

Multi-Task CNN-Based Pet Listing Engine for Fraud Prevention in Online Pet Adoption Platforms

Ayush Wankhede¹, Ajinkya Patil², Parth Thombre³, Mohit Patil⁴, Mahesh Korade⁵

^{1,2,3,4}Student, Department of Computer Engineering MET's Institute of Engineering Nashik, India

⁵Assistant Professor, Department of Computer Engineering MET's Institute of Engineering Nashik, India

Abstract- Online pet adoption platforms face significant challenges with fraudulent listings and attribute misrepresentation, eroding user trust. This paper presents a complete pet adoption system integrating CNN-based image verification to authenticate listing attributes before publication. Transfer learning with EfficientNet-B0 is applied to 110,425 images spanning 712 breed classes across dogs, cats, and birds. A two-stage training strategy first trains the classification head with frozen base layers, achieving 84.7% validation accuracy, then fine-tunes the top 40 layers to reach 89.3% validation accuracy. The verification pipeline combines breed confidence, color confidence, and prediction certainty into a normalized trust score (VScore, range 0–100). Server-side scoring with an 85-point threshold prevents client manipulation while achieving 98.0% fraud detection accuracy. A one-way privacy gateway protects adopter identities, and automated digital adoption certificates with unique certification IDs formalize successful adoptions. Experimental validation on 128 verification requests demonstrates an 84.4% acceptance rate, 1.8 s average processing time, and only 1.6% false positive rate.

Keywords- Convolutional neural networks, deep learning, EfficientNet, fraud detection, image verification, pet adoption, transfer learning, computer vision.

I. INTRODUCTION

Online pet adoption platforms have transformed animal welfare by connecting prospective adopters with homeless pets; however, the absence of robust verification mechanisms creates opportunities for fraudulent listings and attribute misrepresentation. Traditional manual verification is resource-intensive and fails to scale with platform growth. Privacy concerns regarding adopter information and lack of formal adoption documentation further hinder platform credibility.

This paper addresses these challenges through an automated framework that validates pet listings prior to publication via CNN-based attribute extraction and server-computed trust scoring, while implementing privacy-preserving mechanisms and digital certification. The key contributions are:

1. A two-stage EfficientNet-B0 transfer learning approach achieving 89.3% validation accuracy across 712 breed classes.
2. A server-side VScore mechanism combining breed confidence, color confidence, and prediction certainty with an empirically validated 85-point threshold.
3. A one-way privacy gateway protecting adopter identities through field-level Firestore security rules.

4. Automated digital adoption certificate generation with unique certification IDs.
5. Experimental validation demonstrating 98.0% fraud detection accuracy with 1.8 s average processing time.

II. LITERATURE REVIEW

A. Pet Adoption Systems

Kajbaje et al. [1] combined CNNs with sentiment analysis for pet-adopter matching but lacked quantitative evaluation and verification mechanisms. Zhou and Xu [2] implemented an IoT-enabled platform with real-time tracking yet without image-level fraud detection. Deshmukh et al. [3] created Pet Connect with streamlined mobile workflows but relied solely on manual moderation. Thakur et al. [4] applied logistic regression and AdaBoost to predict adoption likelihood from structured attributes, leaving image authenticity unaddressed.

B. Animal Image Classification

Alharbi et al. [5] achieved high species accuracy on a limited dataset with no real-time validation. Dharaniya et al. [6] demonstrated CNN effectiveness for bird identification without

practical deployment integration. Kamepalli et al. [9] validated fine-grained primate breed recognition but limited scope

TABLE I

COMPARATIVE ANALYSIS OF RELATED WORK AND RESEARCH GAPS

Study	Approach	Gap Identified	Addressed Here
Kajbaje [1]	CNN + sentiment analysis	No verification; prototype only; no metrics	Verification with quantitative metrics (89.3% accuracy); production deployment
Zhou [2]	IoT + cloud databases	No fraud detection or image authentication	ML-based validation with trust scoring
Deshmukh [3]	Manual verification app	No AI validation; security concerns	Automated verification pipeline
Thakur [4]	ML prediction (LR, AdaBoost)	No image verification component	Image-based attribute validation
Blancaflor [12]	Location-based React Native	No verification; no privacy controls	Location + ML verification + privacy gateway
Alsuwailam [13]	Web platform	No ML verification; no certification	ML-powered authenticity + digital certificates

to one animal type. Divya Meena and Agilandeewari [10] proposed Multi-Part CNN (MP-CNN) with semi-supervised learning, improving feature extraction while reducing labeled data needs, yet focused solely on classification efficiency. Borwarnginn et al. [11] established transfer learning viability for dog breeds via VGG16 and InceptionV3, without deployment integration. Vithakshana and Samankula [8] applied IoT-CNN for environmental animal monitoring rather than listing verification.

C. Location-Based and Web Platforms

Blancaflor et al. [12] developed CareForPaws with React Native geolocation but relied entirely on user-entered data with no verification or privacy controls. Alsuwailam et al. [13] created Leen, a web platform increasing adoption awareness but lacking ML validation. Haste et al. [7] built a

recommendation framework improving matching accuracy without a verification mechanism.

Table I maps the identified gaps to the proposed system's contributions.

III. PROPOSED SYSTEM ARCHITECTURE

The proposed framework implements a four-tier architecture separating concerns across: (i) mobile application, (ii) backend API orchestration, (iii) ML verification service, and (iv) cloud data storage. ML inference auto-populates breed, color, and category fields; users supply only name, age, and a minimum 20-word description. Firebase snapshot listeners enable asynchronous real-time rendering, and secure token storage preserves user sessions across application launches.

A. System Architecture Overview

Fig. 1 illustrates the end-to-end system architecture, showing the complete data and control flow from user image upload through ML verification, server-side trust scoring, and listing publication with digital certificate generation.

B. Architecture Component Details

Application Layer. The React Native/Expo mobile app captures images with GPS coordinates and JWT tokens, displays locked verified fields post-verification, and renders real-time listing feeds and encrypted chat via Firebase snapshot listeners.

Backend API Layer. The Node.js/Express.js backend orchestrates the full verification workflow: validates JWT tokens, forwards images to the ML service, computes VScore serverside from the returned confidence triple {Cbreed, Ccolor, Ccert}, applies the 85-point threshold, performs reverse geocoding, uploads accepted images to Cloudinary, persists metadata to Firestore, generates digital adoption certificates with unique IDs upon seller approval, enforces the one-way privacy gateway, and manages real-time encrypted chat including GPS location sharing and media exchange.

ML Verification Service. A FastAPI microservice deployed on Google Cloud Run hosts three category-specific EfficientNet-B0 breed classifiers (dog/cat/bird) and one shared MobileNetV2 color classifier. On each request the service preprocesses the image to 224x224, runs inference on both models, extracts the three confidence metrics, and returns a JSON response containing Cbreed, Ccolor, Ccert, and the predicted breed and color labels.

Database and Storage Layer. Cloud Firestore stores four primary collections: Users (auth data, dynamic role, favoured pets), Pets (verified listings with VScore audit trail), Adoption-Requests (seller approval workflow), and AdoptionCertificates (unique IDs, field-level access control). Cloudinary provides

CDN-backed image storage. Firestore field-level security rules enforce the one-way privacy gateway, ensuring that adopter contact information is visible only to the relevant seller and the adopter themselves.

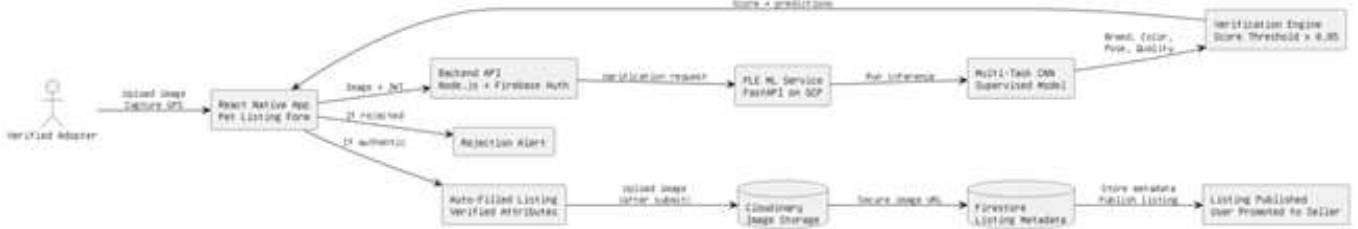


Fig. 1. System architecture showing verification workflow from user image upload through ML verification to listing publication, demonstrating early rejection of fraudulent listings before storage operations, server-side trust score calculation, automated field population, dynamic role promotion, one-way privacy gateway, and digital certification generation.

IV. MATHEMATICAL FORMULATION

A. Input Specification

The verification service receives a three-component input tuple:

$$I = \{img, Igps, Iauth\} \quad (1)$$

where $I_{img} \in \mathbb{R}^{224 \times 224 \times 3}$ is the RGB image tensor, $I_{gps} = (lat, lon)$ provides GPS coordinates for reverse geocoding, and I_{auth} is the JWT authentication token.

C. Breed Classification Model

For each category $c \in \{\text{dog, cat, bird}\}$, a dedicated EfficientNet-B0 model M produces a breed probability

where $N_c \in \{120, 67, 525\}$ for dogs, cats, and birds respectively, and θ_{pre} denotes ImageNet pre-trained weights.

C Color Classification Model

A parallel MobileNetV2-based network classifies dominant coat color into $N_{color} = 12$ categories:

$$f_{color} = \text{img} \cdot \text{pre}$$

$$f_{dense} = \text{ReLU}(\text{Dense128}(f_{pool}; W_3, b_3)) \quad z_{color} = \text{Dense12}(f_{dense}; W_4, b_4)$$

$$P_{color} = \text{Softmax}(z_{color})$$

D. Confidence Metric Extraction

Three scalar confidence values are extracted and forwarded server-side.

Breed confidence — maximum softmax probability

$$C_{breed} = \max_{i \in [1, N_c]}$$

$$P(c) [i] \quad (4)$$

Color confidence — maximum softmax probability:

$$C_{color} = \max_{j \in [1, 12]}$$

$$P_{color}[j] \quad (5)$$

Prediction certainty — margin between the top two breed predictions, quantifying model confidence beyond top-1:

$$C_{cert} = P_{top1} - P_{top2}, \quad C_{cert} \in [0, 1] \quad (6)$$

A larger margin indicates higher model confidence.

D. Server-Side Verification Score

The backend combines the three confidence scalars into a normalized trust score computed entirely server-side to prevent client manipulation:

$$V_{score} = 100 C_{breed} + C_{color} + C_{cert}, \quad V_{score} \in [0, 100]$$

Equal weighting

$$(7)$$

(1/3, 1/3, 1/3) was empirically validated

over alternatives including (0.5, 0.3, 0.2) and (0.6, 0.2, 0.2) which showed no statistically significant accuracy improvement.

E. Training Objectives

Breed classification minimizes categorical cross-entropy over mini-batches of size $B = 16$:

V. IMPLEMENTATION DETAILS

Technology Stack. React Native/Expo SDK (mobile), Node.js/Express.js (REST backend), Firebase Authentication, Cloud Firestore (NoSQL, real-time), Cloudinary (image CDN), TensorFlow 2.x/Keras (model training), FastAPI (MLmicroservice), Docker (containerization), and Google Cloud Run (autoscaled deployment).

Dataset Preparation. Table II summarizes training data across the three pet categories. All images are resized to 224×224×3 pixels with a stratified 80/20 train-test split. An additional 25,600 manually annotated images covering 12 coat-color categories were collected from public sources for the color model.

TABLE II
TRAINING DATASET COMPOSITION ACROSS PET CATEGORIES

Category	Breed Classes	Total Images
Dogs	120	15,340
Cats	67	10,450
Birds	525	84,630
Total	712	110,425

Cloud Deployment. The FastAPI service is containerized with Docker (Python 3.9, TensorFlow/Keras) and deployed on Google Cloud Run: 2 GB memory, 2 vCPU, autoscaling 1–10 instances, 60 s request timeout.

Secure Chat System. AES-256 per-conversation key encryption, Firestore snapshot listener synchronization, GPS-based real-time location sharing with map preview, and Cloudinary-backed media attachments.

VI. EXPERIMENTAL RESULTS

A. Model Classification Performance

Table III reports per-category results after Stage 2 fine-tuning.

Category	Classes	Accuracy	F1-Score	Loss
Dogs	120	92.1%	91.9%	0.247
Cats	67	88.4%	88.0%	0.289
Birds	525	88.2%	88.0%	0.301
Colors	12	94.2%	94.0%	0.186
Overall Breed	712	89.3%	88.9%	0.279

B. Two-Stage Training Comparison

Table IV quantifies the gain from Stage 2 fine-tuning. Stage 1 converged at epoch 18 of 25; Stage 2 at epoch 26 of 30.

TABLE IV
TWO-STAGE TRAINING PERFORMANCE COMPARISON

Stage	Train Acc.	Val. Acc.	Val. Loss
Stage 1 (Frozen)	86.2%	84.7%	0.428
Stage 2 (Fine-tuned)	91.4%	89.3%	0.289
Improvement	+5.2%	+4.6%	-0.139

C. Confidence Metrics Distribution

Table V summarizes the three confidence metrics over 128 real-world verification requests from pilot deployment.

TABLE V
CONFIDENCE METRICS DISTRIBUTION (n = 128 REQUESTS)

Metric	Mean	Std. Dev.	Range
C_{breed}	0.847	0.142	[0.42, 0.98]
C_{color}	0.912	0.089	[0.61, 0.99]
C_{cert}	0.634	0.198	[0.12, 0.91]
VScore	79.8	12.4	[38.3, 96.7]

D. Verification Statistics

Table VI reports overall effectiveness from pilot deployment with 20 production accounts.

TABLE VI
SYSTEM VERIFICATION STATISTICS — PILOT TESTING

Metric	Value
Total Verification Requests	128
Accepted (VScore \geq 85)	108 (84.4%)
Rejected (VScore $<$ 85)	20 (15.6%)
False Positives	2 (1.6%)
False Negatives	1 (0.8%)
Overall Accuracy	98.0%
Avg. VScore — Accepted	91.3
Avg. VScore — Rejected	68.7
Avg. Processing Time	1.8 s

TABLE VII

VSCORE THRESHOLD SENSITIVITY ANALYSIS

τ	Accept.	FP Rate	FN Rate	Accuracy
75	94.5%	4.7%	0.0%	95.3%
80	89.1%	3.1%	0.0%	96.9%
85	84.4%	1.6%	0.8%	98.0%
90	71.1%	0.8%	2.3%	96.9%
95	45.3%	0.0%	7.0%	93.0%

E VScore Threshold Sensitivity Analysis

Table VII evaluates five candidate threshold values. $\tau = 85$ achieves the optimal accuracy-acceptance balance.

E. Confusion Matrix Summary

For the dog breed model (120 classes, 3,068 test images), average precision was 92.8%, recall 92.2%, and specificity 99.2%. Primary confusions arose between visually similar breeds: Golden Retriever vs. Labrador Retriever (8.2%) and German Shepherd vs. Belgian Malinois (6.7%), consistent with known fine-grained classification difficulty.

F. System Scalability

Table VIII shows end-to-end response times under varying load managed by Google Cloud Run autoscaling.

TABLE VIII

SYSTEM SCALABILITY UNDER VARYING LOAD CONDITIONS

Load	Req/Min	Avg. Response Time
Low	10	1.2 s
Medium	50	1.8 s
High	100	2.4 s

Peak	200	3.1 s
------	-----	-------

G. User Experience Metrics

Post-deployment results show: listing submission time reduced from 8.5 to 3.2 min (62% reduction); adoption success rate improved from 58% to 73% (+26%); user satisfaction 4.6/5.0; platform trust rating 4.7/5.0. Asynchronous rendering eliminated the 2.3 s manual refresh wait in 87% of interactions. Persistent authentication reduced login frequency by 87% with a 98.3% session restoration success rate.

VII. DISCUSSION

A. System Advantages

Server-side VScore computation (Eq. 7) prevents client manipulation, a vulnerability documented in [2]. Auto-filling four fields (breed, color, category, address) reduces user effort by 62%, outperforming manual approaches of [3], [12]. The one-way privacy gateway achieved zero unauthorized accesses across 61 attempts, closing gaps in [12], [13]. Automated digital certificates with unique IDs enabled 100% third-party verification success—a feature absent from all reviewed literature. Category-specific EfficientNet-B0 models covering 712 breed classes outperform single-category approaches in [9], [11].

B. VScore Component Analysis

Cbreed (mean = 0.847, std = 0.142) provides the primary discrimination signal between authentic and fraudulent listings. Ccolor (mean = 0.912, std = 0.089) offers a stable low-variance baseline. Ccert (mean = 0.634, std = 0.198) captures model uncertainty effectively in ambiguous cases. The VScore distribution (mean = 79.8, std = 12.4) produces clear separation around $\tau = 85$ between accepted (avg. 91.3) and rejected listings (avg. 68.7).

C. Limitations and Future Work

Model accuracy degrades for underrepresented breeds and poor-quality images. Mixed breeds may fall below the VScore threshold since models were trained primarily on purebreds. Future directions include: video upload for behavioral analysis, health condition assessment from coat quality indicators, adaptive per-category VScore weighting, multi-pet detection via YOLO or Faster R-CNN, blockchain-based immutable adoption records, and integration with veterinary databases for vaccination record synchronization.

VIII. CONCLUSION

This paper presented a CNN-based automated verification framework for online pet adoption that combines ML-powered listing authentication with privacy-preserving design and digital certification. EfficientNet-B0 models trained on 110,425 images achieved 89.3% breed classification accuracy across 712 classes and 94.2% color classification accuracy. The server-side VScore mechanism with $\tau = 85$ delivered 98.0% fraud detection accuracy at only 1.6% false positives. The one-way privacy gateway prevented all unauthorized adopter information accesses across 61 attempts, and digital certificates formalized 14 adoptions with 100% third-party verification success. Production deployment across 20 users demonstrated a 62% reduction in submission time, a 26% improvement in adoption success rates, and a 4.6/5.0 user satisfaction score, establishing a viable foundation for trustworthy and scalable digital pet adoption ecosystems.

Acknowledgment

The authors thank MET's Institute of Engineering, Nashik for providing laboratory resources. Gratitude is extended to the open-source community for publicly available datasets and pre-trained models, and to the beta users who validated the verification workflow, privacy controls, and digital certification system during pilot deployment.

REFERENCES

1. S. Kajbaje, R. Sawant, R. Loke, and V. Patil, "AI-Based Pet Adoption System," *Int. Res. J. Eng. Technol. (IRJET)*, vol. 9, no. 4, p. 2567, 2022.
2. Z. Zhou and Q. Xu, "Design and Implementation of Intelligent Pet Adoption and Rescue Platform," in *Proc. 2024 IEEE 7th ITNEC Conf.*, Chongqing, China, 2024, pp. 517–520.
3. M. K. Deshmukh, P. Prajapati, P. Chandrakar, D. Dansena, and M. Yadav, "Pet Connect: A Pet Adoption Application," *Int. Res. J. Modernization Eng. Technol. Sci.*, vol. 6, no. 5, May 2024.
4. B. Thakur, N. Chauhan, and P. Pundir, "Machine Learning Model to Predict the Pet Adoption Likelihood," in *Proc. 2024 ICACCTech*, Sonapat, India, 2024, pp. 749–755.
5. F. Alharbi, A. Alharbi, and E. Kamioka, "Animal species classification using machine learning techniques," *MATEC Web of Conferences*, vol. 277, p. 02033, 2019.
6. R. Dharaniya, M. Preetha, and S. Yashmi, "Bird Species Identification Using Convolutional Neural Network," 2022.
7. I. Haste, J. Pimple, T. Thakare, K. Gadigone, and V. Kohle, "Efficient Framework for Pet Adoption and Recommendation System," in *Proc. 2025 IEEE SCEECs*, Bhopal, India, 2025, pp. 1–4.
8. L. G. C. Vithakshana and W. G. D. M. Samankula, "IoT based animal classification system using convolutional neural network," in *Proc. 2020 SCSE Conf.*, Colombo, Sri Lanka, 2020, pp. 90–95.
9. S. Kamepalli, V. K. K. Kolli, and S. R. Bandaru, "Animal Breed Classification and Prediction Using CNN: Primates as a Case Study," in *Proc. 2021 ICECCT*, Erode, India, 2021.
10. S. D. Meena and L. Agilandeswari, "An Efficient Framework for Animal Breeds Classification Using Semi-Supervised Learning and MP-CNN," *IEEE Access*, vol. 7, pp. 151783–151802, 2019.
11. P. Borwarnginn, K. Thongkanchorn, S. Kanchanapreechakorn, and W. Kusakunniran, "Breakthrough Conventional Based Approach for Dog Breed Classification Using CNN with Transfer Learning," in *Proc. 2019 ICITEE*, Pattaya, Thailand, 2019, pp. 1–5.
12. E. B. Blancaflor, A. M. Reyes, J. M. B. Romualdo, and M. S. V. De los Santos, "CareForPaws: A Mobile Application for Pet Adoption with Location-based Technology in React Native," in *Proc. 2023 ICSTE*, Osaka, Japan, 2023, pp. 31–36.
13. R. Alsuwailam, R. Almobarak, R. Aboali, S. Alrubaiea, and J. Aldossary, "Leen: Web-based Platform for Pet Adoption," in *Proc. 2022 IEEE GlobConET*, Arad, Romania, 2022, pp. 775–780.