



Comparative Analysis of XGBoost and CatBoost for Detecting AI-Generated Synthetic Faces Using Texture and Color Features

Indra Firmansyah✉

(Universitas Malikussaleh,
Lhokseumawe, Indonesia)

Defry Hamdhana

(Universitas Malikussaleh,
Lhokseumawe, Indonesia)

Wahyu Fuadi

(Universitas Malikussaleh,
Lhokseumawe, Indonesia)

OPEN ACCESS

ARTICLE HISTORY

Received: June 4, 2026

Revised: July 23, 2026

Accepted: August 18,
2026

KEYWORDS

CatBoost;

GLCM;

Image classification;

XGBoost;

YCbCr.

ABSTRACT

Purpose – The rapid advancement of Generative AI produces near-perfect synthetic face images, posing fraud risks. This study compares XGBoost and CatBoost for detecting real versus synthetic faces using texture and color features.

Methods – Texture features were extracted using GLCM with four orientations (0°, 45°, 90°, 135°), generating 16 features, while YCbCr mean and standard deviation provided 6 color features (22 total). A balanced 750-image dataset (real and AI from GPT Image 2, Nano Banana, Leonardo AI, Canva, Dreamina AI) was split 80:20 and normalized with StandardScaler.

Findings – On an asymmetric, source-shifted test split, where real test images were captured independently while synthetic test images came from the same generators as training, CatBoost outperformed XGBoost, achieving 84.67% accuracy (vs. 80.67%), 83.33% precision (vs. 78.05%), 86.67% recall (vs. 85.33%), and 84.97% F1-score (vs. 81.53%), reducing False Positives by 27.78% and False Negatives by 9.09%. However, repeated 5-fold stratified cross-validation (5×10 repeats) revealed only marginal, non-significant differences (0.05-0.87 percentage points, $p > 0.09$).

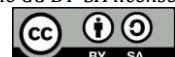
Research implications – Although the moderate dataset excludes recent architectures such as StyleGAN, findings affirm the feasibility of efficient, interpretable detection through feature engineering. Since within-domain cross-validation shows no significant algorithmic difference, the observed cross-domain gap should be regarded as preliminary, pending validation through repeated cross-domain evaluations across diverse AI platforms.

Originality – The novelty lies in comparing XGBoost and CatBoost on multi-orientation GLCM and YCbCr feature fusion for contemporary AI-generated images, identifying `cr_std` and `contrast_0` as stable, interpretable discriminative markers.

Correspondence Author: ✉indra.220170088@mhs.unimal.ac.id

To cite this article : Firmansyah, I., Hamdhana, D., & Fuadi, W. (2026). Comparative Analysis of XGBoost and CatBoost for Detecting AI-Generated Synthetic Faces Using Texture and Color Features. *Journal of Deep Learning, Computer Vision and Digital Image Processing*, 4(3), 77–93. <https://doi.org/10.61255/decoding.v4i3.1677>

This is an open access article under the CC BY-SA license



INTRODUCTION

The development of Artificial Intelligence technology in recent years has shown rapid progress, particularly in the development of Generative Models. These models are capable of creating synthetic images that possess visual characteristics very similar to real images, making them difficult to distinguish with the naked eye [1]. This capability is widely utilized in digital art, creative industries, and entertainment[2]. However, it has also given rise to the Deepfake phenomenon, which has the potential to be misused for fraud, identity theft, and the spread of false information [3]. Currently, distinguishing real and fake face images solely through direct visual observation is extremely difficult. Research by Velásquez-Salamanca et al.[4] proved that humans in general are unable to accurately identify the source of images, where nearly 40% of Generative AI-generated images failed to be recognized and were instead classified as real photos. This is due to the continuously evolving capability of Generative AI to produce high-quality images that appear highly realistic. Although visually perfect, theoretically, AI-generated face images still possess anomalies and statistical distributions that differ from real images. Research by Osińska et al. [5] states that the characteristics of real face images can be recognized from natural skin imperfections, consistent shadow details, and naturalness in human expressions and eyes. Conversely, AI-generated images often leave traces of digital defects, such as overly smooth skin texture, excessively sharp edge details, as well as anomalies in eye light reflections and hair transitions that can be captured statistically by machine learning algorithms.

Various previous studies have attempted to design automatic systems for detecting synthetic face images. The most dominant approach used is Deep Learning architectures, particularly Convolutional Neural Networks (CNNs). Research by Mu et al.[6] showed that CNN models are proven capable of classifying real and deepfake face images with an accuracy rate of 91%. In addition to CNNs, Vision Transformer (ViT) and EfficientNet-based approaches have also been applied for similar tasks and have achieved very high accuracy, albeit with significantly greater computational complexity [7]. Although proven to achieve high accuracy, these Deep Learning approaches have several technical limitations [8]. Deep Learning algorithms tend to require large computational resources and relatively long training times. In addition, the complexity of the models makes them prone to overfitting and gives them a black-box nature, making the decision-making process very difficult to interpret [9].

Meanwhile, feature engineering-based machine learning approaches offer efficiency and transparency compared to Deep Learning [8]. However, research on the combination of multi-orientation texture features and color features simultaneously for deepfake detection remains limited. GLCM is a statistical texture analysis method that characterizes an image by calculating how frequently pairs of pixel intensities occur at a specified distance and orientation, from which descriptors such as contrast, energy, homogeneity, and entropy can be derived[10]. In this study, texture features were extracted using the Gray Level Co-occurrence Matrix (GLCM) method, which has been proven effective in analyzing facial skin texture[11], and has also been successfully applied to various texture-based image classification tasks [12], whereas color features using the YCbCr color space offer high stability in skin color analysis due to their ability to effectively separate luminance and chrominance components [13]. Unlike the RGB color space, in which color and illumination information are mixed together across all three channels, YCbCr separates brightness (Y) from color information (Cb, Cr), making the resulting color features more robust to variations in lighting conditions[14].

For the classification stage, this study relies on gradient boosting, an ensemble learning technique that builds a predictive model incrementally by adding decision trees in sequence, where each new tree is trained to correct the residual errors of the previous ensemble through gradient descent optimization of a loss function [15]. Among the two gradient boosting implementations compared in this study. XGBoost has been validated to offer robust performance in classifying high-dimensional tabular data and has built-in regularization, achieved through a regularized objective function and a parallelized, greedy tree-splitting strategy that improve both predictive accuracy and computational efficiency[16][17]. Meanwhile, CatBoost offers theoretical advantages through Ordered Boosting,

which reduces prediction shift caused by target leakage during training, and symmetric (oblivious) trees that apply the same splitting rule across each depth level, resulting in resistance to overfitting[18]. However, no research has systematically compared the performance of these two gradient boosting algorithms on synthetic face image detection tasks based on GLCM and YCbCr feature fusion. Recent comparative studies have applied XGBoost and CatBoost in other domains, such as air pollution classification [19], yet their comparative performance on synthetic face detection using texture and color features remains unexplored.

This study aims to analyze and compare the performance of two gradient boosting algorithms, namely XGBoost and CatBoost, in classifying real and Generative AI-generated face images. The approach employed relies on GLCM feature extraction to capture skin texture patterns and YCbCr to analyze color distribution, which are then combined into a single feature vector. This feature engineering-based approach is chosen as a more efficient and transparent alternative compared to deep learning architectures, which have a black-box nature and high computational complexity[9]. The novelty of this research lies in the simultaneous use of multi-orientation GLCM and YCbCr feature combinations, which has not been widely explored for synthetic face image detection tasks compared to deep learning approaches that dominate recent literature. Furthermore, the synthetic image dataset in this study was generated using the latest generation of generative AI platforms such as GPT Image 2, Nano Banana, Leonardo AI, Canva, and Dreamina AI, making it relevant to the characteristics of the most current generative artifacts. This study also compares the performance of XGBoost and CatBoost in a controlled manner in the context of this feature fusion, as well as quantitatively identifying the most discriminative features across both models. This study hypothesizes that CatBoost, with its Ordered Boosting mechanism and symmetric trees, will outperform XGBoost in terms of accuracy and resistance to overfitting on medium-sized datasets.

METHOD

This research method consists of seven main stages, including Problem Identification and Literature Study, data collection and preprocessing, feature engineering through texture feature extraction using the Gray Level Co-occurrence Matrix (GLCM) and color feature extraction using the YCbCr color space, feature fusion and data normalization, classification and comparison of XGBoost and CatBoost algorithms, and model evaluation. The approach used is quantitative experimental with a comparative design to compare the performance of two gradient boosting algorithms on binary classification tasks. Figure 1 presents the overall research method flow.

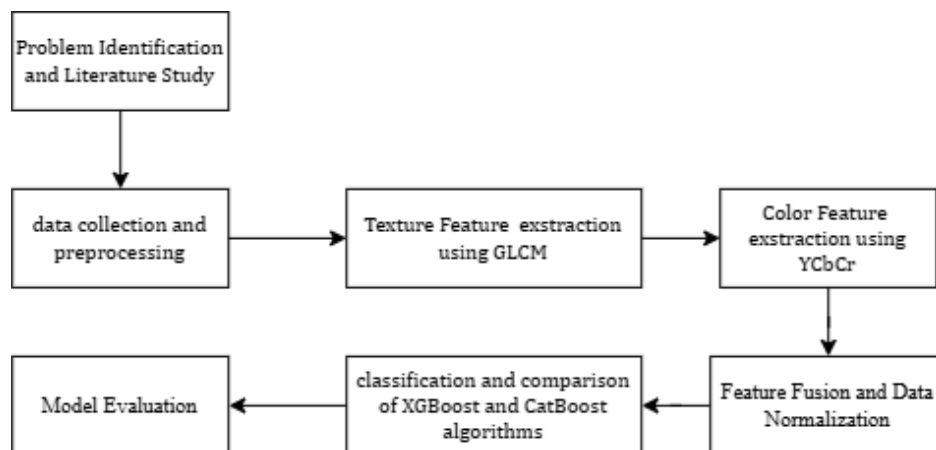


Figure 1. Method Research

The subjects in this study were individuals who served as the source of real face images taken directly using a camera. The direct image capture involved 75 individuals, with variations in lighting conditions, viewing angles, and facial expressions to increase data diversity. As detailed below, this direct data collection was specifically designated as test data to evaluate the model's robustness against real-world conditions.

The dataset used in this study consisted of 750 face images with balanced class composition, consisting of 375 real images and 375 Generative AI-generated images. For the real images, the primary source was the 'Human Faces Dataset' by Kaustubh Dhote, obtained from Kaggle (<https://www.kaggle.com/datasets/kaustubhdhote/human-faces-dataset>), from which 125 real images were sampled. Additional real images were sourced from publicly accessible profiles on Instagram and Pinterest (125 images from Pinterest and 50 from Instagram) to increase data diversity. The remaining 75 real images were obtained through direct camera capture involving 75 individuals, as described above. For the synthetic images, 375 images were generated using five different platforms: GPT Image 2, Nano Banana, Leonardo AI, Canva, and Dreamina AI. The specific distribution across these platforms was 80, 70, 60, 75, and 90 images, respectively, totaling 375 synthetic images. Before entering the feature extraction stage, all images underwent preprocessing in the form of resizing to 224×224 pixels using the OpenCV library. Resizing all images to a uniform dimension is a standard preprocessing step that ensures consistent input size for feature extraction while reducing computational load [20]. The dataset was then partitioned with an 80:20 ratio, resulting in 600 images for training and 150 images for testing. Of the 150 test data, 75 were real face images obtained directly from camera shots, and the remaining 75 were synthetic images. No public source real images were included in the test set to ensure a strict evaluation of the model's performance on independently captured real world data.

It is important to clarify that the evaluation design is asymmetric or partially cross-domain, rather than fully cross-domain. The real-image test set (75 images) was sourced from independent direct camera captures, representing a clear source-domain shift from the publicly sourced real training images. However, the synthetic test images (75 images) were drawn from the same five AI generators used to produce the synthetic training images, split proportionally at the 80:20 ratio (see Table 1, AI Class - Training/Test Split). Consequently, only the real-image class undergoes a true domain shift, while the synthetic class is evaluated within the same generator families. Detailed data characteristics, including the generator-wise distribution for training and testing, are presented in Table 1.

To further address the possibility of content leakage or near duplicate images between the training and test subsets of the synthetic class, each generated image was produced from a distinct textual prompt that varied multiple subject attributes, including age, gender, ethnicity, skin tone, hair color and style, religious attire, facial expression, background setting, and lighting condition. Although most prompts specified a front-facing, half-body (chest-up) portrait, framing was not held constant across the dataset; a subset of images depicts a slightly angled or three-quarter pose rather than a strictly frontal orientation, introducing further variation in pose. Because each image was generated independently from a uniquely varied prompt rather than from repeated prompts, fixed seeds, or a shared subject identity, the likelihood of identical or near-identical subjects, poses, or visual content recurring across the training and test partitions is considered low. In addition, the 80:20 split was performed as a random partition within each generator's image pool rather than by prompt family or subject grouping, further reducing the risk of systematic overlap between the two subsets.

Table 1. Dataset Characteristics

Characteristic	Value
Total Dataset	750 images
Number of Classes	2
Real Class	375 images (125 from Kaggle, 50 from social media, 125 from Pinterest, 75 from direct camera shots)
AI Class	375 images (80 from GPT Image 2, 70 from Nano Banana, 60 from Leonardo AI, 75 from Canva, 90 from Dreamina AI)
AI Class - Training Split (80%)	300 images: 64 (GPT Image 2), 56 (Nano Banana), 48 (Leonardo AI), 60 (Canva), 72 (Dreamina AI)
AI Class - Test Split (20%)	75 images: 16 (GPT Image 2), 14 (Nano Banana), 12 (Leonardo AI), 15 (Canva), 18 (Dreamina AI)
Training Data	600 images (300 real, 300 AI)
Test Data	150 images (75 real, 75 AI)
Split Ratio	80:20

Characteristic	Value
Resized Size	224×224 pixels

This study was conducted with ethical approval from the institutional review board of Malikussaleh University. Written informed consent was obtained from all 75 individuals who participated in direct image capture. Their images are stored in encrypted local storage and used exclusively for the purposes of this academic research. No personal identifying information is published or shared. For the publicly sourced images from Kaggle, Instagram, and Pinterest, the authors adhered to strict data protection protocols in line with the institutional ethics guidelines for secondary use of publicly available data. Only images from public accounts were collected manually, and all personally identifiable information (names, usernames, metadata) was neither collected nor stored. Facial regions were extracted, and the original images and metadata were permanently deleted after feature extraction. This anonymized, non-redistributable use of publicly available facial images for academic research was reviewed and deemed consistent with the ethical approval granted for this study by the institutional review board of Malikussaleh University.

To mitigate potential confounding effects arising from the different sources of real training and test images, all facial images underwent a consistent preprocessing pipeline. Specifically, each image was cropped with a 1:1 aspect ratio centered on the facial region and resized to 224×224 pixels using OpenCV. This cropping procedure ensures that feature extraction using GLCM and YCbCr primarily captures facial texture and skin color, effectively eliminating variations caused by irrelevant backgrounds or inconsistent framing. However, we acknowledge that this preprocessing step does not entirely eliminate source-domain differences, such as JPEG compression levels, camera sensor characteristics, or lighting conditions, which inherently differ between direct camera captures and publicly sourced images.

Despite these residual differences, the asymmetric dataset design in which the training set consists of publicly sourced real images and the test set consists of independently captured real images was intentionally retained. This design serves as a realistic stress test to evaluate the model's cross-domain generalization capability. In real-world forensic applications, detection models are rarely tested on images that share the exact same source characteristics as the training data. Therefore, evaluating the model on an independent, out-of-source test set provides a more rigorous and ecologically valid estimate of its practical performance. Consequently, the reported accuracy scores should be interpreted primarily as indicators of cross-domain robustness rather than as measures of within-domain classification performance. Figure 2 presents sample face images from each class used in this study. Visually, real and Generative AI generated face images are difficult to distinguish with the naked eye, therefore, numerical feature extraction based on texture and color is required for the classification process.

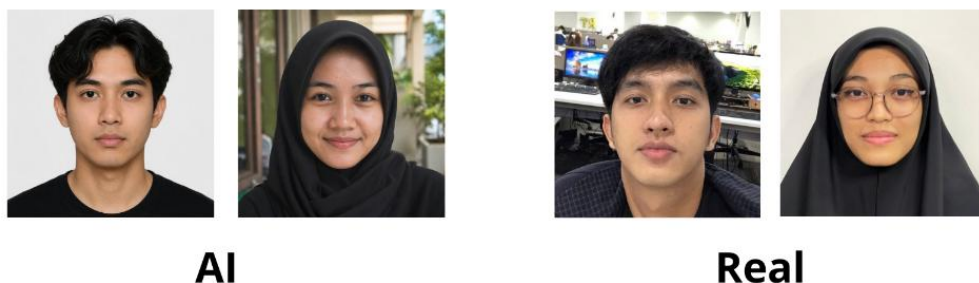


Figure 2. Image Dataset

Texture features were extracted using the Gray Level Co-occurrence Matrix (GLCM) method, which models the spatial relationships between pairs of pixels based on distance and orientation angle [8]. Extraction was performed with a pixel distance of 1 and four orientation angles, namely 0°, 45°, 90°, and 135°. A pixel distance of 1 combined with these four orientations is a widely adopted

configuration, as it captures fine-grained local texture variation while keeping the computational cost of the co-occurrence matrix low. From each orientation, four statistical features were calculated, namely energy, contrast, homogeneity, and entropy [21], resulting in a total of 16 texture features. These four features were calculated using Equations (1) to (4).

$$Energy = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [P(i, j)]^2 \quad (1)$$

$$Contrast = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i - j)^2 P(i, j) \quad (2)$$

$$Homogeneity = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{P(i, j)}{1 + (i - j)^2} \quad (3)$$

$$Entropy = - \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P(i, j) \log_2 P(i, j) \quad (4)$$

Color features were extracted using the YCbCr color space, which separates the luminance component (Y) from the chrominance components (Cb and Cr). Extraction was performed by calculating the mean and standard deviation of each Y, Cb, and Cr channel, resulting in 6 color features. The mean represents the central tendency of each channel, while the standard deviation captures the dispersion of pixel values, together providing a compact yet informative summary of the color distribution [22]. The calculation of the mean and standard deviation follows Equations (5) and (6), respectively.

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (5)$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad (6)$$

All extracted features, namely 16 texture features and 6 color features, were combined into a single feature vector of 22 dimensions. Detailed Composition of Extracted Features in Table 2. To prevent data leakage, the feature normalization procedure was performed exclusively using the training set. Specifically, the StandardScaler was fitted only on the 600 training images to compute the mean and standard deviation for each of the 22 features. The same fitted scaler was then applied to transform both the training set and the 150 test images. This ensures that the test set does not influence the scaling parameters, preserving the integrity of the model evaluation. Feature scaling to zero mean and unit variance prevents features with larger numeric ranges from dominating the learning process and improves the stability of model training [23].

Table 2. Composition of Extracted Features

Feature Type	Extraction Method	Feature Names	Count
Texture	GLCM (0°, 45°, 90°, 135°)	contrast_0, contrast_45, contrast_90, contrast_135, energy_0, energy_45, energy_90, energy_135, homogeneity_0, homogeneity_45, homogeneity_90, homogeneity_135, entropy_0, entropy_45, entropy_90, entropy_135	16
Color	YCbCr (mean & std)	y_mean, y_std, cb_mean, cb_std, cr_mean, cr_std	6
Total Count			22

The research procedure begins with problem identification and literature study, followed by data collection and preprocessing. The third stage is texture feature extraction using GLCM, followed by the fourth stage of color feature extraction using YCbCr. Fifth, all features are combined and normalized. Sixth, classification and comparison of XGBoost and CatBoost algorithms are performed. Finally, model evaluation is performed using a confusion matrix. The entire research process was conducted over a period of three months. Model evaluation uses a confusion matrix with accuracy, precision, recall, and F1-score metrics, which are calculated using Equations (7) to (10). These

metrics are commonly used together in binary classification tasks because accuracy alone can be misleading when the cost of False Positives and False Negatives differs, whereas precision, recall, and F1-score provide a more complete picture of a model's error trade-offs [24].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (7)$$

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (10)$$

Feature importance analysis was performed using the gain metric, which quantifies each feature's average contribution to reducing the loss function whenever it is used to split a node across all trees in the ensemble [25]. Both algorithms were configured with comparable parameters to ensure a fair comparison. Table 3 summarizes the comparison of the parameters for both models. These parameters were selected based on initial trials on the training set only, monitoring the training validation loss curve within the training data (not the held-out test set), to produce a stable model that is not prone to overfitting. Hyperparameter optimization was performed using Grid Search with 5-fold stratified cross-validation (StratifiedKFold, random_state=42) exclusively on the 600 training images. The test set (150 images) was not used at any stage of hyperparameter selection. The search space included n_estimators/iterations [200, 300], max_depth/depth [3, 4, 5], learning_rate [0.01, 0.05, 0.1], and algorithm-specific regularization terms (reg_alpha, reg_lambda for XGBoost; l2_leaf_reg for CatBoost), with subsample and colsample_bytree additionally tuned for XGBoost. The best combination was chosen based on the highest average cross-validation accuracy across the five folds.

Table 3. Model Parameters for XGBoost and CatBoost

Parameter	XGBoost	CatBoost
Tuning Method	Grid Search (5-fold CV)	Grid Search (5-fold CV)
Tuning Data	Training set only (600 images)	Training set only (600 images)
Number of trees	n_estimators=300	iterations=200
Maximum tree depth	max_depth=5	depth=5
Learning rate	learning_rate=0.05	learning_rate=0.05
Sample fraction per tree	subsample=0.9	-
Feature fraction per tree	colsample_bytree=0.8	-
Minimum child weight	min_child_weight=2	-
L2 Regularization	reg_lambda=1.5	l2_leaf_reg=3.0
L1 Regularization	reg_alpha=0.1	-
Loss function	objective='binary:logistic'	loss_function='Logloss'
Evaluation metric	eval_metric='logloss'	eval_metric='Accuracy'
Random seed	random_state=42	random_seed=42

Hyperparameter optimization was performed using Grid Search with 5-fold stratified cross-validation (StratifiedKFold, random_state=42) exclusively on the 600 training images. The held-out test set of 150 images was not used at any stage of hyperparameter selection to prevent data leakage. The search space for XGBoost included n_estimators [200, 300], max_depth [3, 4, 5], learning_rate [0.01, 0.05, 0.1], subsample [0.8, 0.9], colsample_bytree [0.8, 0.9], min_child_weight [1, 2], reg_alpha [0, 0.01, 0.1], and reg_lambda [0.5, 1.0, 1.5]. For CatBoost, the search space included iterations [200, 300], depth [3, 4, 5], learning_rate [0.01, 0.05, 0.1], and l2_leaf_reg [1.0, 3.0, 5.0]. The best combination for each model was chosen based on the highest average cross-validation accuracy across the five folds.

Table 4. Hardware Specifications for Experimental Reproducibility

Component	Specification
Laptop Model	MSI Modern 14 B5M

Component	Specification
Processor	AMD Ryzen 5 5500U @ 2.10 GHz
RAM	8 GB
Storage	512 SSD

The Grid Search identified optimal hyperparameters for XGBoost: `colsample_bytree=0.8`, `learning_rate=0.05`, `max_depth=5`, `min_child_weight=2`, `n_estimators=300`, `reg_alpha=0.1`, `reg_lambda=1.5`, and `subsample=0.9`, achieving a cross-validation accuracy of 75.67%. For CatBoost, the optimal parameters were `depth=5`, `iterations=200`, `l2_leaf_reg=3.0`, and `learning_rate=0.05`, achieving a cross-validation accuracy of 75.83%. The comparable cross-validation scores indicate that both algorithms performed similarly under their respective optimal configurations on the training data. All experiments were conducted using the hardware specifications and software environment summarized in Table 4 and Table 5, respectively. A consistent random seed of 42 was applied to the data splitting procedure to ensure reproducible train-test partitions, while the random seeds for the individual models are specified in Table 3. The training time for each model was measured using Python's time module to support computational efficiency analysis.

Table 5. Software Environment and Library Versions

Component	Version / Setting
Operating System	Windows 11 (64-bit)
Python	3.10.0
OpenCV	4.13.0.92
scikit-learn	1.7.2
XGBoost	3.2.0
CatBoost	1.2.10

This research has several methodological limitations that need to be considered. First, the dataset size used (750 images) is considered moderate, and the range of AI generators tested, while diverse, does not represent the entire spectrum of rapidly growing generative models, such as StyleGAN or Diffusion Models. Consequently, the generalization of these findings still requires further validation on larger and more varied datasets. Second, the process of resizing all images to 224×224 pixels, although necessary for standardization, has the potential to remove fine micro-scale artifacts that could serve as important discriminative markers in high-resolution synthetic images. Third, this study has not conducted a direct comparison with the latest deep learning architectures, such as EfficientNet or Vision Transformer, on the same dataset. Fourth, while Grid Search was used for hyperparameter tuning, the search grids were predefined and may not have covered the global optimum; more extensive searches or Bayesian optimization could potentially yield further marginal gains.

RESULTS AND DISCUSSION

The texture feature extraction process using GLCM with four orientation angles (0°, 45°, 90°, 135°) and color features using the YCbCr color space successfully produced 22 feature vectors per image. The feature extraction results from all images formed a numerical dataset with 22 features and 750 rows, which were then normalized using StandardScaler before being used as input for the classification process. Variations in feature values across images indicated differences in texture characteristics and color distribution between real face images and Generative AI-generated face images.

The Grid Search procedure identified optimal hyperparameters for each model. XGBoost achieved a cross-validation accuracy of 75.67% with parameters `colsample_bytree=0.8`, `learning_rate=0.05`, `max_depth=5`, `min_child_weight=2`, `n_estimators=300`, `reg_alpha=0.1`, `reg_lambda=1.5`, and `subsample=0.9`. CatBoost achieved a cross-validation accuracy of 75.83% with `depth=5`, `iterations=200`, `l2_leaf_reg=3.0`, and `learning_rate=0.05`. The comparable cross-validation scores (75.67% vs 75.83%) suggest that both algorithms achieved similar within-domain performance under their respective optimal configurations, consistent with the subsequent repeated cross-validation findings. The XGBoost model was trained using 600 training data with 22 features. The training process built 300 decision trees sequentially with parameters as listed in Table 3. Evaluation

on 150 test data produced a confusion matrix presented in Figure 3 and evaluation metrics in Table 6.

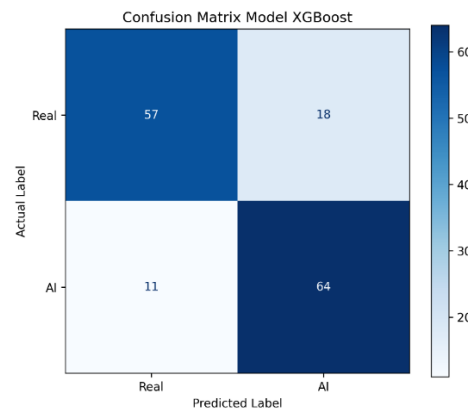


Figure 3. Confusion Matrix XGBoost Model

Based on Figure 3, from 75 real face images, the model successfully predicted 57 images correctly and 18 images were incorrectly predicted as AI (False Positive). Meanwhile, from 75 Generative AI-generated face images, the model successfully predicted 64 images correctly and 11 images were incorrectly predicted as real (False Negative).

Table 6. XGBoost Model Evaluation Metrics

Metric	Value
Accuracy	80.67%
Precision (AI Class)	78.05%
Recall (AI Class)	85.33%
F1-Score (AI Class)	81.53%

The XGBoost model achieved an accuracy of 80.67% on the test data. The recall value of 85.33% indicates that the model was able to capture most of the AI face images in the test data. However, the accuracy gap between the training data (99.83%) and test data (80.67%) of 19.16 percentage points indicates considerable overfitting. The CatBoost model was trained with a parameter configuration comparable to XGBoost (Table 3). Evaluation on the same test data produced a confusion matrix in Figure 4 and evaluation metrics in Table 7.

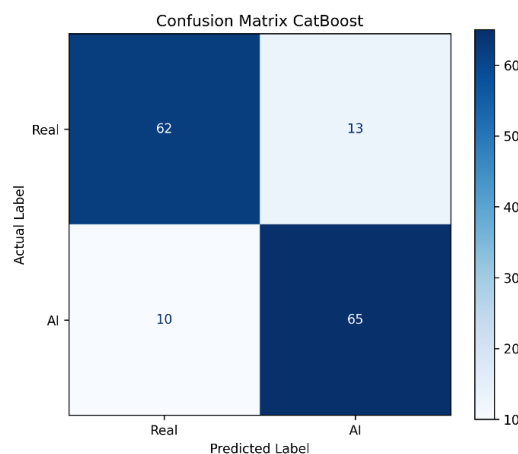


Figure 4. Confusion Matrix CatBoost Model

Based on Figure 4, from 75 real face images, the model successfully predicted 62 images correctly and 13 images were incorrectly predicted as AI. Meanwhile, from 75 Generative AI-generated face images, the model successfully predicted 65 images correctly and only 10 images were incorrectly predicted as real (False Negatives).

Table 7. CatBoost Model Evaluation Metrics

Metric	Value
Accuracy	84.67%
Precision (AI Class)	83.33%
Recall (AI Class)	86.67%
F1-Score (AI Class)	84.97%

The CatBoost model achieved an accuracy of 84.67% on the test data, outperforming XGBoost by 4.00 percentage points. CatBoost also demonstrated higher precision (83.33% vs. 78.05%), higher recall (86.67% vs. 85.33%), and higher F1-score (84.97% vs. 81.53%) on this asymmetric test split. The improvement in recall from 85.33% to 86.67% indicates that CatBoost was slightly better at capturing AI-generated face images on this particular split, while the precision improvement from 78.05% to 83.33% shows that CatBoost was substantially more reliable in avoiding the misclassification of real images as synthetic.

Table 8. Comparison of XGBoost and CatBoost Performance

Metric	XGBoost	CatBoost	Improvement (percentage points)
Accuracy	80.67%	84.67%	+4.00
Precision (AI)	78.05%	83.33%	+5.28
Recall (AI)	85.33%	86.67%	+1.34
F1-Score (AI)	81.53%	84.97%	+3.44
False Negative	11	10	-1 (9.09% rel.)
False Positive	18	13	-5 (27.78% rel.)

Table 8 presents a complete comparison of XGBoost and CatBoost performance on the asymmetric source-shifted test split. CatBoost consistently outperformed XGBoost across all evaluation metrics on this split, with improvements of 4.00 percentage points in accuracy, 5.28 percentage points in precision, 1.34 percentage points in recall, and 3.44 percentage points in F1-score. The 27.78% relative reduction in False Positives (from 18 to 13) is practically relevant because it reduces the manual verification burden of misclassifying real images as AI. Additionally, CatBoost reduced False Negatives from 11 to 10 (a 9.09% relative decrease), meaning it was slightly less prone to missing synthetic images on this particular split compared to XGBoost.

To assess whether this advantage generalizes beyond a single split, repeated 5-fold stratified cross-validation (5×10 repeats, 50 fold-level estimates per model) was performed on the training set. Under this within-domain evaluation, XGBoost achieved 74.03% ± 4.20% accuracy, 73.15% ± 4.17% precision, 76.13% ± 5.61% recall, and 74.53% ± 4.22% F1-score, while CatBoost achieved 74.30% ± 4.74% accuracy, 73.21% ± 4.77% precision, 77.00% ± 5.89% recall, and 74.95% ± 4.59% F1-score. CatBoost was numerically higher on every metric, by margins of 0.05 to 0.87 percentage points, but none of these differences were statistically significant (paired t-test: $p = 0.4934$ for accuracy, $p = 0.8934$ for precision, $p = 0.1033$ for recall, $p = 0.2775$ for F1-score; Wilcoxon signed-rank test: $p = 0.6878$, 0.9705 , 0.0915 , and 0.3545 , respectively). These statistical comparisons should be interpreted cautiously, as the 50 fold-level estimates are not strictly independent due to overlapping training sets across the repeated folds, which may lead to underestimated variance. This indicates that, within a single data source, XGBoost and CatBoost perform comparably, and the two algorithms cannot be reliably distinguished using this dataset size and fold configuration. The larger gap observed on the asymmetric source-shifted test split (Table 8) should therefore be interpreted as a preliminary, split-specific observation of cross-domain behavior rather than as evidence of a general, statistically confirmed superiority of CatBoost over XGBoost.

Table 9. Contingency table for McNemar's test comparing paired predictions

Model	CatBoost Correct	CatBoost Incorrect	Total
XGBoost Correct	118	3	121
XGBoost Incorrect	9	20	29
Total	127	23	150
McNemar's Test Result	Chi-square = 2.0833		df = 1
			p-value = 0.1489

Note. Since $p > 0.05$, the observed 4.00 percentage point accuracy difference is not statistically significant at the 5% significance level.

To further evaluate whether the observed 4.00 percentage point accuracy advantage of CatBoost on the asymmetric test split is statistically reliable, McNemar's test was applied to the paired predictions of both models on the 150 test images. Table 9 presents the contingency matrix of paired predictions along with the test results. The test yielded a Chi-square statistic of 2.0833 ($df = 1$) and a p-value of 0.1489. Since the p-value exceeds the conventional significance threshold of 0.05, the difference is not statistically significant on this split. This finding corroborates the repeated cross-validation results, further supporting the interpretation that CatBoost's numerical superiority on this particular data split is not robust and should be regarded as preliminary rather than conclusively established. Feature importance analysis was performed using the gain metric to measure each feature's contribution to improving model quality. Figure 5 and Figure 6 present the top ten features with the highest importance levels in the XGBoost and CatBoost models, respectively.

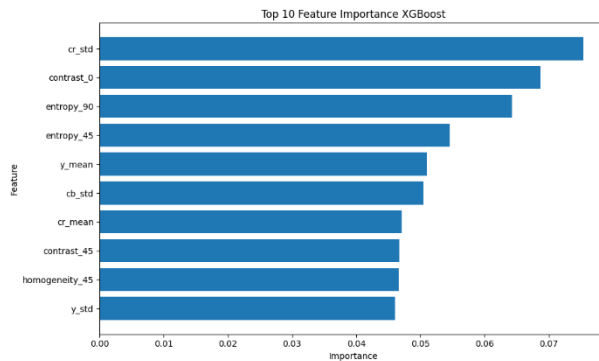


Figure 5. Diagram Feature Importance XGBoost

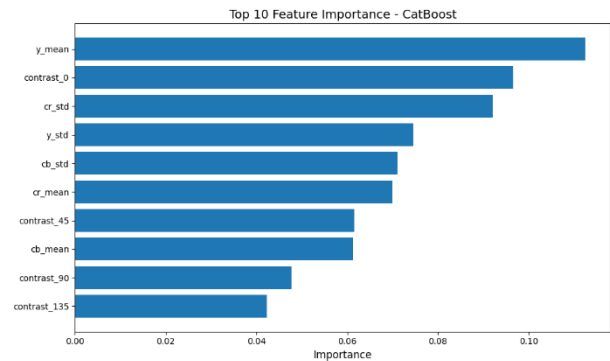


Figure 6. Diagram Feature Importance CatBoost

Feature importance analysis shows that `cr_std` and `contrast_0` were consistently highly ranked across both models. To assess the stability of these rankings, feature importance (gain) was extracted from each of the 50 cross-validation folds (5-fold $\times$ 10 repeats).

Table 10. Stability of feature importance rankings for `cr_std` and `contrast_0`

Feature	Model	Mean Rank	Std. Deviation of Rank	Rank Range (Min – Max)
cr_std	XGBoost	1.42	0.67	1 – 4
	CatBoost	2.38	0.98	1 – 5
contrast_0	XGBoost	1.78	0.58	1 – 3
	CatBoost	2.54	1.02	1 – 5

Note. Lower ranks indicate higher feature importance (Rank 1 = most important).

Table 10 further validates the stability of these rankings. In XGBoost, `cr_std` maintained a low mean rank (Mean Rank = 1.42, Std = 0.67, Range = 1-4), while in CatBoost its mean rank stayed close to the top (Mean Rank = 2.38, Std = 0.98, Range = 1-5). Similarly, `contrast_0` maintained mean ranks below 2 in XGBoost (1.78 ± 0.58 , Range = 1-3) and below 3 in CatBoost (2.54 ± 1.02 , Range = 1-5). These findings confirm that both features are not only important in the final models but also comparatively stable across resampled training data, even though their exact rank fluctuated somewhat across folds.

The main difference lies in `y_mean`, which became the most important feature in CatBoost, indicating its sensitivity to luminance. Notably, this shift in feature importance corresponds to CatBoost's higher recall (86.67% vs. 85.33%) and substantially higher precision (83.33% vs. 78.05%) compared to XGBoost. Quantitatively, 6 out of the top 10 features come from GLCM (texture) and 4 from YCbCr (color). However, considering the initial feature pool 16 texture features and 6 color features, this raw count reveals a more nuanced insight: color features actually appear at a proportionally higher rate (4 out of 6, or 66.7%) compared to texture features (6 out of 16, or 37.5%). This suggests that, feature-for-feature, YCbCr-derived color characteristics are at least as discriminative as GLCM texture characteristics. Consequently, the absolute numerical dominance of texture features in the top ten should not be interpreted as evidence that texture is the most dominant differentiator; rather,

both feature groups are highly complementary, with color features demonstrating remarkable per-feature efficiency. This finding suggests, as a hypothesis for future work rather than a demonstrated result, that a lightweight detection system restricted to the six YCbCr features (y_mean , y_std , cb_mean , cb_std , cr_mean , cr_std) alongside key GLCM indicators ($contrast_0$, $energy_0$, $entropy_0$) might retain much of the current accuracy while reducing computational overhead; this study did not perform the reduced-feature ablation experiment needed to confirm it.

CatBoost's advantage on the cross-domain test split may be related to two internal mechanisms, though the present design does not isolate them individually. First, the Ordered Boosting mechanism is theoretically effective in overcoming overfitting on medium-sized datasets. On this split, XGBoost showed a gap between training accuracy (99.83%) and test accuracy (80.67%) of 19.16 percentage points. CatBoost showed a substantially smaller gap of 14.66 percentage points (training accuracy 99.33% $\rightarrow$ test accuracy 84.67%), or 4.50 percentage points smaller than XGBoost. This suggests that CatBoost's Ordered Boosting mechanism may be more effective at mitigating overfitting on this limited dataset. Ordered Boosting processes samples sequentially in random order to calculate gradients, which is theoretically expected to produce less biased predictions on limited datasets, and is offered here as a plausible, though not directly tested, explanation for the smaller gap. On this same split, CatBoost's AI-class recall was 1.34 percentage points higher than XGBoost's (85.33% to 86.67%), indicating that CatBoost was slightly more effective at correctly identifying AI-generated face images on this asymmetric test split. This improvement in recall, combined with the larger improvement in precision, contributed to CatBoost's higher F1-score on this split. As reported above, however, the repeated cross-validation results did not find this difference to be statistically significant within a single data source, so it should not yet be generalized as a reliable property of CatBoost on this task.

Second, the Symmetric Tree architecture (Oblivious Trees) in CatBoost is naturally more resistant to multicollinearity. The GLCM texture features extracted from four orientation angles have high spatial correlation, for example between $contrast_0$ and $contrast_45$. Symmetric trees force each branch at the same depth level to use the same feature, preventing the model from relying too heavily on correlated features and potentially reducing overfitting risk. This mechanism is offered as a plausible explanation, consistent with a 27.78% decrease in False Positives (from 18 to 13 cases) on the cross-domain test split, meaning CatBoost was more reliable in avoiding the misclassification of real images as synthetic on this particular split; however, the present experiment compares complete algorithms rather than isolating individual architectural components, so this explanation should be read as interpretive rather than causally established. This caution is reinforced by the repeated cross-validation results reported above, which found no statistically significant difference between the two models within a single data source. Thus, although CatBoost requires slightly longer training time than XGBoost and showed a numerical advantage on the cross-domain split, the present evidence supports treating CatBoost as a promising rather than a conclusively superior choice for synthetic face image detection based on manual feature extraction on limited datasets (< 1000 samples), pending confirmation through repeated cross-domain evaluation.

It is important to contextualize these performance differences within the asymmetric dataset design. Since the real training images were drawn from public sources and the real test images from direct camera captures, the reported accuracy, precision, and recall metrics reflect the models' ability to generalize across different image acquisition domains. This design choice makes the observed performance gap between XGBoost and CatBoost particularly meaningful, as it suggests that CatBoost's Ordered Boosting mechanism offers better resistance to domain-specific variations in image compression and sensor noise.

In terms of computational cost, the training time for CatBoost on the 600-image training set was measured at 0.9345 seconds, compared to 0.4309 seconds for XGBoost (excluding feature extraction time), using an MSI Modern 14 B5M laptop equipped with an AMD Ryzen 5 5500U processor and 8GB RAM. While CatBoost requires a longer training time (+0.5036 seconds) compared to XGBoost, this increase is practically negligible (approximately 0.5 seconds) given the notable improvements it offers in accuracy (4.00 percentage points) and precision (5.28 percentage points) on this

asymmetric test split, alongside its theoretical interpretability advantages for synthetic face detection on medium-sized datasets.

To characterize the complete detection pipeline rather than training alone, model size, memory usage, feature extraction time, and inference latency were also measured, as summarized in Table 11. Feature extraction (GLCM and YCbCr computation) took 56.26 ± 4.54 ms per image and is identical for both models since it is performed once upstream of classification. At inference, XGBoost required 0.848 ms per image with a model size of 0.43 MB and 64.63 MB of memory usage, whereas CatBoost required 1.715 ms per image but with a substantially smaller model size of 0.19 MB and lower memory usage of 8.18 MB. Combining feature extraction with inference, the total processing time per image was 57.11 ms for XGBoost and 57.98 ms for CatBoost, a difference of only 0.87 ms (roughly 1.5%) that is negligible relative to the shared 56.26 ms feature extraction cost. This indicates that, for this pipeline, feature extraction rather than classification dominates total latency, and both models are well within the range required for practical, near real-time deployment on modest hardware. CatBoost's smaller model size and lower memory footprint may be advantageous in memory-constrained deployment settings, despite its higher per-image inference latency and longer training time relative to XGBoost. These efficiency gains should be read as descriptive of measured runtime cost rather than as evidence bearing on classification accuracy, which is addressed separately above through the non-significant within-domain cross-validation results.

Table 11. Full-Pipeline Computational Efficiency Metrics

Metric	XGBoost	CatBoost
Training time (s)	0.4309	0.9345
Model size (MB)	0.43	0.19
Memory usage (MB)	64.63	8.18
Feature extraction (ms/image)	56.26 ± 4.54	56.26 ± 4.54
Inference latency (ms/image)	0.848	1.715
Total per image (ms)	57.11	57.98

The findings of this study have several important implications, both theoretically and practically. Theoretically, this study enriches the literature on synthetic image detection by showing that a feature engineering approach based on texture and color feature fusion, combined with gradient boosting algorithms, can achieve competitive performance without requiring substantial computational resources. This opens opportunities for developing more efficient detection systems accessible to researchers with computational limitations. Practically, this study suggests CatBoost as a promising candidate, alongside XGBoost, for synthetic image detection systems that are accurate, efficient, and easy to interpret compared to deep learning approaches, especially for scenarios with limited datasets and the need for decision transparency; because within-domain cross-validation did not find a statistically significant difference between the two algorithms, this recommendation should be treated as preliminary until confirmed by repeated cross-domain evaluation. The identification of 5 core indicators (cr_std, contrast_0, y_mean, y_std, cb_std) as the most discriminative features also points to a potential direction for developing lighter detection systems, although the feasibility of restricting feature extraction to these indicators without compromising accuracy has not yet been directly tested.

This study provides three main contributions. First, a comparative study quantitatively comparing XGBoost and CatBoost in the context of Generative AI-generated face image detection based on GLCM and YCbCr feature fusion. Second, the use of a synthetic image dataset generated using the latest generative AI platforms (GPT Image 2, Nano Banana, Leonardo AI, Canva, and Dreamina AI), making the research results relevant to the characteristics of the most current generative artifacts. Third, the quantitative identification of the most discriminative features across both models, namely cr_std and contrast_0 as the most stable indicators, which offer better interpretability compared to deep learning approaches.

Although this study observed a numerical advantage for CatBoost on the asymmetric test split, this finding was not confirmed by the repeated cross-validation results, which showed no statistically significant difference between the two algorithms. Several limitations need to be considered. First,

the dataset size (750 images) is considered moderate, and the coverage of AI generators tested, while diverse, does not represent the entire spectrum of rapidly developing generative models, such as StyleGAN or Diffusion Models, thus the generalization of these findings still requires further validation on larger and more varied datasets. Second, the process of resizing all images to 224×224 pixels, although necessary for standardization, has the potential to remove fine micro-scale artifacts that may be important discriminative markers in high-resolution synthetic images. Third, this study has not conducted a direct comparison with the latest deep learning architectures, such as EfficientNet or Vision Transformer, on the same dataset. Fourth, while Grid Search was applied for hyperparameter tuning, the predefined search grids may not have covered the global optimum; more extensive searches or alternative optimization techniques such as Bayesian optimization could potentially yield further marginal gains. Additionally, the Grid Search was performed on the same 600-image training set that was later used for repeated cross-validation evaluation, meaning the cross-validation results may retain some model-selection optimism and should be interpreted with this limitation in mind. Fifth, and most importantly, the observed superiority of CatBoost was demonstrated on a single cross-domain train-test split; when the comparison was repeated using 5-fold stratified cross-validation with 10 repeats on the training data, the differences between XGBoost and CatBoost across all metrics were not statistically significant (paired t-test and Wilcoxon signed-rank test, $p > 0.09$). The single-split result should therefore be regarded as an indication of cross-domain behavior on this particular dataset rather than as statistically confirmed evidence of CatBoost's general superiority. These limitations do not invalidate the exploratory findings of this study but indicate that the performance comparison should be treated as preliminary, and they open clear opportunities for improvement and further development in future research.

Based on the identified limitations, there are several suggestions for future research. First, it is recommended to apply cross-validation to test model stability and reduce the overfitting risk still present in XGBoost. Second, testing model generalization on other generative AI platforms that have not been tested, such as DALL-E, Stable Diffusion, Midjourney, or StyleGAN, is necessary to strengthen the robustness claims of the results. Third, exploration at original resolution or multi-scale approaches should be considered to capture fine artifacts that may be lost due to the resizing process. Fourth, future research is suggested to compare this feature engineering approach with the latest deep learning architectures on the same dataset to objectively measure the relative advantages of each approach. Fifth, future research could explore more extensive hyperparameter optimization strategies, such as Bayesian optimization or random search with wider search ranges, to potentially uncover better parameter combinations beyond the predefined Grid Search space used in this study. Additionally, nested cross-validation could be employed to obtain a less biased estimate of tuned-model performance.

CONCLUSION

This study successfully implemented texture (multi-orientation GLCM) and color (YCbCr) feature fusion for detecting synthetic face images generated by the latest generative AI, which aligns with the objectives formulated in the Introduction to analyze and compare the performance of XGBoost and CatBoost. One asymmetric source-shifted test split, CatBoost achieved higher performance than XGBoost, with an accuracy increase of 4.00 percentage points (from 80.67% to 84.67%), a precision improvement of 5.28 percentage points (from 78.05% to 83.33%), a recall improvement of 1.34 percentage points (from 85.33% to 86.67%), and an F1-Score improvement of 3.44 percentage points (from 81.53% to 84.97%). This was accompanied by a reduction in False Negatives from 11 to 10 (a 9.09% relative decrease) and a reduction in False Positives from 18 to 13 (a 27.78% relative decrease). However, when this comparison was subjected to repeated 5-fold stratified cross-validation (10 repeats, 50 fold-level estimates) on the training data, CatBoost was only marginally higher than XGBoost on every metric (0.05 to 0.87 percentage points), and none of these differences reached statistical significance (paired t-test and Wilcoxon signed-rank test, $p > 0.09$ for all metrics). This indicates that, within a single data source, the two algorithms perform comparably, and the initial hypothesis that CatBoost would outperform XGBoost is only partially and preliminarily supported by the present evidence, pending confirmation through repeated cross-domain testing.

This conclusion is further supported by the McNemar test on the single test split ($p = 0.1489$), which confirmed that the 4.00 percentage point accuracy difference is not statistically significant.

Feature importance analysis shows cross-model consistency on core features (cr_std, contrast_0, y_mean, y_std, cb_std), indicating that red/chrominance color variation and horizontal texture complexity are among the most stable and discriminative indicators for distinguishing real and synthetic images across both algorithms, independent of which model performs better overall. Given the non-significant cross-validation results, this study suggests CatBoost as a promising rather than a definitively superior candidate for synthetic image detection systems that are accurate, efficient, and easy to interpret compared to deep learning approaches, particularly for scenarios with limited datasets and the need for decision transparency. The prospects for developing the results of this study include repeated cross-domain evaluation (rather than a single train-test split) to test whether the observed cross-domain advantage of CatBoost is reproducible, testing generalization on other generative AI platforms such as DALL-E, Stable Diffusion, Midjourney, and StyleGAN, as well as exploring multi-scale approaches to capture fine artifacts that may be lost due to the resizing process. Future research is also suggested to compare this feature engineering approach with the latest deep learning architectures on the same dataset and to expand the current grid search space or apply Bayesian optimization to explore a wider range of hyperparameter combinations, potentially improving model performance further.

ACKNOWLEDGMENT

The authors would like to express their gratitude to all parties who assisted in the data collection process, particularly the subjects who volunteered to serve as models for the direct image capture, as well as colleagues in the Department of Informatics who provided valuable input throughout the research process.

AUTHOR CONTRIBUTION STATEMENT

IF, DH, WF: Conceptualization, Methodology, Investigation, Data Curation, Writing, Original Draft, Visualization, Formal Analysis. DH, WF: Resources, Supervision, Project Administration. All authors reviewed and approved the final manuscript.

AI DISCLOSURE STATEMENT

The authors used Grammarly during the preparation of this work for language polishing and grammar checking. After using the tool/service, the authors thoroughly reviewed and edited the content as needed and take full responsibility for the content of the publication.

REFERENCES

- [1] E. J. Miller, B. A. Steward, Z. Witkower, C. A. M. Sutherland, E. G. Krumhuber, and A. Dawel, "AI Hyperrealism : Why AI Faces Are Perceived as More Real Than Human Ones," *Psychol. Sci.*, vol. 34, no. 12, pp. 1390–1403, 2023, doi: 10.1177/09567976231207095.
- [2] A. Hardiyanto, "Classification of Synthetic Images Generated by Diffusion Models Using Gray Level Co-Occurrence Matrix (GlcM) and Convolutional Neural Network (CNN)," *JIKO (Jurnal Inform. dan Komputer)*, vol. 9, no. 3, p. 572, 2025, doi: 10.26798/jiko.v9i3.2033.
- [3] H. Prayoga and H. Tuasikal, "Penyebaran Konten Deepfake Sebagai Tindak Pidana: Analisis Kritis Terhadap Penegakan Hukum Dan Perlindungan Publik Di Indonesia [Deepfake Content Distribution as a Criminal Act: Critical Analysis of Law Enforcement and Public Protection in Indonesia]," *Abdurrauf Law Sharia*, vol. 2, no. 2, pp. 22–38, 2024, doi: 10.70742/arlash.v2i1.194.
- [4] D. Velásquez-salamanca, M. Á. Martín-pascual, and C. Andreu-sánchez, "Interpretation of AI-Generated vs . Human-Made Images," *J. Imaging*, vol. 11, pp. 1–16, 2025, doi: 10.3390/jimaging11070227.
- [5] V. Osińska, W. Kortas, A. Szalach, and M. Welter, "AI Images vs . Real Photographs :

- Investigating Visual Recognition and Perception," *J. Eye Mov. Res.*, vol. 18, p. 61, 2025, doi: 10.3390/jemr18060061.
- [6] J. Mu, M. Adrezo, and A. N. Haikal, "Identifikasi Wajah Asli dan Buatan Deepfake Menggunakan Metode Convolutional Neural Network [Identification of Real and Deepfake Faces Using Convolutional Neural Network Method]," *TEKNIKA*, vol. 13, no. 1, pp. 45–50, 2024, doi: 10.34148/teknika.v13i1.705.
- [7] M. S. A. Aria, C. Slamet, and M. D. Firdaus, "Klasifikasi Fake dan Real Menggunakan Vision Transformer dan EfficientNet-B0 pada Gambar Asli dan Generatif AI [Fake and Real Classification Using Vision Transformer and EfficientNet-B0 on Real and Generative AI Images]," *SMATIKA J.*, vol. 15, no. 1, pp. 179–192, 2025, doi: 10.32664/smatika.v15i01.1531.
- [8] A. Gulati, A. Felahatpisheh, and C. E. Valderrama, "Feature engineering through two-level genetic algorithm," *Mach. Learn. with Appl.*, vol. 21, no. July, p. 100696, 2025, doi: 10.1016/j.mlwa.2025.100696.
- [9] T. Qamar and N. Z. Bawany, "Understanding the black-box : towards interpretable and reliable deep learning models," *PeerJ Comput. Sci.*, vol. 9, p. e1629, 2023, doi: 10.7717/peerj-cs.1629.
- [10] J. Huang and T. Handhayani, "Klasifikasi Telapak Tangan Menggunakan Deep Learning Dan Machine Learning [Palm Classification Using Deep Learning and Machine Learning]," *J. Mnemon.*, vol. 8, no. 2, pp. 293–300, 2025, doi: 10.36040/mnemonic.v8i2.15716.
- [11] R. F. Naibaho and I. P. Sari, "Implementasi Metode Gray Level Co-Occurrence Matrix Menganalisis Tekstur Kulit Wajah [Implementation of Gray Level Co-Occurrence Matrix Method for Analyzing Facial Skin Texture]," *sudo J. Tek. Inform.*, vol. 3, no. 4, pp. 172–182, 2025, doi: 10.56211/sudo.v3i4.668.
- [12] C. I. Purba, A. Alrizal, and Y. Fendriani, "Klasifikasi Kanker Kulit dari Citra Dermoskopi Menggunakan Fitur Gray Level Co-occurrence Matrix (GLCM) dengan Algoritma Machine Learning [Skin Cancer Classification from Dermoscopy Images Using Gray Level Co-occurrence Matrix (GLCM) Features with Machine]," *JFTJ. Fis. dan Ter.*, vol. 12, pp. 30–44, 2025, doi: 10.24252/jft.v12i1.56651.
- [13] A. Prasetio and Y. Utami, "Analisis Warna Kulit Menggunakan Citra Digital Berbasis Ruang Warna YCbCr [Skin Color Analysis Using Digital Images Based on YCbCr Color Space]," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 8, no. 3, pp. 1728–1732, 2025, doi: 10.32672/jnkti.v8i3.9228.
- [14] S. Alharthi and A. Gutub, "Adjusting image stego practicality via YCbCr color space formation," *J. Eng. Res.*, vol. 14, no. 1, pp. 756–764, 2026, doi: 10.1016/j.jer.2025.07.008.
- [15] K. Letou and S. Thiagarajan, "Analysis of Enhanced Adaptive Gradient Boosting Regression in Business Intelligence," *Int. J. Adv. SIGNAL IMAGE Sci.*, vol. 12, no. 2, pp. 1075–1088, 2026, doi: 10.29284/1f9fcb03.
- [16] T. Chen and C. Guestrin, "XGBoost : A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794. doi: 10.1145/2939672.2939785.
- [17] F. I. Adani and H. Amalia, "Penerapan Metode Algoritma XGBoost untuk Prediksi Risiko Penyakit Jantung [Application of the XGBoost Algorithm Method for Heart Disease Risk Prediction]," *J. Insa. J. Inf. Syst. Manag. Innov.*, vol. 5, no. 2, pp. 117–125, 2025, doi: 10.31294/j-insan.v5i2.10000.
- [18] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost : unbiased boosting with categorical features," in *Advances in Neural Information Processing Systems (NeurIPS 2018)*, 2018, pp. 6638–6648.
- [19] G. E. Saputra, M. Hanindia, P. Swari, and A. L. Nurlaili, "Implementasi Algoritma XGBoost, CatBoost, dan LGBM untuk Klasifikasi Pencemaran Udara [Implementation of XGBoost,

- CatBoost, and LGBM Algorithms for Air Pollution Classification],” *JIIIP - J. Ilm. Ilmu Pendidik.*, vol. 8, no. 12, pp. 14135–14139, 2025, doi: 10.54371/jiip.v8i12.10102.
- [20] I. Setiawan, R. Dahlan, A. I. Basuki, H. Susanto, and D. Rosiyadi, “Trade-off between Image Quality and Computational Complexity : Image Resizing Perspective,” *J. Tek. Elektro*, vol. 14, no. 1, pp. 24–28, 2022, doi: 10.15294/jte.v14i1.37629.
- [21] A. Setiawan and A. A. Budiman, “Optimization of Gray Level Co-occurrence Matrix (GLCM) Texture Feature Parameters in Determining Rice Seed Quality,” *Emit. Int. J. Eng. Technol.*, vol. 13, no. 1, pp. 110–123, 2025, doi: 10.24003/emitter.v13i1.928.
- [22] S. K. Abdulateef and A. N. Hasoon, “Comparison of the Components of Different Colour Spaces to Enhanced Image Representation,” *J. Image Process. Intell. Remote Sens.*, vol. 03, no. 01, pp. 11–17, 2023, doi: 10.55529/jipirs.31.11.17.
- [23] R. R. Laska and A. M. Yolanda, “A Comparative Study of Z-Score and Min-Max Normalization for Rainfall Classification in Pekanbaru,” *J. Data Sci.*, vol. 2024, no. 1, pp. 1–8, 2024, doi: 10.61453/jods.v2024no04.
- [24] K. M. Sujon, R. Hassan, K. Choi, and M. A. Samad, “Accuracy, precision, recall, f1-score, or MCC? empirical evidence from advanced statistics, ML, and XAI for evaluating business predictive models,” *J. Big Data*, vol. 12, no. 1, p. 268, 2025, doi: 10.1186/s40537-025-01313-4.
- [25] A. A. Huang and S. Y. Huang, “Use of machine learning to identify risk factors for insomnia,” *PLoS One*, vol. 18, no. 4, pp. 1–16, 2023, doi: 10.1371/journal.pone.0282622.