JOURNAL OF COMPUTER SCIENCE AND INFORMATICS ENGINEERING
VOL. 5, NO. 4 (2026)
DOI: https://doi.org/10.55537/cosie.v5i4.1858

e-ISSN 2827-8356
p-ISSN 2827-9999

Local Binary Pattern for Facial Gender Identification Using a Kivy-Based Application

Wira Pranata Sebayang[1], Tomy Satria Alasi[2], Irwan Jani Tarigan[3]

[1,2,3]*STMIK Methodist Binjai, Binjai, 20711, Indonesia*

[1]wirapranatasebayang0987@gmail.com, [2]tomysatriaalasi@gmail.com, [3]irwanjntarigan@gmail.com

Received: August 2, 2026 | Revised: August 4, 2026 | Accepted: September 16, 2026 | Published: September 17, 2026

Abstract

Automatic gender identification based on facial images still faces challenges due to spatial variations and the complexity of visual features, while deep learning approaches such as Convolutional Neural Networks require high computational resources that are unsuitable for resource-constrained devices. This study aims to design, implement, and evaluate the performance boundaries of the Local Binary Pattern method for facial image-based gender identification integrated into a Python application using the Kivy framework. The research methodology utilized facial images from the UTKFace dataset, undergoing preprocessing through grayscale conversion, micro-texture feature extraction via the Local Binary Pattern operator, and gender classification using a threshold-based decision rule, all of which were fully integrated into the Kivy user interface. The system was evaluated across three scenarios with varying numbers of test images (80, 160, and 320 images). Testing results revealed a peak accuracy of only 43%, with variations in lighting, facial pose, and expression identified as the primary factors limiting overall accuracy. It can be concluded that while the Local Binary Pattern method was successfully integrated technically within the Kivy framework, its simplified mean-value feature representation remains insufficient for reliable gender identification under uncontrolled visual conditions, highlighting the need for richer feature representations in future research.

Keywords: Local Binary Pattern; Gender Identification; Facial Images; Kivy; Image Processing

1. INTRODUCTION

Rapid advancements in computer vision have established digital image processing as a fundamental pillar of modern artificial intelligence[1], [2]. One evolving application in this domain is automatic gender identification based on facial image analysis. Human faces inherently convey rich visual information, ranging from micro-surface skin texture patterns to local intensity gradient variations. Capturing and processing these spatial texture patterns is essential for constructing responsive and robust recognition frameworks within human-computer interaction systems[3], [4]. Although deep learning paradigms, particularly Convolutional Neural Networks (CNNs), have demonstrated superior accuracy in gender classificationa[5], [6], [7], [8], [9], their high computational overhead and memory demands present significant practical challenges[10], [11]. These limitations become especially pronounced when deploying models to mobile devices or resource-constrained hardware environments. Consequently, exploring classic texture feature extraction techniques that operate with low computational complexity remains highly relevant for developing lightweight and efficient visual processing workflows[12].

Various approaches to facial gender recognition have been investigated in recent literature. Fajar [13] implemented a CNN model via the TensorFlow framework for gender classification; however, the study primarily prioritized network architecture without deeply assessing input preprocessing optimization. Similarly, Natun et al. [14] evaluated CNNbased gender identification, but system performance remained constrained by substantial computational requirements. Furthermore, Aditia et al. [15] deployed OpenCV for real-time gender detection, yet classification accuracy dropped noticeably when evaluated on low-resolution facial inputs. From this literature review, a critical research gap emerges: while existing research predominantly concentrates on increasing model complexity, limited empirical attention has been directed toward rigorously measuring the performance boundaries of baseline texture descriptors, such as Local Binary Pattern[16], [17], [18], [19], [20], when directly coupled with lightweight application frameworks like Kivy,[21], [22], [23]. The standard Local Binary Pattern operator provides an efficient mechanism for localized micro-texture extraction, whereas the Python-based Kivy framework offers a portable and responsive platform for interactive deployment. The novelty of this study resides in its application oriented system architecture, which seamlessly unifies a grayscale preprocessing pipeline, histogram equalization, Local Binary Pattern micro-texture feature extraction, and a threshold based numerical decision rule within a unified Kivy interface. This manuscript explicitly does not claim a novel classification algorithm or a modified Local Binary Pattern mathematical formulation. Instead, the primary scientific contribution lies in delivering an architectural blueprint and an empirical performance evaluation of the classification limitations of standard Local Binary Pattern descriptors within an interactive computer vision environment.

The problem formulation addresses how to structure an end-to-end Local Binary Pattern extraction workflow with decision rules and integrate this system into the Kivy framework to evaluate performance limits under uncontrolled visual variability. Accordingly, the primary objective of this research is to design, implement, and empirically evaluate the performance boundaries of the Local Binary Pattern method for facial gender identification using a Python-based Kivy application framework.

2. METHODS

2.1. Datasets and Research Materials

Facial image samples from the UTKFace dataset were selected using a random sampling without replacement method to ensure no identity leakage; meaning each individual is guaranteed to appear only once throughout the evaluation process. Prior to testing, the data was strictly separated into two independent groups: the Training Set and the Testing Set. The Training Set was used exclusively to define the reference texture mean values for each class. Subsequently, the algorithm's performance evaluation was conducted on the Testing Set, which was divided into three balanced and non-overlapping class scenarios: Scenario I (80 images), Scenario II (160 images), and Scenario III (320 images).

2.2. Local Binary Pattern Feature Extraction

Local Binary Pattern is a local feature extraction method that analyzes the spatial surface characteristics of an image by comparing the grayscale intensity of the central pixel against its neighboring pixels. This operator is executed on every pixel within the facial image to extract highly detailed local texture patterns.

2.2.1. Local Thresholding

Local thresholding is the foundational phase within the Local Binary Pattern framework. It computes the grayscale intensity variation between neighboring pixels and the central pixel. The thresholding function assigns a binary bit of 1 to non-negative intensity variations and 0 to negative variations. The mathematical expression for local binary thresholding is defined as:

$$S(I_n - I_c) = \begin{cases} 1, & I_n - I_c \geq 0 \\ 0, & I_n - I_c < 0 \end{cases} \quad \dots (1)$$

$S(I_n - I_c)$: Binary thresholding output (1 or 0).
I_c : Central pixel intensity.
I_n : Neighbor pixel intensity.
$(I_n - I_c)$: Intensity difference.

2.2.2. Local Binary Pattern

Subsequent to local thresholding, the resulting binary bits are aggregated into a single scalar value. Each binary bit is scaled by its corresponding binomial weighting factor according to its spatial sequence and summed to yield a decimal integer representation for the central pixel. The formulation for Local Binary Pattern encoding is expressed as:

$$(x_c, y_c) = \sum_{n=0}^{P-1} S\,(I_n - I_c) \cdot 2^n \quad \dots (2)$$

(x_c, y_c) : Central pixel coordinate.
P : Number of sampling points ($P = 8$).
R : Neighborhood radius
n : Neighbor index
2^n : Binomial weighting factor.

2.2.3. Classification Rule

Since the Local Binary Pattern method primarily functions to extract facial texture patterns, the system requires a clear decision rule to determine gender. This system employs a highly straightforward thresholding approach. In practice, the entire facial texture calculation derived from the Local Binary Pattern method is reduced into a single mean value. This mean value is then compared against a threshold value (θ), which is obtained from the midpoint between the average of the male facial group and the average of the female facial group, calculated separately from the training data.

$$\theta = \frac{\mu_{\text{male}} + \mu_{\text{female}}}{2} \ldots (3)$$

θ : Classification threshold parameter.
μ_{male} : Mean LBP intensity value for the male class.
μ_{female} : Mean LBP intensity value for the female class.
The final gender decision logic (Y) is determined as follows:

The classification decision logic is defined as:

$$Y = \begin{cases} 0 \quad (\text{Male}), & \text{if } X_{\text{LBP}} \geq \theta \\ 1 \quad (\text{Female}), & \text{if } X_{\text{LBP}} < \theta \end{cases} \ldots (4)$$

Y: Gender identification output label (0 = Male, 1 = Female).
X_{LBP}: Mean LBP intensity value.
θ: Classification threshold parameter.

2.2.4. Evaluation Metrics

Performance metrics are computed based on the Confusion Matrix parameters, where the female category ($Y = 1$) is defined as the positive class. The evaluation indicators include Accuracy, Precision, Recall, and F1-Score, formulated as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \ldots (5)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \ldots (6)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \ldots (7)$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \ldots (8)$$

Where TP (True Positive) is the number of female images correctly classified as female, TN (True Negative) is the number of male images correctly classified as male, FP (False Positive) is the number of male images incorrectly classified as female, and FN (False Negative) is the number of female images incorrectly classified as male.

2.3. Research Phases

The gender identification stage is the main process in the designed system to determine a person's gender based on a facial image. This stage consists of three important parts: image upload, gender identification, and result.

DOI: https://doi.org/10.55537/cosie.v5i4.1858

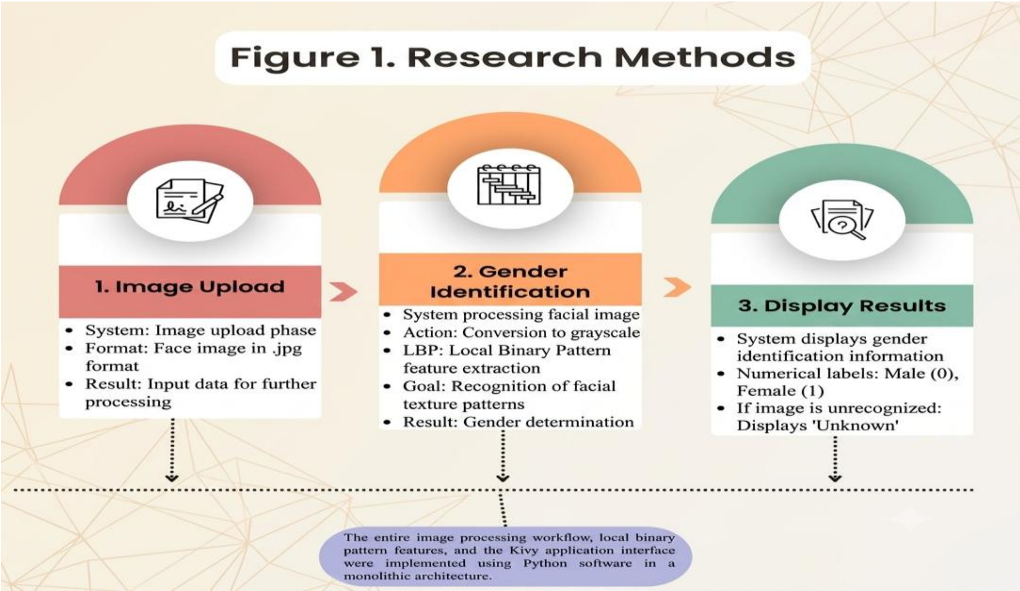

Figure 1. Research Methods

- In the image upload stage, the user inputs a facial image into the system in JPG format. The uploaded image becomes the input data to be processed further.
- The second stage is gender identification, in which the system processes the facial image. This process includes converting the image to grayscale and feature extraction using the Local Binary Pattern method to recognize facial texture patterns. The results of this processing are then used to determine gender.
- The final stage is the result, where the system displays the output in the form of gender identification information. The result is expressed in the form of a numerical label, namely 0 for male and 1 for female. If the facial image cannot be recognized, the system displays the information "Unknown".

3. RESULTS AND DISCUSSION

The Local Binary Pattern values for a grayscale image are determined based on the center position within a 3×3 block because only the central pixel has a complete set of eight neighbors and is used as a reference for comparison to form the binary pattern. This forms the basis of the Local Binary Pattern algorithm to ensure consistency, accuracy, and relevance to local texture patterns in image analysis. Center (g0) = 170 (the one in the middle) If a neighbor's value is ≥ the center (170), set the value to 1; if a neighbor's value is < the center (170), set the value to 0

Table 1. Local Binary Pattern Values

94	144	98
184	170	134
162	134	101

- $94 < 170 = 0$
- $144 < 170 = 0$
- $98 < 170 = 0$
- $134 < 170 = 0$
- $101 < 170 = 0$
- $134 < 170 = 0$
- $162 < 170 = 0$
- $184 < 170 = 1$

DOI: https://doi.org/10.55537/cosie.v5i4.1858

g1	g2	g3
g8	g0	g4
g7	g6	g5

1: Pixel Position Representation

0	0	0
1	170	0
0	0	0

2: Thresholding

1	2	4
128	170	8
64	32	16

3: Weighting and Score Calculation

Figure 2. displays the representation of pixel positions, thresholding, and weight calculation.

The Local Binary Pattern method works by comparing the intensity of the central pixel (g_0) with its eight surrounding pixels (g_1–g_8) in a 3×3 window, assigning a code of 1 if the neighboring pixel is greater than or equal to the central value, and 0 otherwise. After an RGB color image is converted into a grayscale matrix, a central pixel with a value of 170 is used as a reference to be compared with its neighbors through a thresholding process.

$$= (1 \times 2^7) + (0 \times 2^6) + (0 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (0 \times 2^0)$$
$$= 0 + 0 + 0 + 0 + 0 + 0 + 0 + 128 = 128.$$

The results of the calculation process show that the image yields a value of 128 based on the conversion from RGB to grayscale, followed by the Local Binary Pattern process. This value of 128 is obtained by comparing the intensity of the central pixel with that of its neighboring pixels, converting the results to binary form, and then converting them to decimal values according to their weighting.

3.1 Data Processing and Feature Extraction

```
# Deteksi Citra Wajah menggunakan Haar Cascade
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5)
```

Figure 3. Code Snippet for Face Detection Using Haar Cascade

The raw facial images undergo structured sequential preprocessing and extraction.

```
#  Meratakan kontras pencahayaan foto dasar
gray = cv2.equalizeHist(gray)
```

Figure 4. Illumination Histogram Equalization Process

Performing histogram equalization to balance the illumination contrast on the face.

```
#  Mengubah foto menjadi grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Figure 5. Image Color Space Conversion to Grayscale

Converting a color image (BGR) into grayscale.

```
# Menyusun Folder Fisik
dst_dir = os.path.join(path_dataset_out, nama_tahap, nama_gender, status_folder)
os.makedirs(dst_dir, exist_ok=True)
shutil.copy(img_src_path, os.path.join(dst_dir, filename))
```

Figure 6. Code Structure for Test File Storage Directory Management

This code functions to organize the test result files by automatically creating new folders based on the evaluation status, then copying the images into them so that the processed data is neatly arranged without disrupting the original files.

DOI: https://doi.org/10.55537/cosie.v5i4.1858

```python
# Ekstraksi Fitur LBP 3x3
height, width = face_roi.shape
lbp_matrix = np.zeros((height - 2, width - 2), dtype=np.uint8)

for i in range(1, height - 1):
    for j in range(1, width - 1):
        center = face_roi[i, j]
        code = 0
        code |= (face_roi[i-1, j-1] >= center) << 7
        code |= (face_roi[i-1, j]   >= center) << 6
        code |= (face_roi[i-1, j+1] >= center) << 5
        code |= (face_roi[i,   j+1] >= center) << 4
        code |= (face_roi[i+1, j+1] >= center) << 3
        code |= (face_roi[i+1, j]   >= center) << 2
        code |= (face_roi[i+1, j-1] >= center) << 1
        code |= (face_roi[i,   j-1] >= center) << 0
        lbp_matrix[i-1, j-1] = code

return np.mean(lbp_matrix), "SUCCESS"
```

Figure 7. Local Binary Pattern Texture Feature Extraction Algorithm on the Facial Region of Interest (ROI)

This function performs the 3x3 Local Binary Pattern feature extraction process manually on the facial image area (Region of Interest). The process compares the intensity of the central pixel against its eight neighboring pixels to form a binary code through bit-shift operations, then stores all these values into the Local Binary Pattern matrix before returning the mean value of the extracted image texture along with the execution success status.

```python
# Label Jenis Kelamin berdasarkan Threshold Otomatis
if pria_is_higher:
    label_prediksi = 0 if mean_lbp >= threshold else 1
else:
    label_prediksi = 1 if mean_lbp >= threshold else 0

label_text_pred = "Wanita" if label_prediksi == 1 else "Pria"

if label_asli == label_prediksi:
    status_folder = "berhasil"
    evaluasi = "BENAR"
else:
    status_folder = "gagal"
    evaluasi = "SALAH PREDIKSI"

y_true.append(label_asli)
y_pred.append(label_prediksi)
```

Figure 8. Gender Label Determination Logic Based on Automatic Thresholding

This code executes the automatic gender labeling process based on the comparison of the Local Binary Pattern value against the threshold value. The system compares the predicted label with the actual label to evaluate classification correctness, determines the output storage folder, and collects all predicted and actual data into an array for model evaluation metric calculations.

3.2 Implementation of the Application Interface

Figure 9. Interface Kivy

The application interface display illustrates the sequential workflow stages of system testing, starting from the command prompt window when the application is launched, the main user interface menu, the facial image file selection window, the image upload process, to the results screen displaying a green bounding box marking the facial area along with the gender identification status.

3.2.1 Test Scenario I (80 Test Data Points)

Based on the testing, the following results were obtained:

- Men: 7 images were correctly recognized, 33 images were misrecognized.
- Women: 17 images were correctly identified, 23 images were incorrectly identified

Figure 10. Men's Top Results

Table 7. Initial Analysis

	Identified: Male	Identified: Female	Total Actual
Actual: Male	7	33	40
Actual: Female	23	17	40
Total Identified	30	50	80

1. Accuracy

Measures the proportion of data that was correctly Identified compared to the total data.

$$Accuracy = \frac{17 + 7}{17 + 7 + 23 + 33} = \frac{24}{80} = 0{,}3\ x\ 100\% = 30\%$$

2. Precision

Measures the system's accuracy in predicting positive results. The higher the precision, the fewer errors there are in predicting positive results.

$$Precision = \frac{17}{17 + 33} = \frac{17}{50} = 0{,}34\ x\ 100\ \% = 34\%$$

3. *Recall*

Measures the extent to which the system is able to correctly identify positive data. A high recall indicates that the system does not miss many positive data points.

$$Recall = \frac{17}{17 + 23} = \frac{17}{40} = 0{,}42\ x\ 100\ \% = 42\%$$

4. *F1-Score*

It is the harmonic mean of precision and recall. The F1-Score is used to evaluate the balance between accuracy (precision) and completeness (recall).

$$F1 = 2\ \mathrm{X} \frac{(34\ \mathrm{X}\ 42)}{34 +\ 42}$$

$$F1 = 2\ \mathrm{X} \frac{1.428}{76} = 37.57\ x\ 100\ \% = 37\%$$

3.2.2 Test Scenario II (160 Test Data Points)

DOI: https://doi.org/10.55537/cosie.v5i4.1858

Based on the testing, the following results were obtained:
• Men: 25 images were correctly recognized, 55 images were misrecognized.
• Women: 44 images were correctly identified, 36 images were misidentified.

Figure 11. Results for the Two Menu

Table 8. Analysis of the Second Set of Results

	Identified: Male	Identified: Female	Total Actual
Actual: Male	25	55	80
Actual: Female	36	44	80
Total Identified	61	99	160

1. Accuracy
Measures the proportion of data that was correctly Identified compared to the total data.

$$Accuracy = \frac{44 + 25}{25 + 44 + 36 + 55} = \frac{69}{160} = 0.43\ x\ 100\ \% = 43\%$$

2. Precision
Measuring the system's accuracy in predicting positive data. The higher the precision, the fewer errors there are in predicting positive data.

$$Precision = \frac{44}{44 + 55} = \frac{44}{99} = 0.44\ x\ 100\ \% = 44\%$$

3. *Recall*
Measures how well the system can correctly identify positive data. A high recall indicates that the system does not miss many positive data points.
.

$$Recall = \frac{44}{44 + 36} = \frac{44}{80} = 0.55\ x\ 100\ \% = 55\%$$

4. *F1-Score*
It is the harmonic mean between precision and recall. The F1-score is used to assess the balance between accuracy (precision) and completeness (*recall*).

$$F1 = 2\text{X}\frac{(44\ \text{X}\ 55)}{44 +\ 55}$$

$$F1 = 2\text{X}\frac{2.420}{99} = 0.488\ x\ 100\ \% = 49\%$$

DOI: https://doi.org/10.55537/cosie.v5i4.1858

3.2.3 Test Scenario III (320 Test Data Points)
Based on the testing, the following results were obtained:

- Men: 30 images were correctly recognized, 130 images were misrecognized.
- Women: 69 images were correctly identified, 91 images were incorrectly identified.

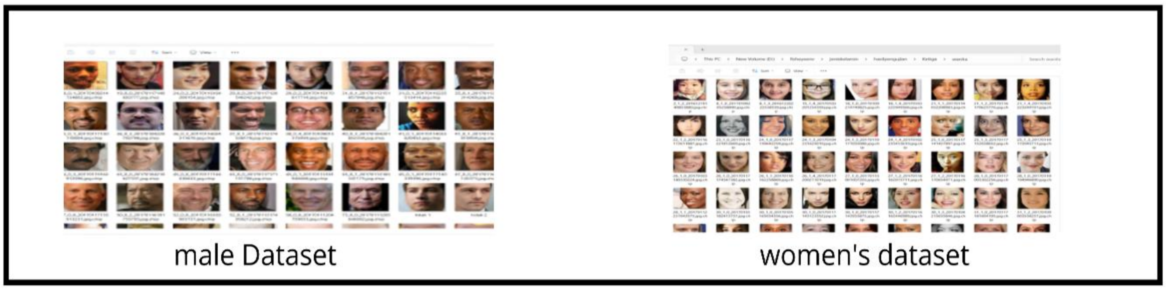

Figure 12. Results for the Three Men

Table 9. Analysis of the Third Set of Results

	Identified: Male	Identified: Female	Total Actual
Actual: Male	30	130	160
Actual: Female	91	69	160
Total Identified	121	199	320

1. Accuracy
Measuring how much of the data was correctly Identified compared to the total data.

$$Accuracy = \frac{69 + 30}{30 + 69 + 91 + 130} = \frac{99}{320} = 0.30\ x\ 100\ \% = 30\%$$

2. Precision
Measuring the system's accuracy in predicting positive results. The higher the precision, the fewer errors there are in predicting positive results.

$$Precision = \frac{69}{69 + 130} = \frac{69}{199} = 0.34\ x\ 100\ \% = 34\%$$

3. *Recall*
Measures how well the system can correctly identify positive data. A high recall indicates that the system does not miss many positive data points.

$$Recall = \frac{69}{69 + 91} = \frac{69}{160} = 0.43\ x\ 100\ \% = 43\%$$

4. *F1-Score*
It is the harmonic mean of precision and recall. The F1-score is used to evaluate the balance between accuracy (precision) and completeness (recall).
.

$$F1 = 2\ \mathrm{X} \frac{(34\ \mathrm{X}\ 43)}{34 + 43}$$

$$F1 = 2\ \mathrm{X} \frac{1.462}{77} = 0.379\ x\ 100\ \% = 38\%$$

3.3 Discussion of Results

3.3.1 Results per Scenario:

- Scenario 1 (80 data points): Accuracy 30%, Precision 34%, Recall 42%, F1-Score 37%
- Scenario II (160 data points): Accuracy 43%, Precision 44%, Recall 55%, F1-Score 49%
- Scenario III (320 data points): Accuracy 30%, Precision 34%, Recall 43%, F1-Score 38%

These three scenarios utilize mutually exclusive and non-overlapping image datasets, making each an independent random sample rather than a gradual expansion of the same dataset. Consequently, this inconsistent accuracy pattern (30%, 43%, 30%) cannot be interpreted as evidence that increasing the test dataset size improves system performance. Instead, this variation reflects the system's sensitivity to the specific composition of each random sample a direct consequence of feature representation that reduces facial texture to a single mean value, thereby discarding local texture distribution information. Factors such as variations in lighting, pose, and expression were observed as potential influences on the results, though they remain hypothetical as they were not evaluated through controlled subgroup experiments.

Despite these limitations in identification accuracy, all processing pipeline stages from grayscale conversion and Local Binary Pattern feature extraction to threshold based decision rules were successfully and consistently integrated into a single, functional Kivy application workflow from image input to label output. This demonstrates that successful software implementation (the integration of image processing stages into a working interface) must be strictly distinguished from the identification effectiveness of the Local Binary Pattern method itself, which remains constrained by the underlying feature representation.

4. CONCLUSION

Based on the implementation stages and technical analysis conducted, this research yields the following conclusions. Full Pipeline Integration The complete image processing workflow from grayscale conversion and Local Binary Pattern feature extraction to threshold based decision rules was successfully and consistently integrated into a fully functional Kivy based application, running seamlessly from image input to gender identification label output. Feature Representation Limitations Gender identification performance using the Local Binary Pattern method remains limited, achieving a peak accuracy of only 43% in the 160 sample testing scenario. The accuracy across the three test scenarios (30%, 43%, and 30%) showed no consistent correlation with dataset size. Instead, performance was primarily constrained by feature representation limitations that reduced facial texture to a single mean value, thereby discarding local spatial texture distribution details. Separation of Software and Algorithmic Performance Successful software implementation must be strictly distinguished from the identification effectiveness of the Local Binary Pattern method itself. This study successfully proves the technical feasibility of system integration; however, it demonstrates that the Local Binary Pattern method with the current simplified feature representation is insufficient for reliable gender identification under variable visual conditions. Uncontrolled Environmental Variables Factors such as variations in lighting, facial pose, and expression are suspected to influence identification outcomes. However, these variables were not evaluated through controlled subgroup experiments in this study and require further empirical validation. Baseline and Future Directions: This study establishes a foundational baseline regarding the performance boundaries of simple Local Binary Pattern feature representation within a Kivy framework. It highlights clear avenues for future research, such as exploring feature representations with richer information density and incorporating comparative baseline identification methods.

FUNDING

This research received no external funding

DATA AVAILABILITY STATEMENT

The data supporting the findings of this study are available from the corresponding author upon reasonable request

CONFLICT OF INTEREST

The author declares that there is no conflict of interest regarding the research, writing, or publication of this article

REFERENCES

[1] P. Tripathi, N. Kumar, M. Rai, P. K. Shukla, and K. N. Verma, "Applications of machine learning in agriculture," in *Smart Village Infrastructure and Sustainable Rural Communities*, IGI Global Scientific Publishing, 2023, pp. 99–118.

[2] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Comput. Sci.*, vol. 2, no. 3, p. 160, 2021.

[3] P. Salzabila, A. Hafiz, and A. Setiawan, "Pengembangan Sistem Evaluasi Otomatis Dalam E-Learning Menggunakan Machine Learning," *Pendas: Jurnal Ilmiah Pendidikan Dasar*, vol. 10, no. 2, pp. 641–649, 2025.

[4] A. P. Purnama, S. Amir, F. Syamsuddin, and F. Fitriansal, "Integrasi Algoritma Kecerdasan Buatan dalam Optimalisasi Efektivitas Komunikasi Visual," *DEIKTIS: Jurnal Pendidikan Bahasa dan Sastra*, vol. 5, no. 4, pp. 3036–3052, 2025.

[5] C. Cahyaningtyas, C. Gudiato, and M. Sari, "Deteksi Ekspresi Wajah Manusia Menggunakan Metode Convolutional Neural Network," *Jurnal Fasilkom*, vol. 15, no. 1, pp. 138–145, 2025.

[6] M. Fansyuri, "Klasifikasi Jenis Kelamin Berdasarkan Citra Wajah Menggunakan Algoritma Neural Network," *Journal of Research and Publication Innovation*, vol. 2, no. 4, pp. 3784–3795, 2024.

[7] M. A. Satriawan and W. Widhiarso, "Klasifikasi Pengenalan Wajah Untuk Mengetahui Jenis Kelamin Menggunakan Metode Convolutional Neural Network," *Jurnal Algoritme*, vol. 4, no. 1, pp. 43–52, 2023.

[8] A. L. S. Guntoro, E. Julianto, and D. Budiyanto, "Pengenalan Ekspresi Wajah Menggunakan Convolutional Neural Network," *Jurnal Informatika Atma Jogja*, vol. 3, no. 2, pp. 155–160, 2022.

[9] P. Musa, W. K. Anam, S. B. Musa, W. Aryunani, R. Senjaya, and P. Sularsih, "Pembelajaran Mendalam Pengklasifikasi Ekspresi Wajah Manusia dengan Model Arsitektur Xception pada Metode Convolutional Neural Network," *Rekayasa*, vol. 16, no. 1, pp. 65–73, 2023.

[10] M. S. Negara, M. Irzan, A. D. Haqqi, and F. Bimantoro, "Implementasi Convolutional Neural Network pada Multi-label Classification Wajah Manusia Berdasarkan Usia, Gender, dan Ras," *DIELEKTRIKA*, vol. 11, no. 2, 2024.

[11] A. Sopian, D. Setiadi, A. Suryatno, and R. Agustino, "Computer Vision: Deteksi Masker Wajah Prediksi Usia Jenis Kelamin dengan Teknik Deep Learning Menggunakan Algoritma Convolutional Neural Network (CNN)," *Jurnal Teknologi Informatika dan Komputer*, vol. 10, no. 2, pp. 720–733, 2024.

[12] C. A. Marcelio, M. A. Azzikra, D. P. Mufazzal, A. R. Illahi, S. Al Husain, and A. Abdiansah, "Aplikasi Analisis Wajah, Klasifikasi Gender dan Prediksi Usia Menggunakan Deep Learning pada Dataset Citra Wajah Manusia," *Jurnal Media Infotama*, vol. 20, no. 1, pp. 378–383, 2024.

[13] Y. I. Fajar, "Penerapan Tensorflow Dalam Prediksi Jenis Kelamin Dengan Menggunakan Algoritma CNN," *JICode: Jurnal Informatika dan Komputer*, vol. 2, no. 1, pp. 122–129, 2025.

[14] N. C. I. Natun, M. A. Santhia, and Y. R. Kaesmetan, "Identifikasi Pengenalan Wajah Berdasarkan Jenis Kelamin Menggunakan Metode Convolutional Neural Network (CNN)," *Journal of Technology and Informatics (JoTI)*, vol. 6, no. 1, pp. 50–57, 2024.

[15] R. Aditia, M. S. Arrafiq, and F. Afandi, "Implementasi Opencv Face Recognition Pada Real-Time Deteksi Umur Dan Jenis Kelamin Menggunakan Python dengan Metode Klasifikasi," *Jurnal Garuda Pengabdian Kepada Masyarakat*, vol. 1, no. 2, pp. 46–55, 2023.

[16] A. L. Arda, M. F. Said, E. Y. Puspaningrum, and S. Alam, "Klasifikasi Motif Batik Yogyakarta menggunakan Neural Network dengan Fitur Local Binary Pattern," *SemanTIK: Teknik Informasi*, vol. 11, no. 2, 2025.

[17] V. Betcy Thanga Shoba and I. Shatheesh Sam, "Empirical mode decomposition and local binary pattern based feature extraction for face recognition," *The Imaging Science Journal*, vol. 72, no. 6, pp. 791–807, 2024.

[18] U. L. Sowjanya and R. Krithiga, "Decoding student emotions: An advanced CNN approach for behavior analysis application using uniform local binary pattern," *IEEE Access*, vol. 12, pp. 106273–106284, 2024.

[19] A. A. Suaib, I. I. Tritosmoro, and N. Ibrahim, "Identifikasi Covid-19 Berdasarkan Citra X-Ray Paru-Paru Menggunakan Metode Local Binary Pattern Dan Random Forest," *Jurnal Teknik Informasi dan Komputer (Tekinkom)*, vol. 5, no. 2, p. 419, 2022.

[20] K. M. Hosny, W. M. El-Hady, F. M. Samy, E. Vrochidou, and G. A. Papakostas, "Multi-class classification of plant leaf diseases using feature fusion of deep convolutional neural network and local binary pattern," *IEEE Access*, vol. 11, pp. 62307–62317, 2023.

[21] N. T. A. Pinem, I. P. G. Budisanjaya, S. Sumiyati, and N. N. Sulastri, “Rancang Bangun Aplikasi Mobile Berbasis Kivy untuk Estimasi Biomassa Tanaman Microgreen: Design and Development of a Kivy Based Mobile Application for Microgreen Plant Biomass Estimation,” *Jurnal BETA (Biosistem dan Teknik Pertanian)*, vol. 13, no. 2, pp. 317–324, 2025.
[22] Y. Aufar and I. S. Sitanggang, “Face recognition based on Siamese convolutional neural network using Kivy framework,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, no. 2, pp. 764–772, 2022.
[23] F. Suryani, “Pemanfaatan Framework Kivy untuk Pengembangan Aplikasi Pendidikan Interaktif Berbasis Android,” in *Prosiding Seminar Nasional Amikom Surakarta*, 2024, pp. 562–573.