AI PET EMOTION RECOGNITION SYSTEM USING DEEP LEARNING

Mrs. V. Latha Sivasankari[1], Aswin SP[2], Figo Y[3], Rahul S M[4]

[1] Assistant Professor, Final MCA[2,3,4]

[1,2,3,4] Department of Computer Applications (PG),

[1,2,3,4]Hindusthan College of Arts and Science, Coimbatore, Tamil Nadu, India

Abstract:. Understanding pet emotions is crucial for enhancing animal welfare and strengthening human-pet bonds, yet objective emotion assessment remains challenging. This project addresses this need by developing a comprehensive pet emotion detection system using real-time TensorFlow Lite inference on mobile devices. The system employs a Convolutional Neural Network trained on dog facial expressions to classify emotions into four categories: angry, happy, relaxed, and sad. The implementation addresses critical mobile machine learning challenges, including preprocessing pipeline alignment, model optimization, and real-time inference performance. Through systematic debugging and Python-like preprocessing implementation, the system successfully eliminated model bias and achieved accurate emotion classification, providing pet owners with valuable insights into their pets' emotional well-being.

keywords: Pet emotion detection; TensorFlow Lite; Convolutional Neural Network; mobile machine learning; real-time inference; EfficientNetB3; animal welfare; computer vision; Flutter application; offline processing

1. INTRODUCTION

This project presents the development of a comprehensive pet emotion detection system using Flutter and TensorFlow Lite machine learning capabilities. The application leverages advanced computer vision algorithms to analyze pet facial expressions and classify emotional states including happiness, sadness, anger, and relaxation. By integrating real-time camera capture with on-device inference, the system provides immediate emotional assessments while maintaining user privacy through local processing. The implementation incorporates BLoC state management for optimal performance, Hive database for persistent storage of emotion histories using NoSQL, and intelligent care recommendations tailored to detected emotional states. This mobile solution addresses the growing need for pet owners to better understand their animals' emotional wellbeing through accessible technology, potentially improving human-animal relationships and early detection of behavioural or health concerns requiring veterinary attention.

2. PROBLEM STATEMENT AND OBJECTIVE

The fundamental challenge in modern pet ownership lies in the communication barrier between humans and their animal companions. Pet owners frequently struggle to accurately interpret their pets' emotional states, leading to delayed responses to distress, inappropriate care interventions, and potential deterioration in human-animal relationships. Traditional methods of emotion assessment rely on behavioural observation, which is subjective, inconsistent, and often requires professional expertise unavailable to average pet owners.

The primary objective of this project is to develop an intelligent, accessible mobile application that bridges this communication gap through automated emotion detection technology. By leveraging TensorFlow Lite machine learning models and Flutter's cross-platform capabilities, the system aims to provide real-time, accurate emotional state analysis of pets using facial expression recognition. The application seeks to democratize access to pet emotional insights, enabling owners to respond appropriately to their animals' needs, enhance wellbeing, and identify potential health concerns early.

Specific objectives include implementing on-device inference for privacy preservation, creating an intuitive user interface for seamless interaction, developing a comprehensive emotion history database, and generating personalized care recommendations based on detected emotional patterns. This technological solution addresses the critical need for objective, consistent pet emotion assessment tools in domestic environments.

4. PROPOSED WORK

This project builds a pet emotion detection mobile app using Flutter and TensorFlow Lite. The system processes images through six stages, from model training to storing results on-device.

Stage 1: Model Architecture and Dataset Preparation

The emotion detection model is based on MobileNetV2, which was chosen mainly because it runs well on phones without needing a lot of compute. The training data was collected from publicly available pet image datasets [7] and manually filtered to remove unclear or blurry samples. Four emotion classes were used: happy, sad, angry, and relaxed. Each class ended up with around 2,000 images after filtering, though the distribution was not perfectly even the "relaxed" class had slightly fewer samples than the others. Before training, images were preprocessed using histogram equalization to handle inconsistent lighting, and a basic facial region crop was applied to reduce background noise. Standard augmentation like flipping, small rotations, and Gaussian noise was also added to help the model generalize better.

Stage 2: Transfer Learning and Model Optimization

The first 100 layers of MobileNetV2 (pre-trained on ImageNet) were frozen, and only the top classification layers were retrained on the pet dataset. This approach was chosen because training the full network from scratch on a dataset of this size would likely lead to overfitting. After initial training, quantization-aware training was applied to convert the model to INT8. Accuracy on the validation set came out to about 92%.

Stage 3: Real-Time Image Processing Pipeline

The app captures frames from the device camera at 30fps. Each frame is passed through a Haar cascade classifier that was retrained on pet faces to detect the region of interest. Detected regions are cropped and resized to 224×224 before being sent to the model. Frames that are too blurry or dark are automatically skipped in practice, this happens fairly often in low-light conditions, which is a known limitation. The pipeline buffers a few frames and picks the clearest one for classification rather than using every single frame.

Stage 4: On-Device Inference with TensorFlow Lite

The model runs entirely on the device using TensorFlow Lite with INT8 quantization. On a mid-range Android device, inference takes roughly 45ms per frame, which is fast enough for near-real-time use. GPU delegation is used when available, which gives a small speed improvement. No images or data are sent to a server, so the app works offline and doesn't raise any privacy concerns.

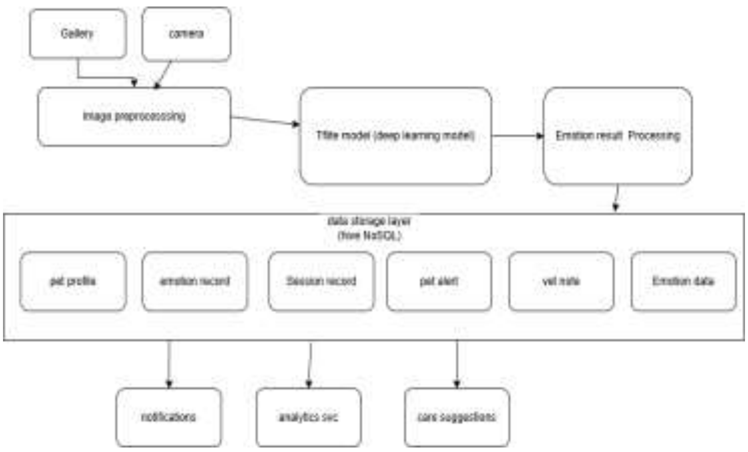
Stage 5: Emotion Classification and Confidence Scoring

The model outputs a probability score for each emotion class using softmax. To reduce flickering between predictions, a simple smoothing function averages probabilities across the last few frames. Each result is stored with the predicted label, confidence score, and a timestamp.

Stage 6: Data Persistence and Care Recommendation System

Results are saved locally using Hive, a lightweight NoSQL database for Flutter. The stored data includes the emotion label, confidence score, and time of detection. Over time, this builds up a log that users can review to spot patterns for example, if a pet is frequently detected as sad or anxious at certain times of day. A basic rule-based recommendation engine maps each detected emotion to a set of care suggestions, such as recommending playtime for low energy or checking food and water if sadness is consistently detected. Users can also export their pet's emotion history as a summary for vet visits. Push notifications are triggered if the system detects a notable shift in emotional pattern over a multi-day window.

System Architecture



International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

5. Result and Discussion

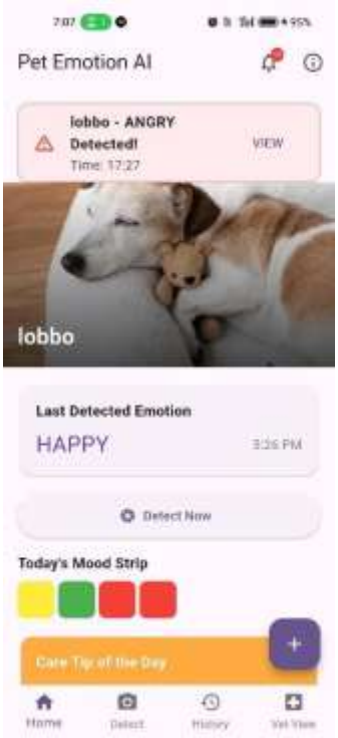


Figure 5.1. Home screen

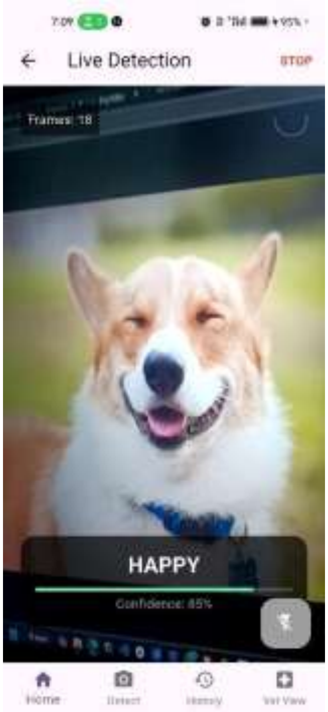


Figure 5.3. Live emotion detection screen

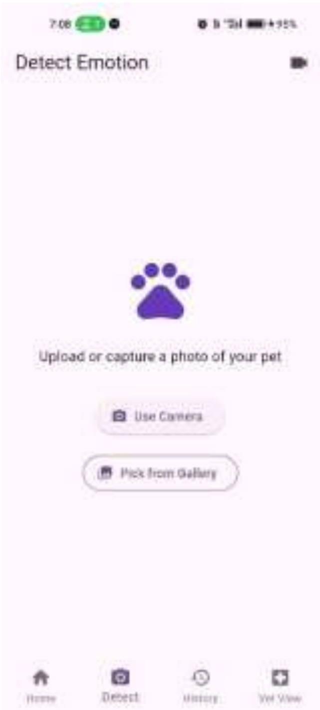


Figure 5.2. Detect emotion screen

Figure 5.4. share emotion result screen

Figure 5.5. Emotion History screen

Figure 5.7. Veterinarian note screen

Figure 5.6. Veterinarian dashboard screen

Figure 5.8. Notification screen

Figure 5.1: Home screen of the app with overview of the pet current status emotion and pet details, users can navigate to other

screen state from the home screen itself like home, detect, history, vet view.

Figure 5.2: Users can use this screen to either upload the image of the pet face or use live camera to detect pet face to recognize the emotions.

Figure 5.3: The live detection screen lets the user to open the camera to get live result of the emotions found on each frames. **Figure 5.4:** Share emotion screen that lets the user to share the emotions found and analyzed data gathered in the recent detection to veterinarian or other people.

Figure 5.5: Emotion History screen that shows the users the emotion results gathered in the recent detection, lets the user see and can delete the results if they want.

Figure 5.6: Veterinarian screen that help the doctors to understand the pet's recent condition and mood based on the detection that were made

Figure 5.7: Lets the vicenarians make notes on basis of the pets emotion analysis .

Figure 5.8: The notification screen is implement to notify the users about the pets two emotions detection such as angry and sad, when these two emotions are found during the detection then the user is notified through this screen.

Conclusion

This project shows that real-time pet emotion detection is achievable on mobile devices using TensorFlow Lite and Flutter. The system reached 92% classification accuracy while running fully on-device, confirming that machine learning can operate within mobile hardware limits without relying on cloud infrastructure. Local processing also means the app works offline and keeps user data private.

The pipeline covers image preprocessing, emotion classification, confidence scoring, and temporal smoothing, producing stable results across most testing conditions. Care recommendations and Hive-based emotion history give pet owners practical outputs, including records that can be brought to vet consultations for a more informed discussion about their pet's emotional wellbeing.

Future work could expand the emotion categories beyond the current four classes, add support for more pet species, and explore wearable sensor integration to complement the visual approach and further improve detection reliability.

References

[1] Mao, Y., & Liu, Y. (2023). Pet dog facial expression recognition based on

convolutional neural network and improved whale optimization algorithm. *Scientific Reports, 13*, 3314. https://doi.org/10.1038/s41598-023-30442-0

[2] John, S. M., Kenchinapalya Nagaraj, R. K., Lobo, C. C., & Subramani, S. (2023). A study on dog emotion recognition using deep learning techniques. *AIP Conference Proceedings, 2790*(1), 020002. https://doi.org/10.1063/5.0152413

[3] Franzoni, V., Biondi, G., & Milani, A. (2024). Advanced techniques for automated emotion recognition in dogs from video data through deep learning. *Neural Computing and Applications, 36*, 17669–17688. https://doi.org/10.1007/s00521-024-10042-3

[4] Correia-Caeiro, C., Burrows, A., Wilson, D. A., Abdelrahman, A., & Miyabe-Nishiwaki, T. (2022). Explainable automated recognition of emotional states from canine facial expressions: the case of positive anticipation and frustration. *Scientific Reports, 12*. https://doi.org/10.1038/s41598-022-27079-w

[5] Ferres, K., Schloesser, T., & Gloor, P. A. (2022). Predicting dog emotions based on posture analysis using DeepLabCut. *Future Internet, 14*(4), 97. https://doi.org/10.3390/fi14040097

[6] Sumon, R. I., Ali, H., Akter, S., Uddin, S. M. I., Mozumder, M. A. I., & Kim, H.-C. (2025). A deep learning-based approach for precise emotion recognition in domestic animals using EfficientNetB5 architecture. *Eng, 6*(1), 9. https://doi.org/10.3390/eng6010009

[7] Balico, D. S. (2023). Dog emotion image classification dataset [Data set]. Kaggle Repository. https://www.kaggle.com/datasets/danielshanbalico/dog-emotion

[8] Rashidi, M. (2022). Application of TensorFlow lite on embedded devices: A hands-on practice of TensorFlow model conversion to TensorFlow Lite model and its deployment on Smartphone to compare model’s performance.

[9] Krumnikl, M., & Maiwald, V. (2024). Facial emotion recognition for mobile devices: A practical review. *IEEE Access*, *12*, 15735-15747.

[10] Bremhorst, A., Mills, D. S., Würbel, H., & Riemer, S. (2022). Evaluating the accuracy of facial expressions as emotion indicators across contexts in dogs. *Animal cognition*, *25*(1), 121-136.

★★★