUGUST 2025
DOI : 10.24843/LKJTI.2025.v16.i2.p06
Accredited Sinta 2 by RISTEKDIKTI Decree No. 158/E/KPT/2021
p-ISSN 2088-1541
e-ISSN 2541-5832

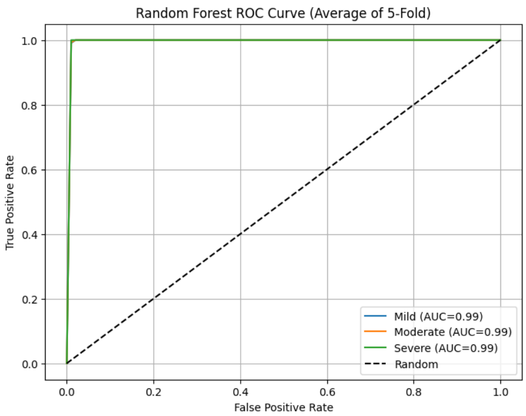

Figure 8. KNN ROC Curve

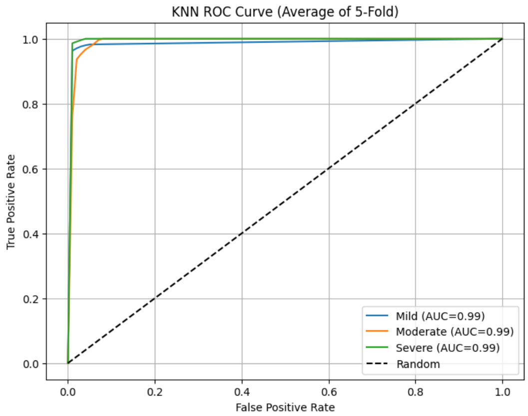

Figure 9. RF ROC Curve

as it builds 100 decision trees, leading to longer training time. Nevertheless, it offers faster inference and better scalability during deployment, achieving 97% accuracy, which is 2% higher than KNN. Despite the seemingly small difference, this improvement can have meaningful implications in clinical dermatology applications.

In addition to standard evaluation metrics such as precision, recall, and F1-score, this study also presents ROC (Receiver Operating Characteristic) curve analysis to further evaluate the classification performance of the KNN and Random Forest models. As shown in Figures 8 and 9, both models show excellent capabilities, with AUC (Area Under the Curve) values reaching 0.99 for all three acne severity classes. These results reinforce the reliability of the models and support the consistency of the classification performance in the cross-validation scenario.

4. Conclusion

Based on the results of this study, the acne severity classification model using two machine learning methods, namely K-Nearest Neighbors (KNN) and Random Forest, was successfully developed with the support of feature extraction from the YOLOv8 model and also the SMOTE method as a data balancer. The evaluation results in Table 7 show that the KNN and RF models can classify acne severity into three categories, namely 'Mild', 'Moderate' and 'Severe' with a relatively high level of accuracy, namely 95% for the KNN model and 97% for the Random Forest model.

The Random Forest model shows relatively high performance compared to KNN, which is indicated by the Precision, Recall, and F1-score values that are almost perfect in all classes. In addition, the confusion matrix also shows that the classification error in Random Forest is tiny compared to KNN, which has several errors. The use of the SMOTE method in this study has proven effective in balancing data so that the model can improve accuracy.

This study highlights the importance of two key features—acne density and average confidence—extracted from YOLOv8 detection results. These features serve as compact and meaningful representations of acne severity and contribute significantly to model accuracy. Furthermore, ROC-AUC analysis shows that both models achieve AUC values of up to 0.99 across severity classes, supporting their reliability and discriminative power.

Overall, the application of the Random Forest method combined with feature extraction from the YOLOv8 model and the SMOTE method as a data balancer can produce an accurate and effective acne severity classification system, which has promising potential as an aid in dermatological assessment.

4.1. Ethical Considerations

This study used an acne image dataset obtained from a public platform (Kaggle) without explicit information regarding ethical approval or medical usage rights. In addition, the data did not include demographic information such as skin color, age, and ethnicity, which could potentially bias the classification results. The developed model has not been validated by medical professionals or clinical trials, so it cannot be used as a direct diagnostic tool. Further validation and more representative data-based testing are required before the system is applied in medical practice.

References

[1] M. Ahammed, M. A. Mamun, and M. S. Uddin, "A machine learning approach for skin disease detection and classification using image segmentation," *Healthcare Analytics*, vol. 2, p. 100122, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772442522000624

[2] R. K Karunanayake, W. M. Dananjaya, M. Y Peiris, B. Gunatileka, S. Lokuliyana, and A. Kuruppu, "Cureto: Skin diseases detection using image processing and cnn," in *2020 14th International Conference on Innovations in Information Technology (IIT)*, 2020, pp. 1–6.

[3] V. A. Yusuf, N. Nurbaiti, and T. O. Permatasari, "Hubungan antara tingkat pengetahuan pelajar sekolah menengah atas tentang acne vulgaris pada wajah dengan perilaku pengobatannya," *Tunas Medika Jurnal Kedokteran & Kesehatan*, vol. 6, no. 2, pp. 83–86, 2020.

[4] S. A. Amania, "Klasifikasi jenis jerawat wajah menggunakan arsitektur inception v3," Ph.D. dissertation, UNIVERSITAS ISLAM SULTAN AGUNG, 2023.

[5] A. A. Azhar, I. A. M. P. Sutema, and P. H. Gunawan, "Deep learning-based exploration of yolov8 for acne vulgaris type classification and lesion counting," in *2024 International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)*. IEEE, 2024, pp. 1377–1382.

[6] P. Bhadra, "A literature review onacne due to hormonal changes and lifestyle," Indian Journal of Natural Sciences, vol. 10, no. 59, pp. 18 507–18 521, 2020

[7] K. Sieradocha, "The mental health implications of acne vulgaris," Quality in Sport, vol. 35, pp. 56 063–56 063, 2024.

[8] S. A. AlDera and M. T. B. Othman, "A model for classification and diagnosis of skin disease using machine learning and image processing techniques," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, 2022. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2022.0130531

[9] A. Sankar et al., "Utilizing Generative Adversarial Networks for Acne Dataset Generation in Dermatology," *BioMedInformatics*, vol. 4, pp. 1059–1070, 2024.

[10] A. A. Aldino, R. R. Suryono, and R. Ambarwati, "Analysis of covid-19 cash direct aid (blt) acceptance using k-nearest neighbor algorithm," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 16, no. 2, pp. 193–204, 2022.

[11] R. Suwanda, Z. Syahputra, and E. M. Zamzami, "Analysis of euclidean distance and manhattan distance in the k-means algorithm for variations number of centroid k," Journal of Physics: Conference Series, vol. 1566, no. 1, p. 012058, jun 2020. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/1566/1/012058

[12] A. Singh and B. Pandey, "An euclidean distance based knn computational method for assessing degree of liver damage," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, vol. 1, 2016, pp. 1–4.

[13] L. Farokhah, "Implementasi k-nearest neighbor untuk klasifikasi bunga dengan ekstraksi fitur warna rgb," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 7, no. 6, pp. 1129– 1135, 2020.

[14] M. R. Romadhon and F. Kurniawan, "A comparison of naive bayes methods, logistic regression and knn for predicting healing of covid-19 patients in indonesia," in *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, 2021, pp. 41–44.

[15] S. Bebortta, M. Panda, and S. Panda, "Classification of pathological disorders in children using random forest algorithm," in *2020 international conference on emerging trends in information technology and engineering (ic-ETITE)*. IEEE, 2020, pp. 1–6.

[16] H. A. Salman, A. Kalakech, and A. Steiti, "Random forest algorithm overview," *Babylonian Journal of Machine Learning*, vol. 2024, pp. 69–79, 2024.

[17] M. Minnoor and V. Baths, "Diagnosis of breast cancer using random forests," *Procedia Computer Science*, vol. 218, pp. 429–437, 2023.

[18] W. Feng, C. Ma, G. Zhao, and R. Zhang, "Fsrf:an improved random forest for classification,"

in 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA), 2020, pp. 173–178.

[19] Y. Yanto, F. Aziz, and I. Irmawati, "Yolo-v8 peningkatan algoritma untuk deteksi pemakaian masker wajah," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 3, pp. 1437–1444, 2023.

[20] A. Aboah, B. Wang, U. Bagci, and Y. Adu-Gyamfi, "Real-time multi-class helmet violation detection using few-shot data sampling technique and YOLOv8," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 5350–5358.

[21] D. J. Marcelleno and M. P. K. Putra, "Performance evaluation of yolov8 in real-time vehicle detection in various environmental conditions," *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 1,pp. 269–279, 2025.

[22] I. Muraina, "Ideal dataset splitting ratios in machine learning algorithms: general concerns for data scientists and data analysts," in *7th international Mardin Artuklu scientific research conference*, 2022, pp. 496–504.

[23] T. Cai, H. He, and W. Zhang, "Breast cancer diagnosis using imbalanced learning and ensemble method," *Applied and Computational Mathematics*, vol. 7, no. 3, pp. 146–154, 2018.

[24] A. Abdullah ALFRHAN, R. Hamad ALHUSAIN, and R. Ulah Khan, "Smote: Class imbalance problem in intrusion detection system," in 2020 International Conference on Computing and Information Technology (ICCIT-1441), 2020, pp. 1–5

[25] S. Sathyanarayanan and B. R. Tantri, "Confusion matrix-based performance evaluation metrics," *African Journal of Biomedical Research*, pp. 4023–4031, 2024.

[26] F. Rahmad, Y. Suryanto, and K. Ramli, "Performance comparison of anti-spam technology using confusion matrix classification," in *IOP Conference Series: Materials Science and Engineering*, vol. 879, no. 1. IOP Publishing, 2020, p. 012076.

[27] S. Swaminathan and B. R. Tantri, "Confusion matrix-based performance evaluation metrics," African Journal of Biomedical Research, vol. 27, pp. 4023–4031, 11 2024.

[28] A. A. Salih and A. M. Abdulazeez, "Evaluation of classification algorithms for intrusion detection system: A review," Journal of Soft Computing and Data Mining, vol. 2, no. 1, pp. 31–40, 2021.

[29] H. Saveria and P. H. Gunawan, "Stunting classification among toddlers using decision tree c4. 5 in berastagi city," in *2025 International Conference on Advancement in Data Science, E-learning and Information System (ICADEIS)*. IEEE, 2025, pp. 1–6.

[30] A. Arias-Duart, E. Mariotti, D. Garcia-Gasulla, and J. M. Alonso-Moral, "A confusion matrix for evaluating feature attribution methods," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3709–3714.