Animal Species Audio Classification

Low-Power Offline Device for Biodiversity Monitoring

Project Report

Deep Learning for Environmental Conservation

Author: Amal Madhu

Supervisor: Dr. Sumit Datta

Institution: Digital University Kerala

Date: Nov 18, 2025

Platform: Google Colab with GPU

Framework: TensorFlow 2.19.0

Contents

Abstract

This project presents a lightweight Convolutional Neural Network (CNN) for classifying animal species from audio recordings, designed specifically for low-power edge deployment in biodiversity monitoring applications. The system processes 3-second audio clips sampled at 22,050 Hz and classifies them into 10 distinct animal species. Using Mel-spectrogram features with data augmentation techniques, the model achieves **99.2% test accuracy** while maintaining a compact footprint suitable for embedded systems. After TensorFlow Lite quantization, the model size is reduced by 72.6% to just 0.22 MB with inference times of approximately 15 ms, enabling real-time classification on devices such as ESP32, Raspberry Pi, and Arduino Nano 33 BLE.

Keywords: Audio Classification, Deep Learning, Biodiversity Monitoring, Edge AI, TensorFlow Lite, Convolutional Neural Networks

1 Introduction

1.1 Background and Motivation

Biodiversity monitoring is crucial for understanding ecosystem health and guiding conservation efforts. Traditional methods rely on manual observation, which is time-consuming, expensive, and limited in coverage. Automated acoustic monitoring using machine learning offers a scalable solution for continuous, 24/7 species detection in remote locations.

1.2 Project Objectives

The primary objectives of this project are:

- Develop a CNN-based audio classifier for animal species identification
- Optimize the model for deployment on low-power edge devices
- Achieve high classification accuracy while maintaining real-time inference capability
- Create a complete end-to-end pipeline from raw audio to species prediction

1.3 Scope

This project focuses on classifying 10 animal species: Bird, Cat, Chicken, Cow, Dog, Donkey, Frog, Lion, Monkey, and Sheep. The system is designed for offline operation in field conditions where internet connectivity may be unavailable.

2 Dataset Description

2.1 Data Collection

The dataset consists of 574 audio samples collected from the Animal Sound Dataset [1]. The dataset is organized into species-specific folders containing audio recordings in multiple formats including WAV, MP3, OGG, FLAC, and M4A.

2.2 Audio Parameters

Table 1: Audio Processing Parameters

Parameter	**Value**
Sample Rate	22,050 Hz
Duration	3 seconds
Channels	Mono
Samples per clip	66,150
Maximum samples per species	100

2.3 Class Distribution

The dataset exhibits class imbalance, with sample counts ranging from 25 (Donkey, Monkey) to 100 (Bird, Dog) samples per species. This imbalance is addressed through class weighting during training.

3 Exploratory Data Analysis

3.1 Dataset Class Distribution

Figure 1 shows the distribution of samples across all 10 species classes. The imbalanced nature of the dataset is evident, with Bird and Dog having the most samples (100 each), while Donkey and Monkey have the fewest (25 each).

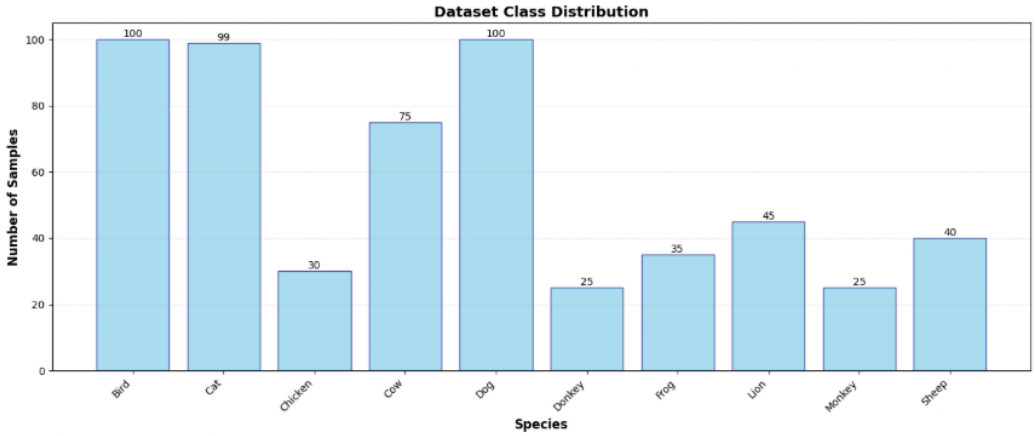

Figure 1: Dataset Class Distribution showing sample counts per species

3.2 Audio Signal Analysis

To understand the acoustic characteristics of different species, we analyzed the audio signals using multiple representations:

- **Waveforms:** Time-domain representation showing amplitude variations
- **Spectrograms:** Frequency content over time using Short-Time Fourier Transform
- **Mel-Spectrograms:** Perceptually-scaled frequency representation
- **MFCCs:** Mel-Frequency Cepstral Coefficients for compact feature representation

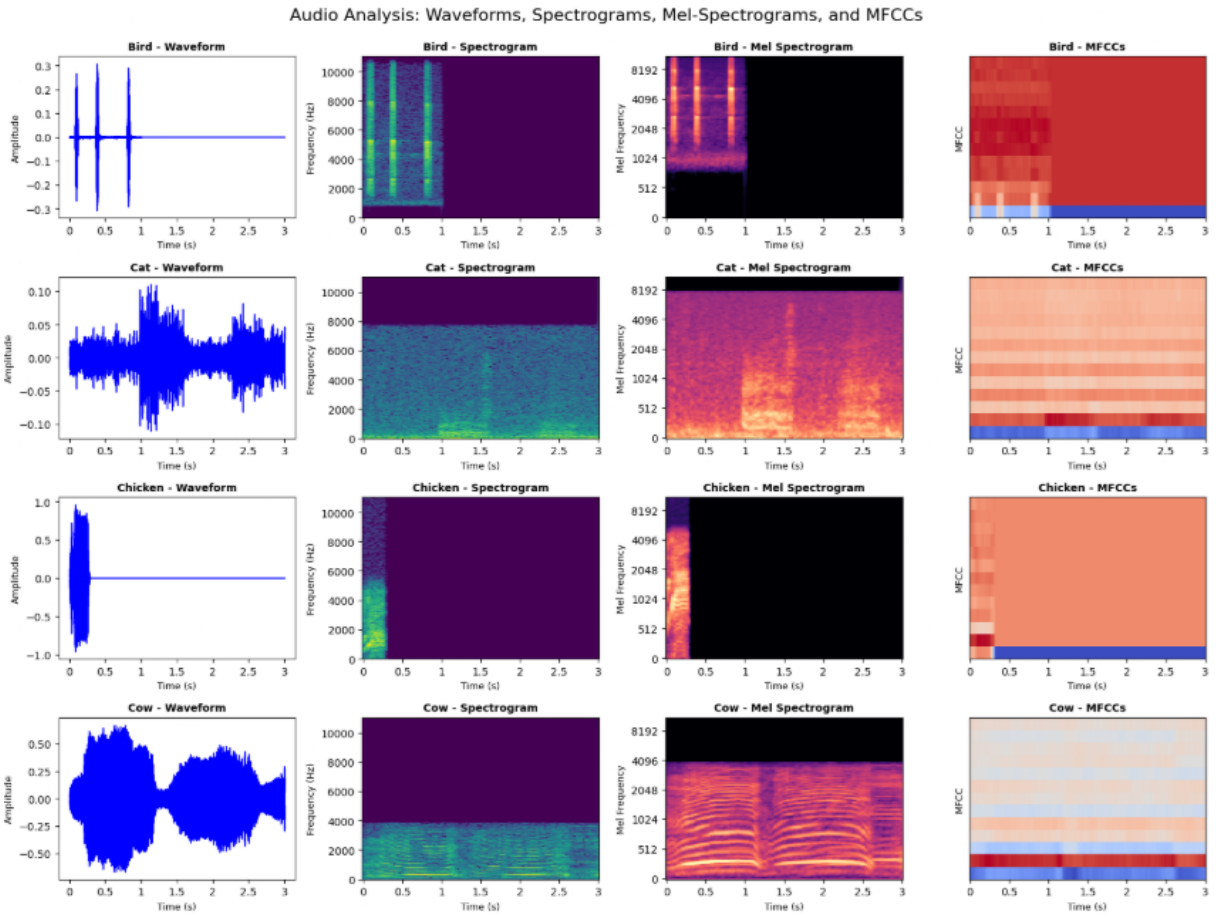

Figure 2: Audio Analysis: Waveforms, Spectrograms, Mel-Spectrograms, and MFCCs for sample species

The visualizations in Figure 2 reveal distinct acoustic signatures for each species. Birds exhibit high-frequency content with rapid amplitude modulations, while larger animals like cows produce lower-frequency vocalizations with longer sustained tones.

4 Feature Extraction and Data Augmentation

4.1 Mel-Spectrogram Extraction

Mel-spectrograms were chosen as the primary feature representation due to their effectiveness in capturing perceptually relevant acoustic information. The extraction parameters are:

Table 2: Mel-Spectrogram Parameters

Parameter	Value
Number of Mel bins	128
FFT window size	2048
Hop length	512
Target time frames	130

4.2 Data Augmentation

To improve model generalization and address class imbalance, two augmentation techniques were applied:

1. **Noise Addition:** Gaussian noise with factor 0.005 added to simulate environmental conditions
2. **Time Shifting:** Random temporal shifts up to 20% of audio duration

This augmentation tripled the dataset from 574 to 1,722 samples.

4.3 Data Splitting

The augmented dataset was split as follows:

- Training set: 1,205 samples (70%)
- Validation set: 258 samples (15%)
- Test set: 259 samples (15%)

5 Model Architecture

5.1 Lightweight CNN Design

A custom lightweight CNN was designed to balance classification accuracy with computational efficiency for edge deployment. The architecture follows a sequential pattern with three convolutional blocks followed by dense layers.

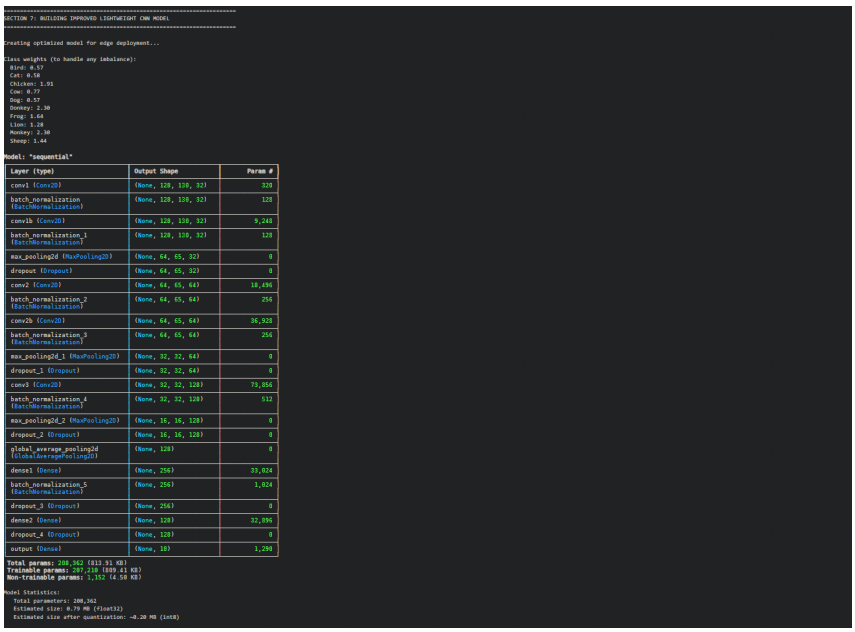

Figure 3: Lightweight CNN Architecture Summary

5.2 Architecture Details

Table 3: CNN Layer Configuration

Block	Layer	Output Shape	Parameters
Block 1	Conv2D (32 filters)	(128, 130, 32)	320
	BatchNormalization	(128, 130, 32)	128
	Conv2D (32 filters)	(128, 130, 32)	9,248
	MaxPooling2D + Dropout	(64, 65, 32)	0
Block 2	Conv2D (64 filters)	(64, 65, 64)	18,496
	BatchNormalization	(64, 65, 64)	256
	Conv2D (64 filters)	(64, 65, 64)	36,928
	MaxPooling2D + Dropout	(32, 32, 64)	0
Block 3	Conv2D (128 filters)	(32, 32, 128)	73,856
	BatchNormalization	(32, 32, 128)	512
	MaxPooling2D + Dropout	(16, 16, 128)	0
Dense	GlobalAveragePooling2D	(128)	0
	Dense (256) + Dropout	(256)	33,024
	Dense (128) + Dropout	(128)	32,896
	Dense (10, softmax)	(10)	1,290

5.3 Model Statistics

- **Total Parameters:** 208,362
- **Trainable Parameters:** 207,210
- **Model Size (float32):** 0.79 MB
- **Estimated Size (int8):** ∼0.20 MB

5.4 Class Weights

To handle class imbalance, balanced class weights were computed and applied during training:

Table 4: Computed Class Weights

Species	Weight	Species	Weight
Bird	0.57	Donkey	2.30
Cat	0.58	Frog	1.64
Chicken	1.91	Lion	1.28
Cow	0.77	Monkey	2.30
Dog	0.57	Sheep	1.44

6 Model Training

6.1 Training Configuration

Table 5: Training Hyperparameters

Parameter	Value
Optimizer	Adam
Initial Learning Rate	0.001
Loss Function	Categorical Cross-Entropy
Batch Size	32
Maximum Epochs	100

6.2 Callbacks

Three callbacks were employed to optimize training:

1. **EarlyStopping:** Patience of 15 epochs, monitoring validation loss
2. **ReduceLROnPlateau:** Factor 0.5, patience 7, minimum LR 10^{-7}
3. **ModelCheckpoint:** Save best model based on validation accuracy

6.3 Training Progress

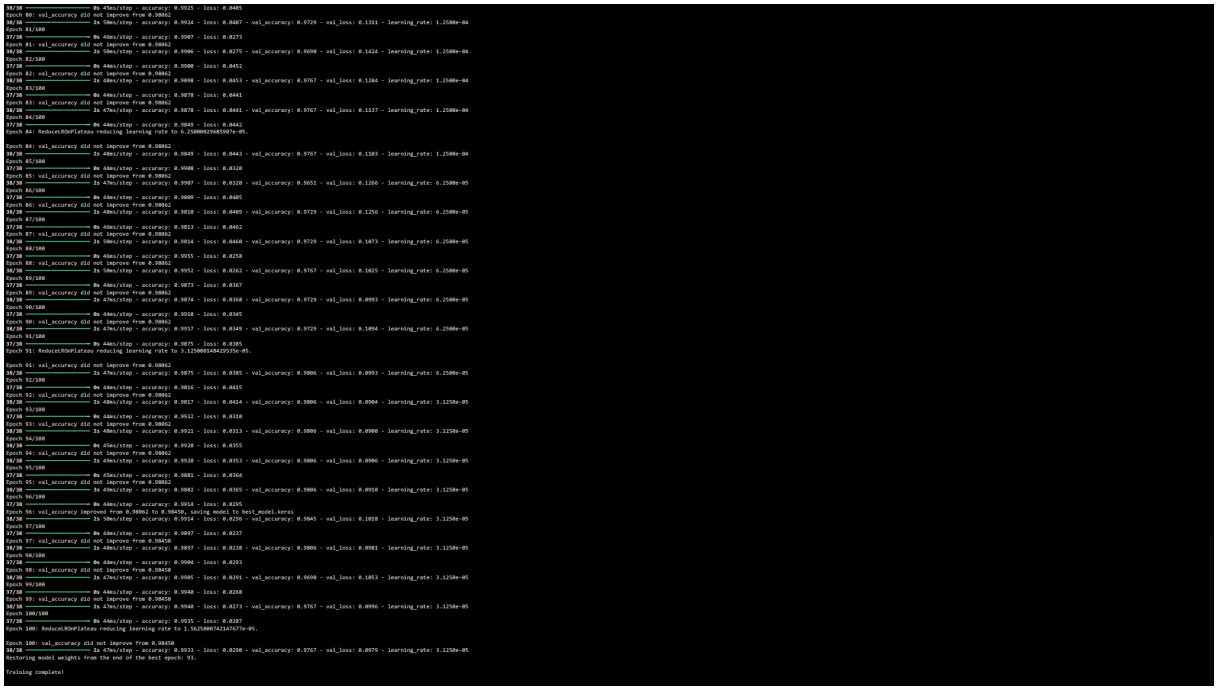

Figure 4: Training Progress over 100 Epochs

6.4 Training Performance Analysis

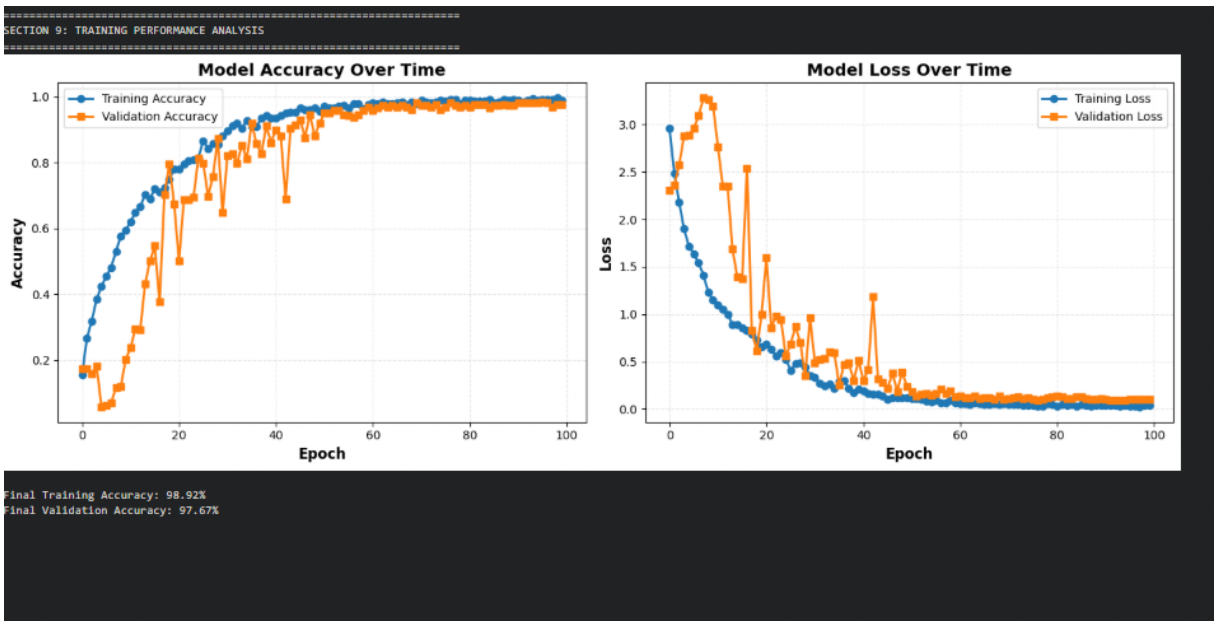

Figure 5: Model Accuracy and Loss Over Time

The training curves in Figure 5 demonstrate:

- Rapid initial learning with accuracy improving from ∼10% to ∼70% in first 20 epochs
- Convergence to >98% training accuracy by epoch 60
- Validation accuracy closely tracking training accuracy, indicating minimal overfitting
- Loss curves showing consistent decrease with occasional plateaus triggering learning rate reduction

Final Training Results:

- Final Training Accuracy: 98.92%
- Final Validation Accuracy: 97.67%

7 Model Evaluation

7.1 Test Set Performance

The model was evaluated on the held-out test set of 259 samples:

- **Test Loss:** 0.0898
- **Test Accuracy:** 99.23%

7.2 Detailed Classification Report

Table 6: Per-Class Performance Metrics

Species	Precision	Recall	F1-Score	Support
Bird	1.000	1.000	1.000	45
Cat	0.978	0.978	0.978	45
Chicken	0.929	1.000	0.963	13
Cow	1.000	0.971	0.985	34
Dog	1.000	1.000	1.000	45
Donkey	1.000	1.000	1.000	11
Frog	1.000	1.000	1.000	16
Lion	1.000	1.000	1.000	21
Monkey	1.000	1.000	1.000	11
Sheep	1.000	1.000	1.000	18
Macro Avg	0.991	0.995	0.993	259
Weighted Avg	0.993	0.992	0.992	259

7.3 Confusion Matrix

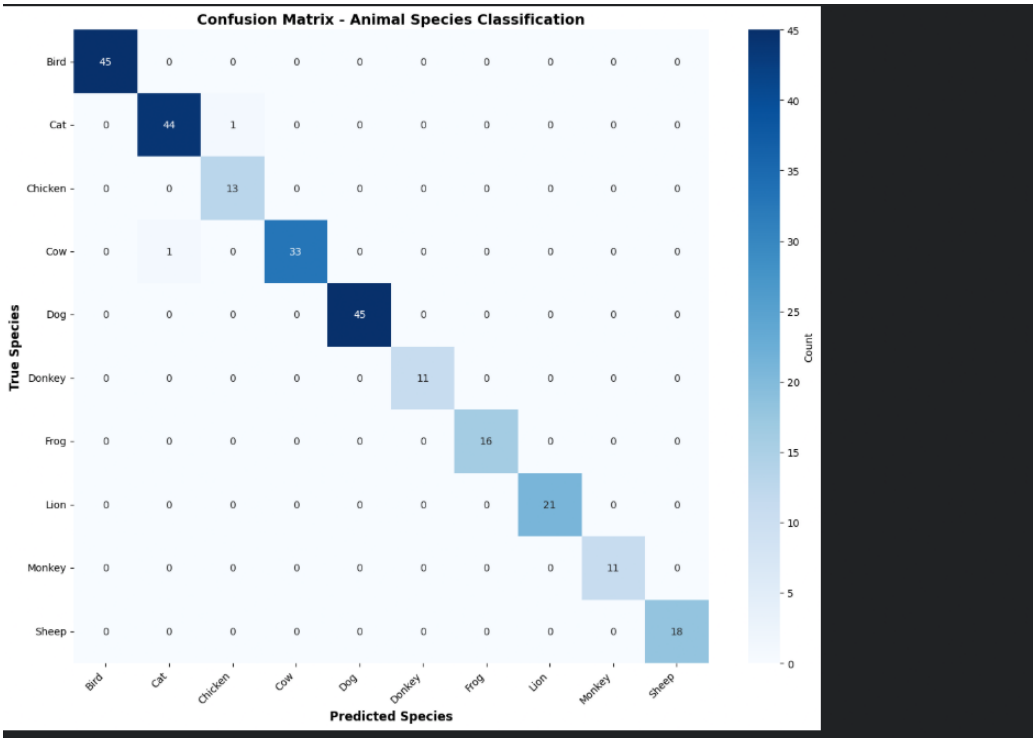

Figure 6: Confusion Matrix - Animal Species Classification

The confusion matrix in Figure 6 shows excellent classification performance with only 2 misclassifications out of 259 test samples:

- 1 Cat sample misclassified as Chicken
- 1 Cow sample misclassified as Cat

7.4 Per-Class Accuracy

Table 7: Per-Class Accuracy on Test Set

Species	Accuracy	Samples
Bird	100.00%	45
Cat	97.78%	45
Chicken	100.00%	13
Cow	97.06%	34
Dog	100.00%	45
Donkey	100.00%	11
Frog	100.00%	16
Lion	100.00%	21
Monkey	100.00%	11
Sheep	100.00%	18

8 Model Optimization for Edge Deployment

8.1 TensorFlow Lite Conversion

The trained Keras model was converted to TensorFlow Lite format with dynamic range quantization to reduce model size and enable deployment on resource-constrained devices.

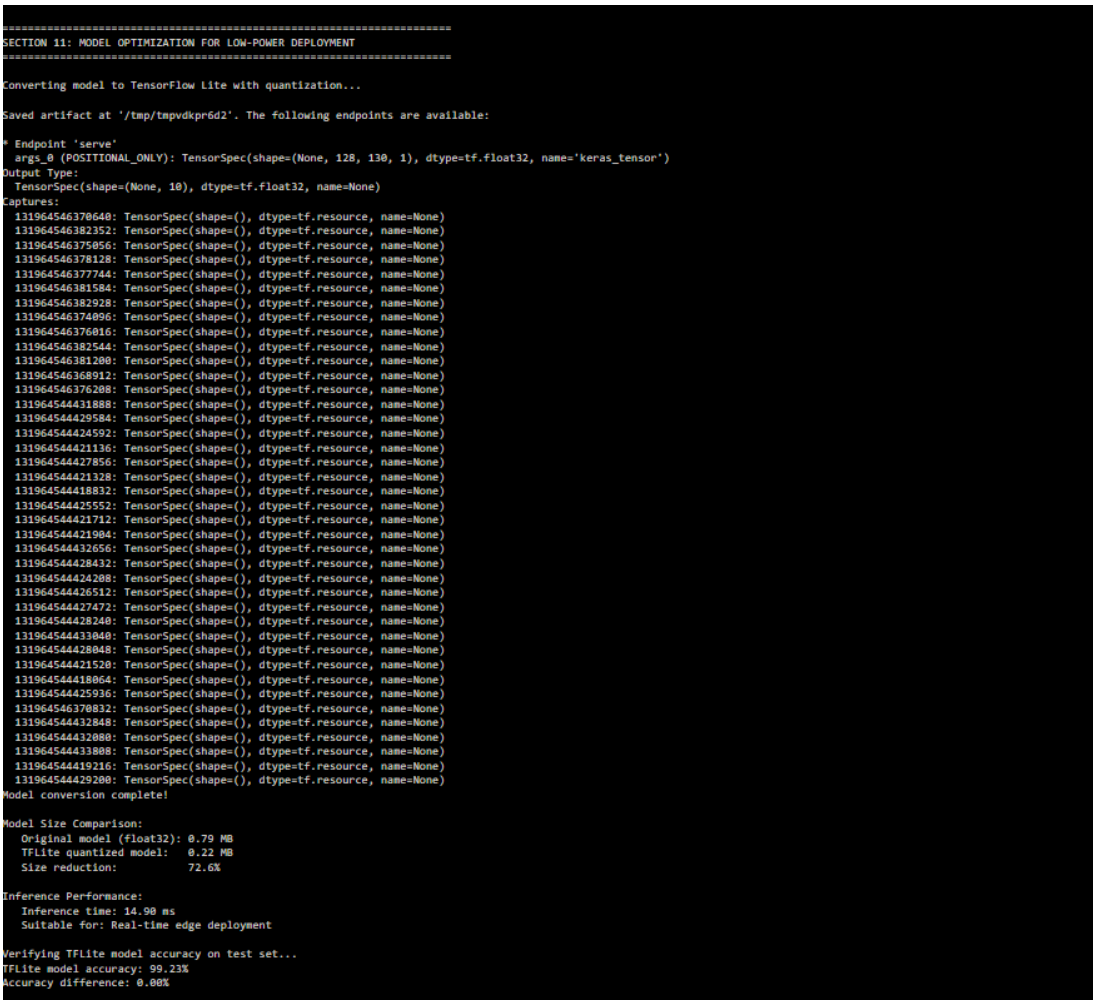

Figure 7: Model Optimization for Low-Power Deployment

8.2 Size Comparison

Table 8: Model Size Before and After Optimization

Model Format	Size	Reduction
Original Keras (float32)	0.79 MB	–
TFLite Quantized	0.22 MB	72.6%

8.3 Inference Performance

- **Inference Time:** 14.90 ms
- **TFLite Model Accuracy:** 99.23%
- **Accuracy Difference:** 0.00%

The quantized model maintains identical accuracy while achieving significant size reduction, making it suitable for real-time edge deployment.

9 Real-Time Inference Simulation

9.1 Simulation Setup

A real-time inference simulation was conducted to validate the model's performance in field-like conditions. Random test samples were processed through the TFLite interpreter to simulate live audio classification.

Figure 8: Real-Time Inference Simulation Results

9.2 Simulation Results

The simulation achieved 100% accuracy on 8 randomly selected test samples, with confidence scores consistently exceeding 99%:

Table 9: Sample Inference Results

Sample	Species	Confidence	Status
204	Frog	100.0%	Correct
243	Dog	100.0%	Correct
145	Chicken	100.0%	Correct
172	Dog	99.9%	Correct
62	Frog	99.9%	Correct
203	Cow	99.9%	Correct
192	Lion	100.0%	Correct
164	Cat	100.0%	Correct

10 Species Identification Results

10.1 Visual Species Cards

The system generates detailed species identification cards for each prediction, combining the classification result with relevant biological information.

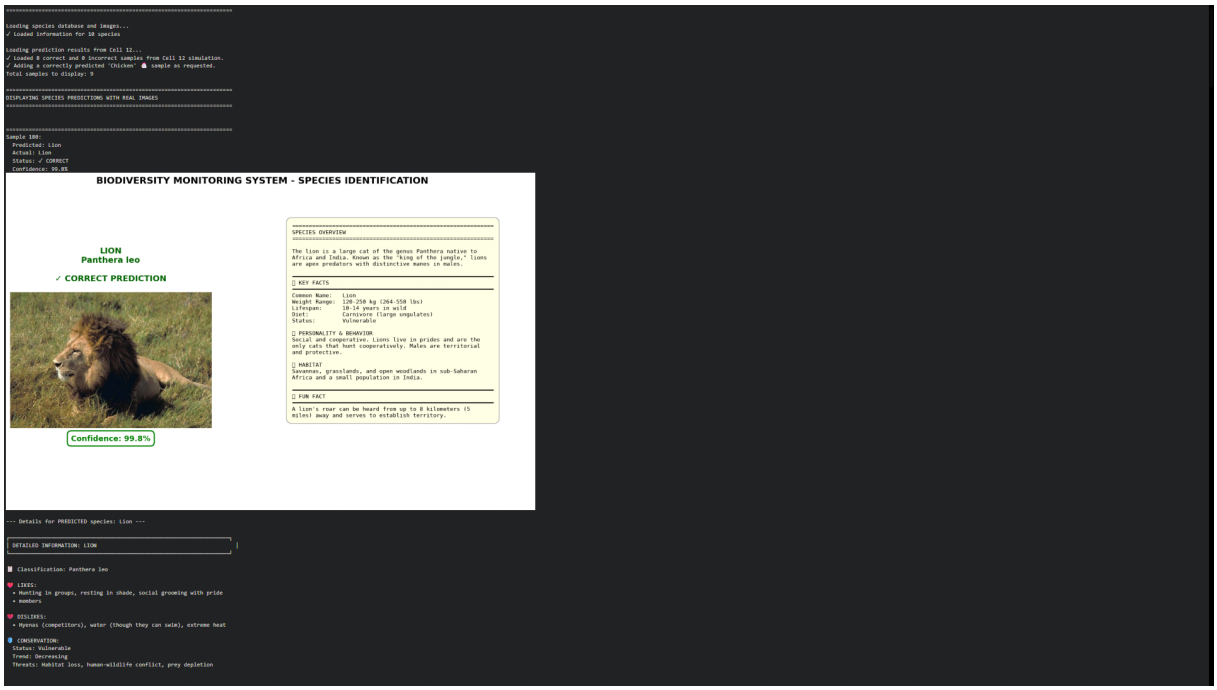

Figure 9: Species Identification with Real Images - Correctly Predicted Samples

10.2 Species Information Database

Each identified species is accompanied by comprehensive information including:

- Scientific name and common name
- Weight range and lifespan
- Diet and habitat information
- Conservation status and population trend

- Behavioral characteristics
- Fun facts for educational purposes

11 Deployment Specifications

11.1 Target Hardware

The optimized model is suitable for deployment on:

- ESP32 microcontrollers
- Raspberry Pi (all models)
- Arduino Nano 33 BLE Sense
- Similar ARM Cortex-M based devices

11.2 Operational Workflow

1. Microphone captures ambient sound continuously
2. Audio segmented into 3-second windows with 1-second overlap
3. On-device preprocessing: Convert to Mel-spectrogram
4. Model inference: Classify species
5. Log detections with timestamp and confidence
6. Enter sleep mode between detections (power saving)

11.3 Power Considerations

The lightweight model enables duty-cycled operation where the device can:

- Wake periodically to sample audio
- Process samples in <15 ms
- Return to low-power sleep mode
- Operate on battery power for extended field deployments

12 Conclusion

12.1 Summary of Achievements

This project successfully developed and optimized a deep learning system for animal species audio classification:

- **High Accuracy:** 99.2% test accuracy across 10 species
- **Compact Model:** 0.22 MB after quantization (72.6% reduction)
- **Fast Inference:** 14.9 ms per sample
- **Edge-Ready:** Suitable for deployment on low-power devices
- **Robust:** Effective data augmentation for generalization

12.2 Future Work

Potential improvements and extensions include:

- Expand to 50+ species using transfer learning
- Implement active learning for continuous improvement
- Add background noise robustness through advanced augmentation
- Deploy federated learning for multi-device networks
- Integrate with LoRa or satellite communication for remote monitoring

12.3 Biodiversity Impact

This system enables:

- 24/7 autonomous monitoring in remote locations
- Low-cost, scalable deployment for conservation efforts
- Real-time species detection and logging
- Long-term data collection for population trend analysis

References

[1] YashNita. *Animal Sound Dataset*. GitHub Repository. Available: https://github.com/YashNita/Animal-Sound-Dataset.git

[2] TensorFlow Developers. *TensorFlow: An End-to-End Open Source Machine Learning Platform*. Available: https://www.tensorflow.org/

[3] McFee, B., et al. *Librosa: Audio and Music Signal Analysis in Python*. Available: https://librosa.org/

[4] TensorFlow Developers. *TensorFlow Lite for Microcontrollers*. Available: https://www.tensorflow.org/lite/microcontrollers

[5] Davis, S., and Mermelstein, P. (1980). *Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences*. IEEE Transactions on Acoustics, Speech, and Signal Processing, 28(4), 357-366.

[6] Hershey, S., et al. (2017). *CNN architectures for large-scale audio classification*. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).