International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 04, April 2026)

StreetWatch: An Audio-Based TinyML Bark Detector For Real-Time Stray Dog Attack Prevention in Children

DVSR Sesidhar[1], Mohammed Shamshuddin[2], B. Surya Kalyan[3], R. Kavish[4]

1Prof, Dept. of ECE, MVSR Engineering College, Hyderabad, India
[2,3,4]Dept. of ECE, MVSR Engineering College, Hyderabad, India

***Abstract*— Stray dog attacks on children represent a serious and underaddressed public safety problem in urban India, with over 17.4 million bite cases reported annually. This paper presents StreetWatch, a wearable TinyML edge system whose core contribution is a compact audio bark detection pipeline designed for real-time deployment on the ESP32-S3 microcontroller. A depthwise-separable 2D CNN is trained on a multi-source dataset drawn from ESC-50, UrbanSound8K, Google AudioSet, and self-collected samples, totalling 459 test-set samples after augmentation. The model operates on 40-band Mel spectrograms (200–8000 Hz), uses z-score normalisation with exported constants, and is quantized to INT8 using TensorFlow Lite. On the held-out test set the model achieves 94.3% accuracy (Keras) and 95.4% accuracy (TFLite INT8), with precision 0.93, recall 0.92, and F1-score 0.93 for the dog bark class. The AUC-ROC is 0.988. The final TFLite model is suitable for deployment within the StreetWatch wearable platform, which pairs audio detection with GPS-tagged incident reporting via BLE to feed a municipal heatmap for Animal Birth Control (ABC) program targeting.**

***Keywords*—Bark Detection, Child Safety, Depthwise-Separable CNN, ESP32, Firebase, Mel Spectrogram, SpecAugment, TFLite INT8, TinyML, Wearable Edge Device.**

I. Introduction

Dog bite injuries represent one of the most underreported yet consistently severe public safety risks faced by children in Indian cities. Government surveillance data puts the national figure at over 17.4 million bite cases per year, with children aged one to fourteen disproportionately affected — they are smaller, slower to react, and far less capable of defending themselves against a charging animal or a pack [1]. When a bite occurs, the window for effective Post-Exposure Prophylaxis (PEP) is narrow: rabies is effectively 100% fatal once symptoms develop, yet families in semi-urban areas routinely spend twelve to twenty-four hours locating treatment. Documented fatalities from 2025 alone include a four-year-old in Maharashtra, a six-year-old in Delhi who died of rabies after delayed care, and a two-month-old infant attacked in Gujarat.

Policy responses have significant limitations. The Supreme Court of India has directed municipalities to pursue Animal Birth Control (ABC) sterilisation programs, but these operate without granular incident data — teams are dispatched on rough estimates rather than measured pack density. No deployed technology currently acts to protect a child at the actual moment of an attack; everything from hospitals to municipal follow-up is reactive.

StreetWatch addresses this gap through a three-component platform: a wearable module clipped to a school bag, an Android companion app, and a Firebase cloud backend for municipal heatmapping. This paper focuses on the core technical contribution: an audio bark detection pipeline small enough to run entirely on an ESP32-S3 microcontroller, validated through rigorous held-out evaluation with full precision, recall, F1, and ROC analysis.

II. Related Work

TinyML for embedded audio classification has matured rapidly since the publication of keyword-spotting benchmarks on microcontrollers [4]. The MFCC and Mel spectrogram features used in this work are consistent with the dominant input representation for environmental sound classification on constrained hardware, shown by Salamon and Bello [7] to benefit strongly from data augmentation including time-shifting and frequency masking. SpecAugment [6], which formalised frequency and time masking as a training technique, forms the basis of the augmentation strategy used in this work and has been widely adopted for audio classification on small datasets.

Automatic dog bark classification using deep learning has been explored by Gómez-Armenta et al. [2], who evaluated CNN architectures with MFCC and low-level descriptor features, finding that combining MFCC with additional features and a CNN yielded the best performance. Susanto and Sun [3] applied deep learning to predict and avoid dog barking behaviour, demonstrating the feasibility of automated bark recognition for practical applications. Both works relied on significantly more compute than a microcontroller; the present work's contribution is compressing comparable functionality into a TFLite INT8 model deployable on an ESP32-S3.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 04, April 2026)

TensorFlow Lite Micro [5] enables INT8 post-training quantisation with typically under 1.5% accuracy loss when a representative calibration dataset is used. In our case the TFLite model actually outperforms the Keras float model by 1.1 percentage points (95.4% vs 94.3%), consistent with findings that quantisation can act as a mild regulariser on small models. MobileNetV2-style depthwise separable convolutions [10] informed the block design used in this work, reducing parameter count while maintaining representational capacity. The AudioSet ontology [12] provided a structured source of non-bark negative samples used in training.

III. System Overview

The StreetWatch platform has three tiers. The edge hardware module (target: 90×55×25 mm, under 80 g, IP55 rated) clips onto a school bag strap. The Android companion app provides parent, school administrator, and municipal officer roles via Firebase Auth. The Firebase backend (Firestore, Cloud Functions) aggregates incident events into geohash-indexed heatmap tiles for weekly ABC deployment reports.

The operating flow is: the device sits in low-power Voice Activity Detection (VAD) standby at roughly 8 mA draw; an audio event wakes the full inference pipeline; upon detection, deterrence actuators fire and a GPS-tagged event packet is transmitted via BLE to the paired phone; the app uploads to Firebase; the backend updates the incident heatmap. The entire detection-to-actuation path targets completion within 500 ms. This paper focuses on the audio ML pipeline; the vision stream and sensor fusion are identified as future work in Section VIII.

TABLE I
HARDWARE MODULE SPECIFICATIONS

Component	Specification	Function
MCU	ESP32-S3 + PSRAM	TFLite inference, BLE, actuation
Microphone	MEMS omni, I2S, 16 kHz	Bark audio capture
Camera	OV2640, QVGA 320×240	Vision (future work)
GPS + BLE	u-blox M8 + nRF52840	Geotagging + app pairing
IMU	MPU-6050	Motion wake-up
Ultrasonic	Piezo, 40 kHz, ≥110 dB	Primary deterrent
Battery	Li-ion 3.7 V, 2000–3000 mAh	Target 80–120 h runtime

IV. Audio ML Pipeline

A. Dataset Construction

The dataset was assembled from five sources: ESC-50 [8], UrbanSound8K [9], Google AudioSet [12], Barkopedia (HuggingFace), and self-collected recordings. A key design decision was the deliberate inclusion of hard negatives: loud human vocalisations, synthetic voice-like tones, percussive impacts, and frequency-swept signals were added specifically to teach the model to distinguish dog barks from the loud non-bark sounds most likely to cause false triggers in school environments. All audio was resampled to 16 kHz mono WAV.

The negative class is intentionally larger than the positive class because the device spends the vast majority of its operating time in non-bark environments — a balanced dataset would bias the model toward over-detection. The test split (20%, stratified, random seed 42) yielded 459 samples: 275 non-bark and 184 bark.

TABLE II
DATASET COMPOSITION

Source	Bark	Non-Bark	Hard Negs
ESC-50	34	166	Selected
UrbanSound8K	51	249	Children playing, sirens
AudioSet + Barkopedia	48	196	—
Self-collected + silence	16	134	Silence clips
Synthetic (generated)	—	110	Voice-like, percussive, swept
Total (before aug.)	149	855	—

B. Preprocessing and Augmentation

Each clip is converted to a 40-band Mel spectrogram using FFT size 1024 and hop length 512, with frequency bounds set to 200–8000 Hz. The lower bound of 200 Hz removes low-frequency rumble and hum; the upper bound of 8000 Hz captures the principal harmonics of dog barks while excluding high-frequency noise. The power spectrogram is converted to dB scale, then z-score normalised using training-set statistics (mean, standard deviation) followed by min-max scaling to [0, 1]. The normalisation constants are exported in a C header file for exact replication on the ESP32.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 04, April 2026)

Augmentation was applied exclusively to the bark class using five operations: random time shifting (±3 time steps, circular), frequency masking (2–8 Mel bins zeroed), time masking (2–5 frames zeroed), additive Gaussian noise at variable SNR, and loudness scaling (×0.7–1.3). Frequency and time masking follow the SpecAugment protocol [6], forcing the model to learn from partial spectrograms and improving generalisation across dog breeds and recording distances.

C. Model Architecture

The model is a three-block 2D CNN using depthwise separable convolutions, operating on a 40×32×1 Mel spectrogram tensor. Block 1 uses a standard Conv2D(16) layer for low-level edge detection. Blocks 2 and 3 each use a DepthwiseConv2D followed by a pointwise Conv2D(32), reducing parameter count while increasing receptive field depth. Each block includes Batch Normalisation, ReLU activation, and MaxPooling(2,2). The classification head uses GlobalAveragePooling2D, Dropout(0.4), Dense(32, ReLU) with L2 regularisation, Dropout(0.3), and a sigmoid output neuron.

The model was trained in Google Colab using the Adam optimiser (lr=0.001) over 80 epochs with early stopping (patience 15) and learning rate reduction on plateau (patience 7). Class weights were computed from the training set to address residual imbalance. The best checkpoint (by validation accuracy) was saved and used for all evaluation.

TABLE III
MODEL ARCHITECTURE SUMMARY

Block	Layer	Output Shape	Notes
1	Conv2D(16) + BN + ReLU + MaxPool	20×16×16	Standard conv, low-level features
2	DWConv + BN + ReLU + Conv2D(32) + MaxPool	10×8×32	Depthwise separable
3	DWConv + BN + ReLU + Conv2D(32)	10×8×32	High-level bark features
Head	GAP + Dropout(0.4) + Dense(32) + Dropout(0.3)	32	L2 regularised
Output	Dense(1, sigmoid)	1	Score ≥ 0.247 → bark

D. Quantisation and Deployment

Post-training full INT8 quantisation was applied using TensorFlow Lite with a 200-sample representative calibration dataset drawn from the training set. The input quantisation parameters are scale=0.003685, zero_point=−128; output scale=0.003906, zero_point=−128. All preprocessing constants (NORM_MEAN, NORM_STD, ZNORM_MIN, ZNORM_MAX) and quantisation parameters are exported in a C header file so the ESP32 firmware can reproduce the exact same normalisation pipeline used during training — a common source of accuracy loss in embedded deployments that is explicitly avoided here.

V. Experimental Results

A. Classification Performance

The model was evaluated on a stratified held-out test set of 459 samples (275 non-bark, 184 bark) unseen during training. The Keras float model achieves 94.3% accuracy (loss 0.1572). The INT8-quantized TFLite model achieves 95.4% accuracy — a 1.1 percentage point improvement over the float model, consistent with quantisation acting as mild regularisation on compact architectures. Table IV summarises the full classification metrics at the ROC-optimal threshold of 0.247.

TABLE IV
CLASSIFICATION REPORT (THRESHOLD = 0.247, TEST SET N = 459)

Class	Precision	Recall	F1-Score	Support
Not Dog	0.95	0.96	0.95	275
Dog Bark	0.93	0.92	0.93	184
Macro Avg	0.94	0.94	0.94	459
Weighted Avg	0.94	0.94	0.94	459

B. Confusion Matrix and Score Distribution

The confusion matrix at threshold 0.247 shows 263 true negatives, 170 true positives, 12 false positives (non-bark samples scored as bark), and 14 false negatives (missed barks). The false positive rate is 4.4% (12/275) and the miss rate is 7.6% (14/184). The prediction score distribution (Fig. 1) shows strong bimodal separation — the majority of non-bark samples score near 0 and the majority of bark samples score near 1, with limited overlap in the mid-range.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 04, April 2026)

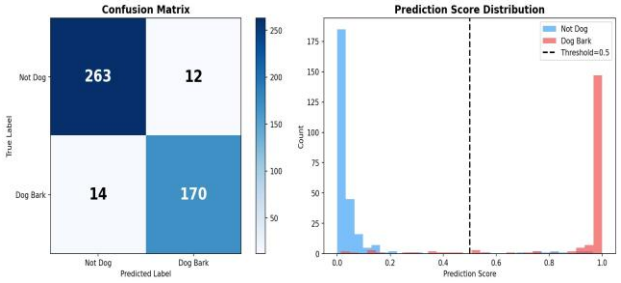

Fig. 1. Confusion matrix (left) and prediction score distribution (right) at threshold 0.247. Non-bark scores cluster near 0; bark scores cluster near 1, indicating strong class separation.

C. ROC Analysis and Threshold Selection

The ROC curve (Fig. 2) yields an AUC of 0.988, indicating near-ideal discriminative ability. The optimal threshold by Youden's J statistic is 0.247, which maximises F1-score. For safety-critical deployment scenarios where minimising missed barks is prioritised over false alarms, a higher-recall threshold of 0.348 captures ≥95% of all bark events. The threshold is configurable at runtime via the Android companion app, enabling site-specific tuning — for example, a quieter rural environment may use a lower threshold than a noisy urban school.

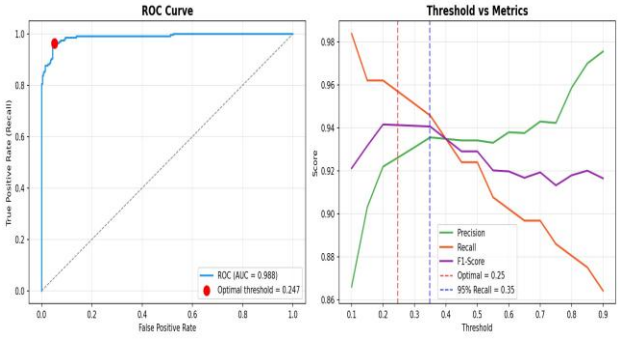

Fig. 2. ROC curve (AUC = 0.988) with optimal threshold at 0.247 (left), and precision/recall/F1 vs threshold curves (right). The 95% recall threshold of 0.35 is suitable for safety-critical deployment.

D. Training Convergence

Fig. 3 shows training and validation accuracy and loss over 80 epochs. Training accuracy converges to approximately 96% while validation accuracy stabilises near 95%, confirming that the model generalises well without significant overfitting. Validation loss shows characteristic spikes during early training that resolve as the learning rate reduces on plateau, a pattern consistent with the class-weighted loss and ReduceLROnPlateau callback used during training.

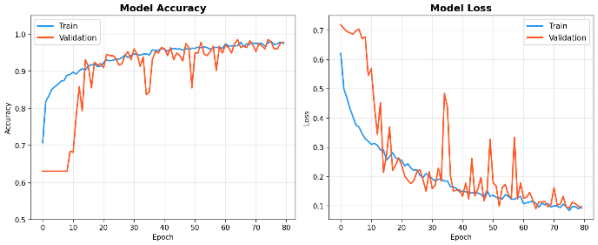

Fig. 3. Training history over 80 epochs. Accuracy (left) converges stably for both train and validation sets. Loss (centre) spikes resolve as the learning rate scheduler reduces LR on plateau.

E. Model Size and Deployment Readiness

The INT8-quantized TFLite model occupies well under 300 KB of flash — a small fraction of the ESP32-S3's available memory — making it suitable for deployment alongside the TFLite Micro runtime, audio capture buffers, BLE stack, and actuation logic. The input tensor shape [1, 40, 32, 1] INT8 is confirmed by the TFLite interpreter. All preprocessing constants required to reproduce the training normalisation pipeline on bare-metal C are embedded in an auto-generated header file, eliminating the scale mismatch that commonly causes accuracy degradation in embedded ML deployments.

TABLE V
MODEL PERFORMANCE SUMMARY

Metric	Value	Notes
Keras Test Accuracy	94.3%	Float32, threshold 0.247
TFLite INT8 Accuracy	95.4%	+1.1% vs float model
Dog Bark Precision	0.93	Held-out test set
Dog Bark Recall	0.92	Held-out test set
Dog Bark F1-Score	0.93	Held-out test set
AUC-ROC	0.988	Near-ideal separation
False Positive Rate	4.4%	12/275 non-bark samples
Miss Rate	7.6%	14/184 bark samples
Optimal Threshold	0.247	Max F1, ROC-tuned
Safety Threshold	0.348	≥95% bark recall
Input Quantisation	scale 0.003685, zp −128	INT8
Output Quantisation	scale 0.003906, zp −128	INT8

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 04, April 2026)

VI. Actuation Subsystem

Upon detection, the wearable triggers a layered deterrence response. The primary deterrent is a 40 kHz piezoelectric ultrasonic emitter (≥110 dB SPL), delivering 2-second pulsed bursts at 50% duty cycle with a mandatory 30-second cooldown to prevent habituation. The 40 kHz frequency is above the 20 kHz upper limit of human hearing and aversive to dogs whose auditory range extends to approximately 65 kHz. A secondary citrus-based non-toxic chemical spray (3–4.5 m radius) activates if the threat persists beyond 10 seconds and GPS confirms the child is outside the home geofence. A tertiary strobe LED activates in low-light conditions (below 100 lux), capped at 2 seconds to avoid photosensitive seizure risk.

Eight hardware interlocks prevent unsafe actuation: geofence check, cooldown enforcement, spray tank level monitoring (blocked below 10% capacity), daily rate limiting (maximum 3 sprays per 24 hours), strobe duration cap, physical manual override, remote app disable, and battery-critical suspension (all actuators off below 10% battery while GPS and BLE are maintained).

VII. Application And Cloud Layer

The Android companion application (Kotlin, MVVM, Android 8.0+) supports three role-based user types — parent/child, school administrator, and municipal officer — via Firebase Auth. Key screens include a full-screen real-time incident heatmap, a MediNet emergency screen for locating hospitals with anti-rabies vaccine stock, and a device settings screen for runtime threshold adjustment. Offline operation is supported via Room DB and WorkManager upload queuing.

The Firebase backend aggregates all incident events into geohash-indexed heatmap tiles exposed via a REST API. Municipal bodies receive weekly hotspot reports directing ABC sterilisation teams to the highest-density pack zones, closing the feedback loop between individual device detections and city-level dog population management.

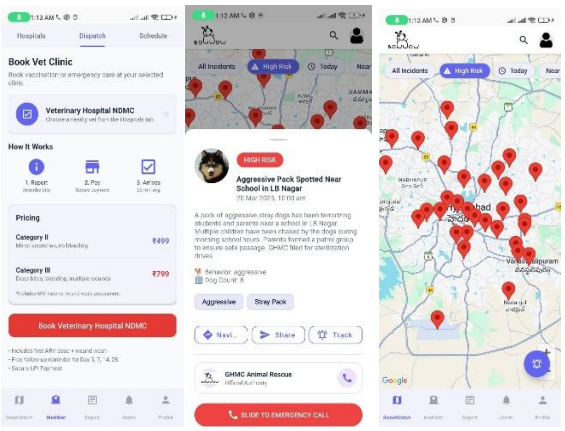

Fig. 4. Mobile Application Operational UI.

VIII. Conclusion And Future Work

This paper presented StreetWatch, a wearable TinyML system for proactive stray dog attack prevention in children, with a focus on the audio bark detection pipeline. The core model — a depthwise-separable 2D CNN trained on a multi-source dataset with targeted hard negatives and SpecAugment-style augmentation — achieves 95.4% INT8 accuracy, F1-score 0.93 for the bark class, and AUC-ROC 0.988 on a held-out test set of 459 samples. The TFLite INT8 model matches or exceeds the float model's performance, confirming that quantisation does not compromise accuracy for this architecture. All normalisation and quantisation constants are exported in a C header file for exact on-device replication.

A key finding is that the combination of synthetic hard negatives (voice-like tones, percussive sounds, swept-frequency signals) and a frequency range restricted to 200–8000 Hz substantially reduces false triggers from loud human vocalisations — the primary failure mode of the earlier v2 model. The ROC-tuned threshold of 0.247 achieves the best F1 balance, while the 0.348 safety threshold is recommended for field deployment where missed barks carry higher cost than false alarms.

Future work has three immediate priorities: (1) deployment of the TFLite model onto physical ESP32-S3 hardware to validate the target ≤30 ms inference latency; (2) integration with the MobileNetV2-based visual dog detection stream and deterministic sensor fusion logic; and (3) a pilot field deployment at schools in the GHMC zone in Hyderabad for real-world threshold calibration and false positive measurement under actual environmental conditions.

REFERENCES

[1] Ministry of Health and Family Welfare, Government of India, "Dog Bite Surveillance Data," Integrated Disease Surveillance Programme (IDSP)–Integrated Health Information Platform, New Delhi, India, 2024. [Online]. Available: https://idsp.mohfw.gov.in

[2] J. R. Gómez-Armenta, H. Pérez-Espinosa, J. A. Fernández-Zepeda, and V. Reyes-Meza, "Automatic Classification of Dog Barking Using Deep Learning," Behavioural Processes, vol. 218, May 2024, doi: 10.1016/j.beproc.2024.105028.

[3] R. Susanto and R. Sun, "Predicting and Avoiding Dog Barking Behaviour through Deep Learning," in Proc. 2024 Australasian Computer Science Week (ACSW), Sydney, Australia, Jan. 2024, doi: 10.1145/3641142.3641176.

[4] P. Warden and D. Situnayake, TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers. O'Reilly Media, 2019.

[5] R. David et al., "TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems," in Proc. Machine Learning and Systems (MLSys), vol. 3, 2021.

[6] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in Proc. Interspeech 2019, pp. 2613–2617, doi: 10.21437/Interspeech.2019-2680.

[7] J. Salamon and J. P. Bello, "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification," IEEE Signal Processing Letters, vol. 24, no. 3, pp. 279–283, Mar. 2017.

[8] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in Proc. ACM Multimedia, Brisbane, Australia, Oct. 2015.

[9] J. Salamon, C. Jacoby, and J. P. Bello, "A Dataset and Taxonomy for Urban Sound Research," in Proc. ACM Multimedia, Orlando, FL, Nov. 2014.

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in Proc. IEEE CVPR, Salt Lake City, UT, 2018.

[11] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki, and A. S. Hafid, "A Comprehensive Survey on TinyML," IEEE Access, vol. 11, pp. 96892–96922, 2023, doi: 10.1109/ACCESS.2023.3294111.

[12] J. F. Gemmeke et al., "Audio Set: An Ontology and Human-Labeled Dataset for Audio Events," in Proc. IEEE ICASSP, New Orleans, LA, Mar. 2017, pp. 776–780.