NextGen Innovations in Computing and Technology (NICT)
e-ISSN: XXXX-XXXX | Vol 1, Issue 1, January 2026
Available Online: https://journal.arfadigitech.com/index.php/nict

Implementation and Evaluation of a Simple On-Device Facial Expression Recognition App Using MIT App Inventor and Personal Image Classifier

Alfin Gimnastiar[1]*, M. Alfi Syahrin[2], Maulana Firjatullah[3], T Muhammad Ricki[4], Maulana Akbar[5]

[1,2,3,4] Faculty of Science and Technology, Universitas Islam Negeri Sumatera Utara, Medan, Indonesia
[5] Faculty of Computer Science, Universitas Royal, Kisaran, Indonesia

*Corresponding Author Email: *Afingimnastiar@gmail.com*

Submitted: November 17, 2025 **Accepted:** December 26, 2025 **Published:** January 31, 2026

Abstract

This study presents the implementation and evaluation of a simple on-device facial expression recognition application developed primarily for educational and introductory learning purposes using MIT App Inventor and the Personal Image Classifier (PIC). The application is designed to classify three basic facial expressions angry, sad, and happy using a machine learning model embedded directly into the mobile device. The development process follows a prototyping approach, starting with the design of the user interface, integration of the PIC extension, and the model training workflow, which includes data collection, training, testing, and exporting the trained model for use within the application. The evaluation was conducted through qualitative testing on various facial images obtained from both AI-generated sources and publicly available images. The model successfully produced probability outputs corresponding to each emotion class and demonstrated correct predictions under favorable conditions such as frontal pose and good lighting. However, its overall accuracy remains limited due to the very small training dataset, consisting of only six images, and the absence of data augmentation. These constraints resulted in inconsistent predictions in several test cases. Despite these limitations, the study demonstrates that integrating a lightweight machine learning model into a mobile application using MIT App Inventor is feasible and effective, making it suitable as an accessible learning tool for beginners exploring artificial intelligence and image-based emotion recognition.

Keywords: Emotion Recognition, Machine Learning, Personal Image Classifier, Image Classification

1. Introduction

The rapid advancement of artificial intelligence (AI) technology in recent years has significantly accelerated developments in image processing and computer vision. One of the widely explored branches is emotion recognition, which refers to the process of identifying human facial expressions using machine learning algorithms[1]. This technology has become increasingly relevant and easier to implement due to improvements in mobile device computing capabilities and the availability of visual programming platforms that support AI-based application development without requiring complex programming skills.

In the context of Human–Computer Interaction, the ability of a system to recognize user emotions can provide substantial added value in various fields, such as education, healthcare, entertainment, and security. However, the development of emotion recognition applications typically requires in-depth understanding of machine learning algorithms, image processing techniques, and complex programming environments. These factors often present challenges for beginners or students who are just starting to learn AI-based application development[2].

MIT App Inventor offers a solution by enabling novice users to create mobile applications visually through a drag and drop approach. With the Personal Image Classifier (PIC) extension, users can train a machine learning model easily through a web-based interface without needing to understand the technical details of classification algorithms[3]. The model training process is carried out through several intuitive steps, including adding labeled image data, training the model, performing initial testing, and downloading the trained model for integration into MIT App Inventor. This flexibility makes the platform suitable for simple experimental projects in computer vision, including facial expression recognition.

Recent studies have also highlighted the growing interest in lightweight machine learning models that can operate directly on mobile devices without relying on cloud-based inference. On-device processing offers several advantages, including reduced latency, enhanced privacy, and broader accessibility for users in low-connectivity environments. Such approaches are particularly relevant for educational or prototype-level applications, where simplicity and ease of deployment are prioritized over high model complexity. Several previous works have demonstrated that even simple classifiers can produce acceptable results for limited-scale image recognition tasks when properly trained with representative datasets[4].

In addition, research on facial expression recognition continues to emphasize the importance of dataset variability, image quality, and consistent preprocessing pipelines in determining model performance. Studies show that small or imbalanced datasets tend to limit a model's generalization ability, leading to inconsistent predictions under different visual conditions [5]. However, most existing studies focus primarily on improving accuracy through complex models or large-scale datasets, while limited attention has been given to evaluating the behavior and limitations of lightweight, beginner-oriented models under extremely constrained training conditions, particularly in fully on-device mobile implementations.

Furthermore, recent educational and applied studies have increasingly emphasized the role of low-code and no-code machine learning platforms as practical alternatives for introducing artificial intelligence concepts to beginners and non-expert users. Unlike conventional deep-learning-based approaches, these platforms prioritize usability, rapid prototyping, and conceptual understanding over algorithmic complexity and high accuracy. As a result, models developed within such environments while inherently limited in performance provide valuable insights into how simplified classifiers behave under real deployment conditions. This perspective justifies the use of Personal Image Classifier as a research object, not as a competitor to CNN-based systems, but as a representative example of beginner-oriented, on-device machine learning tools that remain underexplored in empirical evaluation studies.

Beyond its relevance in educational settings, the development of lightweight emotion recognition systems also addresses practical constraints often encountered in mobile environments, such as limited processing power, restricted memory capacity, and variability in camera quality. Many conventional deep-learning-based approaches require large datasets and high computational resources, making them difficult to implement directly on entry-level devices [6]. This condition highlights the importance of exploring simplified machine learning workflows such as those offered by MIT App Inventor combined with the Personal Image Classifier which allow users to deploy on-device classification models without high hardware demands. Such an approach not only supports accessibility for beginners but also aligns with the broader trend of promoting AI literacy through low-code and no-code platforms[7].

Based on this research gap, the objective of this study is to implement and evaluate a simple on-device facial expression recognition application using MIT App Inventor and the Personal Image Classifier, with a specific focus on analyzing model behavior, performance limitations, and prediction consistency when trained on a very small dataset. To achieve this objective, the following section describes the research methodology, including system architecture, model training procedures, and evaluation methods applied in this study.

2. Methodology

The methodology of this study was designed to systematically describe the processes involved in designing, developing, and evaluating a facial expression recognition application based on MIT App Inventor and the Personal Image Classifier. The methodological approach emphasizes a qualitative exploratory evaluation, aiming to observe model behavior, limitations, and usability under constrained data conditions rather than to measure statistical performance metrics. All stages are presented objectively to ensure clarity, replicability, and transparency of the research process.

2.1 System Overview and Architecture

The developed application implements a simple mobile-based architecture in which the facial expression recognition process is performed entirely on-device using a machine learning model trained with the Personal Image Classifier (PIC). The system operates through three main stages: image acquisition via the smartphone camera and internet sources, image processing by the classification model to determine emotion labels, and presentation of prediction results in the form of probability scores displayed on the application interface. All processing is executed locally on the user's device without requiring an internet connection or external server.

2.2 Development Platform and Tools

The application was developed using MIT App Inventor, a visual programming environment based on block programming that enables the implementation of application logic without manual coding. The machine learning model was trained using the Personal Image Classifier, a web-based service that allows users to create image classification models through data uploading, model training, validation, and model export. An Android smartphone was used as the primary testing device for executing the application.

2.3 Machine Learning Model Training

The development process followed a prototyping approach consisting of several iterative stages until the application achieved functional stability. The first stage involved interface design, including identification of key components such as the classification button, camera component, output display area, and layout arrangement. Next, the Personal Image Classifier extension was added, followed by configuration of essential components related to model invocation and prediction response handling.

The trained model was downloaded as a configuration file and imported into the extension component. Application logic was then constructed using block programming to implement classification function calls, read probability values, and display emotion labels. This iterative process continued until the application ran without system errors.

2.4 Data Acquisition and Processing

Model training was conducted through four stages in the Personal Image Classifier: add data, select model, add testing data, and view result. In the add data stage, facial images were assigned to three emotion classes angry, sad, and happy with each class containing two images, resulting in a total dataset of six images. The model was trained using PIC's default configuration during the select model stage. Initial validation was conducted by uploading unseen images in the add testing data stage. The training results were reviewed in the view result stage, after which the trained model was downloaded and integrated into MIT App Inventor.

2.5 Testing and Evaluation Method

The testing phase consisted of two distinct components: functional testing and model evaluation, each serving a different purpose. Functional testing focused on verifying whether the application operated correctly, including camera activation, model invocation, and result display. In contrast, model evaluation aimed to explore how the trained classifier behaves under various visual conditions, rather than to compute numerical accuracy metrics.

A black-box testing approach was employed by executing the application directly without examining internal system processes. Model evaluation was conducted in a qualitative exploratory manner, observing prediction outputs, probability distributions, and consistency across different facial expressions, lighting conditions, and face orientations. This approach aligns with the study's objective of understanding the practical limitations of a lightweight, beginner-oriented classification model.

2.6 Additional Evaluation Procedures

Application testing was conducted using the black-box method, where the application's main functionalities were evaluated without internal code inspection [8]. Observed parameters included correct label output, system responsiveness, and prediction consistency under different lighting conditions. Accuracy assessment was performed qualitatively by comparing predicted labels and probability trends with the actual facial expressions. While the application functioned as intended, prediction outcomes were found to be unstable due to the limited training dataset.

To strengthen the exploratory evaluation, additional qualitative observations were carried out across multiple testing scenarios involving variations in illumination, background complexity, camera distance, and facial orientation. These scenarios were selected based on prior findings indicating that lightweight models deployed on mobile devices often exhibit performance degradation under non-ideal input conditions, especially when trained with minimal data. The evaluation therefore focused on identifying recurring patterns and behavioral tendencies of the PIC model rather than producing generalized performance claims.

From a system engineering perspective, the application follows a streamlined data-processing pipeline in which captured images are forwarded directly to the embedded model for classification. The PIC model maps input images to probability outputs using parameters learned during training, and these outputs are interpreted by the application logic to determine the most likely emotion label. Environmental consistency was maintained by testing the application on Android devices running Android 10, with additional trials conducted on a secondary device to observe potential hardware-related variationstions[9].

Overall, the evaluation strategy was designed to capture both functional reliability and exploratory insights into model behavior, ensuring that the methodology reflects the study's emphasis on implementation feasibility and qualitative performance analysis under constrained conditions.

3. Results and Discussion

This section presents the results of the implementation of the facial expression recognition system and the evaluation of the model's performance. The findings are described narratively and supported by several figures that illustrate the application interface, model output, and example predictions under various testing conditions. Each figure is placed and referenced according to standard numbering conventions to facilitate understanding.

3.1 System Implementation Results

The application was successfully compiled and executed on an Android device without any system errors. The application interface consists of three main components: the camera preview, a button to initiate the classification process, and a text output area that displays the prediction results. The interface layout of the application is shown in Figure 1.

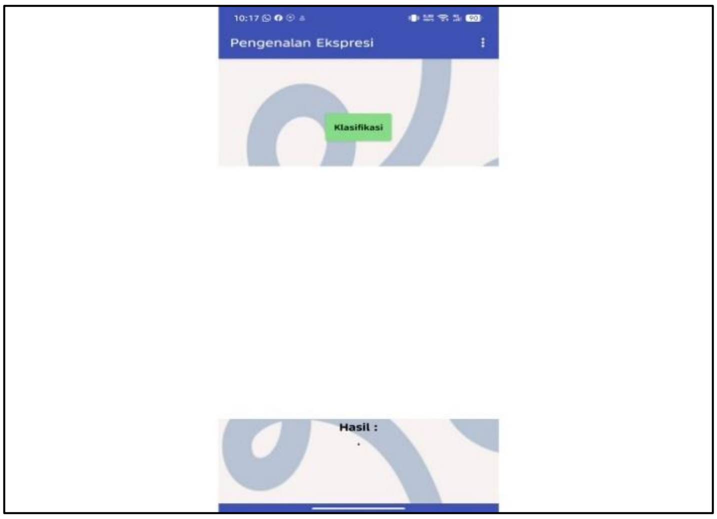

Figure 1. Main interface of the facial expression recognition application

When the classification button is pressed, the application sends the camera image to the embedded Personal Image Classifier model. The model then produces probability scores for each emotion class: angry, sad, and happy. These probability values not only indicate which class is dominant but also show how confident the model is in its decision. This information is important for understanding the accuracy level of the model's output for a given image. An example of the prediction result is shown in Figure 2.

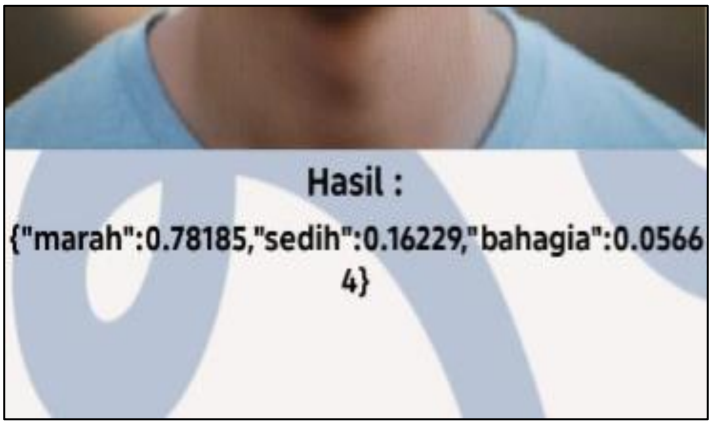

Figure 2. Example of model output showing probability distribution for each emotion class

3.2 Model Performance Analysis

As part of the evaluation of the machine learning model, testing was conducted using several facial expression images from various sources (AI-generated images and publicly available internet images). The model was able to produce predictions that aligned with the trained categories. However, the probability values often showed imbalances between classes and instability when the images had poor lighting or different face angles. This condition is directly influenced by the limited amount of training data, as each class contains only two images.

In addition to the core performance observations, further analysis was conducted to examine how the model behaves under incremental changes to the testing conditions, particularly variations that commonly occur in real mobile usage. When images were captured with moderate noise, such as slight motion blur or inconsistent lighting, the model exhibited a noticeable decrease in confidence scores even when the predicted class remained correct. This phenomenon aligns with previous findings on lightweight machine learning models, which tend to be highly sensitive to small perturbations and lack the robustness of deep convolutional networks trained on large-scale datasets. Moreover, when the background contained distracting elements or colors similar to skin tones, the model occasionally misclassified expressions due to the absence of strong feature extraction mechanisms. This limitation reflects the theoretical constraint that models with minimal parameterization such as the PIC model are more prone to overfitting and rely heavily on simplified feature relationships. These observations further illustrate that while the system performs adequately for basic demonstrations, its reliability decreases significantly in uncontrolled real-world environments

The test results indicate that the model tends to produce more accurate predictions when the facial expression has a frontal pose, adequate contrast, and clearly distinguishable emotional cues. For example, in images where the angry expression is strongly visible, the model assigns the highest probability to the angry class, as shown in Figure 3.

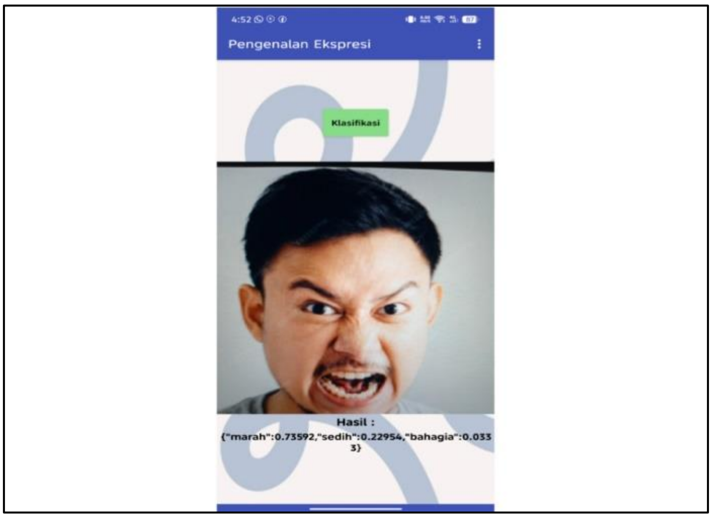

Figure 3. Example of correct classification based on the actual expression

Conversely, in certain test cases, the model failed to correctly recognize the expression. For instance, in images with a slight smile, the model sometimes assigned a higher score to the sad class. This occurs due to the insufficient variability of expressions in the training dataset. An example of an incorrect prediction is illustrated in Figure 4.

In addition to the previously described observations, further evaluation was conducted to examine how the model behaves across different environmental and contextual variations. The results reveal that the Personal Image Classifier demonstrates a high dependency on visual clarity and feature contrast when interpreting facial expressions. When faces appear with strong illumination and distinct shading around the eyes, eyebrows, and mouth region, the model tends to generate more decisive probability distributions. Conversely, when the illumination becomes uneven either too bright or too dim the prediction confidence drops significantly, often resulting in probability scores that are nearly balanced across multiple classes.

Another important finding relates to the model’s handling of subtle emotional expressions. Images showing faint or ambiguous expressions, such as a soft smile or partially raised eyebrows, frequently lead to misclassification. This indicates that the PIC model may be relying more heavily on coarse visual cues rather than fine-grained facial muscle features, which more advanced deep learning models typically capture. Moreover, emotional expressions captured at an angle as opposed to a frontal pose show a noticeable decline in classification accuracy. Even a slight head rotation of 10-20 degrees can disrupt the probability distribution, suggesting limited robustness in the model’s learned representations.

Additional tests were also performed using images with diverse backgrounds, including patterned walls, crowded environments, and images containing partial occlusions such as glasses or hair covering parts of the face. These conditions further challenged the model and frequently reduced prediction accuracy. Such findings reinforce that the classifier is highly sensitive to noise in the visual input and lacks mechanisms to isolate the facial region from the background. The model’s inconsistency across these varied contexts supports the conclusion that the extremely small training dataset limits the model’s generalization capabilities

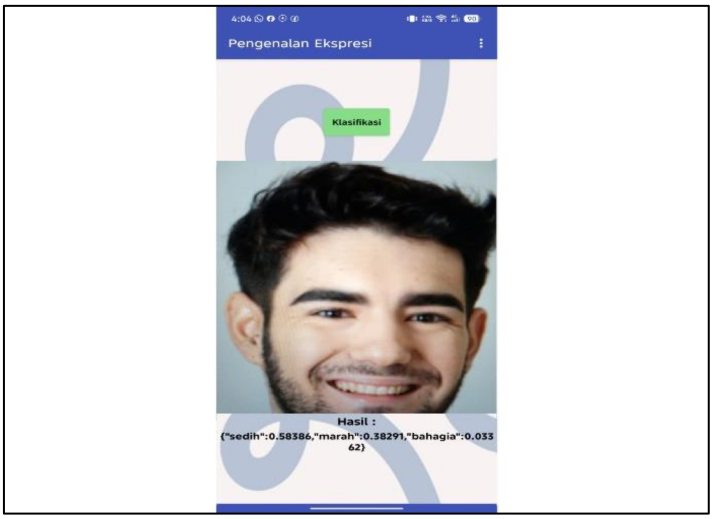

Figure 4. Example of incorrect classification based on the actual expression

3.3 Discussion of Limitations and Technical Factors

The results indicate that the model is capable of performing basic facial expression classification; however, its accuracy is not yet optimal. Several technical factors influencing performance include:

1. Very limited dataset size

The model was trained using only six images (two per class), resulting in inadequate facial feature representation. In machine learning, small training datasets hinder the model’s ability to recognize diverse expressions[10].

2. Non-uniform data variations

The dataset consists of AI-generated images and internet images, which differ in quality, lighting, and expression. These inconsistencies reduce prediction stability[11].

3. Absence of data augmentation.

The Personal Image Classifier does not provide automatic augmentation functions. Without augmentation techniques such as rotation, brightness adjustment, or flipping, the model becomes less robust to pose and lighting variations[12].

4. Limited model complexity of PIC

The Personal Image Classifier is designed for beginners and thus lacks complex deep-learning architectures such as CNNs. This limits the model's capability in extracting detailed facial features[13][14].

In addition to the previously mentioned factors, another limitation observed during testing is the model's sensitivity to background variations and image noise. Facial images with cluttered or highly textured backgrounds tended to produce lower prediction confidence, suggesting that the model was unable to sufficiently isolate salient facial regions during classification. This issue is commonly found in lightweight or shallow classification models that rely primarily on pixel-level patterns rather than deep hierarchical feature extraction. Furthermore, the absence of preprocessing steps such as face detection, cropping, or normalization contributed to inconsistencies in prediction outcomes, as variations in image scale and framing were directly fed into the model without adjustment. These observations emphasize the importance of incorporating basic preprocessing pipelines to improve the model's robustness in real-world applications[15].

Despite these limitations, the implementation results demonstrate that image classification techniques using MIT App Inventor and the Personal Image Classifier can be practically applied for simple applications. The analysis of these limitations further highlights how model constraints and training data influence performance in lightweight machine learning systems.

3.4 Alignment with Research Objectives

The findings indicate that the application successfully meets the research objective of developing a mobile system capable of recognizing facial expressions using an on-device machine learning model. The functionality operates as designed, and the experimental results demonstrate how image classification models can be integrated into mobile applications for educational purposes. Furthermore, the study reinforces that the quantity and quality of training data play a crucial role in AI model accuracy, consistent with machine learning theory that emphasizes the importance of representative datasets for effective generalization.

4. Conclusion

This study successfully developed and evaluated a facial expression recognition application using MIT App Inventor by integrating a machine learning model trained with the Personal Image Classifier. The main objective of the research, which was to produce a simple mobile application capable of recognizing three facial expression categories angry, sad, and happy was achieved effectively. The results show that the model can generate probability-based predictions for each emotion class and display them directly on the user's device.

Testing results indicate that the model performs correctly under favorable conditions, such as good lighting, frontal facial orientation, and clear emotional expressions. However, prediction accuracy remains limited and inconsistent when facing variations in lighting, facial angles, or expression intensity. These limitations are mainly caused by the very small training dataset and the absence of data augmentation during model training.

Overall, this study demonstrates that a simple image classification model can be effectively integrated into a mobile application using MIT App Inventor and easily replicated. This work explicitly contributes to educational purposes by providing an accessible example of on-device machine learning implementation for beginners using low-code tools. The findings also highlight the importance of sufficient and representative training data in improving model performance, suggesting directions for future research to enhance accuracy and robustness.

References

[1] M. B. Septyono, F. T. Anggraeny, and R. Mumpuni, "Pengenalan Ekspresi Wajah dengan LBP dan Multi-Level CNN," *Jurnal Informatika Polinema*, vol. 11, no. 4, pp. 469–476, Aug. 2025, doi: 10.33795/JIP.V11I4.7351.

[2] S. A. Baskara and N. Setiyawati, "Experimental Modeling of Face Emotion Recognition Using Machine Learning Classification (SVM, KNN, Random Forest) and Deep Learning CNN," *Teknika*, vol. 14, no. 2, pp. 280–289, Jul. 2025, doi: 10.34148/TEKNIKA.V14I2.1232.

[3] Michelle, S. Birowo, and M. A. Maulana, "Deteksi Suasana Hati Karyawan Berbasis Deep Learning Menggunakan CNN," *bit-Tech*, vol. 8, no. 1, pp. 18–24, Aug. 2025, doi: 10.32877/BT.V8I1.1882.

[4] A. N. Inaya, A. U. Rahma, M. Jannah, L. R. K. Arafah, L. L. Ishak, and M. R. Edy, "Klasifikasi Citra Dengan Pendekatan Transfer Learning Pada Gambar Fauna Terbang," *Jurnal MediaTIK*, vol. 85, no. 1, pp. 85–89, Jun. 2024, doi: 10.59562/MEDIATIK.V7I1.2785.

[5] I. R. Alfiandi, Gunawan, M. R. Fadhil, and R. Samsinar, "Analisis Performa Convolutional Neural Network (CNN) dan Naive Bayes dalam Face Recognition: Akurasi dan Kompleksitas," *Prosiding Seminar Nasional Hukum, Bisnis, Sains dan Teknologi*, vol. 5, no. 1, pp. 284–294, Jan. 2025.

[6] K. Khotimah, F. Yudistira, and M. Ardiansyah, "Efisiensi Deep learning untuk Analisis Data dan Pengambilan Keputusan," *Jurnal Insan Peduli Pendidikan (JIPENDIK)*, vol. 2, no. 2, pp. 79–82, Dec. 2024.

[7] M. Kholil, R. Akhsani, I. Athaillah, H. Priya Waspada, M. Saiful Muluk, and K. Charisma, "Pelatihan Pengembangan Aplikasi Klasifikasi Gambar Dengan Library Kecerdasan Buatan Bagi Siswa Kelas XI SMAK Diponegoro Blitar – Jawa Timur," *Jurnal Pengabdian Masyarakat Terapan*, vol. 1, no. 2, pp. 83–89, Aug. 2024, doi: 10.20884/1.JUPITER.1.2.20.

[8] E. H. Kusuma Dewi, I. S. Pratama, A. S. Putera, and C. Carudin, "Black Box Testing pada Aplikasi Pencatatan Peminjaman Buku Menggunakan Boundary Value Analysis," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 6, no. 3, p. 315, Apr. 2022, doi: 10.30998/STRING.V6I3.11958.

[9] F. Dinata and F. Antonio, "ANTESEDEN DARI VIRTUAL ORGANIZATIONAL COMMITMENT DAN DAMPAKNYA TERHADAP VIRTUAL CARE PERFORMANCE (STUDI PADA PLATFORM TELEMEDISIN XYZ DI INDONESIA)," *JMBI UNSRAT (Jurnal Ilmiah Manajemen Bisnis dan Inovasi Universitas Sam Ratulangi).*, vol. 11, no. 1, pp. 744–765, Mar. 2024, doi: 10.35794/JMBI.V11I1.55630.

[10] S. Siswanto, M. U. Dewi, S. Kholifah, G. Widhiati, and W. Aryani, "Penggunaan Model Deep Learning Untuk Meningkatkan Efisiensi Dalam Aplikasi Machine Learning," *JURNAL PENELITIAN SISTEM INFORMASI (JPSI)*, vol. 1, no. 4, pp. 215–238, Nov. 2023, doi: 10.54066/JPSI.V1I4.1619.

[11] M. Singh and A. S. Arora, "Varying Illumination and Pose Conditions in Face Recognition," *Procedia Comput Sci*, vol. 85, pp. 691–695, Jan. 2021, doi: 10.1016/J.PROCS.2016.05.255.

[12] B. Baso and N. Suciati, "TEMU KEMBALI CITRA TENUN NUSA TENGGARA TIMUR MENGGUNAKAN EKSTRAKSI FITUR YANG ROBUST TERHADAP PERUBAHAN SKALA, ROTASI DAN PENCAHAYAAN," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 7, pp. 349–358, 2021, doi: 10.25126/jtiik.202072002.

[13] S. A. Adzkia and T. Arifin, "Perbandingan Arsitektur CNN Berbasis Transfer Learning untuk Klasifikasi pada BreastMNIST," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 11, no. 2, pp. 192–200, Sep. 2025, doi: 10.25077/TEKNOSI.V11I2.2025.192-200.

[14] A. Taufiq Akbar, S. Saifullah, and H. Prapcoyo, "Klasifikasi Ekspresi Wajah Menggunakan Covolutional Neural Network," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 6, pp. 1399–1412, Dec. 2024, doi: 10.25126/JTIIK.2024118888.

[15] R. Surya, “Peningkatan Akurasi OCR dalam Pemrosesan Formulir Keuangan melalui Fine-Tuning Transformer dan Strategi Pra- pemrosesan Data,” *Jurnal Inovasi Informatika*, vol. 7, no. 2, pp. 1–12, May 2025, doi: 10.51170/JII.V7I2.85.

[16] T. A. Selian, N. Akbar, and M. I. Hidayat, “Image Classification Based On Color Using Thresholding Method”, *jitcos*, vol. 1, no. 1, pp. 1–6, Jun. 2025, doi: 10.65230/jitcos.v1i1.5.

[17] M. K. Gibran, E. Budhiarti Nababan and P. Sihombing, "Analysis of Face Recognition with Fuzzy C-Means Clustering Image Segmentation and Learning Vector Quantization," *2020 3rd International Conference on Mechanical, Electronics, Computer, and Industrial Technology (MECnIT)*, Medan, Indonesia, 2020, pp. 188-193, doi: 10.1109/MECnIT48290.2020.9166649.

[18] S. F. M. . Mar’ah, H. B. P. . Alam, D. Aulia, N. A. . Meliala, F. . Prayoga, and M. K. . Gibran, “Penerapan Teknik Binarisasi Dan Segmentasi Citra Untuk Digitalisasi Tanda Tangan Di Matlab”, *JMP*, vol. 14, no. 1, pp. 623-630, May 2025.

[19] F. Azmi, M. K. Gibran, and A. Saleh, “Herbal plant image retrieval using HSV color histogram and random forest algorithm,” *Journal of Computer Science, Information Technology and Telecommunication Engineering*, vol. 6, no. 2, pp. 1019–1027, 2025, doi: 10.30596/jcositte.v6i2.26495.

[20] M. I. Rifki, M. K. Gibran, A. H. Hasugian, and M. D. Solihin, “Pemodelan dan evaluasi prediksi RSRP menggunakan artificial neural network untuk optimasi kualitas layanan jaringan komunikasi nirkabel,” *Informatika*, vol. 17, no. 1, pp. 392–401, 2025, doi: 10.36723/juri.v17i1.751.