Utilizing Liquid Neural Network for Efficient Audio Classification

Department of Electronics and Computer Engineering
Thapathali Campus

Team Members
Khemraj Shrestha (THA077BCT020)
Niyoj Oli (THA077BCT029)
Om Prakash Sharma (THA077BCT030)
Punam Shrestha (THA077BCT038)

Supervised By
Er. Kshetraphal Bohara

Co-Supervised By
Er. Rojesh Man Shikhrakar

August 2024

Presentation Outline

- Motivation
- Project Objectives
- Project Scope
- Project Applications
- Methodology
- Results
- Discussion of Results
- Remaining Tasks
- References

Motivation

- Contemporary models uses millions to billions parameters,
- 19 neurons enough for autonomous driving - Liquid Time Constant (LTC) Neural Network,
- Papers suggest LNN to be efficient for temporal data like Audio.

Project Objectives

- To develop LTC neural network model for audio classification and benchmark against contemporary models,
- To achieve comparable accuracy while using less computational power.

Project Scope

- Classify audio events with Liquid Neural Networks (LNN)
- Achieve high accuracy in real-time sound recognition
- Benchmark performance on multiple audio dataset for optimal results
- Revolutionize speech, environmental sound, and anomaly detection
- Optimize network architecture for diverse audio applications

Project Applications

- Voice Authentication and Security
- Medical Data Analysis
- Abnormality Detection
- Enhanced Music Recommendation Systems
- Audio Content Filtering and Moderation
- Interactive Gaming and Virtual Reality
- Speech Emotion Recognition

Methodology - [1]
Liquid Neural Network

- Traditional RNNs, face challenges in adapting to complex time-series dynamics.
- LNNs, still making use of recurrent mechanics, explicitly model time-series dynamics through differential equations that determine neuron states.

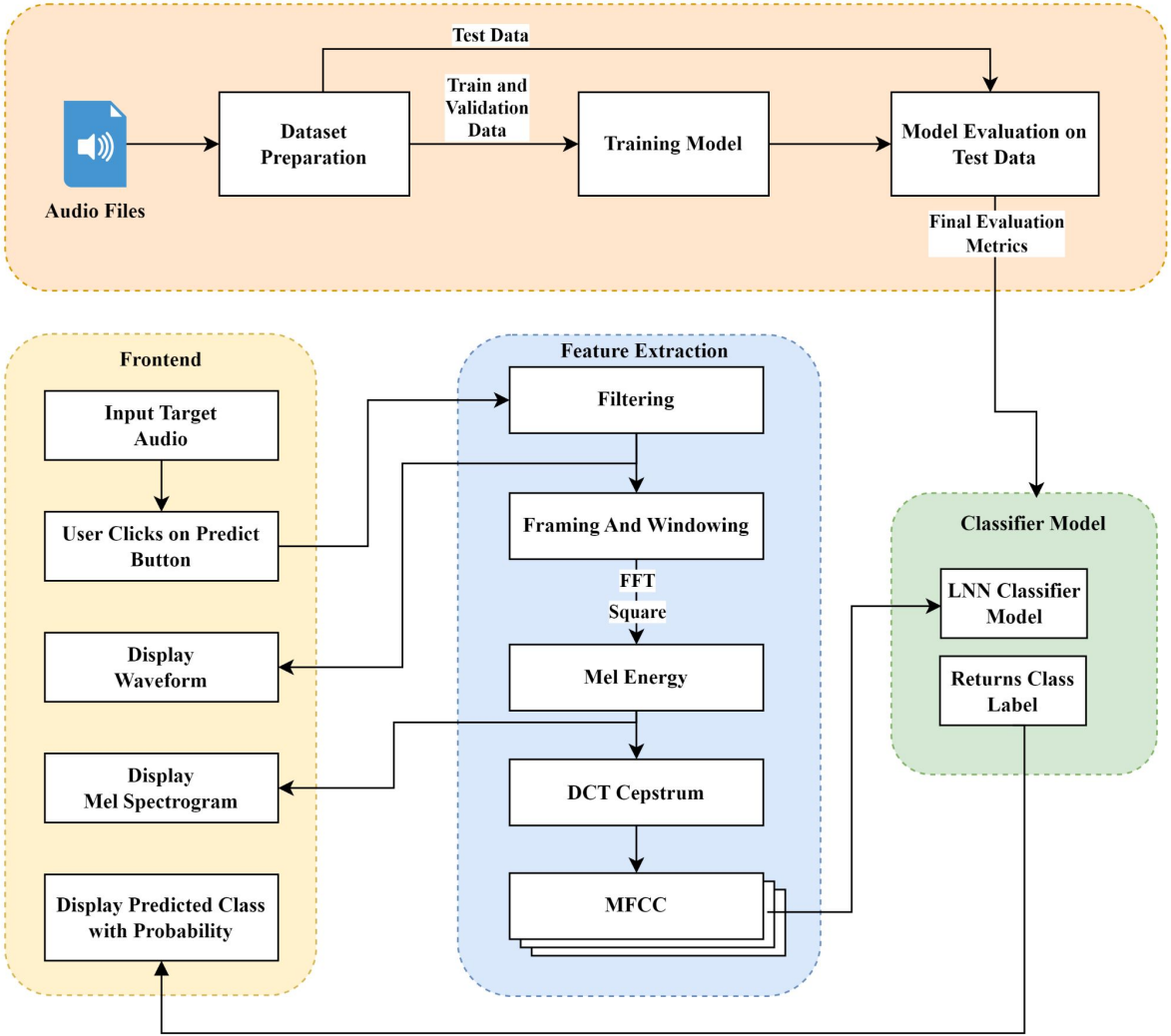
Methodology - [2]
LNN Mathematical Formulation

- Neuron's state is the solution to the differential equation

$$\frac{d\mathbf{x}(t)}{dt} = -\left[\frac{1}{\tau} + f(\mathbf{x}(t), \mathbf{I}(t), \boldsymbol{\theta})\right]\mathbf{x}(t) + f(\mathbf{x}(t), \mathbf{I}(t), \boldsymbol{\theta})\mathbf{A} \quad (1)$$

- Where,
  - *x(t)* is the hidden state
  - *I(t)* is the input
  - τ, time constant is a constant that ensures numerical stability

Methodology-[3] System Architecture



Methodology-[4]
Dataset Exploration

1. **VGG (Visual Geometry Group)**

- Audio-visual dataset with 210,000 data with 310 audio classes,
- Classes like wind noise, sliding door, car, train, etc.
- 10-second audio clips,
- Roughly 200 audio per each class,
- Used by **MirasolI3B** has **69.8%** accuracy on this dataset.

Methodology-[5]
Dataset Exploration

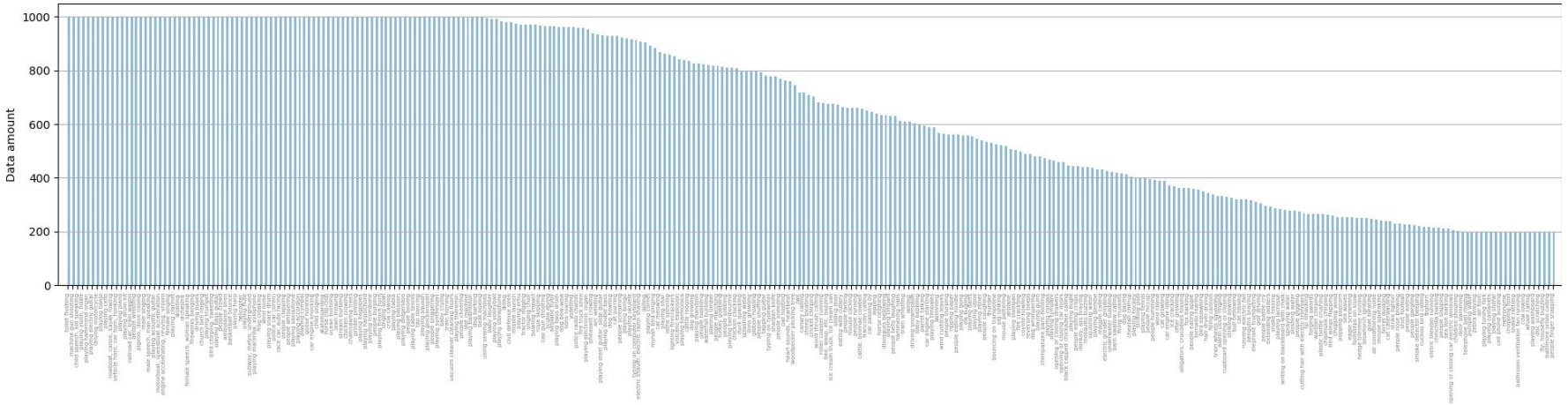


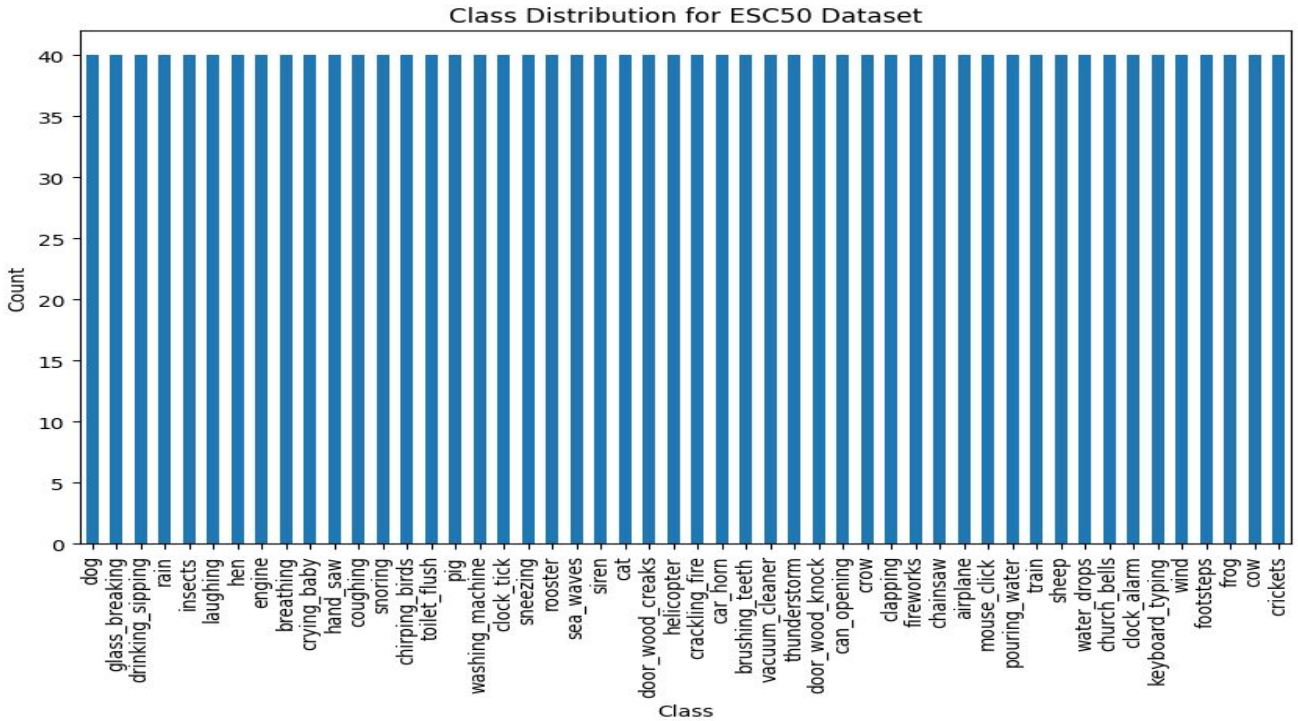
Fig: Dataset distribution for VGG Dataset

Methodology-[6]
Dataset Exploration

2. ESC-50

- 2000 labelled environmental audio recordings,
- Each clip of 5 seconds, covering 50 distinct classes,
- Includes classes like animals, water sound, natural soundscapes, etc.
- Pre-arranged in 5-folds,
- Used by **OmniVec-2 Model** with **99.1%** accuracy.

Methodology-[7]
Dataset Exploration

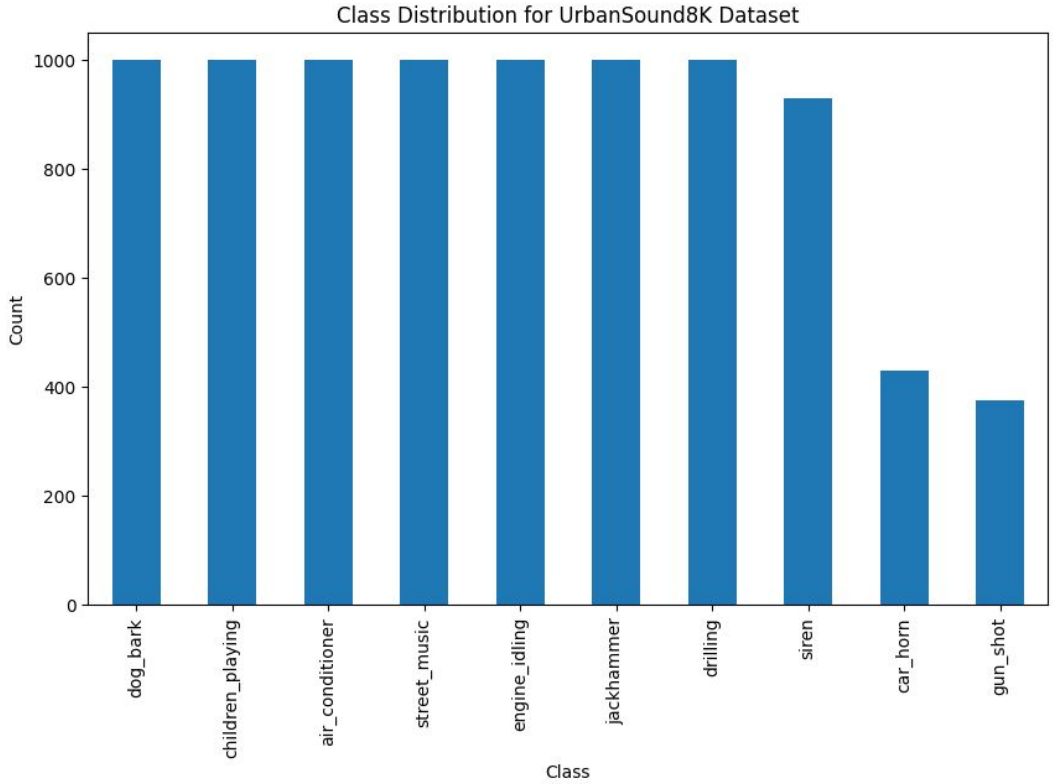


Methodology-[8]
Dataset Exploration

3. UrbanSound8K

- Comprising 8,732 labelled sound excerpts,
- Each clip of 4 seconds, with total 27 hours of audio,
- Includes classes like air conditioner, car horn, children playing, dog bark, etc,
- Used by **ASM-RH-I** with **97.96%** accuracy (10-fold).

Methodology-[9] Dataset Exploration

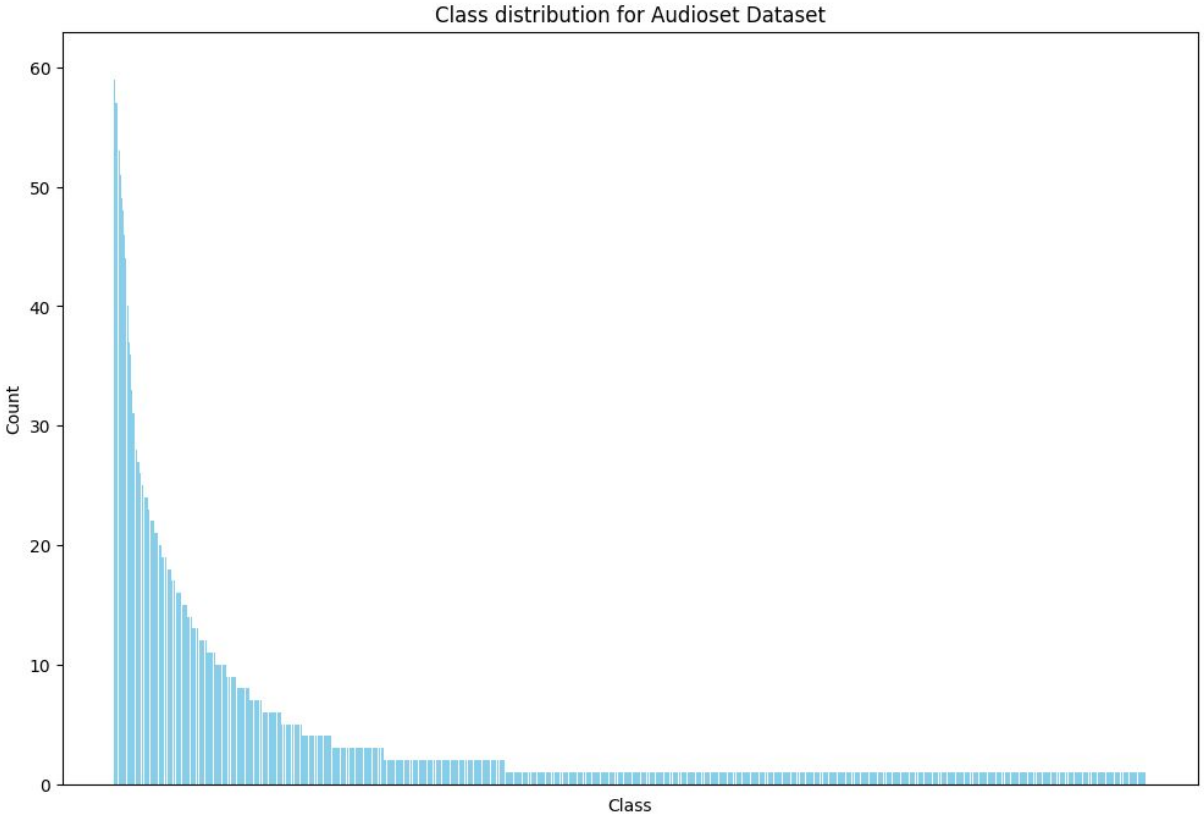


Methodology-[10]
Dataset Exploration

4. AudioSet

- 2,084,320 YouTube videos containing 527 labels,
- 10-second sound clips sourced from YouTube videos and labelled by humans,
- Includes classes like music, speech, vehicle, car, etc.,
- Used by **OmniVec** with **0.548 mAP**.

Methodology-[11] Dataset Exploration



Methodology-[12] Evaluation Metrics

- **F1-Score**
  - Harmonic mean of precision and recall.

$$F1Score = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}$$

Methodology-[13]
Evaluation Metrics

- **Accuracy**

- measures the proportion of correct predictions made by the model out of all predictions.

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{All Predictions}}$$

Methodology-[14]
Evaluation Metrics

- **Mean Average Precision (mAP)**
  - **Average Precision** is the area under the precision-recall curve for a single query or class.

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i$$

Methodology-[15] Instrumentation

1. **Kaggle Notebook**
   - Kaggle Notebooks are essentially Jupyter Notebooks hosted on the cloud
   - Provides 4 CPU cores, 20GB of RAM, and 1 x Nvidia Tesla P100 GPU with 4 cores and 29 GB of RAM,
   - GPU can be used for 30 hours a week and 9 hours per session.

Methodology-[16] Instrumentation

2. Google Colaboratory

- Provides an Intel Xeon CPU with 2 vCPUs (virtual CPUs) and 13 GB of RAM,
- NVIDIA Tesla K80 with 12GB of VRAM (Video Random-Access Memory)

Methodology-[17] Instrumentation

3. Librosa

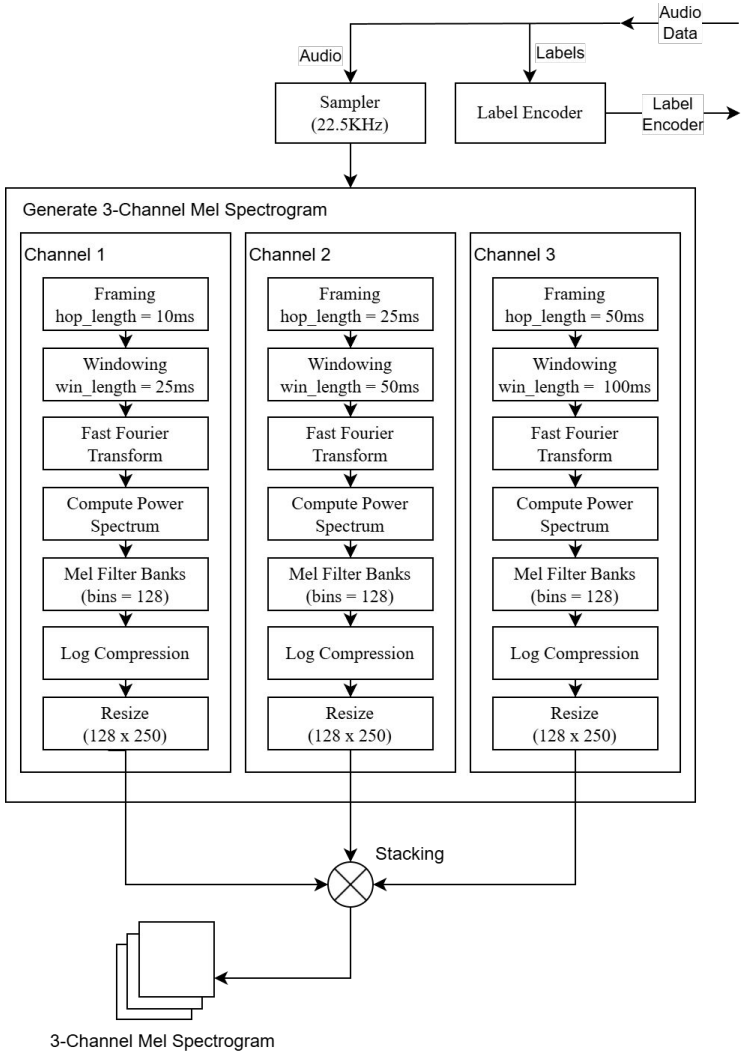
- Python package for music and audio analysis,
- Calculation of time domain features like Zero-crossing rate,
- Calculation of frequency domain features.

Methodology-[18] Instrumentation

4. Pytorch

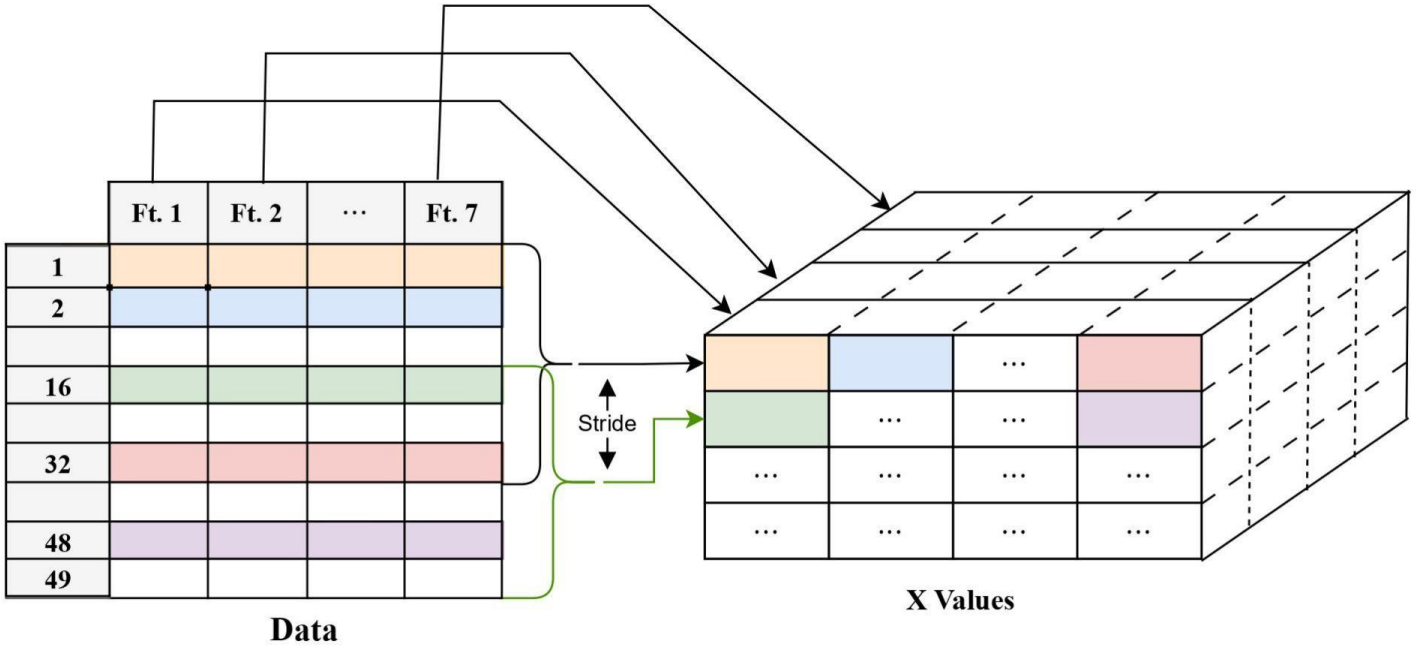
- Open-source deep learning framework developed by Facebook's AI Research lab,
- Uses Dynamic Computation Graph,
- Rich ecosystem and community support.

Implementation-[1]
Data PreProcessing for CNN



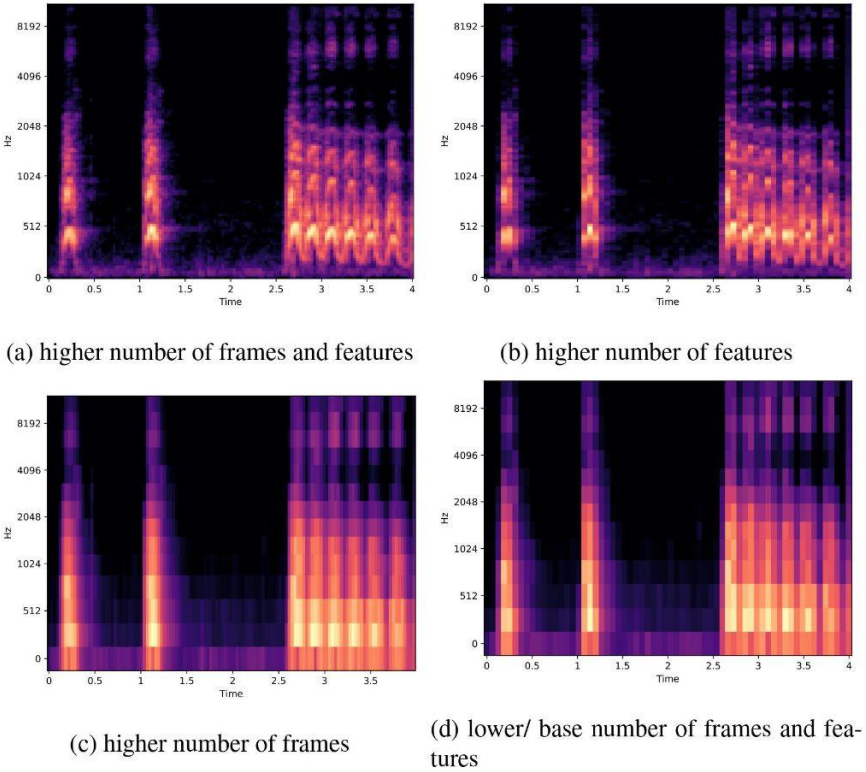
Implementation - [2]

Data PreProcessing For LTC with sequencing



Implementation - [3]

Data PreProcessing For LTC with sequencing



(a) higher number of frames and features

(b) higher number of features

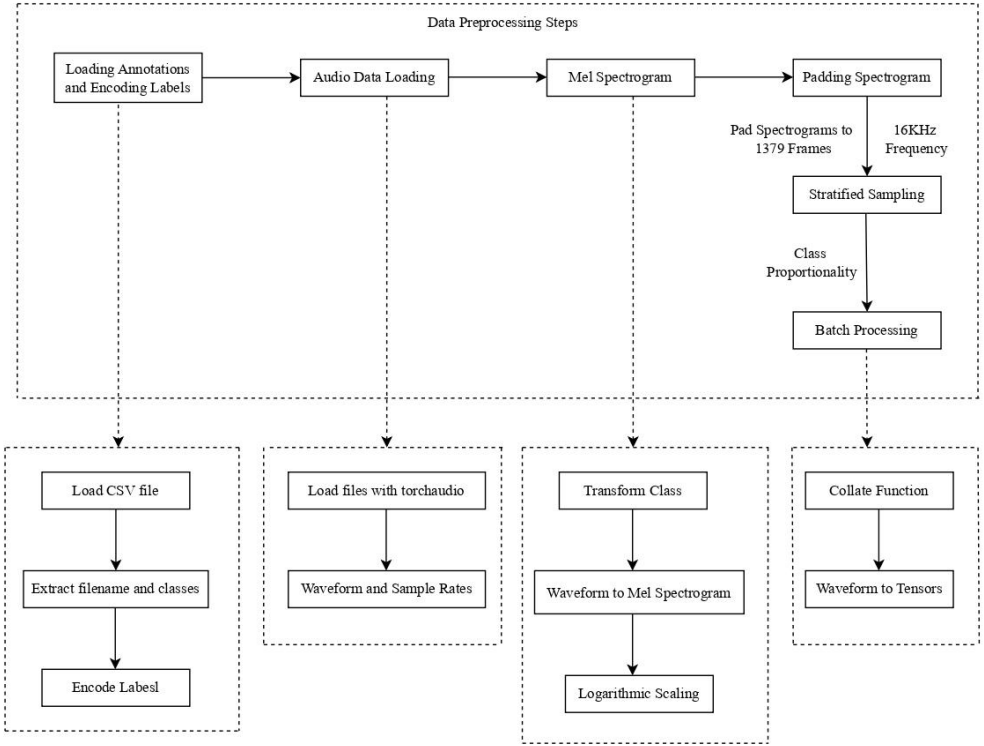
(c) higher number of frames

(d) lower/ base number of frames and features

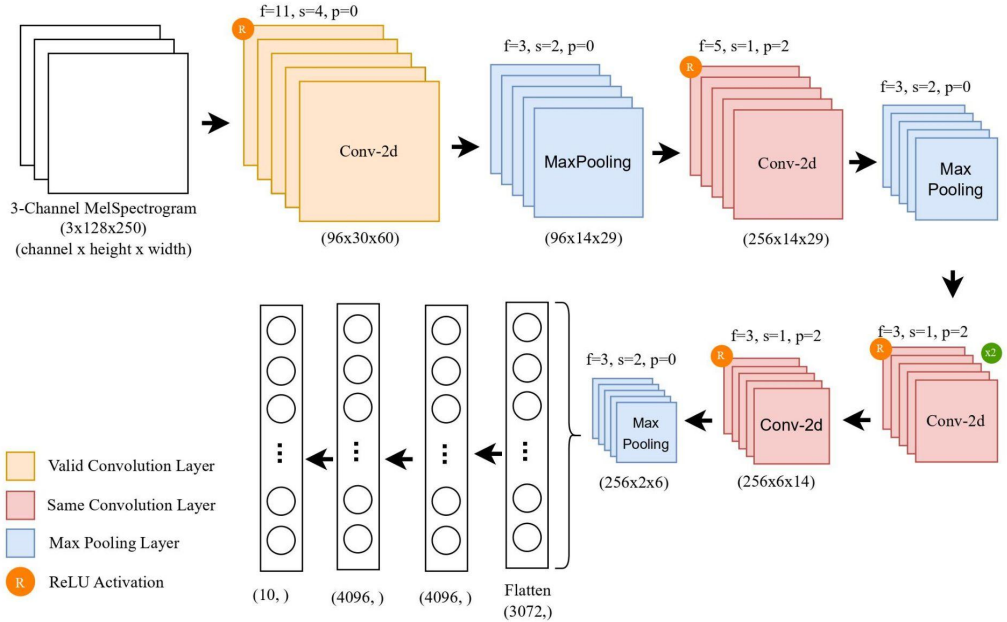
Fig: Impact of Mel bins and hop length on Mel-Spectrogram

Implementation - [4]

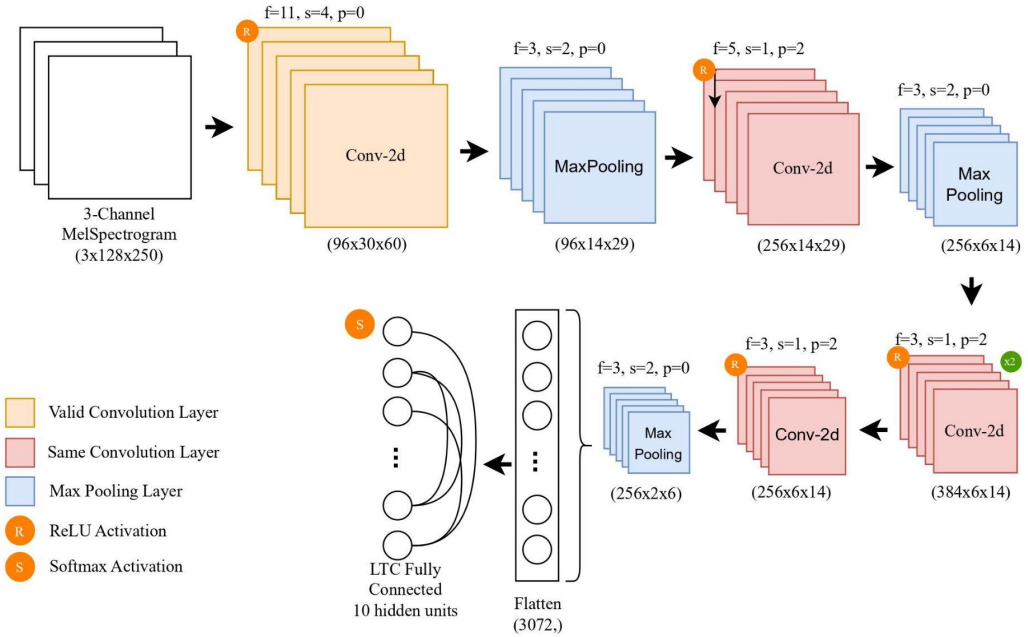
Data PreProcessing for LTC without sequencing



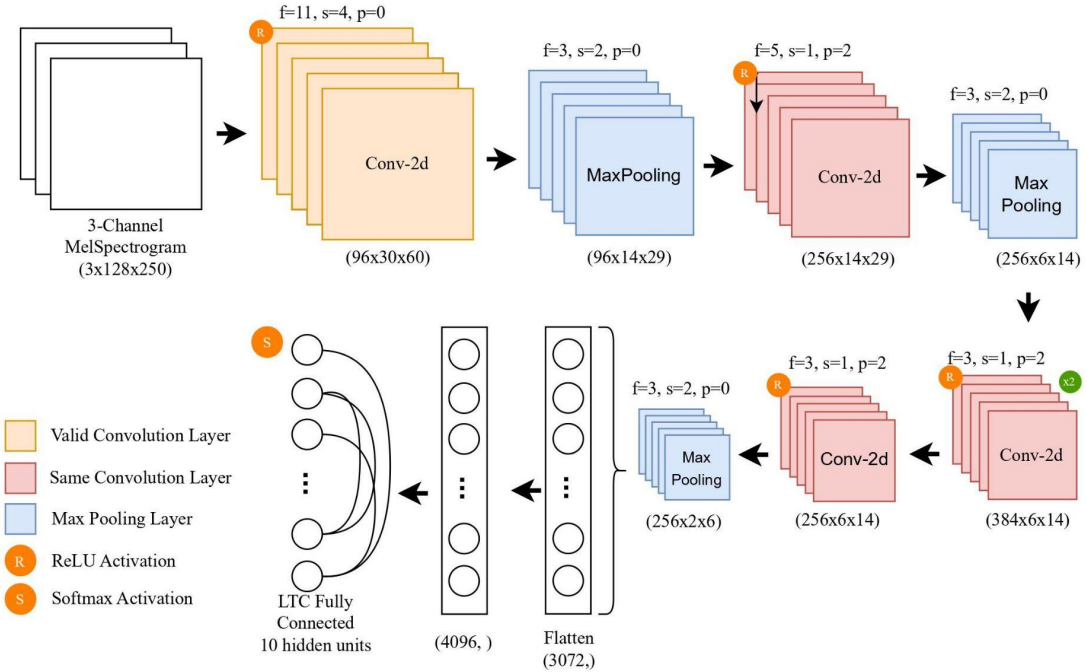
Implementation - [5] AlexNet Architecture



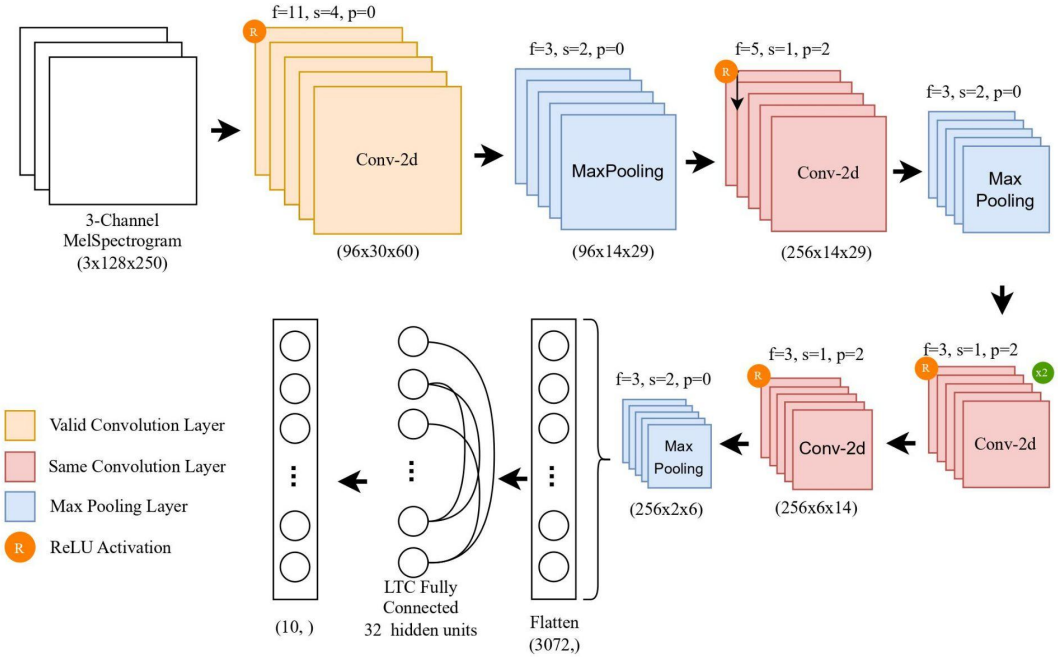
Implementation - [6]
AlexNet LTC Architecture - 1



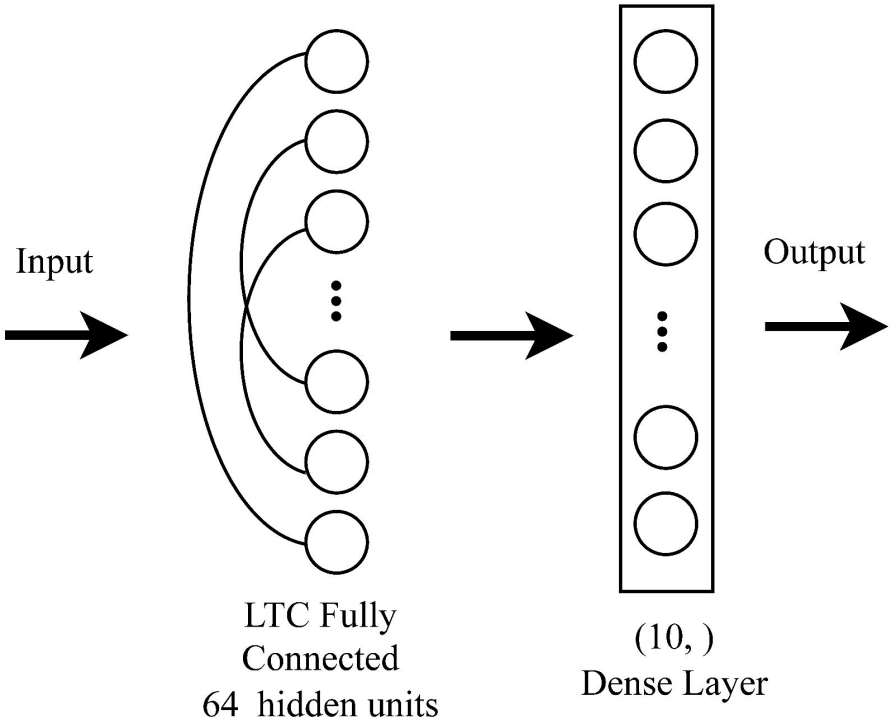
Implementation - [7]
AlexNet LTC Architecture - 2



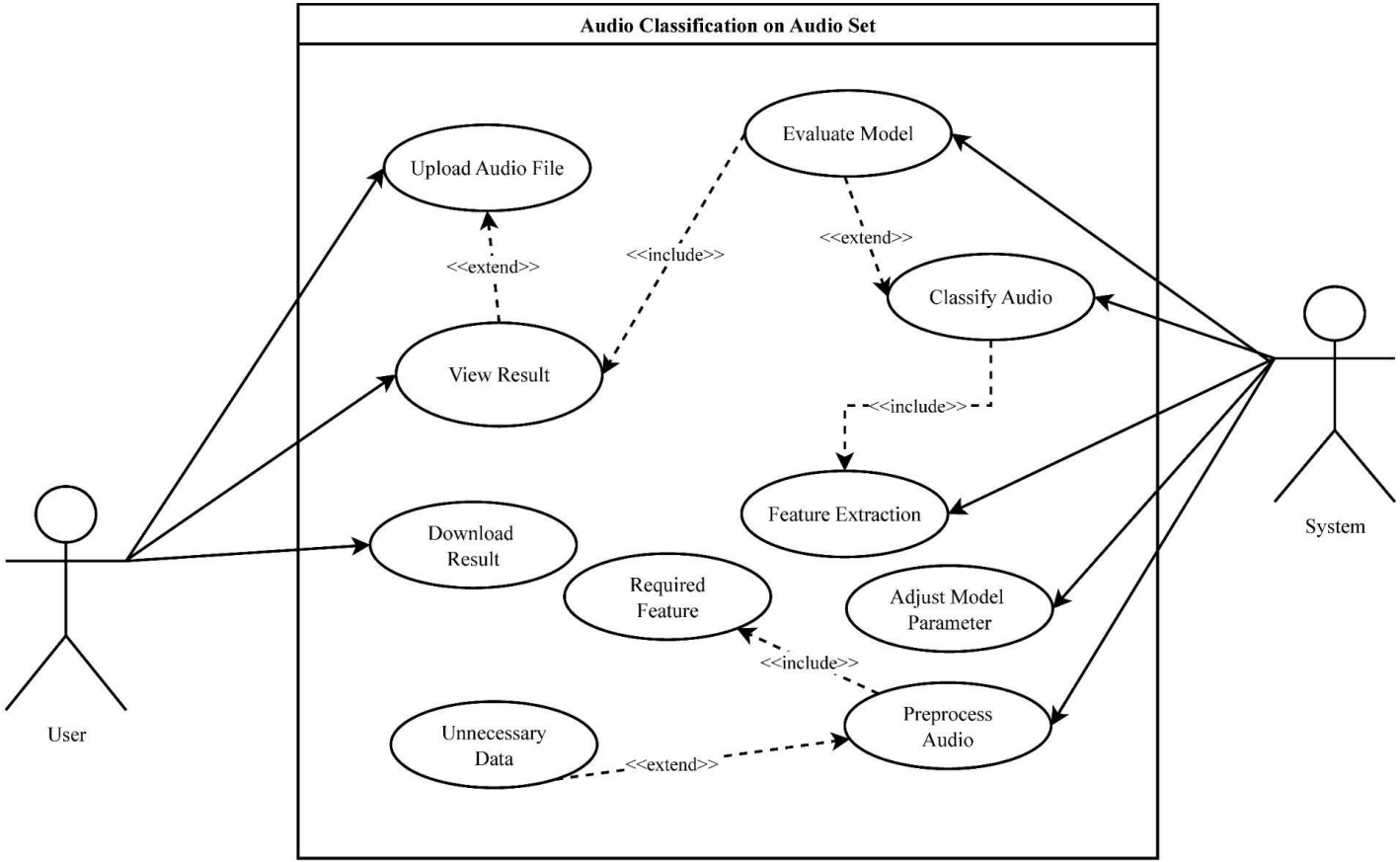
Implementation - [8]
AlexNet LTC Architecture - 3



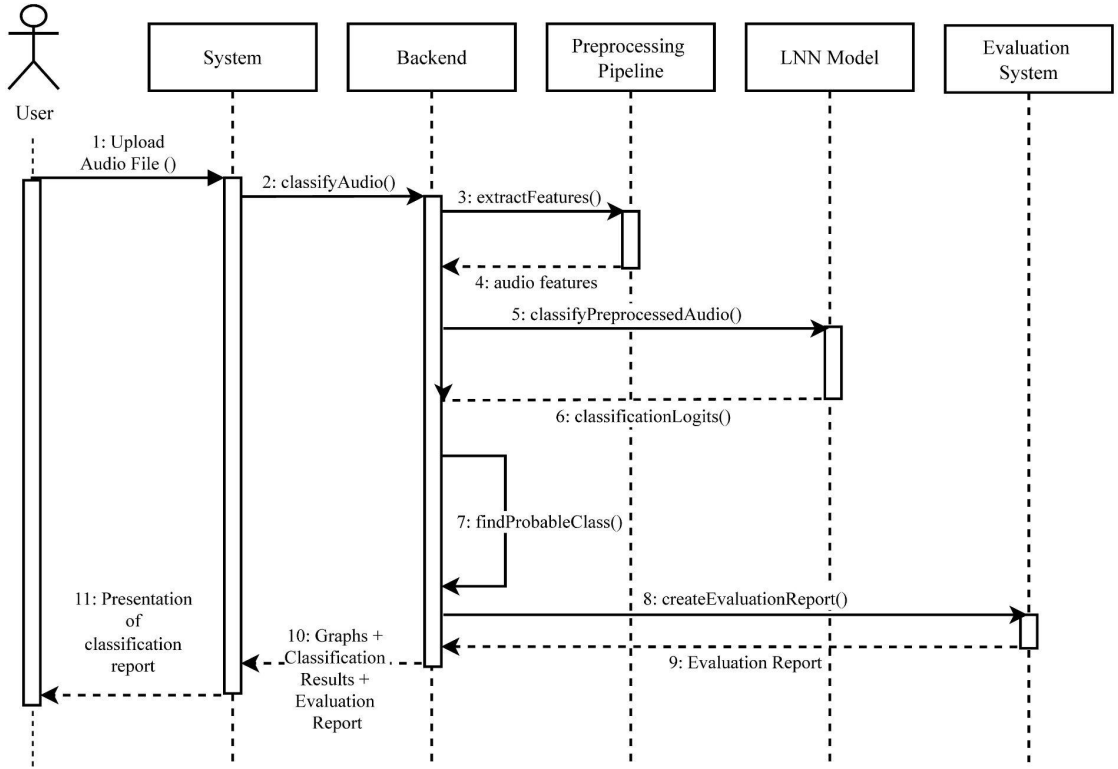
Implementation - [9] Our Architecture



Implementation - [10]



Implementation - [11] Sequence Diagram



Results - [1]

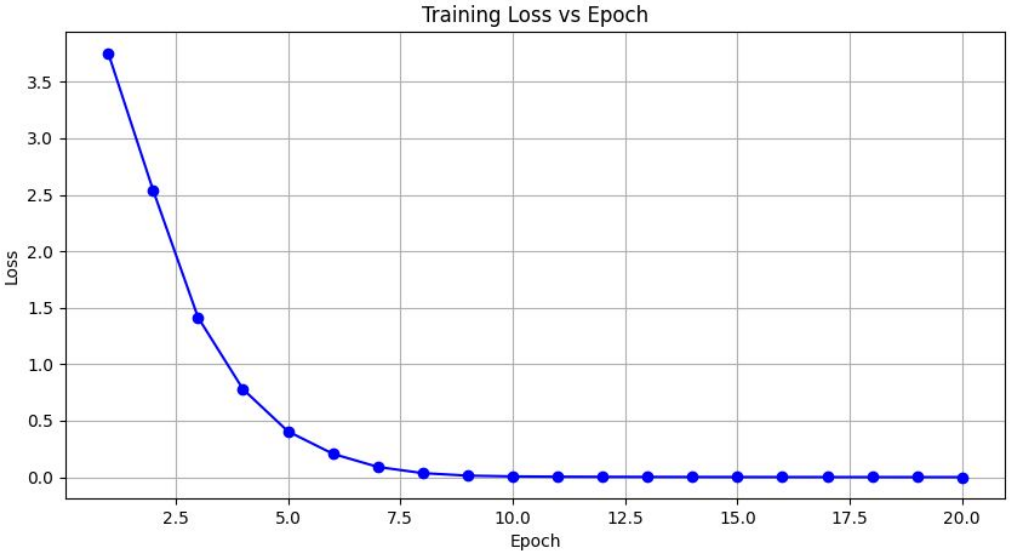
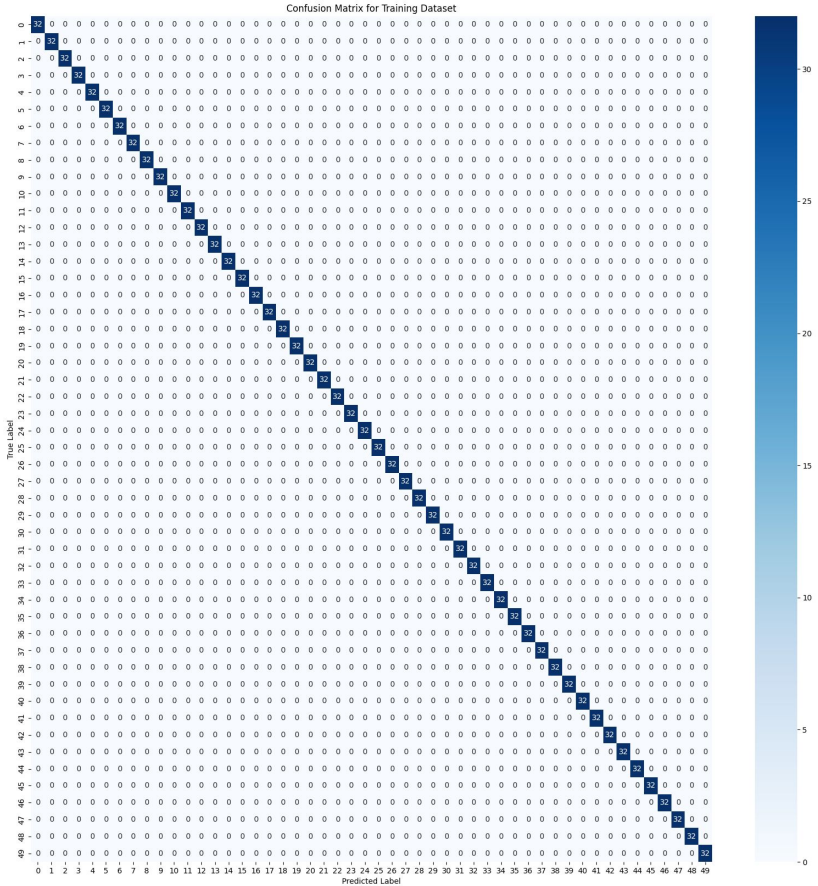


Fig: Training Loss Vs Epoch for LTC without sequencing

Results - [2]

Fig: Confusion Matrix for Training Dataset



Results - [3]

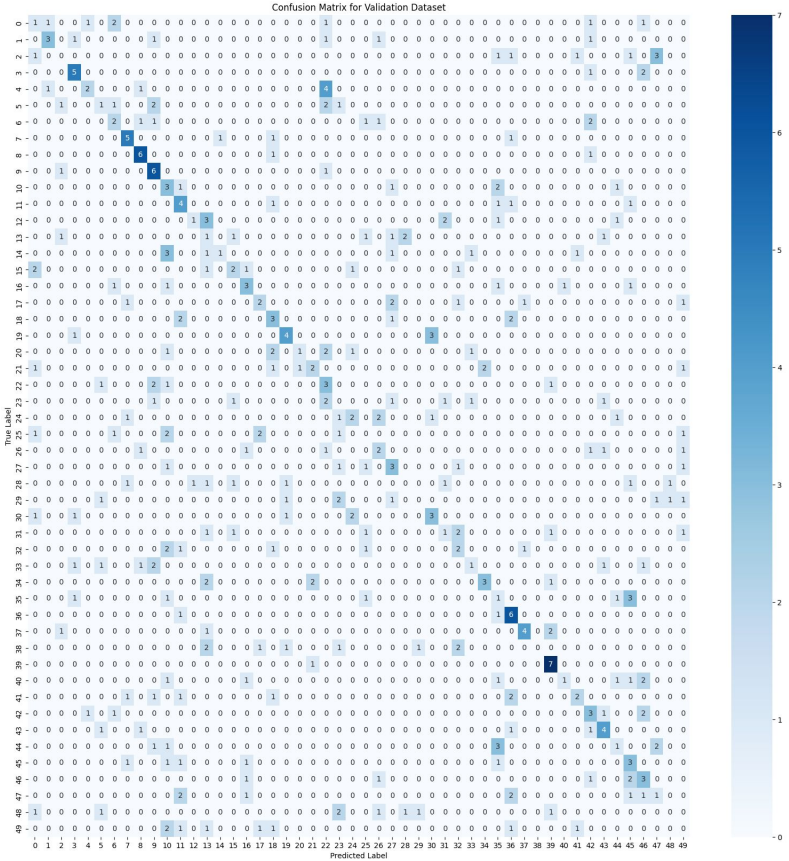


Fig: Confusion Matrix for Validation Dataset

Results - [4]

	Precision	Recall	F1 Score	Support
Accuracy			0.31	400
Macro Avg	0.32	0.31	0.29	400
Weighted Avg	0.32	0.31	0.29	400

Fig: Classification Report for LTC without sequencing

Results - [5]

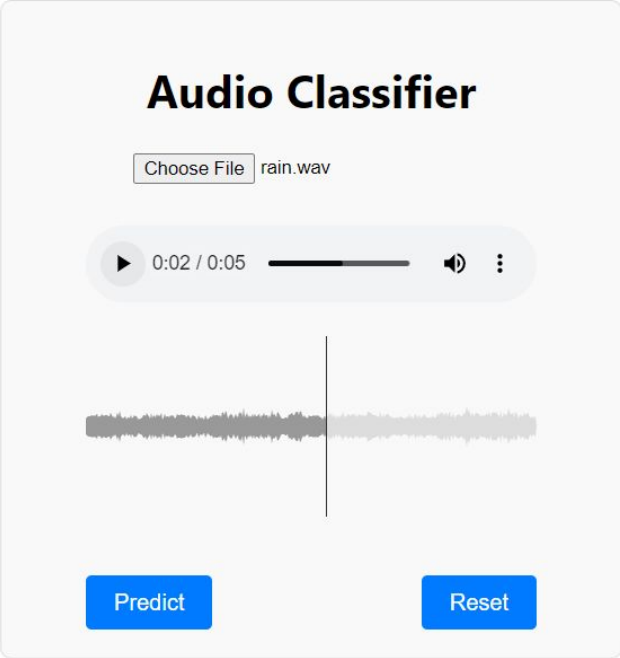
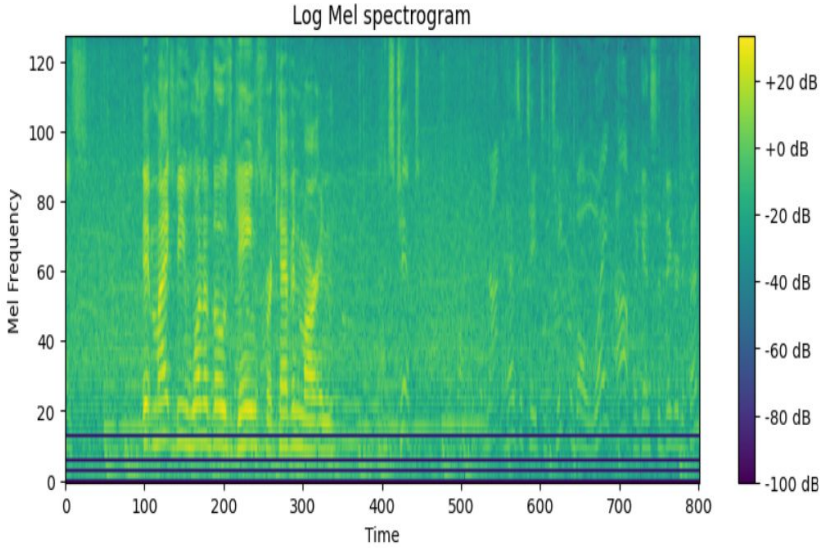


Fig: Web application interface

Results - [6]



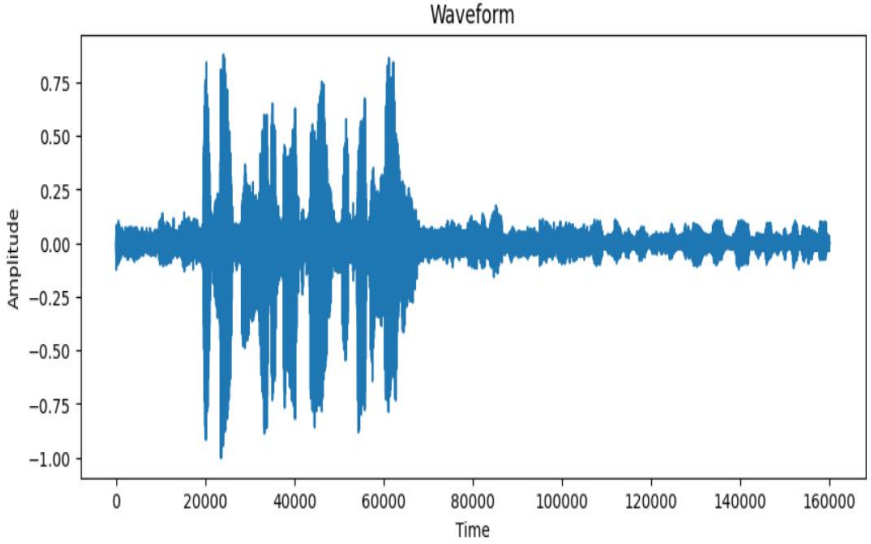


Fig: Web application interface

Results - [7]

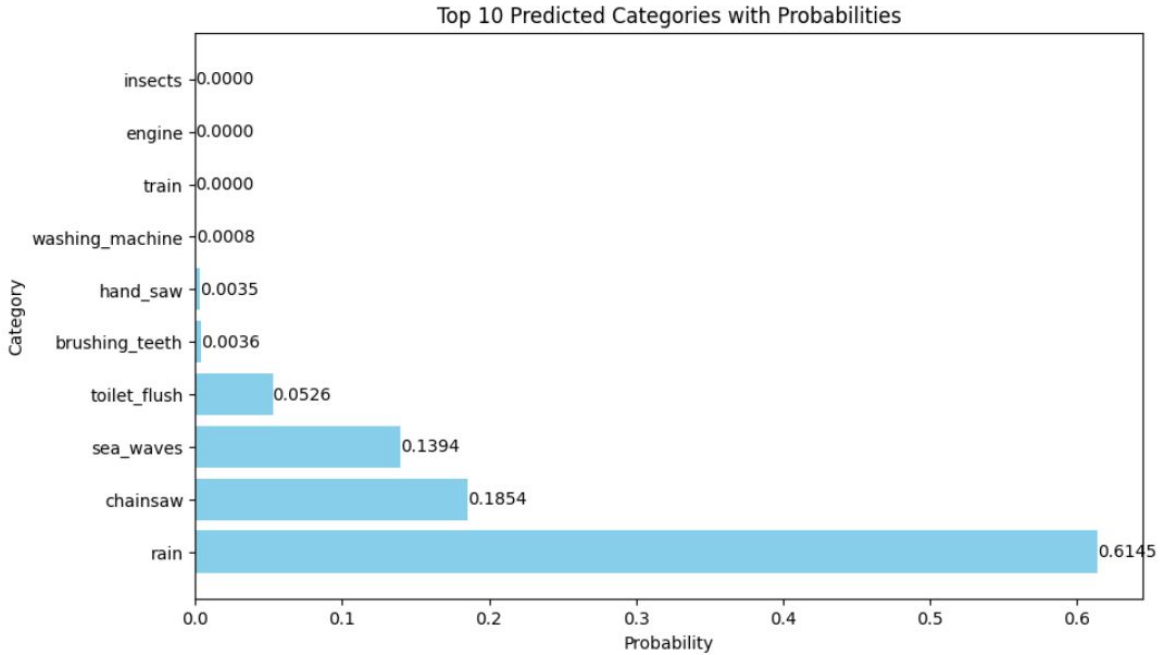


Fig: Web application interface

Results - [8]

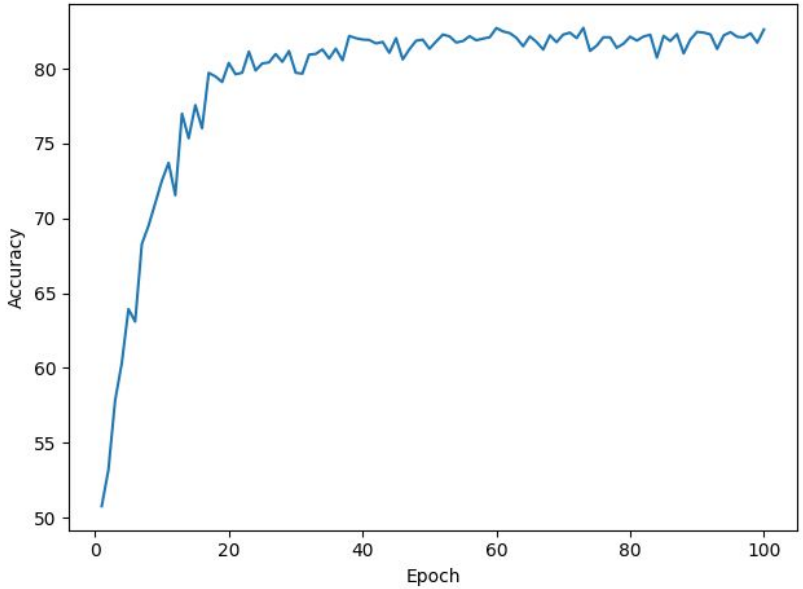


Fig: Accuracy vs. epoch plot on CIFAR dataset

Results - [9]

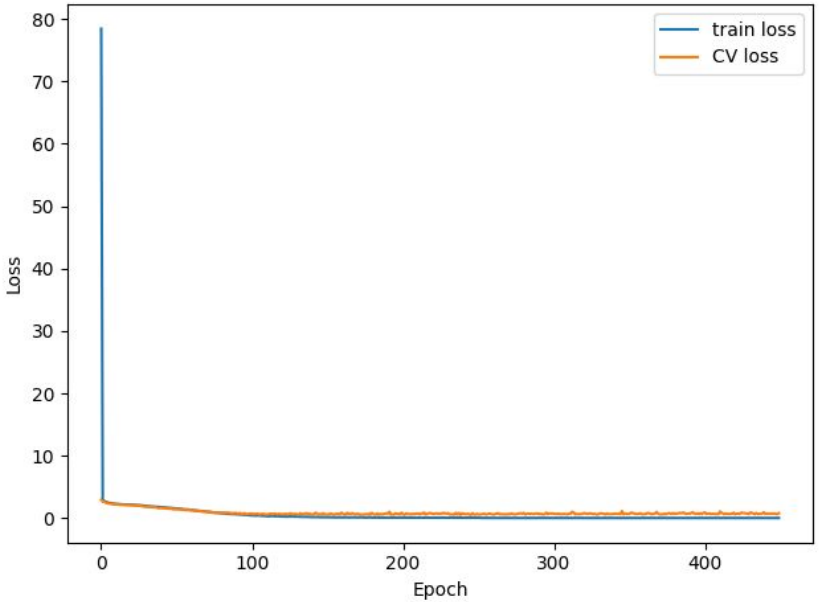


Fig: Accuracy vs. epoch plot on UrbanSound8K dataset with AlexNet architecture (baseline)

Results - [10]

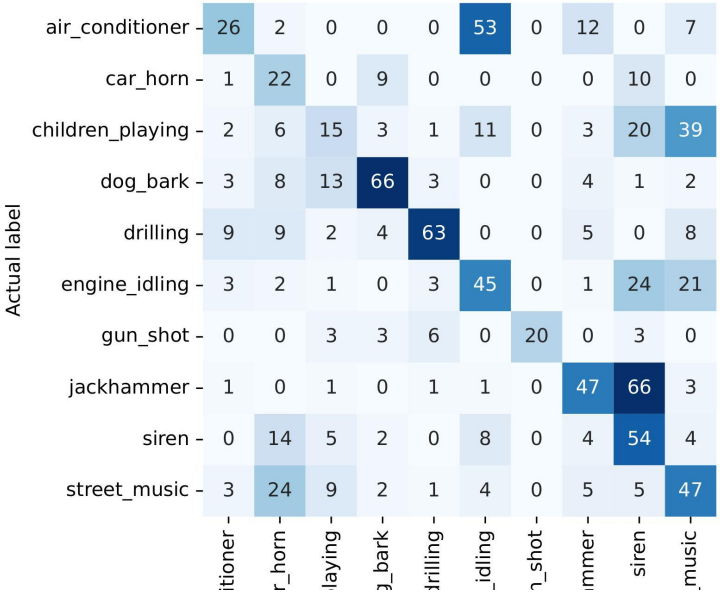


Fig: Baseline model confusion matrix

Results - [11]

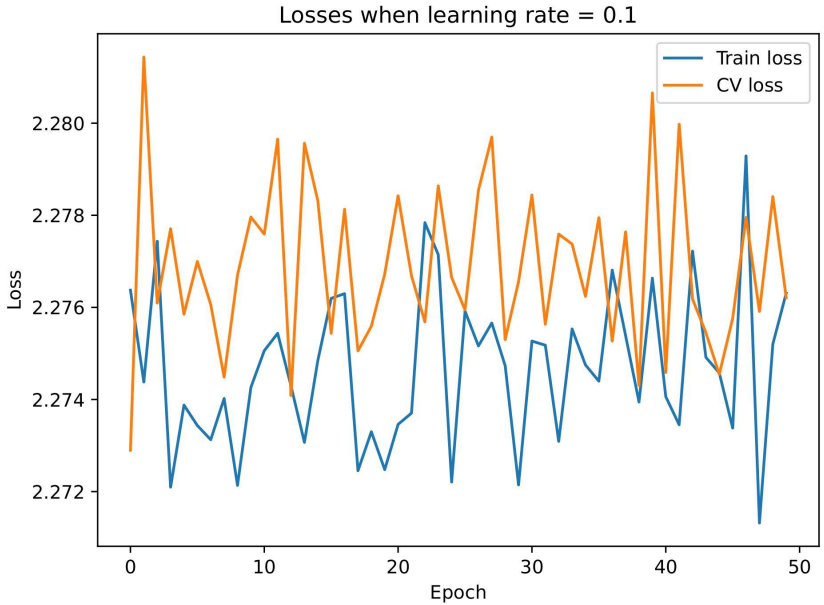


Fig: Loss vs. epoch plot in Hybrid CNN + LTC model (learning rate = 0.1)

Results - [12]

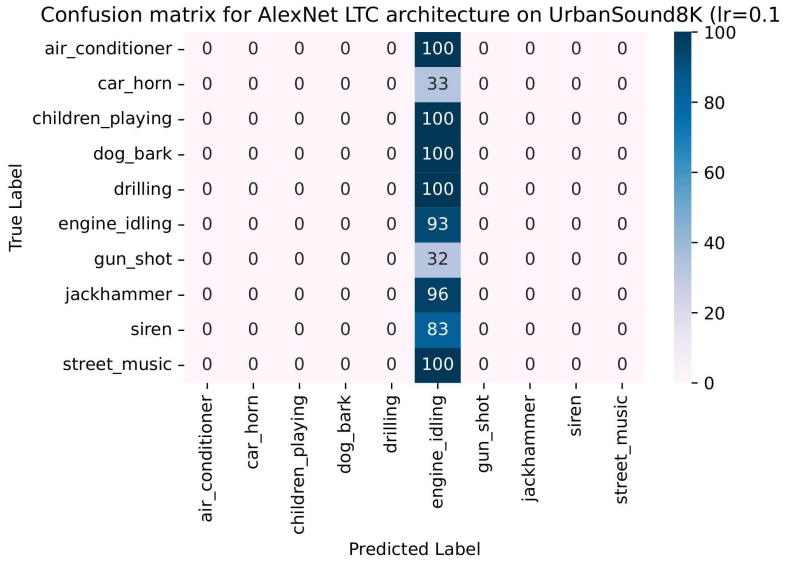


Fig: Confusion matrix of Hybrid CNN + LTC model (learning rate = 0.1)

Results - [13]

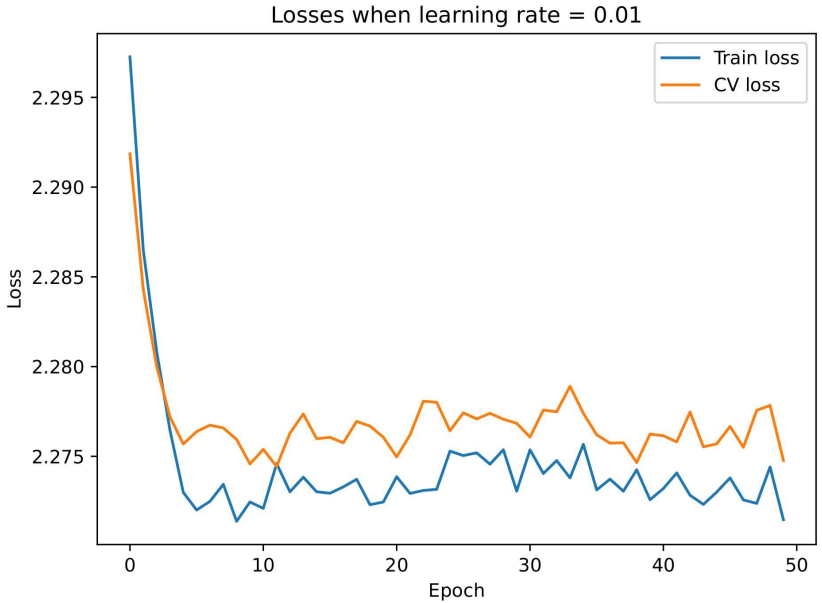


Fig: Loss vs. epoch plot in Hybrid CNN + LTC model (learning rate = 0.01)

Results - [14]

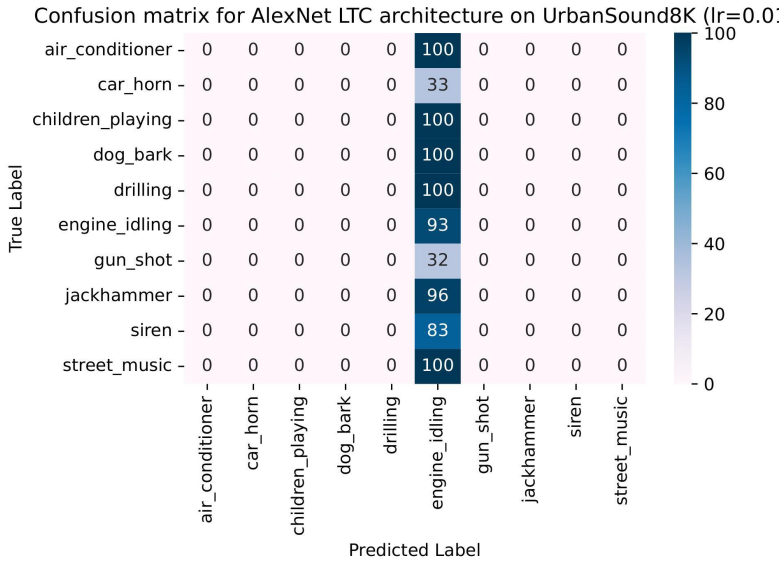


Fig: Confusion matrix of Hybrid CNN + LTC model (learning rate = 0.01)

Results - [15]

	Accuracy	F1 Macro	F1 Micro	mAP
CNN (AlexNet)	**71.08%**	0.7248	0.7109	-
CNN + LTC	11.11%	0.02000	0.1111	0.1000
CNN + Dense + LTC	11.11%	0.02000	0.1111	0.1000
CNN + LTC + Dense	11.11%	0.02000	0.1111	0.0312

Fig: Evaluation metrics for different hybrid architecture

Results - [16]

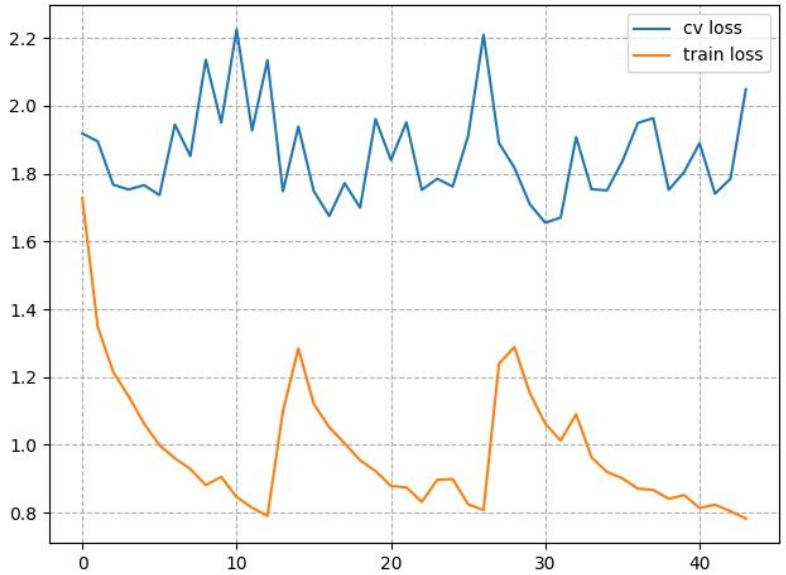


Fig: Losses Vs. Epoch in LTC model

Results - [17]

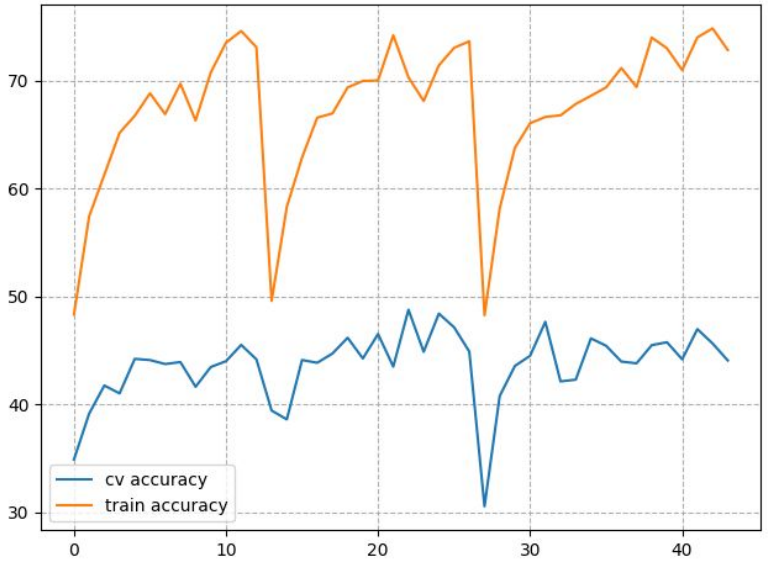


Fig: Accuracy Vs. Epoch in LTC model

Results - [18]

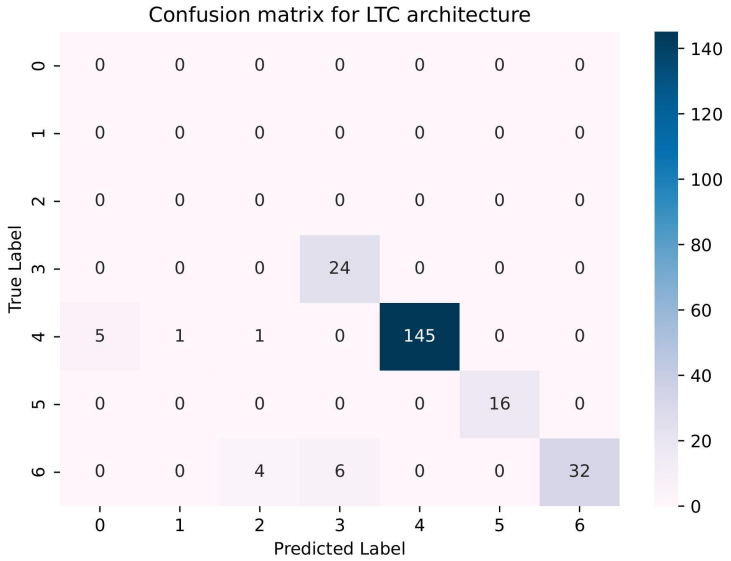


Fig: Confusion matrix of LTC model

Discussion of Results - [1]

LTC without sequencing

- **Overfitting:** Training accuracy at 100%, but validation accuracy only 30.75%, indicating poor generalization.
- **Variable Performance:** Precision, recall, and F1-scores vary widely across classes, showing inconsistent performance.
- **Low mAP:** Mean Average Precision (mAP) of 0.14, indicating poor class ranking and classification.

Discussion of Results - [2]

- LTC model with sequencing achieved **92.73% accuracy**, 0.5329 macro F1 score and 0.9274 micro F1 score,
- Parameter count of model was **25,958** (31M - AlexNet on Urban Sound 8K),
- Need of either early stopping or, solution of overfitting is seen,
- Stratified sampling should be considered for splitting dataset.

Remaining Tasks

- More experiments in LTC architecture
- Benchmarking of LNN in All the Dataset
- Exploration of different Liquid Network architectures
- Benchmarking of LNN in All the Dataset

Reference - [1]

[1] R. Hasani, M. Lechner, A. Amini, D. Rus, and R. Grosu, “Liquid time-constant networks,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 9, 7657–7666, May 2021. DOI: 10.1609/aaai.v35i9.16936. https://ojs.aaai. org/index.php/AAAI/article/view/16936.

[2] S. Srivastava and G. Sharma, Omnivec: Learning robust representations with cross modal sharing, 2023. arXiv: 2311.05709 [cs.CV].

Reference - [2]

[3] M. Chahine, R. Hasani, P. Kao, et al., “Robust flight navigation out of distri bution with liquid neural networks,” Science Robotics, vol. 8, no. 77, eadc8892, 2023, Published online 2023 Apr 19, ISSN: 2470-9476.

[4] H. Ju, J.-X. Xu, and A. M. VanDongen, “Classification of musical styles using liquid state machines,” in The 2010 International Joint Conference on Neural Networks (IJCNN), 2010, 1–7.