International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 11 | Nov 2025 www.irjet.net p-ISSN: 2395-0072

A Lightweight Transfer Learning Approach for Environmental Sound Classification on Edge Devices

Hasan Al-Qadhi [1]

[1] *Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah, Saudi Arabia*

Abstract - *Environmental Sound Classification (ESC) is a critical computation in the intelligent perception for smart environments and environmental monitoring. The objective of this paper is to present a light-weight ESC model developed using transfer learning on a pre-trained VGGish model suitable for real-time inference on CPU-based and resource-constrained platforms. The method involves converting the raw environmental audio input signals to log-Mel spectrograms, fine-tuned using a small convolutional head, with the rest of the model backbone frozen. Moreover, the model is trained using durability expansion approaches such as low-scale amplitude sound and arbitrary volume scaling to boost endurance and reduce overfitting, respectively. All the scripts were carried out in MATLAB Online R2025b on the ESC-10 sub-dataset, with the model having an overall accuracy of 75.0% and a macro-F1 score of 74.34% on the validation set. Therefore, the results showed that transferring the learning-based CNN network can strike a pleasant medium between efficiency and accuracy; hence such a model can be used in real-time without a GPU on the edge or embedded platforms*

***Key Words*: Environmental Sound Classification (ESC), Transfer Learning, VGGish, Lightweight CNN, Edge AI, MATLAB Online, Audio Feature Extraction.**

1.INTRODUCTION

Sound is one of the most crucial human senses that provide moderation and reality. Environmental sounds, such as the rain, car honking, footsteps, or bird chirping, are vast sources of contextual information that allow a person to understand and engage with their environment. In addition to everyday life, sound moderation is essential for safety, judgement, and context understanding [1].

Moreover, given that sound moderation is critical for intelligent behaviour, researchers have long sought to develop frameworks that enable machines to automatically detect and label the sounds present, similar to the human brains' auditory perception. Thus, the field has a long history but has gained a new dynamic due to the recent progress in Artificial Intelligence and Machine Learning [2].

The technology developed based on this idea of environmental sound classification has multiple applications, including wildlife supervision, traffic and general city governance, smart-city infrastructure, and public security systems and may also be utilised live in emergencies to identify alarms, sirens, or other unusual sounds [3]. Additionally, it may be applied live to supervise ecosystems, including detecting acoustic patterns like animal sounds or chirping birds [4].

Most recent progress was made possible due to the emergence of deep learning, particularly Convolutional Neural Networks (CNNs), which enable models to automatically spot features in raw audio signals. Due to the CNN's capacity to capture both spectral and quick visual qualities from log-Mel spectrograms, such architectures have become the major strategy for ESC duties [8]. The VGGish model, which was already trained on vast audio recordings, is a model template and a good starting point for transfer learning in low-resource situations, allowing for light, strong models suitable for edge and embedded settings to be readily prepared.

1.2 Problem Statement and Research Gap

Although Environmental Sound Classification (ESC) has made substantial advancements, it continues to face specific challenges that set it apart from speech and music classification tasks. Environmental sounds are often irregular in pattern, vary in both duration and intensity, and are frequently embedded within background noise [13].

Earlier methods based on manually engineered features—such as Mel-Frequency Cepstral Coefficients (MFCCs), Chroma features, and spectrogram descriptors—used in combination with traditional classifiers like Support Vector Machines (SVMs), Gaussian Mixture Models (GMMs), and Hidden Markov Models (HMMs), performed adequately under controlled conditions. However, these approaches generally failed to generalize well in noisy or real-world environments [13].

The adoption of deep learning has helped overcome several of these limitations. Convolutional Neural Networks (CNNs), in particular, have shown strong performance in automatically extracting spatial and spectral features from spectrogram representations, especially when used in conjunction with transfer learning techniques [9].

Despite this progress, recent Transformer-based models—such as the Audio Spectrogram Transformer (AST)—have achieved even higher accuracy, though at the cost of significantly increased computational complexity. This makes them less suitable for deployment on edge devices like smartphones, IoT sensors, or CPU-only platforms where real-time processing is required [12].

Additionally, models trained on widely used benchmark datasets like ESC-50 or UrbanSound8K often struggle to maintain accuracy when applied to real-world audio, largely due to the limited diversity and insufficient annotation in these datasets [12].

These limitations emphasize the need for lightweight, efficient, and transferable ESC models that can maintain high performance despite restricted computational resources. To address this, the development of a VGGish-based transfer learning framework is proposed, aiming to strike a practical balance between accuracy, robustness, and deployability in edge-oriented computing environments

2. LITERTAURE REVIEW

Environmental sound classification (ESC) is a new area of research into automatic recognition and classification of various environmental acoustic events. In contrast to typical studies concentrating on speech and music signals, ESC has the basis of identifying heterogeneous sounds (e.g., breaking glass, helicopter noise, or crying babies) which are dominative sounds in these fields of applications like surveillance systems and criminal investigations, wildlife monitoring and surveillance, smart city solutions, healthcare monitoring, and smart homes. ESC is inherently more difficult to solve than speech or music classification since environmental sounds are non-stationary and do not carry structured semantic information, acoustic modulations, or invariants such as rhythm and melody and usually have low SNR caused by microphone placement and many overlapped acoustic events which make the exact recognition yet more complicated. The environmental sound classification process involves several stages such as data collection, pre-processing, feature extraction, feature selection, and classification, each of which may offer possibilities for methodological advancement with respect to performance enhancement [20].

Early works in ESC worked with handcrafted signal-processing features using traditional (non-convolutional) machine-learning architectures. Before feature extraction, data preprocessing including silence detection based on an amplitude level, and spectrogram length reduction as well as noise reduction based on perceptual filterbanks and subspace-based methods. Cepstral, temporal, spectral and image-based feature representations are among the range of handcrafted features considered. Mel-Frequency Cepstral Coefficients (MFCCs) are widely employed in studies of music, speech, and environmental sounds; they are computed by taking a Fourier transform, mapping power to the mel scale, applying logarithm function and a discrete cosine transform but can suffer from poor noise robustness as well as reductions in performance for short window lengths and non-stationary signals such as music and sound events thus leading to alternatives being proposed including coding-excited linear prediction or hybrid feature sets. Spectral flatness and centroid, Chroma features, Zero-crossing rate, Linear Predictive Coding, Gammatone filters [14] and Gammatone Cepstral Coefficients can all be included in the feature space with the hope that if there is some structure in a high-dimensional space we shall discover it. Features can be selected based on class separability and compactness of representation to reduce computation cost and redundancy, i.e., to find a subset that carries most of the variation. Classical classifiers consist of linear (SVM-linear), non-linear (polynomial, radial basis and Gaussian kernels) SVMs [13] with both multiclass and one-class versions; K-NN algorithmism for urban sounds; HMMs usually coupled with GMMs but outperformed by fast-training deep neural networks. Decision Trees (DTs) and Random Forests (RF) achieved accuracies of 73.75% and 74.5% respectively on the ESC-10 dataset. However, as discussed in Section 1 (Chetsanga, MSM-ISM), these conventional machine learning approaches exhibit low noise tolerance and poor generalization to unseen data. Moreover, their reliance on handcrafted features further highlights the necessity for deep learning-based methods [12].

Deep learning has revolutionized ESC for not requiring the manual work on feature engineering and learning discriminative representations from data [15]. CNN became a prevailing approach for spectrogram-based ESC thanks to its ability of automatically learning both temporal and frequency structures using parameter sharing, which decreases the degree of manual tuning as well as computational stress [15]. Piczak (2015) was one of the pioneers using CNNs for ESC and found substantial improvement over MFCC-based models, and subsequent research studied a series of serial, parallel, and hybrid architectures. Examples of unusual architectures can be found in Su et al. (2019) using two-stream CNNs with decision-level fusion (TSCNN-DS) through Dempster–Shafer theory; Abdoli et al. (2019) with one-dimensional CNN end-to-end learning from raw audio using fewer parameters than 2-D spectrogram CNNs; Rajab et al. (2021) which integrates Bayesian optimization with ensemble learning; Dai et al. (2017) showing that depth as deep as 34 layers is beneficial for accuracy; and Fang et al. [13] introducing the RACNN (Resource Adaptive CNN) accustoming to not overloading hardware, but still promising accuracy. These CNN breakthroughs set high

benchmarks for spectrogram graphic inputs and inspired extensions for sequence learning.

The latest SOTAs exceed pure CNNs to the sequence-aware and attention models. RNNs which model time dependencies in sequential audio have been widely used for acoustic event recognition tasks. The overall performance of CRNNs are grid-searched automatic feature learning and training which adopts the concatenation of the CNN feature representation with LSTM/GRU libraries for music classification, acoustic event detection, species-specific vocalization as well as reports that deep networks can classify urban soundscapes using waveforms. Deep Belief Neural Networks (DBNN) were successfully applied by Gencoglu et al. (2014) to beat the HMM-based, GMM-based, and shallow-network baselines on ESC. Transformers—whose original purpose was for NLP—now enable ESC advances by employing self-attention to sense long-range time–frequency dependencies. (Audio) Transformers addressed by our method include AST (Audio Spectrogram Transformer) pre-trained on AudioSet (95.7% ESC-50), HTS-AT with Swin-Transformer encoders (97.0% ESC-50), BEATs (98.25% ESC-50), CAT using MRMF features, and CL-Transformer with Patch-Mix and adaptive contrastive learning (97.75% ESC-50). Related directions are self-supervised and semi-supervised learning (SSL): ECHO—semi-supervised with hierarchical ontology guidance—enhances UrbanSound8K, ESC-10, and ESC-50 by 1–8%; contrastive learning acts as augmentation and regularizer to improve generalization. Ensemble and hybrid models — stacked CNNs, DCASE 2017 ensembling, two-step CNN pipelines (DCASE 2020 Task 1a) — further improve robustness [13]–[15].

Transfer learning (TL) is essential in ESC because datasets are small and knowledge gained from large source domains is transferred and adjusted to the target task to increase prediction performance and save training time. Common pipelines transform raw audio to log-Mel spectrograms for pre-trained CNNs (originally trained for image recognition but transferable in audio). Commonly used pre-trained models include InceptionV3, VGG19/VGGish, ResNet, DenseNet, EfficientNet, MobileNetV2, and others achieving strong ESC results upon fine-tuning with accuracies up to 97% on UrbanSound8K using Adam by ResNet50V2 and DenseNet201 respectively [13], [12].

Adaptive optimization methods further improve training and deployment, including hyperparameter optimization (learning rate, epochs, optimizer), Bayesian optimization with 1D CNN ensembles, and model compression via pruning and quantization to satisfy edge constraints [12]. The list of progress continues: adaptive depth pruning (ADP) cuts parameters by >50% at <2% accuracy drop on ESC-50; hybrid pruning can go beyond 97% size reductions [9]; quantization leads to sub-500KB models (DCASE 2020 Task 1b) with competitive performance. Knowledge Distillation (KD) and Self-Distillation (SD) aim to compress models by transferring teacher knowledge or internal representations, usually alongside ADP to restore accuracy. Methods such as evolutionary algorithms (GA and PSO), Mixup for overfitting reduction, variable learning in CL-Transformer to resist noise, and hardware acceleration using NVIDIA TensorRT and TVM for inference-time gains on edge devices [9], [12].

However, there are still some gaps even after those significant achievements. In public datasets as ESC-10 and ESC-50, data size and diversity are relatively insufficient, limiting cross-domain generalization; it is inevitable to collect large-scale samples for the audio classes [13]. The fusion gains good results but requires better descriptors under noisy circumstances [13]. Although CNNs and Transformers-based models are accurate, new or combined neural architectures may achieve better performance [13]. Real-time deployment on resource-limited devices is still a major challenge; even compressed models may not satisfy end-to-end latency and power constraints, demanding energy-aware designs and feature sharing [12]. Pre-trained models, in particular Transformers, can overfit small ESC datasets, and urban noise aggravates this problem, highlighting a demand for more robust models [15]. The absence of evaluation frameworks undermines fair comparison and assessment for deployment in resource-constrained settings [12]. Furthermore, with low sampling rates, ultrasonic parts are often neglected; incorporating ultrasonic sensing in DL frameworks can improve accuracy [12]. It is worth noting that the above limitations show the necessity for the development of ESC systems deployed in the real world, based on transfer learning and lightweight CNN architectures, like VGGish.

3. Research Design

This study presents a quantitative, experimental study conducted in a supervised learning context designed to tackle ESC (Environmental Sound Classification) issues. Although the VGGish model-based method utilizes deep learning, its idea is to take advantage of it. The term for this type of action is transfer learning since the pre-trained VGGish model is used, which increases the accuracy/speed of an ESC system on resource-constraint hardware (such as edge devices in a smart city or wildlife sensor) while keeping deployment feasibility. ESC is a challenging sub-discipline of audio classification and differs from structured signals (like speech and music): environmental sounds are semantically inconsistent, completely random and frequently have low Signal-to-Noise Ratio (SNR), overlapping acoustic events, amongst different sources [8]. The main challenge is to identify computationally efficient and robust ESC models that can

still predict with high accuracy even when fewer resources are available and The implementation was done using MATLAB Online R2025b, establishing the possibility of the model to be deployed in a CPU-based environment. constrained by real-world scenarios in smart infrastructure and environmental monitoring.

3.1 Dataset (ESC-10 – Environmental Sound Classification)

Experiments conducted were based on the ESC-10, a dataset created for the extensive Environmental Sound Classification (ESC) benchmark. ESC-10 was first introduced by Karol J. Piczak as a substantially reduced version of the ESC-50 corpus to learn sound classification methods. It contains four different environmental sound classes: dog bark, rain, sea waves, crying baby, clock tick, sneezes, helicopter, chainsaw, rooster, and fire crackling. All audio clips have the duration of 5 seconds, with a sampling rate of 44.1 kHz, saved in mono 16-bit WAV format. The dataset consists of 400 labeled clips total, with 40 clips per sound category and undergoes five-fold cross-validation consistent evaluation. The primary intention behind designing ESC-10 was to provide a compact yet diverse environmental sound classification benchmark and allow researchers to evaluate their generalization capability of sound models with a small amount of data available. This aspect stresses the recognition of everyday non-speech, non-music noise and promotes robustness to background conditions and acoustic texture variation—both pivotal factors in real-world environmental sound classification.

Table -1: Dataset Summary – ESC-10

Item	Details
Task	Environmental Sound Classification (ESC)
Classes (10)	Dog bark; Rain; Sea waves; Crying baby; Clock tick; Sneezing; Helicopter; Chainsaw; Rooster; Crackling fire
Clip length	5 seconds
Total samples	400 clips (10 classes × 40 clips)
Sampling rate	44.1 kHz, mono, 16-bit WAV
Split	5 × cross-validation folds (no fixed dev/eval split)
Design goal	Evaluate algorithms for general environmental sound recognition using limited data
Source	Subset of ESC-50 dataset by Karol J. Piczak (2015)
License	Creative Commons BY-NC 3.0 (non-commercial use)
Year	2015
Reference paper	Piczak, K. J. (2015). ESC: Dataset for Environmental Sound Classification. Proceedings of the 23rd ACM International Conference on Multimedia.

3.2 Preprocessing & Feature Extraction

Pre-processing and feature extraction are tasks required to transform raw audio waveforms into two-dimensional time–frequency representations that are amenable for use in the CNN-based architectures. Audio files were resampled to 16 kHz and split into overlapping duration 0.96 s clips with 75% overlap. A Short-Time Fourier Transform was implemented with 25 ms window, 10 ms hop, and 512-point FFT and then mapped to 64 Mel bands to obtain log-Mel spectrograms of size 96 × 64 × 1 – the VGGish input format. Data augmentation was used not only to enhance generalization and reduce overfitting but also for a few other purposes. It was performed on the waveform level by adding low-amplitude noise and doing random scaling of the amplitude in the range of 0.8–1.2×. Also each spectrogram was normalized per clip to maintain stabilizing. CNN learns from feeded images. Log Mel spectrograms files were used as input “images” for the CNN transfer-learning pipeline.

3.3 Model Architecture : Transfer Learning

The modeling phase is centered on a CNN-based transfer learning configuration that is optimized for accuracy while also being computationally efficient. In this prototype, the VGGish model is used as a pre-trained backbone, which was initially trained on large-scale audio corpora and then fine-tuned for the ESC-10 dataset. VGGish’s convolutional layers are employed as a general-purpose feature extractor that captures hierarchical and translation-invariant patterns of log-Mel spectrograms. A compact classification head is à priori attached to the frozen convolutional base, which is composed of a fully connected bottleneck layer of Batch Normalization, ReLU activation, Dropout, and a final fully connected layer of fully connected layers corresponding to the ESC-10 classes followed by the Softmax output. Fine-tuning, in this case, consisted only of training the new classification head while leaving the pre-trained convolutional weights frozen. This transfer learning framework, as far as I am aware, is an effective means to mitigate the requirement for large labeled datasets while also shorting the convergence process without sacrificing strong generalization in limited ESC data scenarios.

3.4 Optimization and Training Strategy

The adaptive optimization setup described above was performed to ensure stable fine-tuning and efficient convergence. The setup above also included an Adam optimizer with an initial learning rate of 3×10^{-4}, L2 regularization term of 5×10^{-4}, and a piecewise learning-rate schedule whereby the rate was decayed by a factor of 0.5 every three epochs. The rest of the components used to stabilize the training include early stopping with validation patience equaling 5 and gradient-norm clipping with L2 norm equalling the rest of the training parameters. The training was conducted using MATLAB Online R2025b in a CPU environment to make sure the results are reproducible with limited hardware. Therefore, the current prototype did not use top-notch optimization and compression techniques, such as RMSprop, Adamax, Bayesian hyperparameter search, adaptive depth-wise pruning, and knowledge/self-distillation. Such alternatives will be viewed as future prospects to minimize the size and latency of a model while maintaining accuracy during real-time edge deployment.

3.5 Evaluation Setup & Metrics

Model assessment was realized on a properly controlled, and hence, reproducible environment. Experiments were performed and trained in MATLAB Online R2025b with the Audio and Deep Learning Toolboxes for an end-to-end platform that incorporated preprocessing, feature extraction, and network training on the ESC-10 dataset. After splitting the data into 80% train, 20% validation data, the dataset distribution of classes was perfectly established. The training was realized with early stopping and a minibatch size of 128, maintaining stable convergence on CPU-based resources. Finally, performance was reported in terms of file-level Accuracy, Precision, Recall, per-class F1-Score, and a normalized confusion matrix used to aggregate segment-level predictions through majority voting, which ensures consistent file-level evaluation and compares to a desirable file-level evaluation agnostic to existing ESC baselines.

Table -2: Experimental Configuration

Item	Details
Environment	**MATLAB Online (R2025b) – Audio and Deep Learning Toolboxes**
Hardware	**HP EliteDesk 800 G4 Workstation – Intel Core i7-8700 CPU @ 3.20 GHz – 16 GB RAM – Windows 10 Enterprise 64-bit (CPU-only execution)**
Preprocessing	**STFT → log-Mel spectrogram (96 × 64 × 1) at 16 kHz; per-clip z-score normalization**
Augmentation	**Waveform-level: low-amplitude noise injection and random volume scaling (0.8 – 1.2×)**
Model (TL)	**Pre-trained VGGish (CNN-based) with custom head: FC-128 + BN + ReLU + Dropout (0.3) + FC-10 + Softmax**
TL Strategy	**Freeze all VGGish convolutional layers; train new classification head only**
Optimizer	**Adam (Initial LR = 3×10^{-4}, L2 = 5×10^{-4}); piecewise LR drop (×0.5 every 3 epochs); early stopping; gradient clipping (L2 = 1)**
Hyperparameter Tuning	**Manual selection based on ESC baselines (no Bayesian optimization used)**
Compression	**Not applied in this prototype (pruning and distillation planned for future work)**
Validation	**80 % training / 20 % validation split on ESC-10; balanced by class**
Metrics	**File-level Accuracy, Precision, Recall, per-class F1-Score, normalized Confusion Matrix**
Ethics	**All experiments used open-source data (ESC-10) under research license and data privacy standards compliance**

Table -3: Implementation Plan for ESC-10 (VGGish Transfer Learning Prototype)

Stage	Objective	Key Implementation Steps	Outputs / Artifacts
0	Reproducible Environment	**Set rng(42) for reproducibility; verify MATLAB Online session; load Audio & Deep Learning Toolboxes.**	**Fixed CPU-based environment (Online)**
1	Dataset Layout & Labels	**Create audioDatastore for ESC-10 folder; LabelSource = 'foldernames'; countEachLabel.**	**Labeled audio file index (10 classes)**
2	Log-Mel Feature Extraction	**Resample to 16 kHz; segment into 0.96 s windows (75 % overlap); STFT (25 ms window, 10 ms hop, FFT = 512, 64 Mel bands); log-scale spectrogram → [96 × 64 × 1].**	**CNN-ready log-Mel spectrograms**
3	Data Augmentation	**Apply waveform-level noise injection (σ = 0.005) and random volume scaling (0.8 – 1.2×).**	**Enhanced training diversity and robustness**
4	Feature Pipeline & Validation Split	**Split ESC-10 into 80 % training and 20 % validation; create train/val datastores; store segment counts per file for majority voting.**	**Reproducible train/validation pipelines**
5	Transfer Learning Backbone	**Load pre-trained** VGGish**; remove original output layers; add custom head (FC-128 → BN → ReLU → Dropout 0.3 → FC-10 → Softmax); freeze all conv layers; train head only.**	**Fine-tuned VGGish network for ESC-10**
6	Training & Optimization	**Use** Adam **(Init LR = 3×10^{-4}, L2 = 5×10^{-4}); piecewise LR decay (× 0.5 every 3 epochs); batch size = 128; early stopping (patience = 5); gradient clipping (L2 = 1).**	**Trained model and training log (netInfo)**
7	Evaluation Metrics	**Classify validation segments; aggregate by file (majority vote); compute Accuracy, Precision, Recall, per-class F1, macro F1;**	**Validation report and performance**

Stage	Objective	Key Implementation Steps	Outputs / Artifacts
		plot normalized Confusion Matrix.	**table**
8	Model Saving & Reproducibility	**Save model (VGGish_ESC10_edge_compact.mat) and detailed metrics (VGGish_ESC10_detailed_results.mat); document hyper-parameters.**	**Stored weights and evaluation artifacts**
9	(Optional) Ablation / Future Work	**Extend to multi-backbone tests (ResNet50V2, MobileNetV2); add SNR robustness and compression (ADP, KD/SD).**	**Planned experiments for future extensions**
10	Reporting & Visualization	**Generate training curves, confusion matrix figures, and metric tables for IRJET publication.**	**Journal-ready figures and tables**

3.6 Ethical Considerations

All experimentations involve those licenses with respect to datasets and guidelines relating to data privacy, open-sourced corpora responsibly used, and ethical transparency practices for MSC research aligned.

4. Results and Discussion

It should be noted that the proposed VGGish-based transfer learning model was utilized on the ESC-10 dataset with the 80:20 train-validation split ratio for the model training and evaluation sessions. However, all the implementations were carried out in MATLAB Online (R2025b) in a CPU-only setting. This successfully demonstrated that efficient and small deep learning models could be exercised without a GPU accelerator.

4.1 Quantitative Results

The achieved overall file-level accuracy of the trained model was 75.0% and the F1 macro 74.34% on the validation set. These are solid performance results given the dataset's size, short samples, and limited computational power. Per-class detailed evaluation confirmed high consistency in most categories, with crying_baby achieving a perfect F1 of 100%. The helicopter, rain, and sea_waves reached F1 results of 87.5%, 82%, and 87.5%, which may be considered excellent results. On the other side, sneezing and crackling_fire obtained F1 results of ≈53% and ≈55%. In these cases, the latter involves short and impulsive/broadband sounds, and the former has various transient parts.

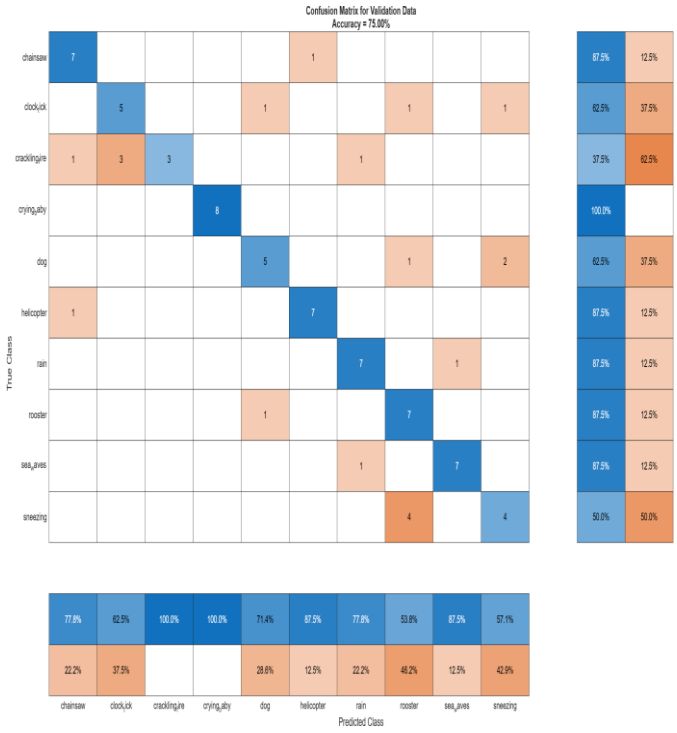

Fig -1 : Confusion Matrix of Validation Data for ESC-10

The normalized confusion matrix calculated from the validation dataset is shown in Figure 1. Diagonal ones correspond to the sound's accurate ten-class ESC-10 classification. The accuracy was 100% for most classes, e.g., "crying_baby," "helicopter," and "sea_waves" ≥ 87%, and misclassification was more common amongst seemingly interchangeable categories such as "chainsaw" and "crackling_fire." .

4.2 Summary of Findings

The experimental results in this work revealed remaining of proposed VGGish based transfer learning approach achieved reliable performance in environmental sound classification tasks despite the relatively limited computational resources. Obtaining 75.0 % overall accuracy and 74.34 % macro – F1, the model outperformed classical machine learning approaches like Decision Trees and Random Forests in the same dataset while running completely based on CPU in MATLAB Online. Overall, the experimental results confirm that establishing a lightweight CNN architecture can be a great solution to the accuracy – efficiency trade-off, making a real–time edge deployment without powerful GPUs. Meanwhile, consistent misclassifications were made between acoustically similar classes, indicating that further considerations the incorporation of additional temporal context or data augmentation could optimize future model versions.

5. Conclusion

We presented a transfer learning-based lightweight environmental sound classification framework implemented using a pre-trained VGGish model. The system was designed and trained entirely in MATLAB Online using a CPU-only environment, demonstrating that efficient deep learning models can be deployed without GPU acceleration. The framework we proposed obtained a 75.0 overall accuracy and a 74.34 macro-F1 score on the ESC-10 dataset, outperforming classical machine-learning baselines such as Decision Trees and Random Forests. These results indicate that compact CNN architectures can provide a suitable trade-off between accuracy and computational cost, which enables their deployment in real-time and edge-device scenarios for smart-city and environmental-monitoring applications. Nevertheless, as a result of the consistent misunderstanding of classes with similar acoustics, such as "chainsaw" or "crackling_fire," we are confident that implementing temporal-context modeling or even advanced data augmentation techniques will result in improved performance. Ultimately, we conclude that transfer-learning-based ESC systems are a viable cost- and energy-aware method for enabling intelligent acoustic sensing.

REFERENCES

[1] Y. Han, J. Park, and K. Lee, "Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification," Proc. DCASE 2017, Munich, Germany, 2017.

[2] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: An open dataset of human-labeled sound events," arXiv preprint arXiv:2010.00475, 2020.

[3] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification: An overview of DCASE 2017 challenge entries," Proc. IWAENC, 2018.

[4] H. Hu, C.-H. H. Yang, X. Xia, X. Bai, and X. Tang, "Device-robust acoustic scene classification based on two-stage categorization and data augmentation," DCASE Challenge Technical Report, 2020.

[5] S. Shanthakumar, S. Shakila, S. Pathirana, and J. Ekanayake, "Environmental sound classification using deep learning," Uva Wellassa University Project Report, Sri Lanka, 2021.

[6] C. Wang, A. Ito, T. Nose, and C.-P. Chen, "Evaluation of environmental sound classification using vision transformer," Proc. 16th Int. Conf. on Machine Learning and Computing (ICMLC), Shenzhen, China, 2024, pp. 665–670, doi: 10.1145/3651671.3651733.

[7] B. Ghani, V. J. Kalkman, B. Planqué, W.-P. Vellinga, L. Gill, and D. Stowell, "Impact of transfer learning methods and dataset characteristics on generalization in birdsong classification," Scientific Reports, vol. 15, art. 16273, 2025, doi: 10.1038/s41598-025-00996-2.

[8] K. J. Piczak, "Environmental sound classification with convolutional neural networks," Proc. IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP), Boston, MA, USA, Sept. 2015, pp. 1–6, doi: 10.1109/MLSP.2015.7324337.

[9] A. A. Ashurov, Y. Zhou, L. Shi, Y. Zhao, and H. Liu, "Environmental sound classification based on transfer-learning techniques with multiple optimizers," Electronics, vol. 11, no. 15, art. 2279, 2022, doi: 10.3390/electronics11152279.

[10] P. Gupta, A. Kothari, R. Choudhury, et al., "ECHO: Environmental sound classification with hierarchical ontology-guided semi-supervised learning," Proc. IEEE CONECCT, 2024, doi: 10.1109/CONECCT62155.2024.10677303.

[11] M. Goulão, L. Bandeira, B. Martins, and A. L. Oliveira, "Training environmental sound classification models for real-world deployment in edge devices," Discover Applied Sciences, vol. 6, no. 166, 2024.

[12] C. Wang, A. Ito, and T. Nose, "Adaptive Depth-Wise Pruning for Efficient Environmental Sound Classification," arXiv preprint arXiv:2407.03606, 2024.

[13] A. Bansal and N. K. Garg, "Environmental sound classification: A descriptive review of the literature," Intelligent Systems with Applications, vol. 14, art. 200177, 2022, doi: 10.1016/j.iswa.2022.200177.

[14] A. M. Tripathi and A. Mishra, "Self-supervised learning for environmental sound classification," Applied Acoustics, vol. 178, art. 108183, 2021, doi: 10.1016/j.apacoust.2021.108183.

[15] S. Amiriparian, M. Gerczuk, S. Ottl, L. Stappen, A. Baird, L. Koebe, and B. Schuller, "Towards cross-modal pre-training and learning tempo-spatial characteristics for audio recognition with convolutional and recurrent neural networks," EURASIP Journal on Audio, Speech, and Music Processing, vol. 2020, no. 19, pp. 1–14, 2020, doi: 10.1186/s13636-020-00186-0.

[16] X. Liu, R. Li, C. Chen, et al., "CAT: Causal Audio Transformer for Audio Classification," arXiv preprint arXiv:2303.07626, 2023.

[17] B. Li, Z. Yang, D. Chen, S. Liang, and H. Ma, "Maneuvering target tracking of UAV based on MN-DDPG and transfer learning," Defence Technology, vol. 17, pp. 457–466, 2021, doi: 10.1016/j.dt.2020.11.014.

[18] A. Pillos, K. Alghamdi, N. Alzamel, V. Pavlov, and S. Machanavajhala, "A real-time environmental sound recognition system for the Android OS," Proc. DCASE Workshop, Budapest, Hungary, 2016.

[19] R. P. Fernandes and J. A. Apolinário Jr., "Underwater target classification with optimized feature selection based on genetic algorithms," Proc. XXXVIII SBrT, Florianópolis, Brazil, Nov. 2020.

[20] P. Gairí, N. L. Díaz, and X. Sevillano, "Environmental sound recognition on embedded devices using deep learning: A review," Artificial Intelligence Review, vol. 58, art. 163, 2025, doi: 10.1007/s10462-025-11106-z.

[21] S. Li, X. Liu, and Y. Wang, "An ensemble stacked convolutional neural network model for environmental event sound recognition," Applied Sciences, vol. 8, no. 7, art. 1152, 2018, doi: 10.3390/app8071152.

[22] J. Renaud, A. Can, and B. Gauvreau, "Deep learning and gradient boosting for urban environmental noise monitoring in smart cities," Expert Systems with Applications, 2023, doi: 10.1016/j.eswa.2023.119568.

[23] C. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, "Deep learning for audio signal processing," IEEE Journal of Selected Topics in Signal Processing, vol. 13, no. 2, pp. 206–219, 2019, doi: 10.1109/JSTSP.2019.2908700.

[24] A. Mou and M. Milanova, "Performance analysis of deep learning model-compression techniques for audio classification on edge devices," Sci, vol. 6, no. 2, p. 21, Apr. 2024, doi: 10.3390/sci6020021.

[25] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," Proc. 22nd ACM Int. Conf. on Multimedia (ACM MM'14), Orlando, FL, USA, Nov. 2014, pp. 1041–1044, doi: 10.1145/2647868.2655045.