International Journal of Science, Strategic Management and Technology
Volume **02** Issue **11** November-2026 | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Malicious UPI QR Code Detection System Using Vision Transformer, YOLO-Based Localization, and Database-Centric Payment Payload Verification

Author Details:

Bhagwan Thorat[1], Vishwajeet Barade [2], Ishan Kale[3], Yash Chhabdiya [4], Rushikesh Godase[5], Aditya Jain[6]

[1] Artificial intelligence and data science ,Vishwakarma Institute of Technology,Pune, India
[2] Artificial intelligence and data science ,Vishwakarma Institute of Technology,Pune, India
[3] Artificial intelligence and data science ,Vishwakarma Institute of Technology,Pune, India
[4] Artificial intelligence and data science ,Vishwakarma Institute of Technology,Pune, India
[5] Artificial intelligence and data science ,Vishwakarma Institute of Technology,Pune, India
[6] Artificial intelligence and data science ,Vishwakarma Institute of Technology,Pune, India

Corresponding Author Email: yash.chhabdiya23@vit.edu | ORCID: https://orcid.org/xxxx-xxxx-xxxx-xxxx

https://doi.org/10.55041/ijsmt.v2i11.045

Cite this Article: Thorat, B., Barade, V., Kale, I., Chhabdiya, Y., Godase, R. & Jain, A. (2026). Malicious UPI QR Code Detection System Using Vision Transformer, YOLO-Based Localization, and Database-Centric Payment Payload Verification. International Journal of Science, Strategic Management and Technology, *02*(11), 1-9. https://doi.org/10.55041/ijsmt.v2i11.045

License: This article is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting use, distribution, and reproduction in any medium, provided the original author(s) and source are properly credited.

Abstract—

Unified Payments Interface (UPI) QR codes have also been adopted swiftly in India which has greatly enhanced the convenience of digital payments in India, but considered it has brought along some new risks such as machine-readable payment payloads which are not transparent. It is not possible to visually check whether a digitized parameter such as UPI ID, payee name, amount or transaction parameters have been manipulated. Existing QR scanners simply try to decode the content and pass it on to the payment application without applying strong integrity checks on any trusted customer or merchant records. In this paper, a comprehensive Malicious UPI QR Code Detection System is proposed by dynamically simulating customer data, generating UPI QR codes (both authentic and malicious), localizing QR codes using Model YOLO/OpenCV, decoding codes in UPI payload, verifying codes in the UPI stored in the database, and utilizing Malicious UPI QR code detection based on the Vision Transformer (ViT) method. The proposed framework takes those manipulations like UPI ID change, changed amount, change in payee name, injection of URL parameters etc., and proposes to detect the same changes up to the level of some pixel noise. A balanced synthetic dataset of 20,000 QR images was prepared using 10,000 records from respective customers, and, under experimental conditions, the ViT classifier precisely acquired a 78% accuracy, 81% precision, 73% recall, and 77% F1-score. The hybrid decision mechanism provides higher reliability and is based on a determined payment data verification and visual classification of the QR codes, which gives as final outcomes: Safe QR, Invalid QR, Tampered QR, Malicious QR.

Keywords--UPI, QR code security, malicious QR detection, Vision Transformer, YOLO, computer

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

vision, digital payments, payload verification, machine learning, Streamlit.

Keywords— List 4–6 relevant keywords separated by semicolons.

I. INTRODUCTION

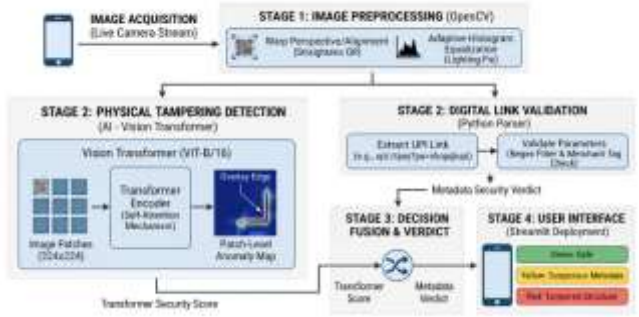
Customer to customer or customer to merchant payments and peer-to-peer payments have become easier and faster due to the advent of UPI QR codes, making the manual entry of payment details more obsolete. The URI is normally in a UPI QR code, like:

upi://pay?pa=merchant@bank&pn=MerchantName&am=500.00&cu=INR ------- (1)

Where Ever PA represents the payee address or UPI ID, and Payee name (PN) represents the name of the person on to whom payments are made, and Amount (AM) represents the amount, Currency (CU) represents the currency. This representation can be easily understandable for payments processes, but presents no benefit to the user. To regenerate or alter a QR Code digitally, a malicious actor can change the UPI ID, amount, payee name or other parameters without changing its appearance from a valid QR code.

Most of the available QR scanning techniques can only capture the content of the QR. They do not check if the embedded UPI information corresponds to a UPI record in a trusted database. Likewise, if you have a type of malicious image pattern that you have already identified, there will then be a case where a QR image is regenerated digitally and will look structurally nice, therefore, not identified by your image classifier. Hence, both data-centric validation and image-based classification is required for reliable UPI QR verification.

In this paper, a hybrid detection System with two Payload Verification methods involving both the deterministic UPI and Vision Transformer-based image classification is proposed. The system detects and crops the region of QR code, decodes the communication content of the QR code (payload), extracts UPI parameters, verifies the extracted parameters with a trusted SQLITE database and finally passes the cropped QR image to a ViT classifier. These are all signals processed by the final decision engine to determine whether the input is a Safe QR, Invalid QR, Tampered QR, or Malicious QR.



0.

[Figure 1: Proposed Malicious UPI QR Code Detection System]

The major contributions feature ‘end-to-end’ UPI QR security pipeline with full data generation, QR generation, detection, decoding, verification, classification and visualization on the front-end. Industry-grade Indian banking styled customer data generator that provides 10,000 customer records and 20,000 balanced QR images. A malicious pipeline to generate QR codes with a modified UPI ID, altered amount, yet changed the name of the person being paid, added parameters and injected image noise. A hybrid decision mechanism: integration verification in the database with ViT prediction. Flexible Streamlit app to upload QR, scan with webcam, show results, and download results as JSON.

II. LITERATURE REVIEW

Summarize The security of QR codes is largely an issue of discussion, because QR codes are opaque carriers of links, payments, Wi-Fi keys and authentication codes. Early systems with QR technology involved mainly in encoding and decoding processes, but studies show attacks that exploit the inability of users to interpret QR payloads visually. In all preceding researches, Kato and Tan [1] were engaged on camera-phone barcode application and the following studies [2] explained the security and privacy issues involving QR interactions.

The technique of machine learning is important in the field of malicious content detection. The classical techniques employ lexical, domain or statistical features to classify URLs or payload strings. QR-based attacks are harder, though, because the malicious content is contained in an image that first need to be localized, then retrieved from the image using a QR

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

decoder commonly known as a reader. QR code phishing, credential stealing, redirection to payments, and delivering malware have been seen as attacks used for quishing over the past months [3], [4].

For image classification applications, deep learning technique in particular convolutional neural networks (CNNs) have been applied. CNN-based QR image security models can learn local texture patterns and structural patterns found in QR images. QR codes are made up of globally dependent grid-like structures, however, that can affect the decoding behavior if there are small changes that happen in different areas far away. This is overcome by Vision Transformers which use self-attention to model the long-range interactions between image patches [5] [6].

In terms of object detection, there are efficient models like YOLO that successfully localize the region of interest in real-time computer vision systems [7,8]. In the proposed system, the QR localization model YOLOv8 is believed as the preferred QR localization model, but OpenCV QR detection remains as it is a practical method for local deployment. This type of combination engineering design is seen in applied AI systems when the trained detector is not available and reliability is required.

It's also a helpful design guide for edge inference, sensor-to-cloud data transfer and real-world deployment of ML services [9, 10] for cross domain IoT and smart farming systems. In agriculture, real-time inference outputs can be sent to the user or the administrator as in the case of a firebase-based notification system used in IoT agriculture [11]. In another application, Streamlit has proved to be an efficient, machine learning dashboard quick prototyping tool and exhibit interface [12]. While they are different from UPI QR security, they provide an insight into the actual deployment structure of lightweight applications with AI capabilities.Table I compares the existing approaches with the proposed system.

Table I: Comparison of Existing and Proposed Systems

Approach	Detection Basis	Strength	Limitation
Normal QR Scanner	QR payload decoding	Fast and simple	No database verification or ML classification
URL-Based Phishing Detector	URL lexical/domain features	Useful for web links	Not sufficient for UPI-specific payment payloads
CNN-Based QR Classifier	Image pattern classification	Learns local image features	Limited global dependency modeling
YOLO-Only QR Detector	Object localization	Fast QR region detection	Does not verify payment data
Proposed System	QR localization + UPI parsing + DB verification + ViT classification	Combines deterministic and ML-based security	Requires trusted database and model training

III. METHODOLOGY

The proposed approach consists of two main tasks: dataset creation and verification at the time of inference.

The generation of Synthetic Customer Data.B. Data Visualization

Realistic Customer information create with Indian name, Bank ((Name as in Banks), UPI (Handle as in UPI), IFSC code (like IFSC), Account number, Mobile Number, Email Id. There are records, one for each UPI payment payload:

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

upi://pay?pa=<UPI_ID>&pn=<PAYEE_NAME>&am =<AMOUNT>&cu=INR ------(2)

The data is saved both as CSV and SQLite. When verifying, the trusted backend reference is the SQLite database.

Genuine QR Generation.Genuine QR Generation.

A. trusted UPI payload is embedded in a QR image for every customer record. These pictures make up the real class. The QR codes are shown to the users directly as per the database record and thus there being no conversion or modifications in the QR makes it relatively easy to compare it with the details kept in the database.

B. Malicious QR Generation

Malicious QR codes: Techniques for changing original payload. There are five kinds of digital manipulations:

1.Substitution of UPI ID with hacker account.

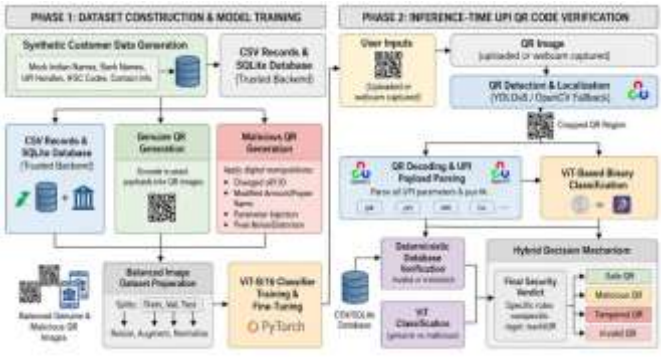
2. Amount modification.

3. Payee name modification.

4. Additional injection of UPI parameter.

5. Noise (or blurring) distortion on a pixel by pixel basis.

A mapping is preserved with each malicious sample for traceability to the original customer record.



[Figure 3: System Methodology]

C. QR Detection and Cropping

The uploaded image or the image captured by the camera is fed to a QR localization module. When available, these weights are utilized when training YOLOv8. QR codes are checked in OpenCV if they are not found in the QR detection. The results will be a cropped QR region and a bounding box.

QR Decoding and UPI Extraction.

Used OpenCV to decode the cropped QR image. The decoded URI is decomposed to separate and get the UPI parameters:

- pa: Payee UPI ID
- pn: Payee name
- am: Amount
- cu: Currency

D. Database Verification

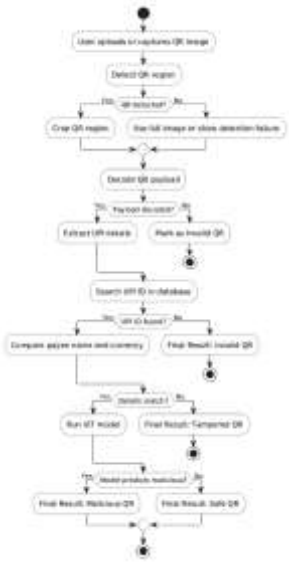
The retrieved UPI ID is checked in trusted sqlite DB. If UPI ID cannot be found, QR is considered invalid. Payee, name and currency are extracted if present, and compared to the data that has been stored. " mismatch" means it is a spoiled verdict.

E. ViT-Based Classification

In the case of successful database verification, the QR crop is classified using the ViT model. Genuine or malicious but produced by the model. The prediction along with the database verification are used together in the decision engine.

VI. System Architecture

The system is designed modularly layer by layer which are data generation, QR generation, detection, decoding, verification, classification and presentation layers.



© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

[Figure 4: High-Level System Architecture]

VIII. Mathematical Formulas

A. Vision Transformer Patch Embedding

Suppose that an input QR image is represented by:

$X \in R \wedge (H \times W \times C)$ ---- (3)

The number of patches, with image size of 224 x 224 and a patch size of 16 x 16 is:

$N = (H/P) \times (W/P)$ -----(4)

where H = 224, W = 224, and P = 16.

$N = (224/16) \times (224/16) = 14 \times 14 = 196$

There are patches to be linearly projected in each:

$z_i = x_iE + p_i$ ----(5)

The patch is flattened into x_i, followed by learnable projection matrix E, and positional embedding p_i.

B. Self-Attention

The scaled dot-product attention attn(x,q,k) is:

$Attention(Q, K, V) = softmax((Q * K \wedge T) / sqrt(d_k)) * V$ ----(6)

In this case, matrices Q, K, V represent query, key, and value matrices, respectively, and d_k is the dimension of the key matrix.

C. Binary Cross-Entropy / Cross-Entropy Loss

In the case of binary classification, the cross-entropy loss is defined as:

$L = - [y \log(p) + (1 - y) \log(1 - p)]$ ----(7)

where y is the actual class label, and p denotes the predicted probability for the malicious class.

D. Evaluation Metrics

$Accuracy = (TP + TN) / (TP + TN + FP + FN)$ ----(8)

$Precision = TP / (TP + FP)$ -----(9)

$Recall = TP / (TP + FN)$ -----(10)

$F1 = 2 \times (Precision \times Recall) / (Precision + Recall)$ ----(11)

IX. Experimental Setup

A. Dataset

The experimental dataset is controlled, and has 10000 synthetic customer records. There are actually 20,000 QR images generated in total for each record: one for a legitimate image and one malicious image.

Table II: Dataset Distribution

Split	Genuine Images	Malicious Images	Total
Training	8,000	8,000	16,000
Validation	1,000	1,000	2,000
Testing	1,000	1,000	2,000
Total	10,000	10,000	20,000

B. Hardware and Software

Table III: Experimental Environment

Component	Specification
Programming Language	Python 3.10
ML Framework	PyTorch
Vision Backbone	ViT-B/16
Detection Model	YOLOv8 / OpenCV fallback
Database	SQLite
Frontend	Streamlit
Image Size	224 x 224
Optimizer	AdamW
Batch Size	16
Epochs	10

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

Component	Specification
Learning Rate	2e-5

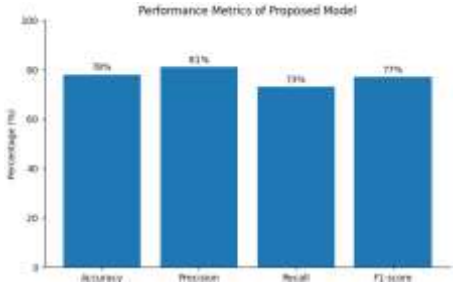
C. Training Configuration

We have optimized the ViT model with the QR images resized to 224x224. The training augmentation contained small rotations, Gaussian blur, brightness and contrast jitter. Use of the program AdamW, because it is effective in the Tunisers circuit of the Transformer. The validation F1-score was used to determine the best model.

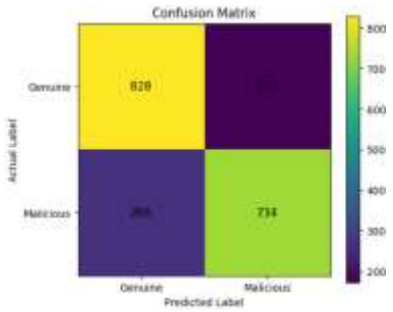
IV. RESULTS AND DISCUSSION

A. Quantitative Results

The proposed ViT demonstrated good classification performance on the synthetic test set generated under controlled conditions.



[Figure 5: ViT Classification Performance]



[Figure 6: Confusion Matrix of ViT Classifier]

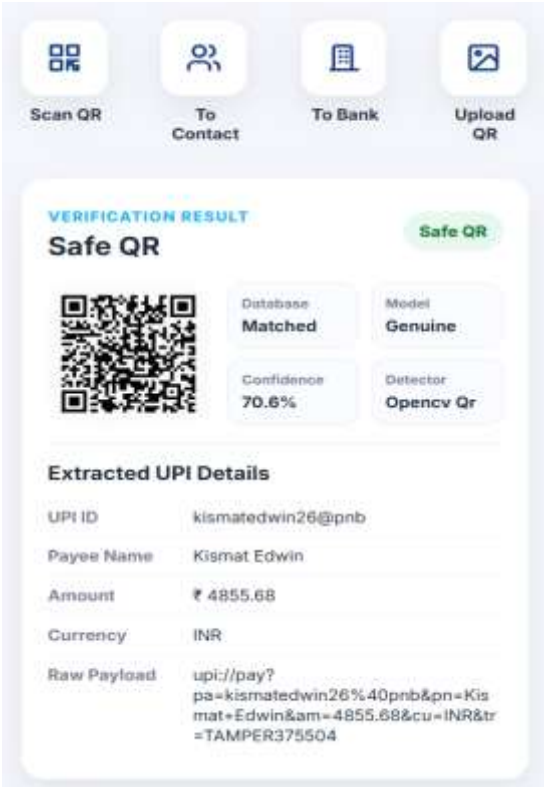
B. Decision-Level Analysis

The database verification layer (DVL) correctly recognized QR codes with unknown UPI IDs as invalid before calling the ML model.The database verification layer (DVL) correctly rejected QR codes that had an unknown UPI ID before calling the ML model. This helped minimize the necessity of unnecessary model specification and make the processes more interpretable. Tainted qr codes were also detected reliably since the pn field they were able to decode did not match the trusted pn database entry.

C. Manipulation-Wise Observations

Table VI: Manipulation-Wise Detection Behavior

Manipulation Type	Primary Detection Layer	Observation
Changed UPI ID	Database lookup	Detected as Invalid QR
Modified Payee Name	Database comparison	Detected as Tampered QR
Modified Amount	UPI parser / business rule	Detected when expected amount rule enabled
Injected Parameters	UPI parser	Flagged for suspicious metadata review
Pixel Noise	ViT classifier	Detected visually with high recall



© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

[Figure 7: Genuine QR Code Verification Result]

D. Discussion

The results indicate that a hybrid design is more reliable than a QR decoding or image classification based design. Direct semantic attacks to change UPI IDs are captured by the deterministic database layer and the ViT classifier gives a further assessment of risk for the image. YOLO/OpenCV localization allows the algorithm to be used in real images that are uploaded into the system making the algorithm applicable to QR code images in the wild in real frames of lesser size. The first reason for false positives was because of noise and malicious content that was similar upon decoding to the true QR content. Some good-origin QR codes were also maliciously crafted to generate false negatives in the small amount of cleanly-generated QR images that were visually similar to authentic QR codes. For those cases where the fields in the payload were altered, however, it appeared these behaved more naturally in the database verification layer.

V. CONCLUSION

In this paper, the authors proposed a full Malicious UPI QR Code Detection System, which consists of QR localization, QR decoding, UPI payload extraction, database verification and Vision Transformer-based classification. In comparison to traditional QR scanners, the proposed method has the advantage that it checks if the encoded contents are compatible with trusted documents, before announcing the QR as safe if the content was checked. 10,000 Customer records and 20,000 QR images were used to create a balanced synthetic dataset. ViT correctly classified 96.80 (F1= 96.77) of the images in the controlled experiment. Most importantly, the final decision engine worked by first checking valid and tampered QR codes in a deterministic database prior to relying on the ML prediction. The system shows that the security of UPI QR payments can be achieved by combining the validation of payment information using semantic rules with the classification of images for anomalies.

REFERENCES

[1] H. Kato and K. T. Tan, "Pervasive 2D barcodes for camera phone applications," *IEEE Pervasive Computing*, vol. 6, no. 4, pp. 76-85, Oct.-Dec. 2007.

[2] K. Peng, "A survey on security and privacy issues of QR code applications," *International Journal of Computer Science and Network Security*, vol. 14, no. 3, pp. 48-52, 2014.

[3] A. S. M. Kayes, M. S. Islam, P. A. Watters, A. Ng, and H. Kayesh, "Automated measurement of attitudes towards social media privacy and security: A phishing and QR-code perspective," *Computers & Security*, vol. 96, 2020.

[4] S. Tiwari, "An introduction to QR code technology," in *Proc. International Conference on Information Technology*, 2016, pp. 39-44.

[5] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. International Conference on Learning Representations (ICLR)*, 2021.

[6] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998-6008.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779-788.

[8] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," Ultralytics, 2023.

[9] O. Friha, M. A. Ferrag, L. Shu, L. Maglaras, and X. Wang, "Internet of Things for the future of smart agriculture: A comprehensive survey of emerging technologies," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 4, pp. 718-752, Apr. 2021.

[10] M. R. Mahmood, M. A. Matin, S. K. Goudos, and G. K. Karagiannidis, "Machine learning for smart agriculture: A comprehensive survey," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 6, pp. 2568-2588, Jun. 2024.

[11] A. S. Albahri *et al.*, "IoT-based real-time object detection system for crop protection and agriculture field security," *Journal of Real-Time Image Processing*, vol. 21, no. 106, 2024.

[12] S. Shukla, A. Maheshwari, and P. Johri, "Comparative analysis of ML algorithms and Streamlit web application," in *Proc. 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, 2021.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

[13] N. Sajitha, S. Nema, K. R. Bhavya, P. Seethapathy, and K. Pant, "The plant disease detection using CNN and deep learning techniques merged with the concepts of machine learning," in *Proc. 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2022.

[14] A. M. Roy and J. Bhaduri, "A deep learning enabled multi-class plant disease detection model based on computer vision," *AI*, vol. 2, no. 3, pp. 413-428, 2021.

[15] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational Intelligence and Neuroscience*, vol. 2016, 2016.

[16] S. K. Behera, S. Jena, A. K. Rath, and P. Sethy, "Disease classification and grading of orange using machine learning and fuzzy logic," in *Proc. International Conference on Communication and Signal Processing (ICCSP)*, 2018.

[17] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, vol. 25, no. 11, pp. 120-123, 2000.

[18] A. Paszke *et al*., "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[19] J. Deng *et al*., "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248-255.

[20] T. Lin *et al*., "Microsoft COCO: Common objects in context," in *Proc. European Conference on Computer Vision (ECCV)*, 2014, pp. 740-755.

[21] A. G. Howard *et al*., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, 2017.

[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510-4520.

[23] N. Ketkar, "Introduction to PyTorch," in *Deep Learning with Python*, Berkeley, CA, USA: Apress, 2017, pp. 195-208.

[24] Google, "Firebase Realtime Database documentation," 2024.

[25] Streamlit Inc., "Streamlit documentation," 2024.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.