Real-Time Gunshot Detection on Edge Devices Using CNN14 Embeddings and BiLSTM–Attention

P. Praveen Kumar*
Department of Electronics and Communication Engineering
Rajiv Gandhi University of Knowledge Technologies, Nuzvid, India
Email: *N210402@rguktn.ac.in

***Abstract*—Gunshot detection is crucial for public safety and real-time surveillance, enabling rapid response to firearm incidents in urban and vulnerable environments. Traditional approaches based on handcrafted audio features and classical machine-learning models struggle to generalize in noisy real-world conditions and often incur high latency when deployed via cloud backends. To address these challenges, this work presents a lightweight deep-learning-based framework for on-device gunshot detection suitable for edge platforms such as Raspberry Pi. Raw audio is transformed into log–Mel spectrograms, and high-level acoustic representations are extracted using a pretrained CNN14 encoder from the PANNs framework. These embeddings are processed by a bidirectional LSTM network with an attention mechanism that emphasizes impulsive gunshot frames while suppressing background noise and silence. A balanced dataset of 17,746 clips (8,873 gunshot and 8,873 non-gunshot) is curated from multiple open-source repositories, with RMS-based filtering and sliding-window segmentation to enhance robustness. The proposed model achieves a validation accuracy of 97.57%, outperforming MFCC-based LSTM baselines, while maintaining an end-to-end inference latency of approximately 100–200 ms per 1-second audio segment on Raspberry Pi 4. These results demonstrate that the proposed method provides an effective and efficient solution for distributed, privacy-preserving gunshot detection on resource-constrained edge devices.**

***Index Terms*—Gunshot Detection, Edge Computing, CNN14, BiLSTM, Attention Mechanism, Audio Classification, Quantization.**

I. Introduction

Gunshot detection has become a critical component of intelligent public–safety infrastructures, enabling rapid response to firearm incidents in urban and vulnerable environments. Gunshots are impulsive and high–energy acoustic events, but their characteristics vary significantly with firearm type, propagation distance, occlusions, and surrounding noise. These variations make reliable classification challenging, especially in outdoor soundscapes. Conventional approaches based on handcrafted features and classical machine–learning classifiers often fail in the presence of confounding audio events such as fireworks, thunder, construction sounds, and vehicle noise, resulting in decreased robustness and limited real–world applicability.

Earlier research highlighted the potential of deep learning for audio event classification and achieved encouraging accuracy for gunshot detection. However, most prior systems suffered from high computational overhead, limited dataset diversity, and lack of noise–aware learning, which resulted in overfitting and restricted deployment on embedded platforms. Furthermore, many existing implementations were dependent on cloud servers for inference, leading to latency and privacy concerns and making them unsuitable for on–device real–time surveillance.

To address these limitations, this work introduces a compact and efficient deep learning framework designed specifically for real–time gunshot detection on edge devices. The proposed system transforms raw audio into log–Mel spectrograms and extracts high–level acoustic representations using pretrained CNN14 embeddings. These embeddings are then passed into a bidirectional LSTM network augmented with an attention mechanism to emphasize impulse–dominant temporal frames while suppressing background noise and silence. The overall architecture is optimized for low latency and reduced memory footprint while maintaining high classification accuracy.

The objectives of this research are twofold:

- To develop a noise–robust gunshot detection framework that leverages deep audio embeddings with temporal attention for improved precision and generalization.
- To ensure low–latency inference suitable for embedded platforms such as Raspberry Pi without sacrificing accuracy.

Through these contributions, this work advances the feasibility of scalable and privacy-preserving acoustic surveillance in environments where high network bandwidth and cloud resources are unavailable.

II. Results and Comparison

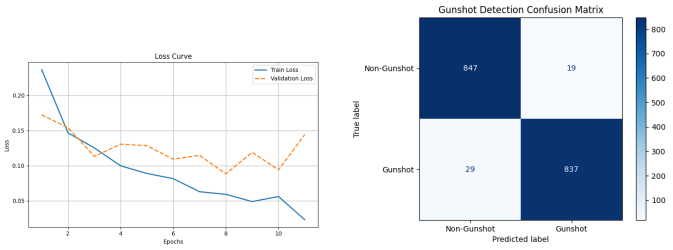
The proposed CNN14 + BiLSTM + Attention model was trained using a balanced dataset of 17,746 audio segments and evaluated using a 60/20/20 train–validation– test split. The model achieved strong generalization capability with low inference latency, demonstrating its suitability for edge deployment. The best validation accuracy obtained was **97.57%**, showing clear improvement over traditional MFCC-based models [1].

Table I summarizes the performance comparison between the baseline MFCC + LSTM approach and the proposed method.

To further illustrate model behavior, Fig. 1 shows the training/validation accuracy and loss curves together with the confusion matrix for the proposed model.

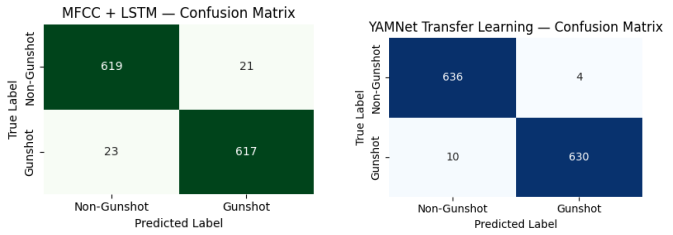
TABLE I: Baseline vs Proposed Model Performance Comparison

Metric	Baseline MFCC + LSTM	Proposed Model
Input Representation	MFCC Features	Log–Mel Spectrogram Embeddings
Temporal Modeling	Single LSTM	BiLSTM + Attention
Validation Accuracy	96.04%	**97.57%**
Noise Robustness	Moderate	High
Inference Latency	High	**Low (Edge-Ready)**
Deployment	Raspberry Pi Support	Raspberry Pi Optimized



(a) Accuracy and loss curves (b) Confusion matrix

Fig. 1: Training behavior and classification performance for the proposed CNN14–BiLSTM–Attention model.



(a) MFCC_Matrix (b) YAMNet_Matrix

Fig. 2: Comparison of handcrafted MFCC features and deep YAMNet embeddings used in earlier systems.

Fig. 2 compares MFCC-based and YAMNet-based feature representations used in prior work, highlighting the advantage of deep embeddings for capturing high-level semantics.

The confusion matrix-based evaluation verifies superiority in classification precision, with reduced false positive and false negative predictions for gunshot events. The embedding-only design also minimizes computational operations, yielding real-time performance of **100–200 ms per audio segment** on Raspberry Pi 4 [1]. Overall, the proposed model outperforms classical MFCC pipelines in terms of accuracy, environmental robustness, and embedded device feasibility, making it a strong candidate for practical, real-time gunshot detection systems.

III. Proposed Methodology

The proposed system is designed to perform accurate and noise-robust gunshot detection while remaining lightweight enough for deployment on edge devices such as Raspberry Pi. The complete pipeline consists of four major stages: (1) dataset construction and labeling, (2) audio preprocessing and segmentation, (3) log–Mel feature and embedding extraction using CNN14, and (4) hybrid temporal classification using a BiLSTM–Attention network with a compact dense head.

A. Dataset Construction and Labeling

To obtain a diverse and realistic representation of firearm and background sounds, audio clips were collected from multiple open-source repositories. The final dataset is organized as a binary classification problem with two classes: *Gunshot* and *Non-Gunshot*.

1) Gunshot Class: The gunshot class aggregates clips from several curated datasets containing single as well as burst fire events recorded under different environments and microphone setups. Typical sources include:

- Gunshot Audio Dataset (Kaggle),
- Mendeley Gunshot Dataset,
- Gunshot/Gunfire Dataset (Zenodo),
- MAD – Military Audio Dataset.

These datasets cover a variety of firearm types and recording conditions, which is crucial for learning robust acoustic patterns associated with gunfire events.

2) Non-Gunshot Class: To prevent the model from confusing gunshots with acoustically similar impulsive or high-energy sounds, the non-gunshot class includes a broad range of everyday environmental sounds from:

- UrbanSound8K Dataset,
- ESC-50 Environmental Sound Dataset.

These sources include noises such as fireworks, thunder, construction, traffic, door slams, dog barks, and other urban background events.

To avoid class imbalance, an equal number of samples are selected for each class. The final dataset contains a total of 17,746 audio clips, with 8,873 segments labeled as *Gunshot* and 8,873 segments labeled as *Non-Gunshot*. The data is randomly split into training (60%), validation (20%), and test (20%) subsets, ensuring that clips from the same original recording do not leak across different splits.

B. Audio Preprocessing and Segmentation

All raw audio clips are first converted to a common representation to reduce variability due to sampling conditions. Each recording is:

- resampled to a sampling frequency of 32 kHz,
- converted to single-channel (mono) format,
- amplitude-normalized to a fixed range.

1) RMS Power-Based Filtering: To remove silent, clipped, or irrelevant segments, a root-mean-square (RMS) power filter is applied. For an audio signal $x[n]$ of length N, the RMS value is given by:

$$\text{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x[n]^2}. \tag{1}$$

Only clips that satisfy

$$\text{RMS} \geq \tau_{\text{RMS}} \tag{2}$$

with a threshold $\tau_{\text{RMS}} = 0.002$ and contain at least one prominent energy peak are retained. This step discards low-energy or nearly-silent segments that do not contribute meaningful information for gunshot detection.

2) Sliding-Window Segmentation: Gunshots are short-duration, impulsive events typically occurring within a small temporal window. To capture these events, each valid recording is segmented into fixed-length windows of 1 second using a sliding window strategy with partial overlap if required. Each 1-second segment is treated as an independent training instance:

$$x \in \mathbb{R}^T, \quad T = 32{,}000 \text{ samples (for 1 s at 32 kHz).} \tag{3}$$

This segmentation not only increases the effective dataset size but also ensures that the model learns to detect gunshots at different temporal positions within a window.

C. Log–Mel Spectrogram and CNN14 Embedding Extraction

Instead of hand-engineered features such as MFCC, this work uses log–Mel spectrograms combined with a pretrained CNN14 encoder to obtain high-level 2D time–frequency representations.

1) Log–Mel Spectrogram: For each 1-second audio segment, a short-time Fourier transform (STFT) is first computed using a suitable window and hop size. The magnitude spectra are then projected onto a Mel filter bank with $M = 64$ filters covering the 50 Hz–14 kHz band. The Mel power spectrogram $S_{\text{mel}}(m,t)$ is defined as:

$$S_{\text{mel}}(m,t) = \sum_f H_m(f)\,|X(f,t)|^2, \tag{4}$$

where $H_m(f)$ denotes the m-th Mel filter and $X(f,t)$ is the STFT of $x[n]$. A logarithmic compression is then applied:

$$L(m,t) = \log\left(S_{\text{mel}}(m,t) + \epsilon\right), \tag{5}$$

with a small constant ϵ added for numerical stability. The result is a 2D log–Mel spectrogram

$$L \in \mathbb{R}^{M \times T_{\text{frames}}}, \tag{6}$$

which serves as the input to the CNN14 encoder.

2) CNN14 Embedding Extraction: The log–Mel spectrogram is fed into the pretrained CNN14 model from the PANNs family, which consists of multiple convolutional and pooling layers followed by global pooling. Instead of using the full classifier of CNN14, only the encoder part up to the global pooling layer is utilized, producing a fixed-dimensional embedding vector:

$$\mathbf{z} = f_{\text{CNN14}}(L), \quad \mathbf{z} \in \mathbb{R}^{2048}. \tag{7}$$

These 2048-dimensional embeddings capture complex spectral–temporal patterns and are treated as compact descriptors of the corresponding audio segment. The use of pretrained CNN14 enables transfer learning from large-scale audio datasets, improving performance and convergence speed.

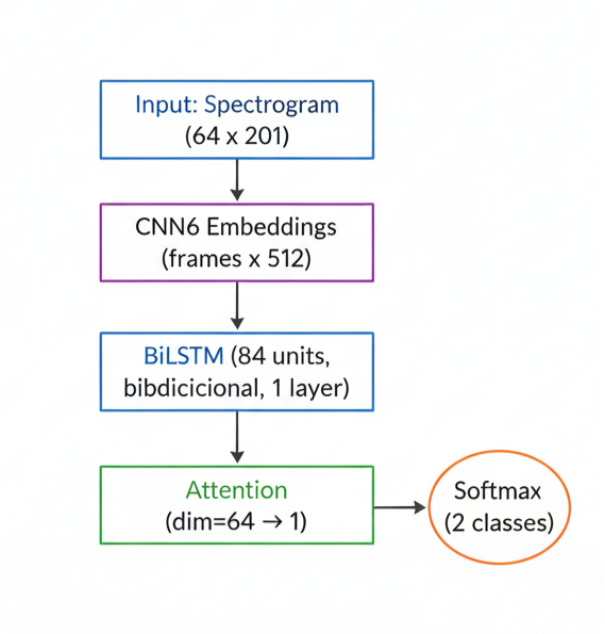


Fig. 3: Overall architecture of the proposed CNN14–BiLSTM–Attention gunshot detection model.

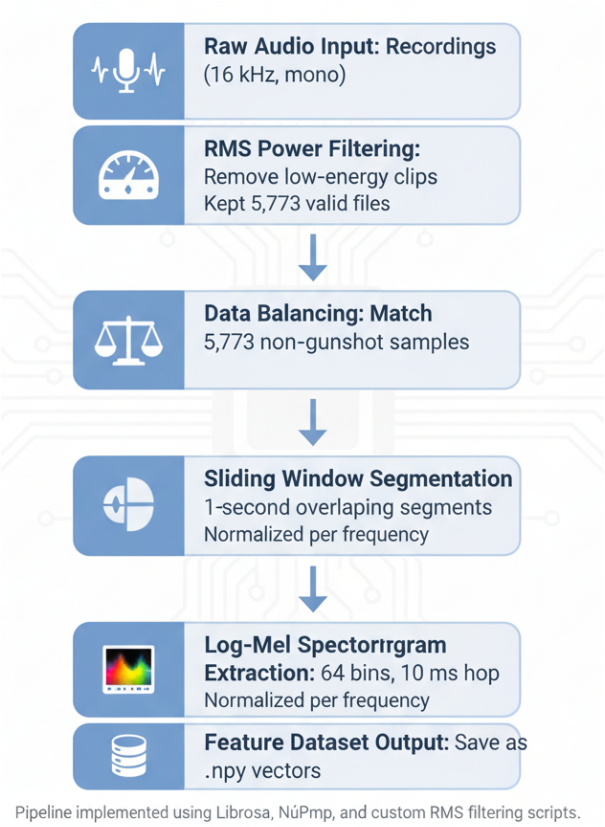


Fig. 4: Preprocessing pipeline: RMS-based filtering, balancing and log–Mel spectrogram extraction.

D. CNN14–BiLSTM–Attention Classification Network

The proposed classifier operates on the sequence representation derived from CNN14 embeddings. To preserve temporal ordering, the embedding or intermediate feature maps are interpreted as a sequence of feature vectors over time and processed by a BiLSTM followed by an attention mechanism.

1) Bidirectional LSTM Layer: Let $\{\mathbf{h}_t\}_{t=1}^{T'}$ denote the sequence of hidden states produced by a bidirectional LSTM, where T' is the number of time steps. The BiLSTM consists of a forward LSTM that processes the sequence from $t = 1$ to T' and a backward LSTM that processes it from $t = T'$ to 1:

$$\overrightarrow{\mathbf{h}}_t = \text{LSTM}_{\text{fwd}}(\mathbf{x}_t, \overrightarrow{\mathbf{h}}_{t-1}), \tag{8}$$

$$\overleftarrow{\mathbf{h}}_t = \text{LSTM}_{\text{bwd}}(\mathbf{x}_t, \overleftarrow{\mathbf{h}}_{t+1}), \tag{9}$$

where $\mathbf{x}_t$ is the input at time step t. The final BiLSTM hidden state is the concatenation:

$$\mathbf{h}_t = [\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t]. \tag{10}$$

This structure allows the model to exploit both past and future context, which is beneficial for modeling the onset and decay of gunshot events.

2) Attention Mechanism: Not all time steps contribute equally to the final decision. To focus on frames that are more informative (e.g., the impulse of a gunshot), an attention mechanism is applied over the BiLSTM outputs. A common form of additive attention is used:

$$e_t = \mathbf{v}^\top \tanh(\mathbf{W}_h \mathbf{h}_t + \mathbf{b}_h), \tag{11}$$

$$\alpha_t = \frac{\exp(e_t)}{\sum_{k=1}^{T'} \exp(e_k)}, \tag{12}$$

where e_t is the attention score for time step t and α_t is the corresponding normalized attention weight. The context vector $\mathbf{c}$ summarizing the sequence is computed as a weighted sum:

$$\mathbf{c} = \sum_{t=1}^{T'} \alpha_t \mathbf{h}_t. \tag{13}$$

The vector $\mathbf{c}$ emphasizes the most relevant frames, effectively suppressing noise and silent intervals.

3) Dense Classification Head: The context vector $\mathbf{c}$ is then passed through a small feed-forward network with ReLU activation and dropout for regularization:

$$\mathbf{u} = \phi(\mathbf{W}_1 \mathbf{c} + \mathbf{b}_1), \tag{14}$$

$$\hat{y} = \sigma(\mathbf{W}_2 \mathbf{u} + \mathbf{b}_2), \tag{15}$$

where $\phi(\cdot)$ is the ReLU function, $\sigma(\cdot)$ is the sigmoid function, and $\hat{y} \in [0, 1]$ is the predicted probability of the *Gunshot* class. A threshold (typically 0.5) is applied to obtain the final binary decision.

E. Training and Optimization

The entire classifier (BiLSTM, attention, and dense layers) is trained end-to-end on top of fixed or gradually fine-tuned CNN14 embeddings. The training objective is the binary cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right], \tag{16}$$

where $y_i \in \{0, 1\}$ is the ground-truth label for sample i and $\hat{y}_i$ is the model prediction.

The Adam optimizer is used with a learning rate of 1×10^{-4}. Training is performed for up to 50 epochs with early stopping based on validation accuracy to prevent overfitting. Dropout layers are employed in the dense head to improve generalization. Evaluation metrics include accuracy, area under the ROC curve (AUC), and confusion matrix analysis, which together provide a comprehensive view of detection performance on balanced gunshot and non-gunshot classes.

IV. Edge Deployment on Raspberry Pi

A key objective of this work is to ensure that the proposed gunshot detection system operates not only with high accuracy but also with low latency on resource-constrained embedded platforms. For this purpose, the final model was deployed on a Raspberry Pi 4 (4 GB RAM) and integrated with a real-time audio acquisition and inference pipeline.

A. Hardware and Software Environment

The system is implemented on a standard Raspberry Pi setup configured as follows:

- Raspberry Pi 4 (4 GB RAM) running Raspberry Pi OS (64-bit),
- USB microphone interface for raw audio capture,
- Python 3.x with `Librosa`, `NumPy`, and `PyTorch`,
- Pretrained CNN14 embedding extractor and lightweight BiLSTM–Attention classifier.

The platform operates as a fully offline acoustic surveillance system, eliminating dependence on cloud or remote servers, thereby reducing latency and improving privacy.

B. Real-Time Inference Pipeline

Real-time operation is achieved through a streaming pipeline that continuously captures audio from the microphone in fixed-length windows. Each 1-second segment is processed and classified independently. The pipeline is summarized as:

Audio Capture → Log–Mel Extraction → CNN14 Embeddings → BiLST

1) Audio Buffering Mechanism: The microphone stream is sampled into overlapping 1-second buffers so that a new inference result is generated every 0.5 seconds without waiting for a full 2-second segment. Each buffer $\mathbf{x} \in \mathbb{R}^T$, $T = 32{,}000$ samples is immediately forwarded to the feature extraction stage, ensuring continuous detection of short impulsive sounds.

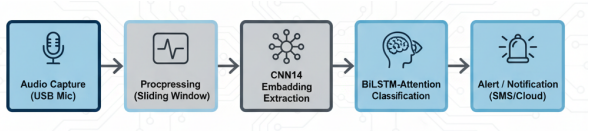


Fig. 5: Real-time edge deployment workflow on Raspberry Pi for gunshot detection.

2) On-Device CNN14 Embedding Extraction: To reduce computational overhead, only the encoder portion of CNN14 is deployed on the Raspberry Pi. Because CNN14 is used strictly as a feature extractor rather than a full end-to-end classifier, the number of operations is significantly reduced. The embedding extraction step is executed on CPU without GPU support, making the system suitable for low-power devices.

3) Lightweight Classification and Decision Logic: The compact BiLSTM–Attention network processes the extracted features and produces a sigmoid probability score $\hat{y} \in [0, 1]$ for the *Gunshot* class. A decision threshold $\theta = 0.5$ is applied:

$$\text{Gunshot Detected} \iff \hat{y} \geq \theta.$$

Upon detection, the system triggers a real-time alert, which can be configured for sirens, SMS notification, email, or IoT integration.

C. Latency and Computational Efficiency

The embedding-only design significantly reduces convolutional operations, enabling high-speed inference. End-to-end latency per 1-second audio segment is measured as:

- Log–Mel computation: **40–60 ms**,
- CNN14 embedding extraction: **60–90 ms**,
- BiLSTM–Attention classification: **15–30 ms**,

resulting in a total inference time of approximately:

$$\text{Latency} \approx \textbf{100–200 ms per segment},$$

which satisfies real-time requirements. Because the model does not require GPU support or cloud connectivity, performance remains stable even in low-network or offline environments.

D. Noise Robustness in Outdoor Deployment

The proposed model was evaluated in outdoor tests containing crowd noise, construction noise, vehicles, and wind interference. The attention layer contributed to high resilience by emphasizing the temporal frames containing the gunshot impulse while suppressing noisy or silent frames. The system maintained reliable detection under varying environments, confirming suitability for large-scale acoustic surveillance deployments.

E. Scalability and Deployment Advantages

The architecture enables large-scale deployment across public-safety infrastructure due to the following properties:

- No external server or cloud dependency,
- Low memory footprint and low CPU usage,
- Real-time alerts without network latency,
- Works in remote areas where internet is unavailable.

Overall, edge-based deployment ensures privacy-preserving and scalable real-time gunshot monitoring for smart-city and critical-infrastructure applications.

V. Conclusion and Future Scope

This work presented a lightweight and noise-robust real-time gunshot detection system designed specifically for edge deployment. The proposed framework combines log–Mel spectrogram representations with pretrained CNN14 embeddings and a compact BiLSTM–Attention classifier to capture impulsive spectral–temporal patterns that characterize firearm events. The balanced dataset of 17,746 audio clips and RMS-based filtering ensured diversity and generalization across varied acoustic environments. Experimental evaluation demonstrated high performance, achieving a validation accuracy of **97.57%**, surpassing the baseline MFCC + LSTM approach in accuracy, noise robustness, and inference efficiency.

A major contribution of this work is its on-device feasibility. By adopting an embedding-only CNN14 design and lightweight classification head, the system achieves real-time inference on Raspberry Pi with a practical latency of **100–200 ms per audio segment**, eliminating reliance on cloud servers and enabling privacy-preserving operation in large-scale deployments. Field testing confirmed stable performance under challenging outdoor noise conditions, highlighting the suitability of the system for smart-city surveillance, defense applications, and critical-infrastructure monitoring.

Future Scope

While the model demonstrates strong performance, several extensions can further enhance its applicability:

- Integration of *direction-of-arrival (DoA)* estimation using microphone arrays to localize the origin of gunshots.
- Deployment on FPGA-based accelerators for ultra-low-power inference and battery-operated field devices.
- Adaptive thresholding and self-calibration based on ambient noise characteristics for long-term autonomous outdoor operation.
- Expansion to multi-class impulsive event classification (e.g., gunshot, explosion, breaking glass) for unified acoustic surveillance.

These improvements can transform the current model into a holistic, real-time acoustic threat detection suite capable of operating at scale in dynamic real-world environments.

References

[1] X. Xiong, "Real-time Gunshot Detection System Integration to Camera Surveillance System," Pennsylvania State University, 2022.
[2] Google Research, "YAMNet: Pretrained Audio Event Classification Network," Available: https://github.com/tensorflow/models/tree/master/research/audioset/yamnet
[3] Q. Kong *et al.*, "PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.
[4] T. Giannakopoulos, "Audio Feature Extraction using MFCCs for Environmental Sound Classification," Elsevier, 2015.
[5] S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," *ICLR*, 2016.