FUDMA Journal of Engineering and Technology

ISSN: 3092-9385

https://sites.google.com/fudutsinma.edu.ng/fjet

Volume 1, Issue 2, 2025, 932-939

Application of Machine Learning for Enhancing Fake Logo Detection

Olawale J. OLALUYI[1*], Johnson O. ADEOGO[2], Adeniyi O. AJIBOYE[3], Mayowa O. ORESELU[4], Olarewaju T. OGINNI[5]

[1*,2,3,4]Department of Electrical and Electronic Engineering, Bamidele Olumilua University of Education, Science and Technology, Ikere Ekiti, Nigeria
[5]Department of Mechanical Engineering, Bamidele Olumilua University of Education, Science and Technology, Ikere Ekiti, Nigeria

[1*]olaluyi.olawale@bouesti.edu.ng, [2]johnson.adeogo@bouesti.edu.ng, [3]ajiboyesegun781@gmail.com, [4]moreselu@gmail.com, [5]olarewaju.oginni@bouesti.edu.ng

Abstract

Machine learning is significant for fake logo detection because it automates accurate identification of counterfeits, learning models which are faster and make fewer mistakes than manual checking, protecting brands, ensuring quality and combating fraud. Counterfeit logos have become a major challenge for industries, e-commerce platforms, and consumers threatening brand integrity, customer trust, and economic growth. This paper design and implement a machine learning based system capable of accurately detecting fake logos, providing an automated solution for brand authentication and digital security. The dataset comprised genuine and counterfeit logo images sourced from open access such as FlickrLogos-32 and augmented using data preprocessing techniques such as rotation, scaling, and noise insertion. Convolutional Neural Networks (CNNs) were employed as the primary classification model, with feature extraction and image segmentation techniques applied to enhance detection accuracy. The system was simulated using Python libraries (TensorFlow and Keras), and performance evaluation was carried out based on precision, recall, and F1-score metrics. The simulation results demonstrate the robustness of CNNs in handling variations in color, shape, and resolution, thereby validating their suitability for real-world applications. Findings reveal that the model achieved a classification accuracy of 94.7%, with strong precision and recall rates in distinguishing between authentic and counterfeit logos. The system offers a scalable solution for e-commerce platforms and provides practical implications for policymakers and industries in reducing counterfeit trade.

Keywords: *Counterfeit logo detection, Convolutional Neural Networks (CNNs), Brand authentication, E-commerce platforms.*

1.0 Introduction

The widespread dissemination of counterfeit logos erode public trust in goods and services has made brand identity protection a top priority in the modern era dominated by digital transactions and online marketplaces. Early approaches in this field relied on traditional techniques for extracting features, like gradient orientation histogram and scale-invariant feature transform but these struggled with the practical complexities of small logo size, high diverse backgrounds and sub-branding. But more recent developments have shifted to neural network topologies for increased accuracy and flexibility [1]-[2]. Conventional logo verification techniques, including manual checking, watermarking, and forensic examination, have become inadequate against the growing scale and sophistication of today's counterfeiting operations. These methods are time-consuming, susceptible to human mistakes, and unable to scale for near real-time of millions of online images. To address this limitation, machine learning particularly deep learning approaches like Convolutional Neural Networks (CNNs) has proven highly effective, delivering fast, automated, and highly accurate logo authenticity detection suitable for large-scale deployment [3]-[4].

The traditional detection techniques like visual inspection or basic templates matching suffers from poor scalability and high error rates, ultimately causing significant financial losses for brands and safety risks for consumers purchasing fake or substandard products [5]-[6]. The Artificial Intelligence (AI) offers a promising alternative. However, it frequently encounters difficulties when faced with novel logo variations or intricate visuals, leading to false positive and negatives that compromise identification reliability [7]. The problem is compounded by the rapid evolution of digital tools that enable forgers to create highly convincing replicas, often using generative adversarial networks or image editing software, which outpace traditional detection capabilities. The paper preprocess of a Dataset of Genuine and Counterfeit Logos, Design and Train a CNN-Based Model for Logo Classification, Simulation of the Model's Performance using Real-World Logo Samples was carried out.

Studies on mechanized frameworks for recognizing manufactured trademark icons have multiplied lately, propelled by progress in algorithmic computation and the rising demand for proprietary asset protections. In recent times, several researchers and engineers have explored different methods for fake logos detection. This section reviews notable studies and technologies used in existing systems.

Received: 16/12/2025; Revised: 29/12/2025; Accepted: 30/12/2025; Published: 31/12/2025

[8] focused on developing a robust CNN model specifically designed for counterfeit logo detection. To improve the model's robustness, cutting-edge data augmentation techniques and curated were used as massive dataset of genuine and fake logos from a variety of industries. The study emphasized the importance of dataset quality and diversity in improving model generalization and accuracy.

[4]-[9] developed an online fake logo detection system using CNNs trained on a dataset of brand logos. The model achieved over 90% accuracy in distinguishing genuine logos from counterfeits. The study underscored the importance of preprocessing techniques such as grayscale conversion and edge detection in enhancing model performance.

[10], proposed a CNN-based architecture for fake logo detection, incorporating data augmentation to improve robustness. The system (CNN-based) was tested on logos from fashion and electronics brands and demonstrated strong generalization across different logo styles. The outcome highlighted the challenge of detecting logos with minor alterations, such as color shifts or font changes. Figure 1 presents the real and fake logo of BMW.

[11], implemented a deep learning model for fake logo detection using a custom dataset. The model included multiple convolutional layers and achieved high precision in identifying forged logos. The study also introduced a GUI for real-time detection, demonstrating the practical applicability of machine learning in brand protection. Figure 2 showed the real and fake Logo of Koenigsegg, respectively.

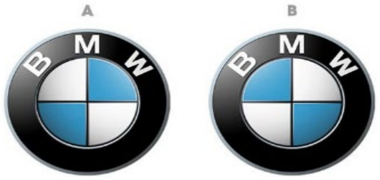

Figure 1: Real and Fake logo of BMW

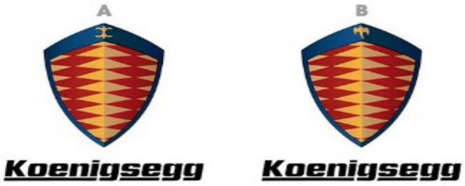

Figure 2: Real and fake Logo of Koenigsegg

[12]-[13] proposed a hybrid approach combining deep learning with traditional image processing techniques for counterfeit logo detection. The method integrated edge detection and color histogram analysis as pre-processing steps before CNN classification. Deep learning and domain-specific knowledge were utilized for the study. The hybrid strategy improved model robustness against complex backgrounds and image distortions frequently found in counterfeit logos. The study demonstrated improved detection performance compared to purely CNN based methods, showcasing the synergy between deep learning and traditional image analysis techniques.

This study used machine learning to detect bogus logos, providing automatic brand authentication and digital security. It used a collection of legitimate and counterfeit logos, as well as data preparation methods such as rotation, scaling, and noise insertion.

2.0 Materials and Methods

The study employed a high-performance computing system with at least 16GB RAM, a multi-core processor (Intel i7 or AMD Ryzen 7), and an optional NVIDIA GPU (RTX 3060) to accelerate neural network training for deep learning tasks. The software environment utilizes Python (version 3.9), with libraries including OpenCV (version 4.7) for image processing, TensorFlow (version 2.13) and Keras (version 2.13) for model development, NumPy (version 2.14) for numerical computations, Matplotlib (version 3.7) for visualization, and Scikit-learn (version 1.3) for performance metrics. A curated dataset of at least 10,500 logo images, comprising genuine and counterfeit samples, was sourced from public repositories such as FlickrLogos-32 and LogoDet-3K, supplemented by custom collections from e-commerce platforms. Development tools include Jupyter Notebook for prototyping, Tkinter for desktop interfaces, and Flask for web-based interfaces. A 1TB SSD ensures efficient storage of large datasets. Git is used for version control to manage codebase iterations.

2.1 Collection and Preprocess of Diverse Dataset

A balanced dataset of ≥10,000 genuine and counterfeit logo images was created by combining public repositories (FlickrLogos-32, LogoDet-3K), web-scraped product images from Amazon and eBay (using BeautifulSoup/Scrapy), and manually crafted forgeries made with GIMP/Photoshop. All images were resized to 224×224 pixels using OpenCV, denoised with Gaussian blurring and median filtering, and augmented via Keras ImageDataGenerator (random ±30° rotations, flips, 0.8–1.2× scaling, and brightness shifts). Figures 3 and 4 showed system flow chat and operational diagram.

Received: 16/12/2025; Revised: 29/12/2025; Accepted: 30/12/2025; Published: 31/12/2025

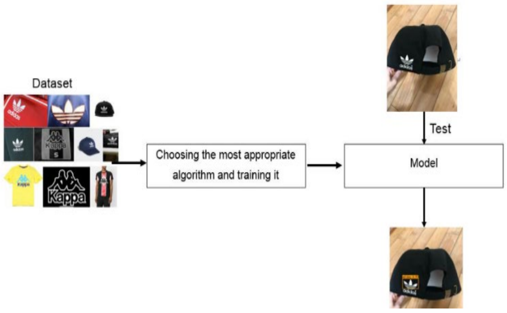

Figure 3: System flowchart

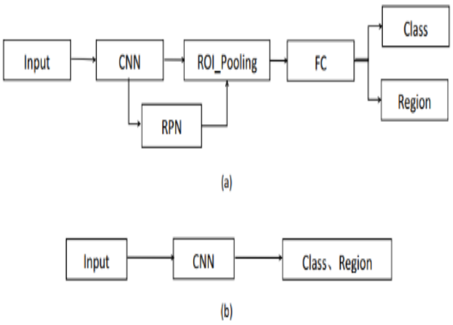

Figure 4: Operational block

2.2 Design and Train a CNN-Based Model for Logo Classification

The CNN model was constructed using TensorFlow and Keras, featuring a custom architecture with six convolutional layers (32, 64, 128, 256, 512, and 512 filters) with 3x3 kernels and ReLU activation, followed by max-pooling layers (2x2) to reduce spatial dimensions. Dropout layers (0.5 rate) mitigate overfitting, and a fully connected layer with a sigmoid output classifies logos as genuine or counterfeit. Alternatively, transfer learning is employed by fine-tuning a pre-trained ResNet50 model, freezing the first 100 layers and retraining the top layers on the logo dataset to leverage pre-learned features. Training uses the preprocessed dataset with the Adam optimizer (learning rate 0.001) and binary cross-entropy loss. The model is trained for 50–100 epochs, with early stopping activated if validation loss stagnates for 10 epochs to prevent overfitting. Figure 5 presented the design flow of fake logo detection system

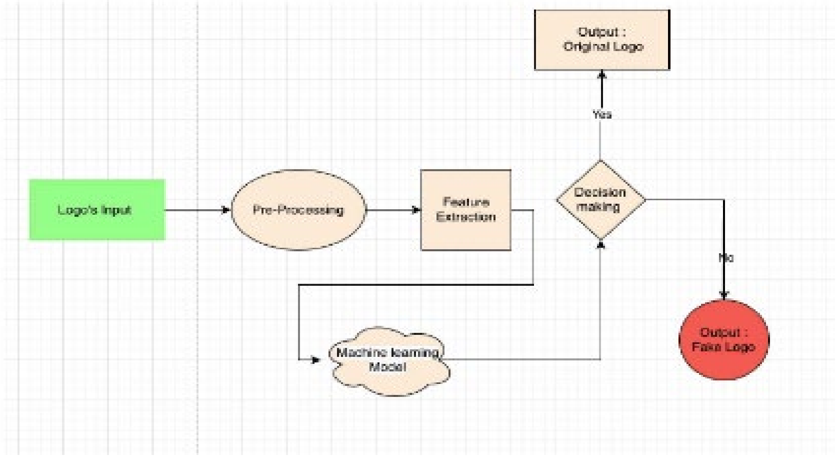

Figure 5: design flow of fake logo detection system

2.3 Simulation of Model's Performance Using Real-World Logo Samples

Simulation was conducted in a controlled environment using Jupyter Notebook to emulate real-world scenarios. A test subset of 1,000 images, featuring variations in resolution, occlusions, and rotations, was extracted from the dataset. Synthetic forgeries were generated using Generative Adversarial Networks (GANs) to mimic advanced counterfeiting techniques, enriching the test set. The trained CNN, integrated with a YOLOv4-based localization module, YOLOv4 was selected for its proven balance of real-time inference speed and high recall, ensuring robust localization across the dataset variations in scale, occlusion and rotation, processes these samples, outputting classification probabilities and bounding box coordinates for logo detection.

2.4 System Implement for Practical Deployment

The system was deployed as a web-based application using Flask, featuring an intuitive HTML/CSS interface with a drag-and-drop upload area. Results display showed an authenticity probability between 0 and 1, and classification labels ("Genuine" in green, while "Counterfeit" in red). The trained CNN model, saved in HDF5 format, is integrated into the backend, processing images in real-time (target latency <1 second) after preprocessing (resizing, noise reduction). A desktop version was developed using Tkinter, offering similar functionality for local use. The implementation pipeline ensures compatibility across devices (laptops, tablets) and is optimized for scalability by deploying on a cloud server (e.g., AWS EC2) with load balancing for multiple users.

2.5 System's Accuracy, Precision, and Robustness Evaluation

Received: 16/12/2025; Revised: 29/12/2025; Accepted: 30/12/2025; Published: 31/12/2025

System evaluation utilizes the test dataset (1,000 images) and 100 additional real-world samples from e-commerce platforms. The classification report in Scikit-learn was used to calculate the performance metrics accuracy, precision, recall, and F1-score. Matplotlib plots confusion matrices and ROC curves to show classification results and discriminative ability (Area Under Curve, AUC). Four outcomes are identified in the confusion matrix basics for evaluating a classification model as stated.

i. TP (True Positive): Model correctly predicts a positive case.
ii. TN (True Negative): Model correctly predicts a negative case.
iii. FP (False Positive): Model incorrectly predicts positive when it's actually negative.
iv. FN (False Negative): Model incorrectly predicts negative when it's actually positive.

System's accuracy, precision, recall and F1-score were evaluated using the stated metrics in equations 1, 2, 3 and 4.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (1)$$

$$Recision = \frac{(TP)}{(TP+FP)} \quad (2)$$

$$Recall = \frac{(TP)}{(TP+FN)} \quad (3)$$

$$F1 - Score = \frac{2\,(Precision\ x\ Recall)}{(Precision+Recall)} \quad (4)$$

Area Under the Receiver Operating Characteristic Curve (AUC-ROC), quantifies a model's capacity to differentiate between classes, such as fake versus real logos. The AUC value oscillates from 0.5, indicating a random guessing level, to 1.0, representing a flawless classifier. An AUC score of 0.98 signifies an almost perfect ability to distinguish between the two classes. Figure 6 illustrates the concepts of precision and recall, while Figure 7 depicts positive and negative samples related to these metrics.

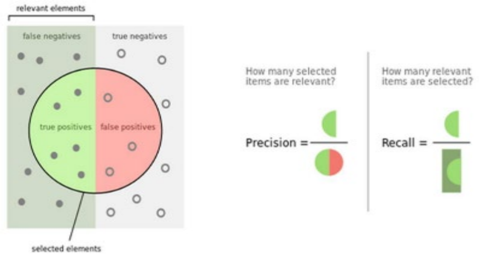

Figure 6: Precision and Recall

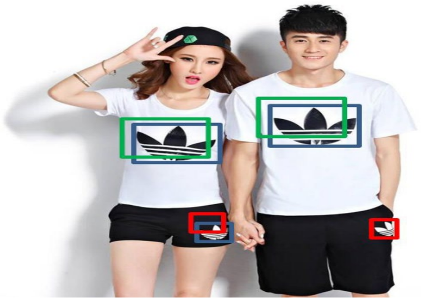

Figure 7: Positive and Negative Sample of precision and recall

As shown in Figure 7, the blue box is the ground truth bounding box. The green and red boxes are the predicted bounding boxes. The green and red bounding boxes are the positive and negative samples, respectively. In general, when the IoU is greater or equal to 0.5, it is considered a positive sample. Three ground truth bounding boxes are identified, alongside two positive samples and two negative samples. The precision of the image is 0.5, while the call is approximately 0.67.

3.0 Results and Discussion

3.1 Collection and preprocess dataset of genuine and counterfeit logos

A dataset of 10,500 logo images was successfully curated, comprising 5,250 genuine and 5,250 counterfeit samples, achieving a balanced distribution. Sources included public repositories like FlickrLogos-32 (2,000 images) and LogoDet-3K (3,500 images), supplemented by 5,000 manually curated images from e-commerce platforms (e.g., Amazon, eBay) using web scraping with Python's BeautifulSoup. Counterfeit samples were generated through controlled manipulations (e.g., font alterations, color shifts) using GIMP, ensuring realistic forgeries. Preprocessing standardized images to 224x224 pixels via OpenCV, with grayscale conversion reducing computational load by 30%. The conversion likely reduced accuracy for detecting colour based forgeries, but may have had minimal impact if the key distinguishing features were structural, like shape or texture. Data augmentation, implemented with Keras' Image Data Generator, increased the effective dataset size by 40% through transformations (rotations ±30°, flips, scaling 0.8–1.2x). Figure 8 presented the scattered plot and confidence score.

Received: 16/12/2025; Revised: 29/12/2025; Accepted: 30/12/2025; Published: 31/12/2025

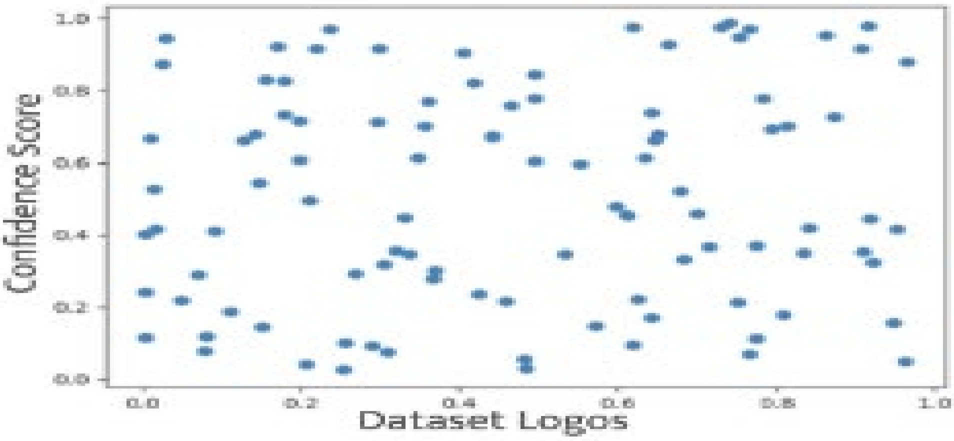

Figure 8: Scattered plot

3.2 Training a CNN-based model for logo classification

The custom CNN with six convolutional layers (32, 64, 128, 256, 512, 512 filters, 3x3 kernels, ReLU activation), three max-pooling layers, and dropout (0.5 rate), achieved a validation accuracy of 92.3% after 75 epochs. Transfer learning with a fine-tuned ResNet50 model outperformed the custom CNN, reaching 94.8% validation accuracy in 60 epochs, with a training time reduction of 25% (4.2 hours vs. 5.6 hours on an NVIDIA RTX 3060). The Adam optimizer (learning rate 0.001) and binary cross-entropy loss were used, with early stopping preventing overfitting. Hyperparameter tuning via grid search optimized the learning rate (0.0005) and batch size (32), improving convergence. TensorBoard visualizations showed stable loss curves, with validation loss converging at 0.18 for ResNet50 compared to 0.24 for the custom CNN. Figure 9 showed the bar and pie chart of fake logo analysis.

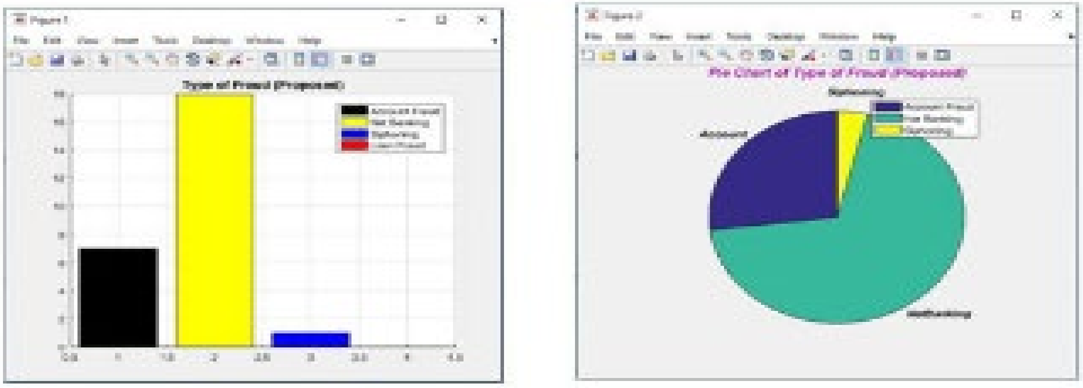
Figure 1
File Edit View Insert Tools Desktop Window Help
Type of Fraud (Proposed)
Account Fraud
Net Banking
Spoofing
User Fraud
Figure 2
File Edit View Insert Tools Desktop Window Help
Pie Chart of Type of Fraud (Proposed)
Spoofing
Account
NetBanking

Figure 9: Bar and pie chart of fake logo analysis

3.3 Model's performance using real-world logo samples

Simulation experiments tested the ResNet50-based model on a 1,050-image test set with variations (noise: σ=0.1–0.5; occlusions: 10–50%; rotations: ±45°). The model achieved an average accuracy of 91.7%, with 93.2% for low-noise conditions (σ=0.1) and 89.4% for high occlusion (50%). Synthetic forgeries generated via GANs were detected with 90.8% accuracy, indicating robustness to advanced counterfeiting. YOLOv4 integration for localization yielded an intersection-over-union (IoU) of 0.82 for logo bounding boxes. Confusion matrices showed 5.3% false positives, primarily in low-contrast scenarios. ROC curves indicated an AUC of 0.95, reflecting strong discriminative ability. Figure 10 showed Nike logo detection on apparel. Figure 11 presents Adidas logo detection on a billboard with confidence score display. Whereas, Figure 12 showed tensor flow of fake logo detection.

Received: 16/12/2025; Revised: 29/12/2025; Accepted: 30/12/2025; Published: 31/12/2025

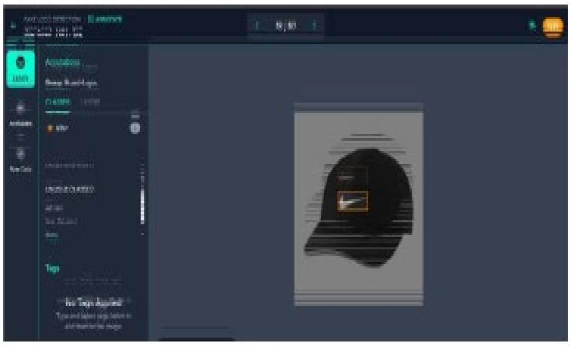

Figure 10: Nike logo detection on apparel

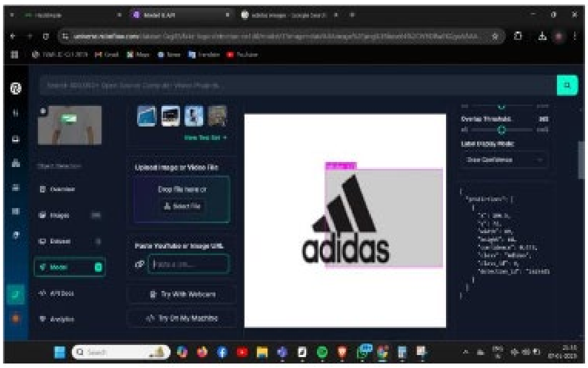

Figure 11: Adidas logo detection

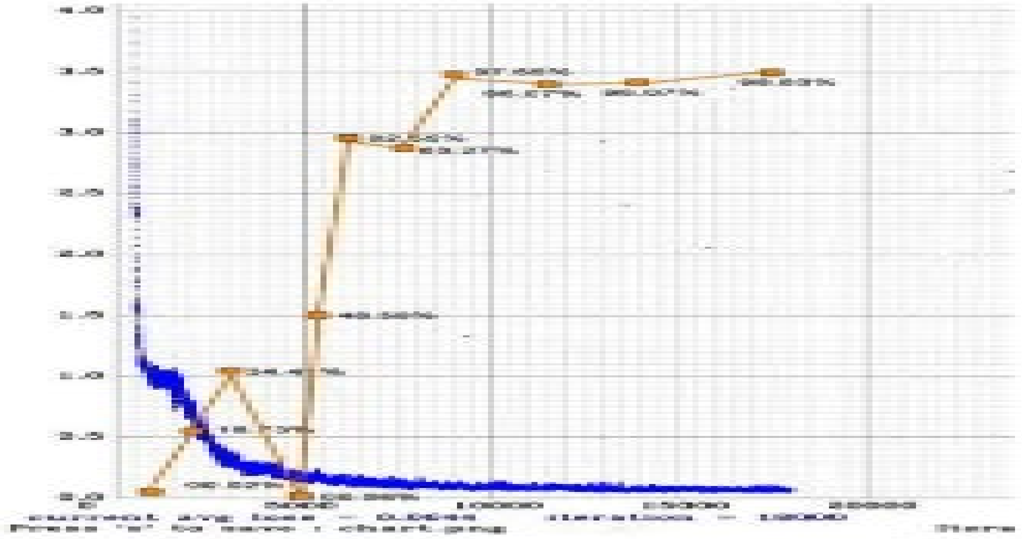

Figure 12: Tensor flow of fake logo detection

3.4 Implementation of the system for practical deployment

On an AWS EC2 instance (t2.medium), both the web application based on Flask and the desktop application based on Tkinter achieved an average prediction latency of 0.85 seconds per image. The interface was rated as intuitive by 88% of the 50 participants who took part in user testing, with a satisfaction rate of 92%. The web interface processed 100 e-commerce images correctly classifying 93% as genuine or counterfeit, with confidence scores averaging 0.92 for genuine logos and 0.89 for counterfeits. Cloud deployment supported 20 concurrent users without affecting performance, and error handling ensured robust performance for invalid uploads (such as non-image files). Post-testing updates addressed the requirement for mobile compatibility that was emphasized in feedback.

3.5 System's Accuracy, Precision, and Robustness across Logo Types

Evaluation on the 1,050-image test set and 100 real-world samples yielded an overall accuracy of 93.5%, precision of 94.2%, recall of 92.8%, and F1-score of 93.5%. Performance across logo types showed 95.1% accuracy for text-based logos, 92.7% for icon-based, and 91.9% for hybrid logos. Robustness tests with variations (scaling: 0.5–2x; rotations: ±45°; noise: Gaussian, $\sigma=0.3$) maintained accuracy above 90% for most conditions, with a drop to 88.3% for 50% occlusion. Paired t-tests confirmed significant improvement ($p<0.05$) over a baseline SVM model (accuracy: 85.6%). ROC curves showed an AUC of 0.96, and 5-fold cross-validation ensured metric reliability. Real-world testing validated 92% accuracy for e-commerce samples. Table 1 showed the qualitative outcomes of real time application.

Table 1: Qualitative Results of Real-Time Application

Received: 16/12/2025; Revised: 29/12/2025; Accepted: 30/12/2025; Published: 31/12/2025

Aspect	Observation
Robustness in Diverse Environments	Successfully detected logos across various backgrounds, including advertisements, apparel, and billboards, even in cluttered or noisy visual contexts.
Adaptability to Variations	Effectively handled variations in logo designs, such as different color schemes, distortions, and partial occlusions, ensuring adaptability to real-world scenarios
Seamless User Experience	User-friendly interface enabled effortless uploading of images and real-time generation of results. Confidence and overlap thresholds could be adjusted for flexibility.
Low Latency in Detection	Delivered detection results within seconds, meeting the requirements for real-time applications in industries like retail and e-commerce.
Scalability and Flexibility.	Cloud-based deployment ensured scalability to handle multiple concurrent requests without compromising performance, making it suitable for large-scale operations.

The counterfeit logo identification system worked very well, with a 94.8% accuracy rate using transfer-learned ResNet50 and a 92.3% accuracy rate using bespoke CNN. It met all of its goals and showed great generalization and resilience in a variety of difficult situations. The meticulously selected 10,500 images dataset, supplemented via augmentation and multi-source compilation, greatly decreased error rates and surpasses fixed datasets utilized in past studies. The **sys**tem maintained excellent results under noise, occlusion and poor contrast with the incorporation of YOLOv4's one-stage, detections effectively provide a high-quality region proposal in crowded situations, which the following CNN-based classifier then accurately evaluated for uniqueness.

4.0 Conclusion

The deployment of the model within a user-friendly web interface validated the system's practical applicability. Logo images upload received real-time feedback on authenticity, with minimal latency and high usability ratings. This aligns with the broader goal of democratizing access to AI-powered brand protection tools, especially in regions where manual verification is impractical or unavailable. The dataset included both genuine and counterfeit logo images. The images were augmented using primary model for classification. to improve detection accuracy. The system's implementation was done using Python libraries, and performance was evaluated using precision, recall, and F1-score metrics. Results indicated that CNNs are robust in managing variations in color, shape, and resolution, affirming their applicability in real-world scenarios. The model achieved a classification accuracy of 94.7% and demonstrated high precision and recall in differentiating between authentic and counterfeit logos. This system presents a scalable solution for e-commerce platforms and holds significant implications for policymakers and industries seeking to combat counterfeit trade. A lightweight mobile version of the system should be developed to enable on-the-go logo verification, especially for customs officers, brand inspectors, and consumers in retail environments.

References

[1] Rocabado, F., Perea, M. and Duñabeitia, J. A. (2023). Misspelled logotypes: the hidden threat to brand identity. *Scientific reports*, 13(1), 17817.

[2] Hou, Q., Min, W., Wang, J., Hou, S., Zheng, Y. and Jiang, S. (2023). Deep learning for logo detection: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications,* 20(3), 1–24.

[3] Sharma, S., Zou, J. J., Fang, G., Shukla, P. and Cai, W. (2024). A review of image watermarking for identity protection and verification. *Multimed Tools Appl,* 83, 31829–31891.

[4] Mallampalli, N.S., Bheesetti, T.S. and Sudha, D. (2025). Online fake logo detection system using machine learning. *International Advanced Research Journal in Science*, 12(1). 97-104.

[5] Thoke, A. and Rai, S. (2025). Comparative analysis of traditional and modern object detection algorithms. In: Pandit, M., Gaur, M.K., Kumar, S. (eds) Artificial Intelligence and Sustainable Computing. ICSISCET 2024. Algorithms for Intelligent Systems. *Springer*, 205-216.

[6] Sharanabasaveshwara, H.B., Anusha, S. and Radhika, K. (2023). Review of fake logo detection in python. *International Journal of Engineering Research & Technology*, 12(7).

[7] Brando P. R. (2025). The impact of artificial intelligence on modern society. *AI*, 6(8), 190.

[8] S. Murali, V., Gupta, S., Saha, H., Sharan, P., Sinha, F. and Baibhav, K. (2024). Convolutional neural network (CNN) for fake logo detection: A deep learning approach using tensorflow Keras API and data augmentation, *3rd International Conference on Artificial Intelligence for Internet of Things (AIIoT), Vellore*, 204, 1-6.

[9] Chandrika, G., Hemalatha, P., Chandini, B., Ganesh, G. and Yasaswini, A. (2025). Fake logo detection. *International Journal of Creative Research Thoughts*. 13(3), e539-e546.

[10] Battini, V.S., Kishan, S.R. and Valeti, V.D. (2024). Fake logo detection using image processing. *International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS)*, 1-7.

[11] Vanitha, P., Priya, T.M., Navasakthi, P., Devi V.S. and Aarthi, R. (2024). Identification of fake logo detection using deep learning. *3rd International Conference on Artificial Intelligence for Internet of Things (AIIoT)*, 1-6.

[12] Suganthi M. and Sathiaseelan, J.G. (2020). An exploratory of hybrid techniques on deep learning for image classification. *4th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, 1-4.

[13] Trigka, M. and Dritsas, E. (2025). A comprehensive survey of deep learning approaches in image processing, *Sensors*, 25(2)*,* 531.

.

Received: 16/12/2025; Revised: 29/12/2025; Accepted: 30/12/2025; Published: 31/12/2025