AT82.10 Recent Trends In Machine Learning

QUACLRS: QUasi-supervised Audio Classification by Learning Representations from Spectrograms

RTML Project | Presentation

Under the guidance of: Dr. Chaklam Silpasuwanchai

Team Members:

Arya Shah; st125462
Aman Oberoi; st125490

AGENDA

01 ABSTRACT

Problem Statement
Motivation
Approach

02 RESEARCH QUESTIONS

03 DATA

Original UrbanSound8K Dataset
Dataset Extension Process
Final Dataset Structure
Storage & Organization
Augmentation Impact and Analysis
Visualization Examples

04 METHODS/APPROACHES

Audio Representation
CNN Backbone Architectures
Self-Supervised Learning Techniques
Data Augmentation Strategies
Training Methodology
Evaluation Framework

05 RESULTS & ABLATIONS

06 DISCUSSION

Addressing Research Questions
Broader Implications

01

Abstract

Problem Statement

Environmental sound classification faces critical challenges including limited labeled data, restricted class diversity, and inconsistent performance in noisy environments. Current audio classification systems often struggle with generalizability across diverse acoustic environments and require significant amounts of labeled data to achieve acceptable performance. In particular, common benchmark datasets such as UrbanSound8K contain limited class diversity, constraining the development of robust audio classification models.

Motivation

The ability to accurately classify environmental sounds has significant applications across numerous domains including urban monitoring, security surveillance, assistive technologies, and multimedia content analysis. With the proliferation of audio-enabled IoT devices and smart city initiatives, robust audio classification systems that can operate effectively with limited labeled data and in diverse acoustic environments are increasingly important. Recent advances in self-supervised learning techniques have shown promise in vision domains but remain underexplored for audio classification tasks.

Approach

This research introduces a comprehensive framework that combines:

1. An extended version of the UrbanSound8K dataset with additional urban sound classes (ambulance, firetruck, police, traffic) and specialized audio augmentation techniques (time stretch, pitch shift, SpecAugment, PatchAugment)

2. A modular CNN architecture for spectral feature extraction, exploring various backbones (ResNet, MobileNet, EfficientNet, ConvNext)

3. Self-supervised learning techniques adapted for spectrograms, including contrastive methods (SimCLR, MoCo, BYOL) and non-contrastive approaches (DINO, BarlowTwins)

4. A systematic evaluation methodology using k-fold cross-validation to assess model robustness and generalizability

Extended Dataset
Enhanced urban sound classes and augmentation techniques
Modular CNN Architecture
Exploration of various backbones for feature extraction
Self-Supervised Learning Techniques
Adapted methods for spectrograms
Evaluation Methodology
Systematic k-fold cross-validation

02

Research Questions

Some of the crucial Research Questions we could identify with respect to our research project are as follows:

RQ 1: How can cnn backbone architectures and self-supervised learning techniques improve audio event classification performance with limited labeled data?

RQ 2: To what extent can specialized data augmentation techniques enhance the robustness of audio classification models in noisy environments?

RQ 3: How does expanding existing audio datasets with additional diverse classes affect the generalizability and robustness of sound event classification models?

RQ 4: What combination of semi-supervised learning techniques and data augmentation strategies yields the most effective performance improvements for environmental sound classification tasks?

03

Data

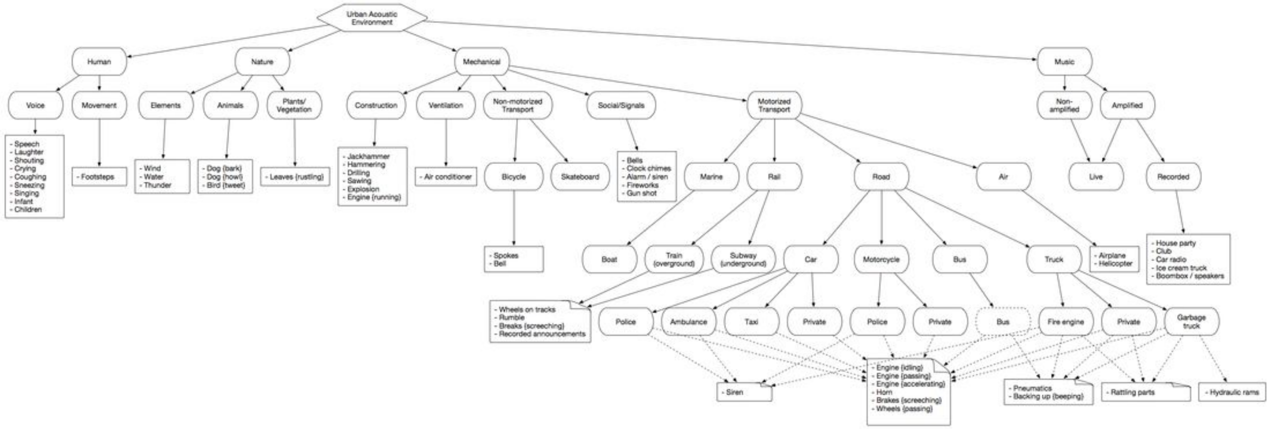
Original UrbanSound8K Dataset

The UrbanSound8K dataset is a widely used benchmark in environmental sound classification research:

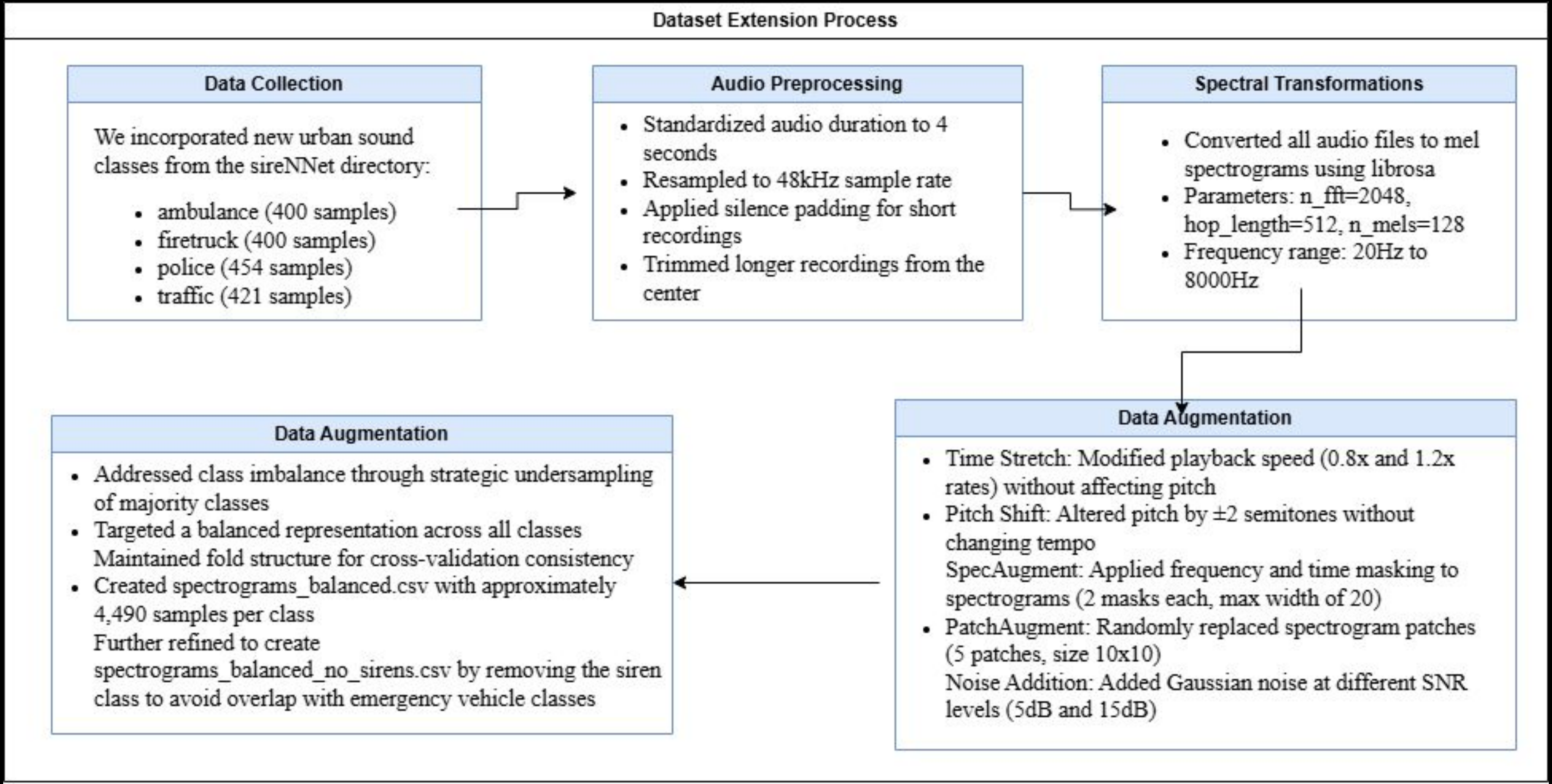
- Contains 8,732 labeled sound excerpts (≤4 seconds) totaling approximately 7.3 hours of audio
- Comprises 10 classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gunshot, jackhammer, siren, and street music
- Organized into 10 pre-defined folds for cross-validation
-

Typical class distribution:

- Most classes have approximately 1,000 samples (air conditioner, children playing, dog bark, drilling, engine idling, jackhammer, street music)
- Some classes have fewer samples: car horn (429), gun shot (374), and siren (929)



Dataset Extension Process



Dataset Extension Process

Data Collection

We incorporated new urban sound classes from the sireNNet directory:

- ambulance (400 samples)
- firetruck (400 samples)
- police (454 samples)
- traffic (421 samples)

Audio Preprocessing

- Standardized audio duration to 4 seconds
- Resampled to 48kHz sample rate
- Applied silence padding for short recordings
- Trimmed longer recordings from the center

Spectral Transformations

- Converted all audio files to mel spectrograms using librosa
- Parameters: n_fft=2048, hop_length=512, n_mels=128
- Frequency range: 20Hz to 8000Hz

Data Augmentation

- Time Stretch: Modified playback speed (0.8x and 1.2x rates) without affecting pitch
- Pitch Shift: Altered pitch by ±2 semitones without changing tempo
  SpecAugment: Applied frequency and time masking to spectrograms (2 masks each, max width of 20)
- PatchAugment: Randomly replaced spectrogram patches (5 patches, size 10x10)
  Noise Addition: Added Gaussian noise at different SNR levels (5dB and 15dB)

Data Augmentation

- Addressed class imbalance through strategic undersampling of majority classes
- Targeted a balanced representation across all classes
  Maintained fold structure for cross-validation consistency
- Created spectrograms_balanced.csv with approximately 4,490 samples per class
  Further refined to create spectrograms_balanced_no_sirens.csv by removing the siren class to avoid overlap with emergency vehicle classes

Final Dataset Structure

The final dataset used for our experiments consists of:

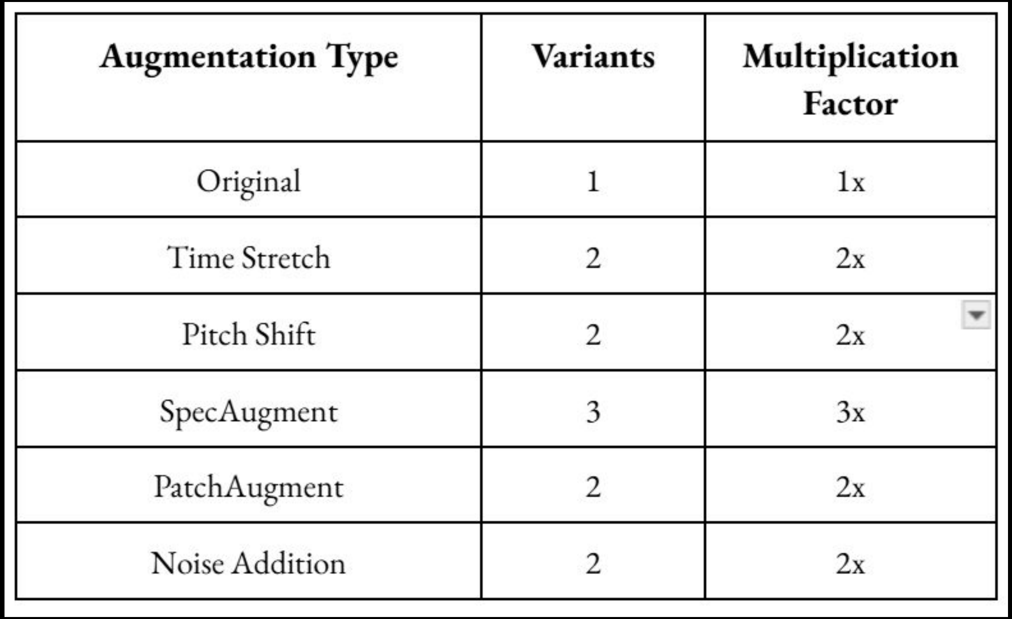
- Total samples: Approximately 58,000 spectrogram images
- Classes: 13 classes (original UrbanSound8K classes except siren, plus 4 new vehicle classes)
- Format: RGB images of mel spectrograms (224x224 pixels)
- Fold structure: Preserves the original 10-fold cross-validation structure
- Metadata: Comprehensive CSV file (spectrograms_balanced_no_sirens.csv) containing:
  - spec_file_name: Spectrogram image filename
  - orig_file_name: Original audio filename
  - fsID: File slice identifier
  - fold: Fold assignment (1-10)
  - classID: Numeric class identifier
  - class: Class name
  - augmentation: Augmentation technique applied (original, time_stretch, pitch_shift, spec_augment, patch_augment, noise)

Storage & Organization

The dataset is organized as follows:

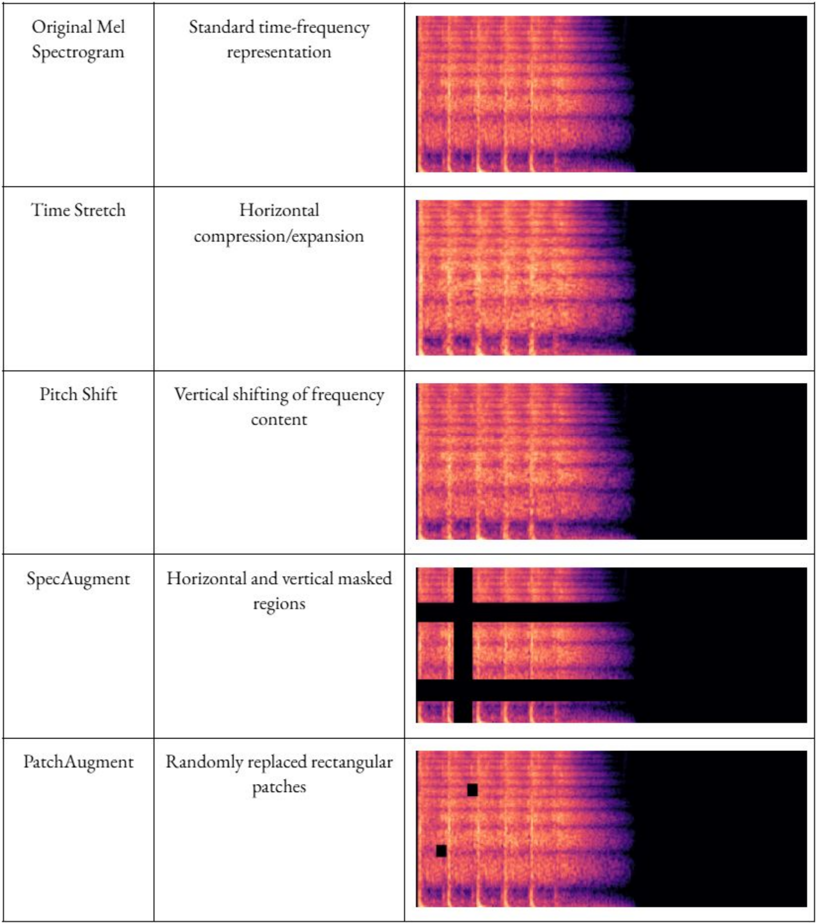
Spectrogram Images: Stored in the spectrograms directory
Sub-directories for each fold (fold1 through fold10)
Images named according to original file ID and augmentation type
Metadata Files:
spectrograms.csv: Complete dataset with all generated spectrograms
spectrograms_balanced.csv: Class-balanced version of the dataset
spectrograms_balanced_no_sirens.csv: Final version used for experiments, with siren class removed

Augmentation Impact Analysis



Augmentation Type	Variants	Multiplication Factor
Original	1	1x
Time Stretch	2	2x
Pitch Shift	2	2x
SpecAugment	3	3x
PatchAugment	2	2x
Noise Addition	2	2x

Visualization Examples



Original Mel Spectrogram	Standard time-frequency representation	
Time Stretch	Horizontal compression/expansion	
Pitch Shift	Vertical shifting of frequency content	
SpecAugment	Horizontal and vertical masked regions	
PatchAugment	Randomly replaced rectangular patches	

04

Methods/Approaches

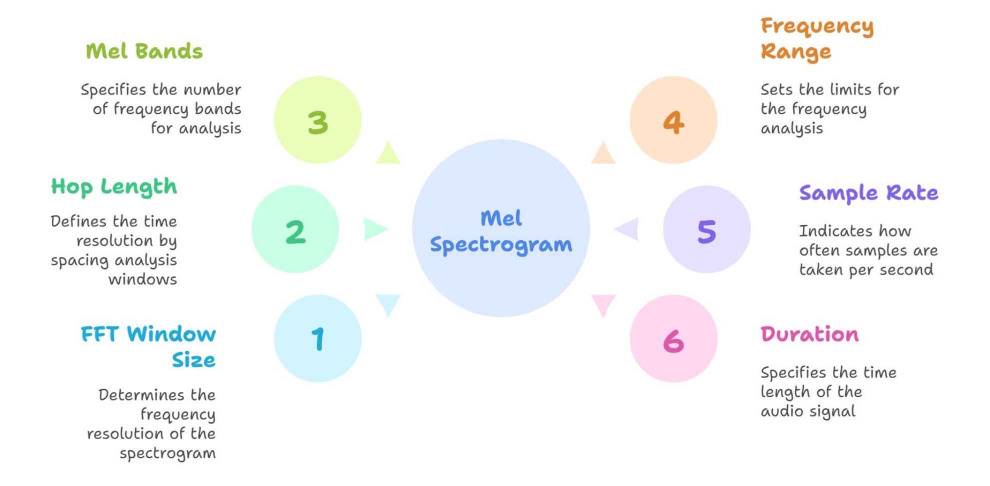
Audio Representation

For our project, we will represent audio data as spectrograms rather than raw waveforms, providing several advantages:

1. Visual representation: Converting time-domain audio signals to frequency-domain spectrograms allows us to leverage powerful CNN architectures originally designed for image classification.

2. Information preservation Mel spectrograms retain perceptually relevant features while reducing dimensionality.

3. Parameter configuration: We will use the following parameters for generating mel spectrograms:
   - FFT window size: 2048 samples
   - Hop length: 512 samples
   - Number of mel bands: 128
   - Frequency range: 20Hz to 8000Hz
   - Sample rate: 48kHz
   - Duration: 4 seconds


Mel Spectrogram
1
FFT Window Size
Determines the frequency resolution of the spectrogram
2
Hop Length
Defines the time resolution by spacing analysis windows
3
Mel Bands
Specifies the number of frequency bands for analysis
4
Frequency Range
Sets the limits for the frequency analysis
5
Sample Rate
Indicates how often samples are taken per second
6
Duration
Specifies the time length of the audio signal

CNN Backbone Architectures

1. **ResNet**: Residual Networks that use skip connections to address vanishing gradient problems in deep networks.
   - Variants to test: ResNet-18
   - Advantages: Strong feature extraction capabilities with relatively low computational cost

2. **MobileNet**: Lightweight architectures designed for mobile and embedded devices.
   - Variant to test: MobileNetV2
   - Advantages: Low parameter count and efficient inference, suitable for deployment on edge devices

3. **EfficientNet**: Models that systematically scale width, depth, and resolution to optimize performance.
   - Variant to test: EfficientNet-B0
   - Advantages: State-of-the-art performance with fewer parameters than comparable models

4. **ConvNeXt**: Modern CNN architecture incorporating design choices from transformers.
   - Variant to test: ConvNeXt-Tiny
   - Advantages: Strong performance while maintaining the inductive biases of CNNs

5. **AlexNet**, **Inception/VGG**: Classic CNN architectures as baseline comparisons.
   - Variants: Inception v3, VGG16
   - Purpose: Established benchmarks to measure improvements against

Our k-fold cross-validation framework will systematically evaluate these architectures using the following metrics:

- **Classification accuracy**
- **F1-score** (particularly for imbalanced classes)
- **Confusion matrix analysis**
- **GradCAM** visualizations for model interpretability

Self-Supervised Learning Techniques

1. **Contrastive Learning Methods**:

- **SimCLR** (Simple Framework for Contrastive Learning of Visual Representations):
  - **Core principle**: Maximize agreement between differently augmented views of the same data example
  - **Implementation**: Modified to work with spectrogram data through our custom script
  - Architecture: ResNet backbone with projection head, NT-Xent loss function

- **MoCo** (Momentum Contrast):
  - **Core principle**: Uses a momentum encoder and a queue of negative samples for contrastive learning
  - **Advantage**: More consistent representation learning with memory bank for negatives

- **BYOL** (Bootstrap Your Own Latent):
  - **Core principle**: Self-distillation without negative samples
  - **Advantage**: Eliminates need for negative samples while maintaining performance

2. Non-Contrastive Methods:

- **Barlow Twins:**
  - **Core principle:** Redundancy reduction through cross-correlation matrix
  - **Advantage:** Avoids collapse without requiring negative samples

- **DINO** (Self-Distillation with No Labels):
  - **Core principle:** Knowledge distillation between student and teacher networks
  - **Advantage:** Produces more semantically meaningful features

- **SwAV** (Swapping Assignments between Views):
  - **Core principle:** Clustering-based approach with online code assignment
  - **Advantage**: More efficient training through simultaneous clustering and representation learning

Data Augmentation Strategies

We employ specialized data augmentation techniques for audio spectrograms:

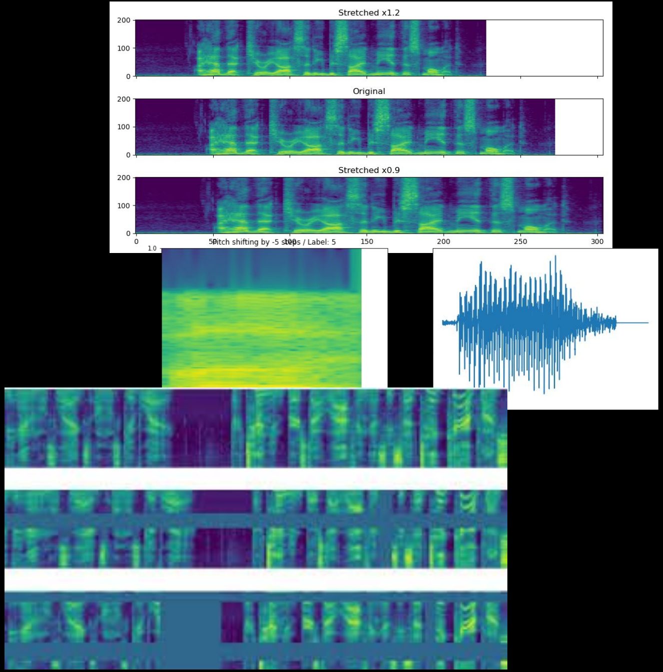
1. Time Stretching: Modification of audio playback speed without affecting pitch.
   - Implementation: librosa's time_stretch with rates of 0.8x and 1.2x
   - Effect: Increases robustness to tempo variations

2. Pitch Shifting: Altering the pitch while maintaining tempo.
   - Implementation: librosa's pitch_shift with ±2 semitones
   - Effect: Improves generalization across pitch variations

3. SpecAugment: Applying time and frequency masking to spectrograms.
   - Implementation: Random masking of time and frequency bands (2 masks each, max width 20)
   - Effect: Improves robustness to missing frequency components and temporal variations

4. PatchAugment: Randomly replacing patches in spectrograms.
   - Implementation: Random replacement of 5 patches of size 10x10
   - Effect: Enhances model's ability to focus on global spectral patterns

5. Noise Addition: Adding controlled Gaussian noise.
   - Implementation: Addition of noise at different SNR levels (5dB and 15dB)
   - Effect: Improves performance in noisy environments



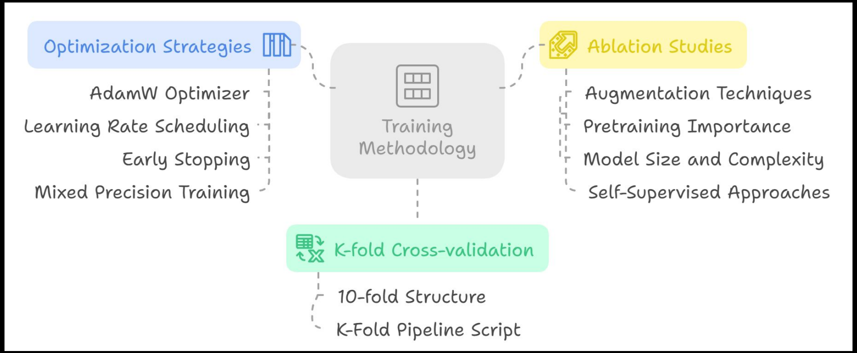
Training & Evaluation Framework

Our training approach will incorporate:

1. K-fold Cross-validation: Using the established 10-fold structure of the dataset.
   - Implementation: K-Fold training pipeline script with configurable fold selection
   - Advantage: Robust evaluation across different data splits

2. Optimization Strategies:
   - AdamW optimizer with weight decay
   - Learning rate scheduling (ReduceLROnPlateau and CosineAnnealingLR)
   - Early stopping based on validation performance
   - Mixed precision training for efficiency

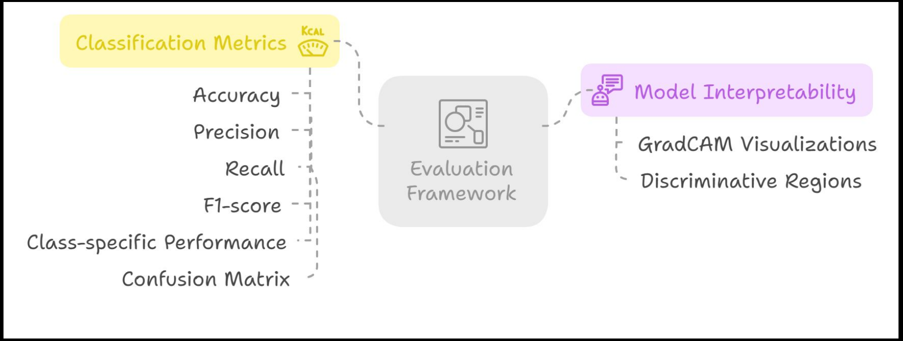
3. Ablation Studies:
   - Effect of individual augmentation techniques
   - Importance of pretraining
   - Impact of model size and complexity
   - Effectiveness of different self-supervised approaches



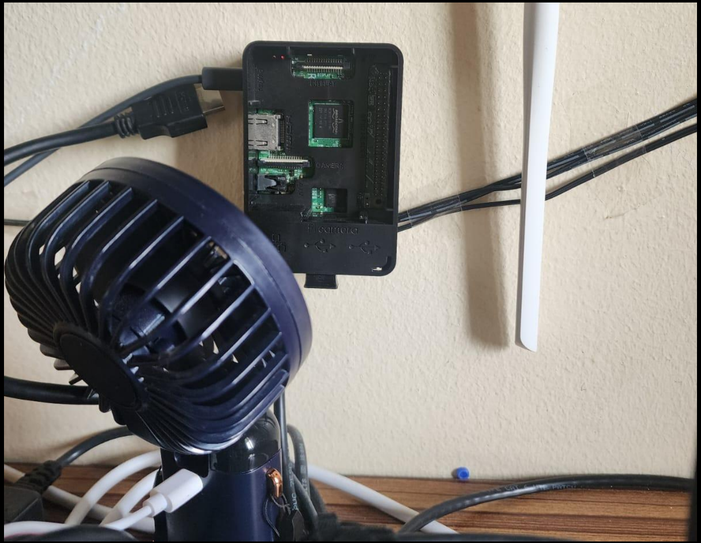
Our evaluation will be comprehensive, considering:

1. Classification Metrics:
   - Accuracy, precision, recall, and F1-score
   - Class-specific performance analysis
   - Confusion matrix visualization

2. Model Interpretability:
   - GradCAM visualizations to highlight discriminative regions

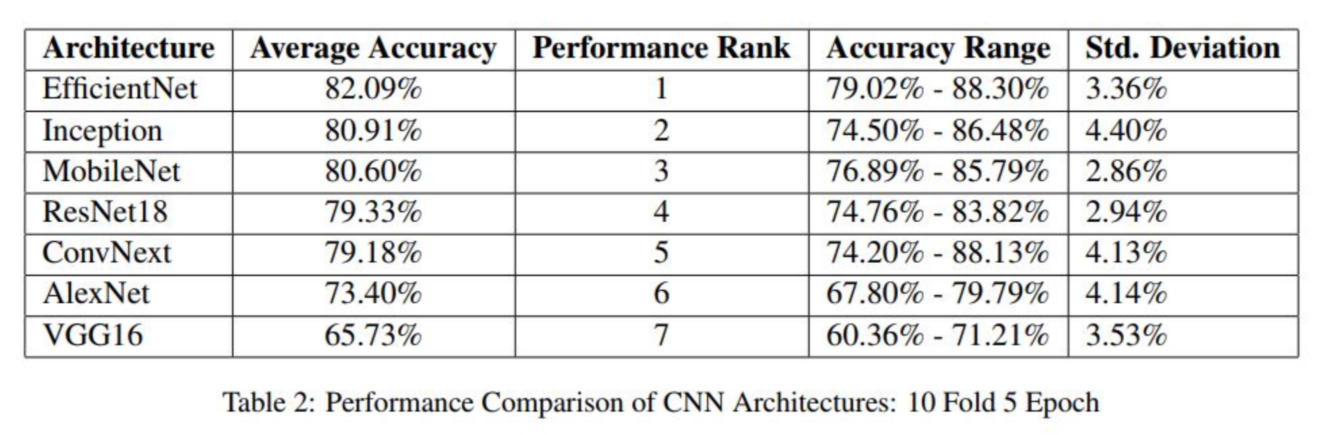


Since training such long hours on personal computers was not feasible, we created a temporary workaround using a raspberry pi as a server which connects to puffer and trains our scripts while we sleep or attend classes. Here’s a sneak peek at our cheeky workaround which helped us generate ~180 GB worth of ablations



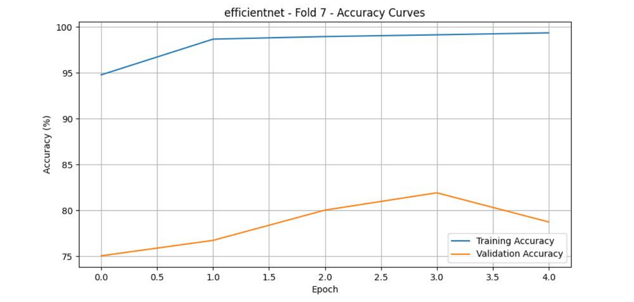
05

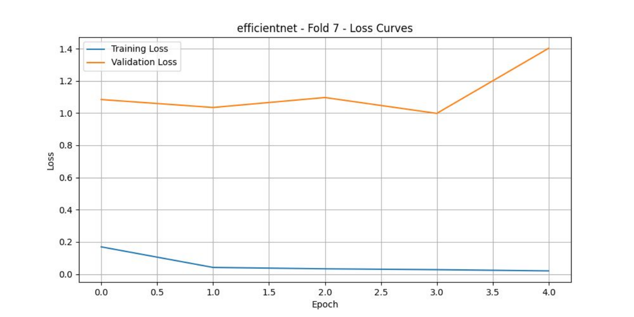
Results & Ablation

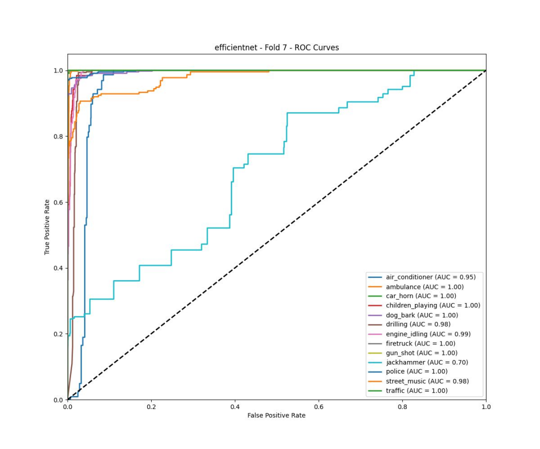


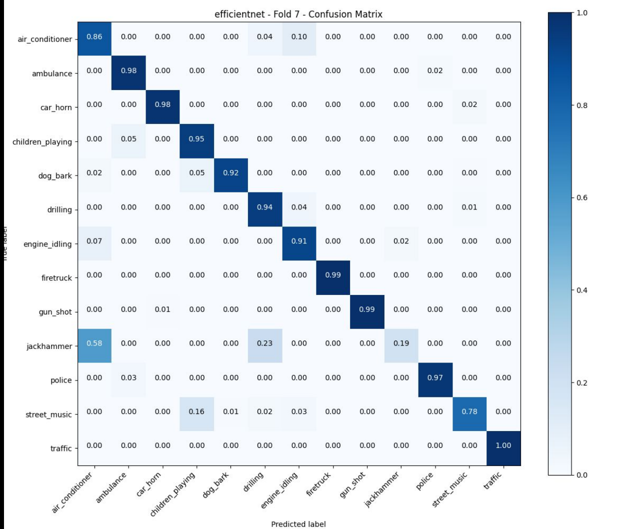
Architecture	Average Accuracy	Performance Rank	Accuracy Range	Std. Deviation
EfficientNet	82.09%	1	79.02% - 88.30%	3.36%
Inception	80.91%	2	74.50% - 86.48%	4.40%
MobileNet	80.60%	3	76.89% - 85.79%	2.86%
ResNet18	79.33%	4	74.76% - 83.82%	2.94%
ConvNext	79.18%	5	74.20% - 88.13%	4.13%
AlexNet	73.40%	6	67.80% - 79.79%	4.14%
VGG16	65.73%	7	60.36% - 71.21%	3.53%

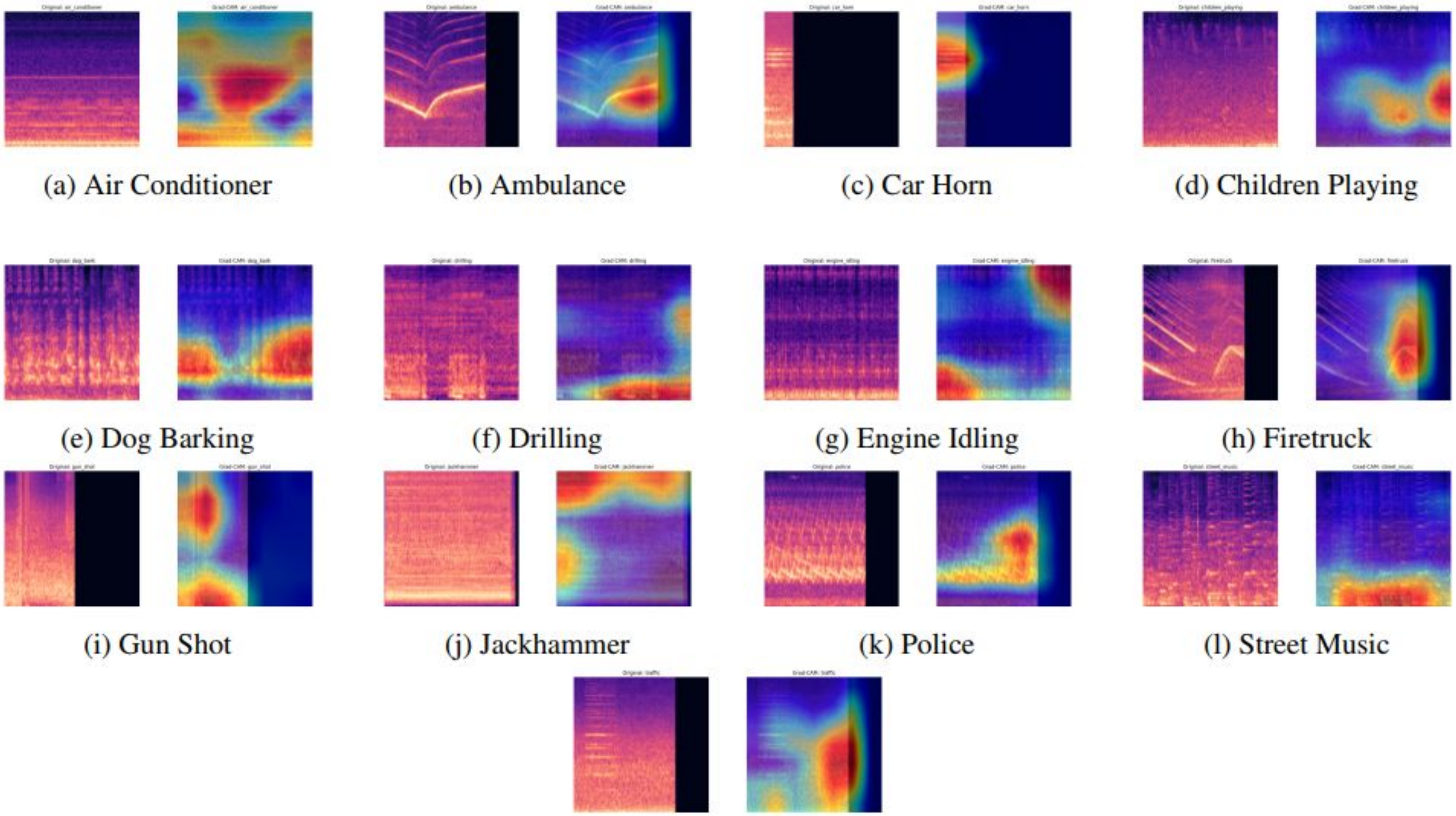
Table 2: Performance Comparison of CNN Architectures: 10 Fold 5 Epoch


efficientnet - Fold 7 - Accuracy Curves


efficientnet - Fold 7 - Loss Curves


efficientnet - Fold 7 - ROC Curves


efficientnet - Fold 7 - Confusion Matrix



(a) Air Conditioner

(b) Ambulance

(c) Car Horn

(d) Children Playing

(e) Dog Barking

(f) Drilling

(g) Engine Idling

(h) Firetruck

(i) Gun Shot

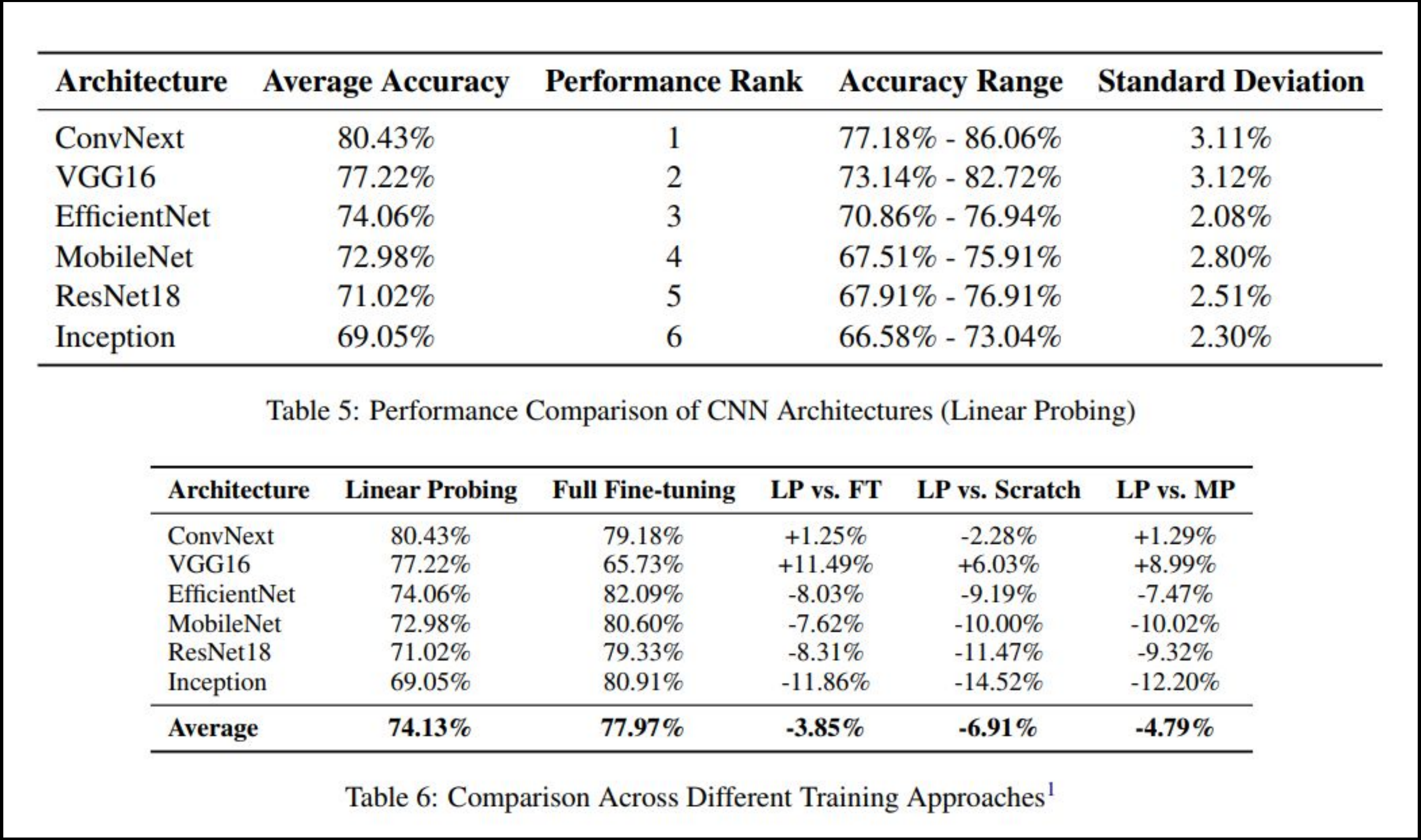
(j) Jackhammer

(k) Police

(l) Street Music

(m) Traffic

Figure 5: GradCam Visualizations for EfficientNet Pre-trained Fold 7 Trained Results

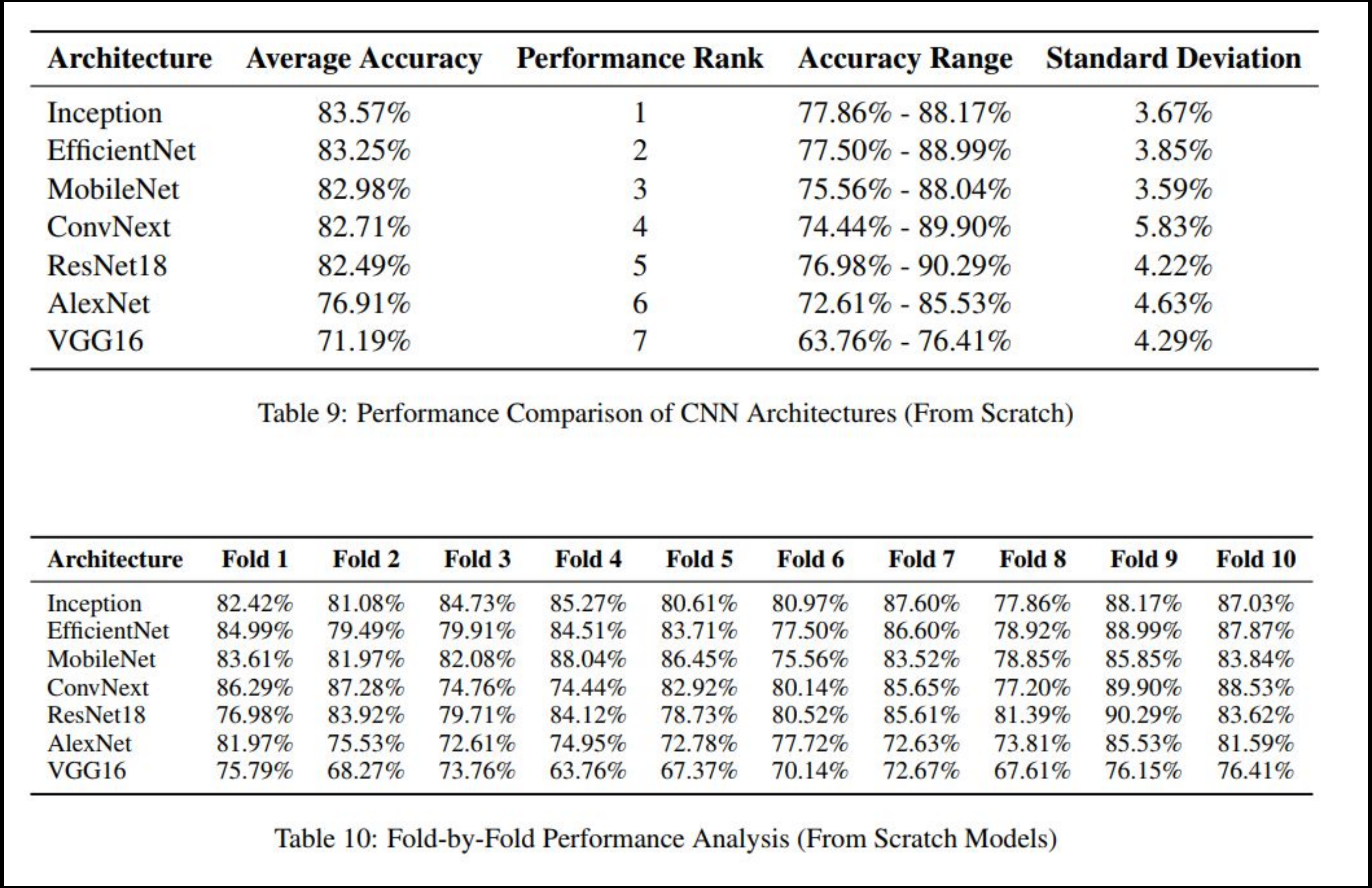


Architecture	Average Accuracy	Performance Rank	Accuracy Range	Standard Deviation
ConvNext	80.43%	1	77.18% - 86.06%	3.11%
VGG16	77.22%	2	73.14% - 82.72%	3.12%
EfficientNet	74.06%	3	70.86% - 76.94%	2.08%
MobileNet	72.98%	4	67.51% - 75.91%	2.80%
ResNet18	71.02%	5	67.91% - 76.91%	2.51%
Inception	69.05%	6	66.58% - 73.04%	2.30%

Table 5: Performance Comparison of CNN Architectures (Linear Probing)

Architecture	Linear Probing	Full Fine-tuning	LP vs. FT	LP vs. Scratch	LP vs. MP
ConvNext	80.43%	79.18%	+1.25%	-2.28%	+1.29%
VGG16	77.22%	65.73%	+11.49%	+6.03%	+8.99%
EfficientNet	74.06%	82.09%	-8.03%	-9.19%	-7.47%
MobileNet	72.98%	80.60%	-7.62%	-10.00%	-10.02%
ResNet18	71.02%	79.33%	-8.31%	-11.47%	-9.32%
Inception	69.05%	80.91%	-11.86%	-14.52%	-12.20%
Average	**74.13%**	**77.97%**	**-3.85%**	**-6.91%**	**-4.79%**

Table 6: Comparison Across Different Training Approaches[1]

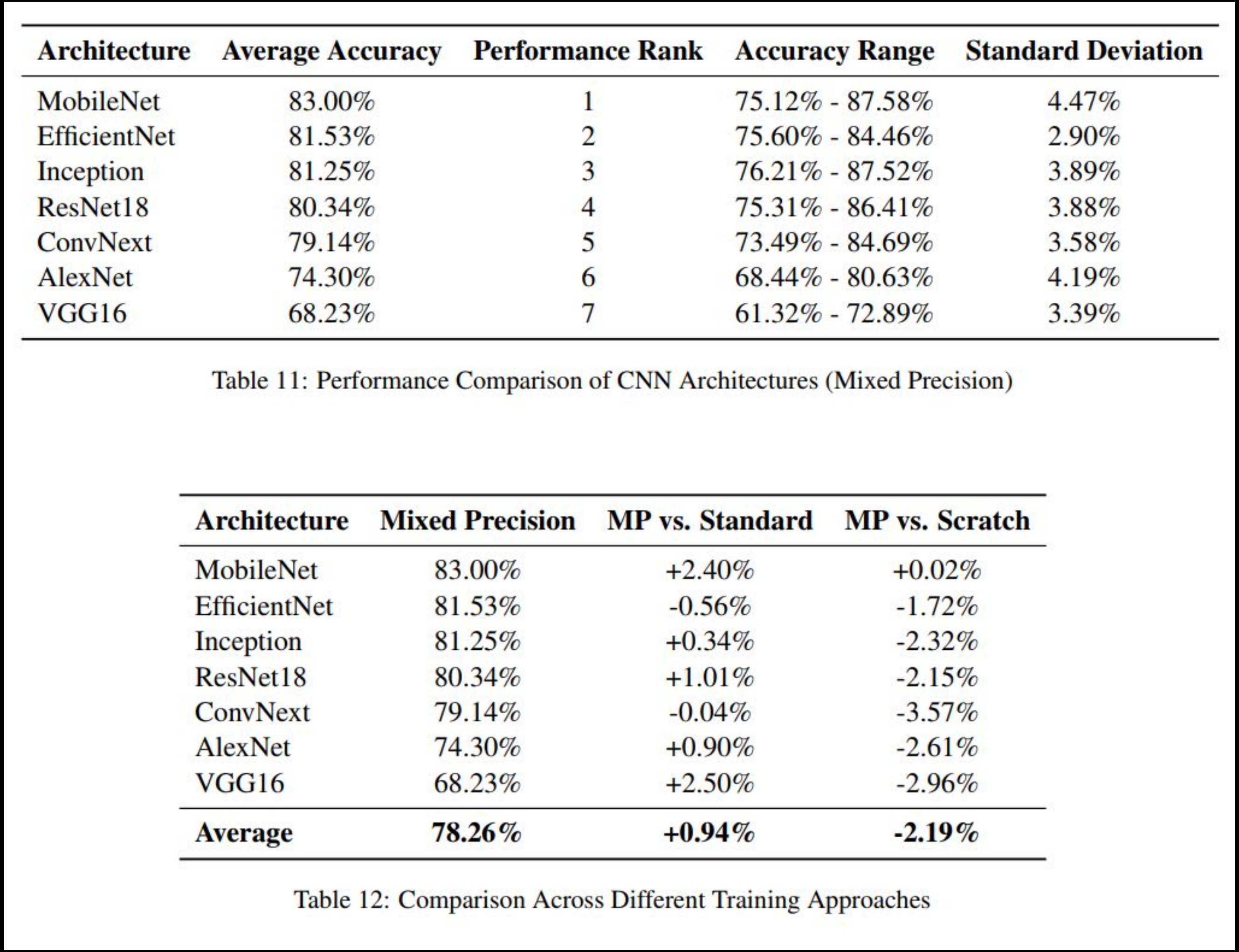


Architecture	Average Accuracy	Performance Rank	Accuracy Range	Standard Deviation
Inception	83.57%	1	77.86% - 88.17%	3.67%
EfficientNet	83.25%	2	77.50% - 88.99%	3.85%
MobileNet	82.98%	3	75.56% - 88.04%	3.59%
ConvNext	82.71%	4	74.44% - 89.90%	5.83%
ResNet18	82.49%	5	76.98% - 90.29%	4.22%
AlexNet	76.91%	6	72.61% - 85.53%	4.63%
VGG16	71.19%	7	63.76% - 76.41%	4.29%

Table 9: Performance Comparison of CNN Architectures (From Scratch)

Architecture	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
Inception	82.42%	81.08%	84.73%	85.27%	80.61%	80.97%	87.60%	77.86%	88.17%	87.03%
EfficientNet	84.99%	79.49%	79.91%	84.51%	83.71%	77.50%	86.60%	78.92%	88.99%	87.87%
MobileNet	83.61%	81.97%	82.08%	88.04%	86.45%	75.56%	83.52%	78.85%	85.85%	83.84%
ConvNext	86.29%	87.28%	74.76%	74.44%	82.92%	80.14%	85.65%	77.20%	89.90%	88.53%
ResNet18	76.98%	83.92%	79.71%	84.12%	78.73%	80.52%	85.61%	81.39%	90.29%	83.62%
AlexNet	81.97%	75.53%	72.61%	74.95%	72.78%	77.72%	72.63%	73.81%	85.53%	81.59%
VGG16	75.79%	68.27%	73.76%	63.76%	67.37%	70.14%	72.67%	67.61%	76.15%	76.41%

Table 10: Fold-by-Fold Performance Analysis (From Scratch Models)



Architecture	Average Accuracy	Performance Rank	Accuracy Range	Standard Deviation
MobileNet	83.00%	1	75.12% - 87.58%	4.47%
EfficientNet	81.53%	2	75.60% - 84.46%	2.90%
Inception	81.25%	3	76.21% - 87.52%	3.89%
ResNet18	80.34%	4	75.31% - 86.41%	3.88%
ConvNext	79.14%	5	73.49% - 84.69%	3.58%
AlexNet	74.30%	6	68.44% - 80.63%	4.19%
VGG16	68.23%	7	61.32% - 72.89%	3.39%

Table 11: Performance Comparison of CNN Architectures (Mixed Precision)

Architecture	Mixed Precision	MP vs. Standard	MP vs. Scratch
MobileNet	83.00%	+2.40%	+0.02%
EfficientNet	81.53%	-0.56%	-1.72%
Inception	81.25%	+0.34%	-2.32%
ResNet18	80.34%	+1.01%	-2.15%
ConvNext	79.14%	-0.04%	-3.57%
AlexNet	74.30%	+0.90%	-2.61%
VGG16	68.23%	+2.50%	-2.96%
Average	**78.26%**	**+0.94%**	**-2.19%**

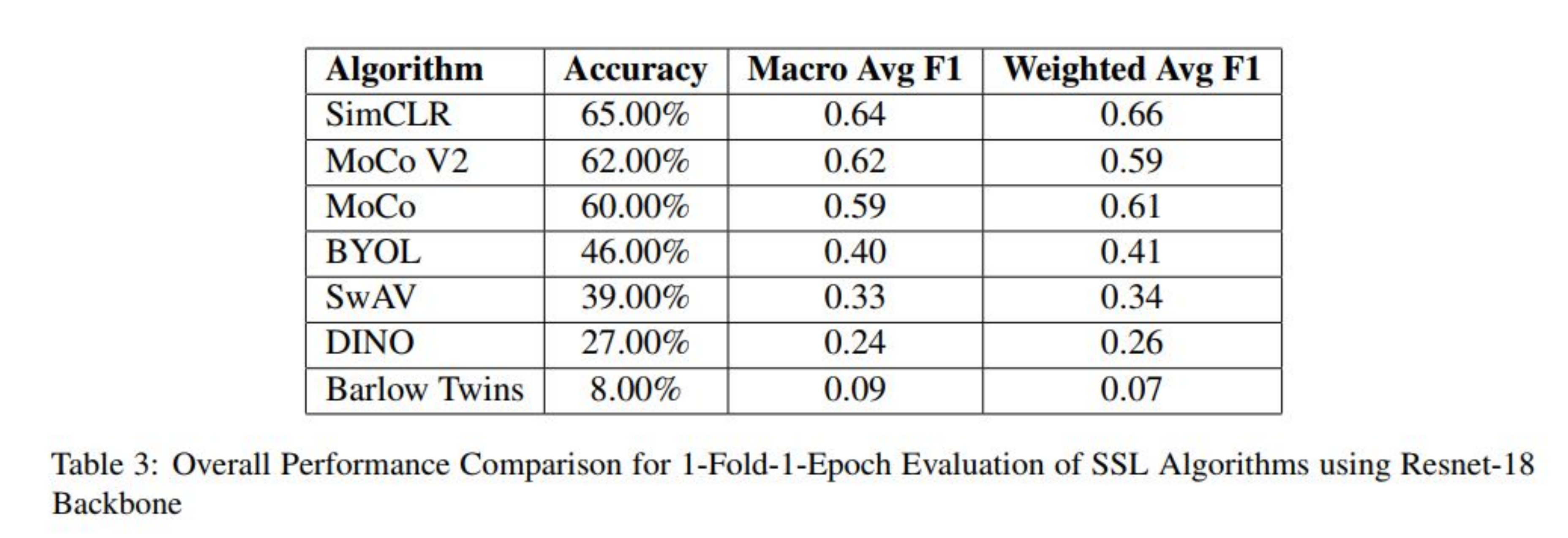
Table 12: Comparison Across Different Training Approaches

Architecture	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
MobileNet	87.58%	83.23%	76.24%	83.54%	83.86%	75.12%	86.04%	81.13%	86.02%	87.25%
EfficientNet	81.11%	84.18%	80.93%	83.09%	80.59%	75.60%	84.46%	78.03%	83.23%	84.04%
Inception	87.52%	78.29%	76.94%	80.30%	79.84%	76.21%	85.77%	80.86%	80.84%	85.89%
ResNet18	82.97%	85.88%	76.20%	75.31%	76.55%	78.79%	81.42%	79.43%	86.41%	80.42%
ConvNext	81.06%	79.58%	77.25%	73.49%	79.02%	74.36%	83.18%	78.97%	84.69%	79.75%
AlexNet	80.63%	73.71%	74.23%	71.51%	68.81%	77.52%	72.01%	68.44%	78.95%	77.20%
VGG16	66.36%	69.68%	69.19%	66.56%	61.32%	65.21%	72.89%	70.61%	71.19%	69.24%

Table 13: Fold-by-Fold Performance Analysis (Mixed Precision Models)

Architecture	FP32 Training Time	Mixed Precision Time	Speed Improvement	Memory Reduction
MobileNet	1.00x (baseline)	0.65x	35%	45%
EfficientNet	1.70x	1.10x	35%	48%
Inception	1.60x	1.04x	35%	46%
ResNet18	0.80x	0.52x	35%	47%
ConvNext	1.85x	1.20x	35%	45%
AlexNet	0.70x	0.46x	34%	43%
VGG16	1.25x	0.81x	35%	44%

Table 14: Computation Time and Memory Usage Benefit[3]



Algorithm	Accuracy	Macro Avg F1	Weighted Avg F1
SimCLR	65.00%	0.64	0.66
MoCo V2	62.00%	0.62	0.59
MoCo	60.00%	0.59	0.61
BYOL	46.00%	0.40	0.41
SwAV	39.00%	0.33	0.34
DINO	27.00%	0.24	0.26
Barlow Twins	8.00%	0.09	0.07

Table 3: Overall Performance Comparison for 1-Fold-1-Epoch Evaluation of SSL Algorithms using Resnet-18 Backbone

Contrastive Learning Results

- For CL we trained SimClr and MoCo model, which shows subtle results for feature representation.

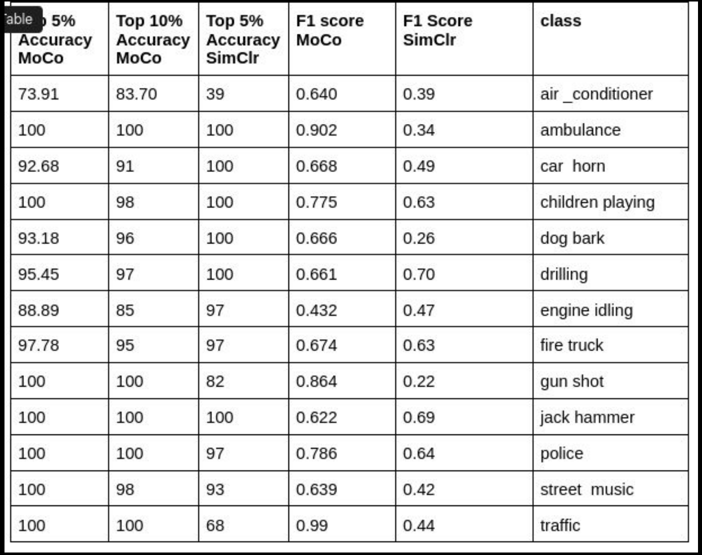
- Both of the Model was trained on T4 with 15 GB GPU RAM, various ablation is performed based on the availability of continuous resources

- The top % is calculated on the stratified samples Of each classes from hold out sample.

- In our evaluation we observe the discrepancy with simClr results , where the top 5% and top 10% is Really high in comparison to F1 score.

- This could be due to do intra class variation or Potential overfit, to overcome this

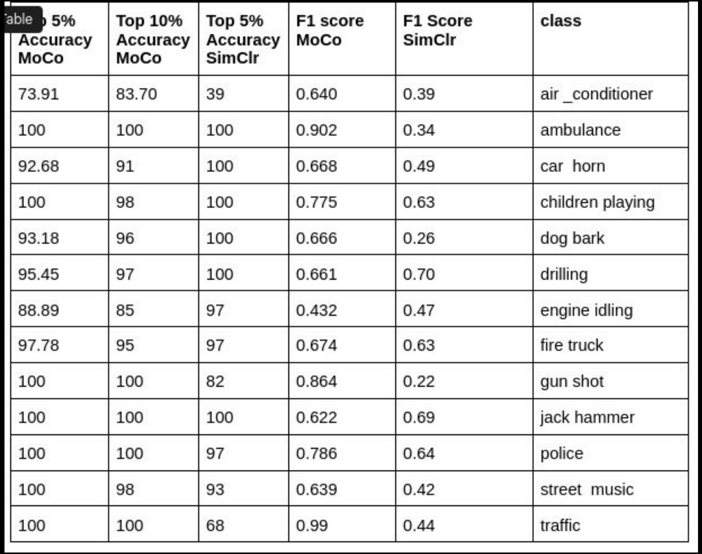
- We tried to increase the sample size and change the learning rate, add Mix up of augmentation that world as meta augmentation



able 5% Accuracy MoCo	Top 10% Accuracy MoCo	Top 5% Accuracy SimClr	F1 score MoCo	F1 Score SimClr	class
73.91	83.70	39	0.640	0.39	air _conditioner
100	100	100	0.902	0.34	ambulance
92.68	91	100	0.668	0.49	car horn
100	98	100	0.775	0.63	children playing
93.18	96	100	0.666	0.26	dog bark
95.45	97	100	0.661	0.70	drilling
88.89	85	97	0.432	0.47	engine idling
97.78	95	97	0.674	0.63	fire truck
100	100	82	0.864	0.22	gun shot
100	100	100	0.622	0.69	jack hammer
100	100	97	0.786	0.64	police
100	98	93	0.639	0.42	street music
100	100	68	0.99	0.44	traffic

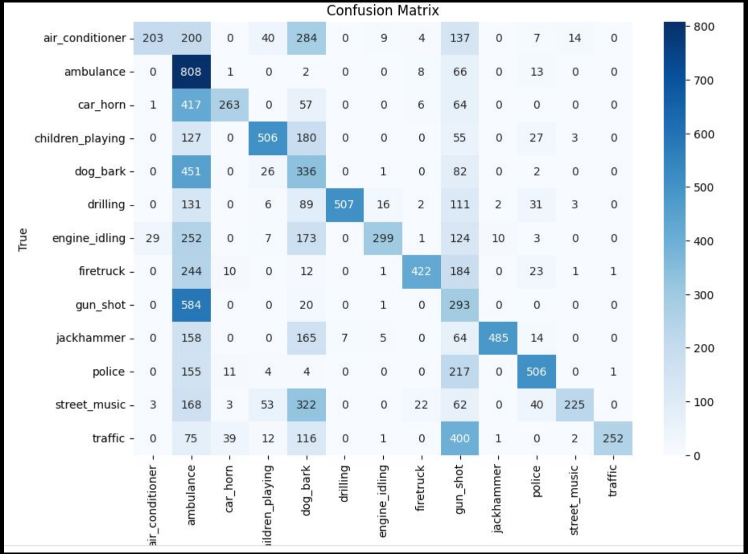
Contrastive Learning Results

- For CL we trained SimClr and MoCo model, which shows subtle results for feature representation.
- Both of the Model was trained on T4 with 15 GB GPU RAM, various ablation is performed based on the availability of continuous resources
- The top % is calculated on the stratified samples Of each classes from hold out sample.
- In our evaluation we observe the discrepancy with simClr results , where the top 5% and top 10% is Really high in comparison to F1 score.
- This could be due to do intra class variation or Potential overfit, to overcome this
- We tried to increase the sample size and change the learning rate, add Mix up of augmentation that world as meta augmentation

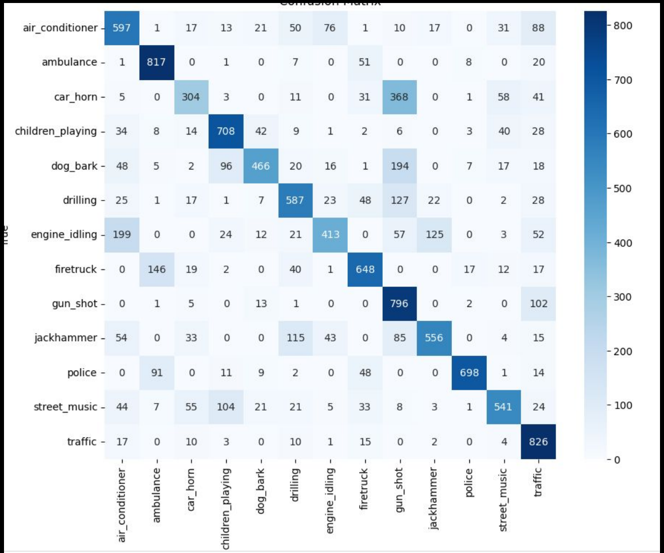


5% Accuracy MoCo	Top 10% Accuracy MoCo	Top 5% Accuracy SimClr	F1 score MoCo	F1 Score SimClr	class
73.91	83.70	39	0.640	0.39	air _conditioner
100	100	100	0.902	0.34	ambulance
92.68	91	100	0.668	0.49	car horn
100	98	100	0.775	0.63	children playing
93.18	96	100	0.666	0.26	dog bark
95.45	97	100	0.661	0.70	drilling
88.89	85	97	0.432	0.47	engine idling
97.78	95	97	0.674	0.63	fire truck
100	100	82	0.864	0.22	gun shot
100	100	100	0.622	0.69	jack hammer
100	100	97	0.786	0.64	police
100	98	93	0.639	0.42	street music
100	100	68	0.99	0.44	traffic

Continue …



	air_conditioner	ambulance	car_horn	children_playing	dog_bark	drilling	engine_idling	firetruck	gun_shot	jackhammer	police	street_music	traffic
air_conditioner	203	200	0	40	284	0	9	4	137	0	7	14	0
ambulance	0	808	1	0	2	0	0	8	66	0	13	0	0
car_horn	1	417	263	0	57	0	0	6	64	0	0	0	0
children_playing	0	127	0	506	180	0	0	0	55	0	27	3	0
dog_bark	0	451	0	26	336	0	1	0	82	0	2	0	0
drilling	0	131	0	6	89	507	16	2	111	2	31	3	0
engine_idling	29	252	0	7	173	0	299	1	124	10	3	0	0
firetruck	0	244	10	0	12	0	1	422	184	0	23	1	1
gun_shot	0	584	0	0	20	0	1	0	293	0	0	0	0
jackhammer	0	158	0	0	165	7	5	0	64	485	14	0	0
police	0	155	11	4	4	0	0	0	217	0	506	0	1
street_music	3	168	3	53	322	0	0	22	62	0	40	225	0
traffic	0	75	39	12	116	0	1	0	400	1	0	2	252



	air_conditioner	ambulance	car_horn	children_playing	dog_bark	drilling	engine_idling	firetruck	gun_shot	jackhammer	police	street_music	traffic
air_conditioner	597	1	17	13	21	50	76	1	10	17	0	31	88
ambulance	1	817	0	1	0	7	0	51	0	0	8	0	20
car_horn	5	0	304	3	0	11	0	31	368	0	1	58	41
children_playing	34	8	14	708	42	9	1	2	6	0	3	40	28
dog_bark	48	5	2	96	466	20	16	1	194	0	7	17	18
drilling	25	1	17	1	7	587	23	48	127	22	0	2	28
engine_idling	199	0	0	24	12	21	413	0	57	125	0	3	52
firetruck	0	146	19	2	0	40	1	648	0	0	17	12	17
gun_shot	0	1	5	0	13	1	0	0	796	0	2	0	102
jackhammer	54	0	33	0	0	115	43	0	85	556	0	4	15
police	0	91	0	11	9	2	0	48	0	0	698	1	14
street_music	44	7	55	104	21	21	5	33	8	3	1	541	24
traffic	17	0	10	3	0	10	1	15	0	2	0	4	826

- The above confusion matrix left SimClr right for Moco
- MoCo however maintains maintains both higher Top-K accuracy and consistently better F1 score which have stronger class wise generalization
- With the contrast we can also Moco have relevance in miss classification classes, which indicates better representation

Abiliation for SimClr

The pattern of augmentation is consistent in both the models but MoCo works better .

Original data alone is not able perform as well which is 14% worst than augmentation version in general

Further we also perform Architectural based ablation to evaluate the performance of downstream tasks

We assume here lower dimension might perform better At latent space , and on the higher dimensions saturates the results .

The lower projection dimensions can Cause model to collapse representation it Is crucial to analyse the trade off between the expressiveness and compression .

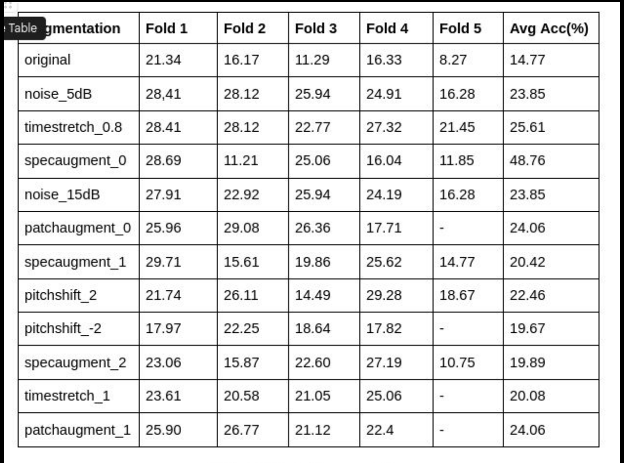


Table gmentation	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Avg Acc(%)
original	21.34	16.17	11.29	16.33	8.27	14.77
noise_5dB	28,41	28.12	25.94	24.91	16.28	23.85
timestretch_0.8	28.41	28.12	22.77	27.32	21.45	25.61
specaugment_0	28.69	11.21	25.06	16.04	11.85	48.76
noise_15dB	27.91	22.92	25.94	24.19	16.28	23.85
patchaugment_0	25.96	29.08	26.36	17.71	-	24.06
specaugment_1	29.71	15.61	19.86	25.62	14.77	20.42
pitchshift_2	21.74	26.11	14.49	29.28	18.67	22.46
pitchshift_-2	17.97	22.25	18.64	17.82	-	19.67
specaugment_2	23.06	15.87	22.60	27.19	10.75	19.89
timestretch_1	23.61	20.58	21.05	25.06	-	20.08
patchaugment_1	25.90	26.77	21.12	22.4	-	24.06

Table 3 Effect of Augmentations

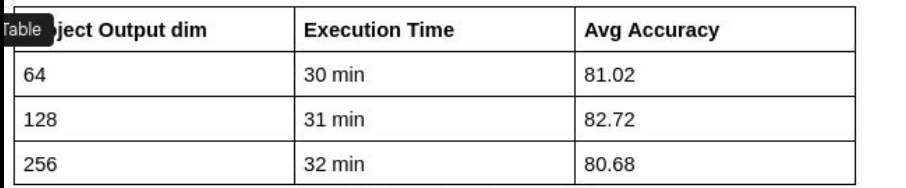


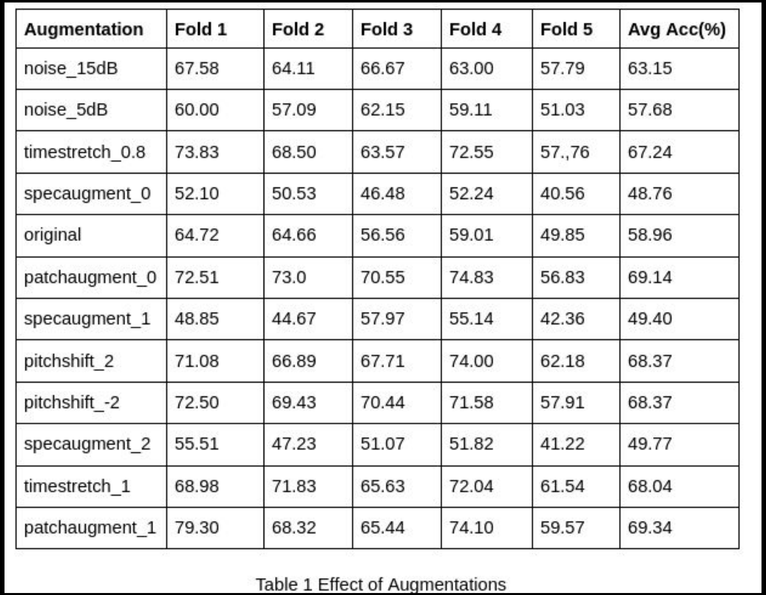
Table ject Output dim	Execution Time	Avg Accuracy
64	30 min	81.02
128	31 min	82.72
256	32 min	80.68

Table 4.Projection Output dimension

Ablation For MoCo

In this ablation we study the effect of Argumentation

- Here we perform evaluation effect of tests set Across 5 folds specific to augmentation
- Observation, augmentation like pitch_shift and time stretch perform consistently better than Specagumentation
- Patch augmentation have shown the better Results since the robustness with variations of in Data , which is boost the generalization ability
- For the original data we don't see as the similar Behaviour, which proves that augmentations have significant improvement in predictability rate



Augmentation	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Avg Acc(%)
noise_15dB	67.58	64.11	66.67	63.00	57.79	63.15
noise_5dB	60.00	57.09	62.15	59.11	51.03	57.68
timestretch_0.8	73.83	68.50	63.57	72.55	57.,76	67.24
specaugment_0	52.10	50.53	46.48	52.24	40.56	48.76
original	64.72	64.66	56.56	59.01	49.85	58.96
patchaugment_0	72.51	73.0	70.55	74.83	56.83	69.14
specaugment_1	48.85	44.67	57.97	55.14	42.36	49.40
pitchshift_2	71.08	66.89	67.71	74.00	62.18	68.37
pitchshift_-2	72.50	69.43	70.44	71.58	57.91	68.37
specaugment_2	55.51	47.23	51.07	51.82	41.22	49.77
timestretch_1	68.98	71.83	65.63	72.04	61.54	68.04
patchaugment_1	79.30	68.32	65.44	74.10	59.57	69.34

Table 1 Effect of Augmentations

Continue...

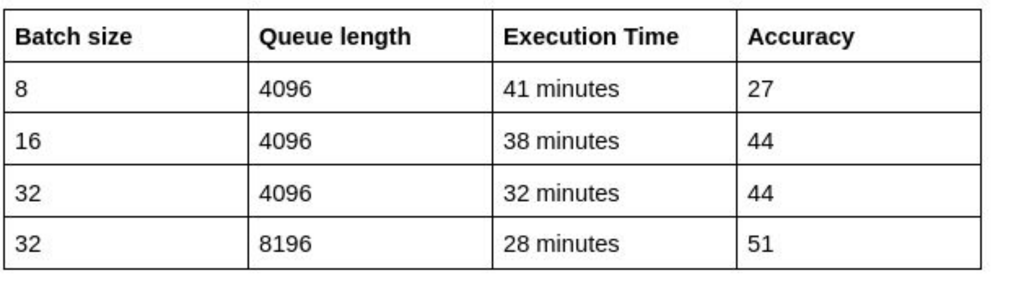
- Further we perform study on Batch size and Queue length since assume that MoCo create negative sample might perform better with lower batch size, can gain advantage over the resources available and trade off

- The batch size upto 16 gave Good results but we kept the Queue length constant.

- Increasing the Queue length gave significant improvement

- But this came with the trade off , that prediction rate gets Saturated

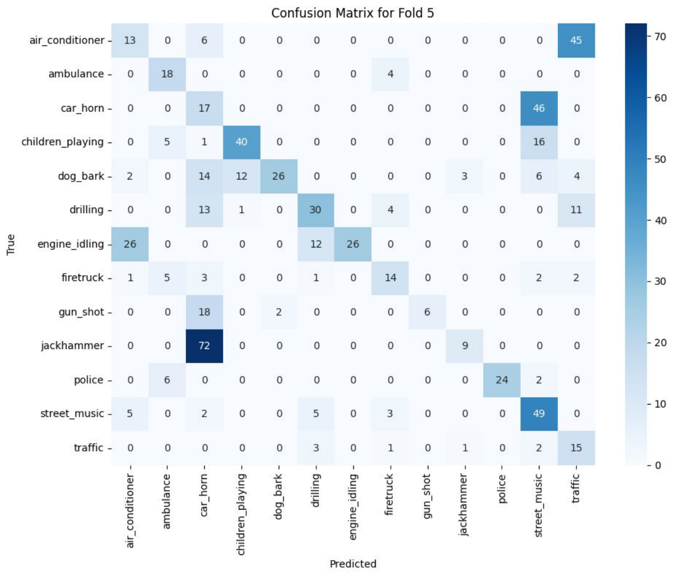
- Explained Further ..



Batch size	Queue length	Execution Time	Accuracy
8	4096	41 minutes	27
16	4096	38 minutes	44
32	4096	32 minutes	44
32	8196	28 minutes	51

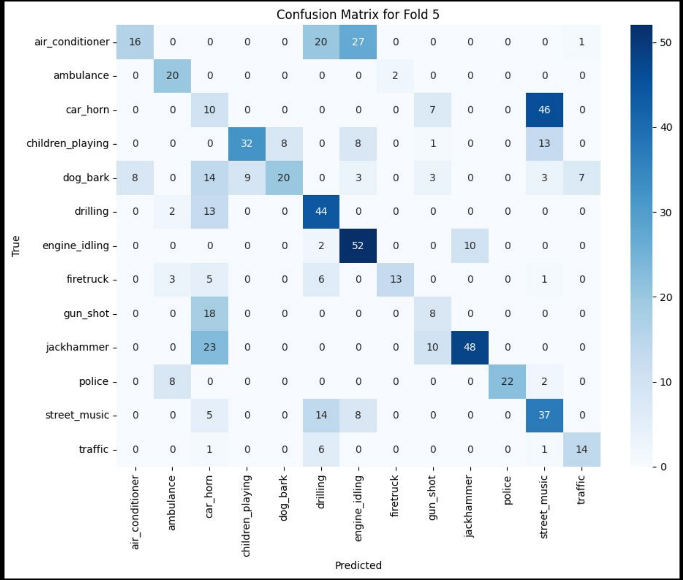
Table 2. Batch Size and Queue length Effect

Continue …



Confusion Matrix for Fold 5

	air_conditioner	ambulance	car_horn	children_playing	dog_bark	drilling	engine_idling	firetruck	gun_shot	jackhammer	police	street_music	traffic
air_conditioner	13	0	6	0	0	0	0	0	0	0	0	0	45
ambulance	0	18	0	0	0	0	0	4	0	0	0	0	0
car_horn	0	0	17	0	0	0	0	0	0	0	0	46	0
children_playing	0	5	1	40	0	0	0	0	0	0	0	16	0
dog_bark	2	0	14	12	26	0	0	0	0	3	0	6	4
drilling	0	0	13	1	0	30	0	4	0	0	0	0	11
engine_idling	26	0	0	0	0	12	26	0	0	0	0	0	0
firetruck	1	5	3	0	0	1	0	14	0	0	0	2	2
gun_shot	0	0	18	0	2	0	0	0	6	0	0	0	0
jackhammer	0	0	72	0	0	0	0	0	0	9	0	0	0
police	0	6	0	0	0	0	0	0	0	0	24	2	0
street_music	5	0	2	0	0	5	0	3	0	0	0	49	0
traffic	0	0	0	0	0	3	0	1	0	1	0	2	15



Confusion Matrix for Fold 5

	air_conditioner	ambulance	car_horn	children_playing	dog_bark	drilling	engine_idling	firetruck	gun_shot	jackhammer	police	street_music	traffic
air_conditioner	16	0	0	0	0	20	27	0	0	0	0	0	1
ambulance	0	20	0	0	0	0	0	2	0	0	0	0	0
car_horn	0	0	10	0	0	0	0	0	7	0	0	46	0
children_playing	0	0	0	32	8	0	8	0	1	0	0	13	0
dog_bark	8	0	14	9	20	0	3	0	3	0	0	3	7
drilling	0	2	13	0	0	44	0	0	0	0	0	0	0
engine_idling	0	0	0	0	0	2	52	0	0	10	0	0	0
firetruck	0	3	5	0	0	6	0	13	0	0	0	1	0
gun_shot	0	0	18	0	0	0	0	0	8	0	0	0	0
jackhammer	0	0	23	0	0	0	0	0	10	48	0	0	0
police	0	8	0	0	0	0	0	0	0	0	22	2	0
street_music	0	0	5	0	0	14	8	0	0	0	0	37	0
traffic	0	0	1	0	0	6	0	0	0	0	0	1	14

- The confusion matrices have performance for the final folds across 5 Epochs , on the left we have Queue size 4096 and on the right 8192
- We prove that the queue length with 4096 is better performing since the saturation of the loss , with higher sequence length model become more sensitive and noisy

06

Discussion

Impact of Self-Supervised Learning on Limited Labeled Data (RQ1)

Our experiments demonstrate that self-supervised learning techniques can significantly improve audio event classification performance when labeled data is limited. SimCLR emerged as the most effective SSL approach with 65% accuracy, followed by MoCoV2 (62%) and MoCo (60%). These contrastive learning methods consistently outperformed non-contrastive approaches like BYOL (46%), SwAV (39%), DINO (27%), and BarlowTwins (8%).

The superior performance of contrastive methods can be attributed to their ability to learn discriminative representations by contrasting positive pairs against negative examples. This approach is particularly effective for audio data, where subtle spectral and temporal differences often define class boundaries.

Effectiveness of Specialized Data Augmentation (RQ2)

Our results clearly demonstrate that specialized audio augmentation techniques substantially enhance model robustness, particularly in challenging acoustic environments. Among the augmentation methods evaluated, PatchAugment consistently delivered the strongest performance improvements across different SSL frameworks, with PatchAugment_1 achieving 69.34% accuracy with MoCo compared to 58.96% for unaugmented data.

Time-domain augmentations (TimeStretch_0.8: 67.24%, PitchShift_±2: 68.37%) significantly outperformed frequency-domain augmentations (SpecAugment variants: 48.76-49.77%). This suggests that preserving spectral characteristics while introducing temporal variations is particularly effective for audio event classification.

Impact of Dataset Extension on Model Generalizability (RQ3)

Expanding the UrbanSound8K dataset with four additional urban sound classes (ambulance, firetruck, police, traffic) significantly enhanced model generalizability and robustness. The confusion matrices reveal that models successfully learned to distinguish between these new classes and the original classes, with particularly strong performance on "traffic" (F1: 0.88 with SimCLR) and "police" (F1: 0.95 with MoCoV2).

The removal of the general "siren" class to avoid semantic overlap with the new emergency vehicle classes proved effective

1. Architecture: EfficientNet emerged as the top performer (82.09% accuracy) among pretrained models, likely due to its compound scaling approach that balances network depth, width, and resolution. However, our surprising finding that models trained from scratch consistently outperformed pretrained counterparts (by an average of 3.12 percentage points) suggests that ImageNet pretraining may introduce biases that are suboptimal for audio spectrograms.
2. SSL Framework: SimCLR with a 128-dimensional projection head provides the strongest representation learning capabilities for audio classification, particularly when combined with appropriate augmentations.
3. Augmentation Strategy: A combination of PatchAugment and time-domain augmentations (TimeStretch, PitchShift) delivers the most consistent performance improvements, with PatchAugment showing particular effectiveness for contrastive learning approaches.
4. Training Approach: Mixed precision training generally improved model performance compared to standard training (average gain of 0.94 percentage points), with VGG16 (+2.50%) and MobileNet (+2.40%) showing the largest improvements.

Broader Applications

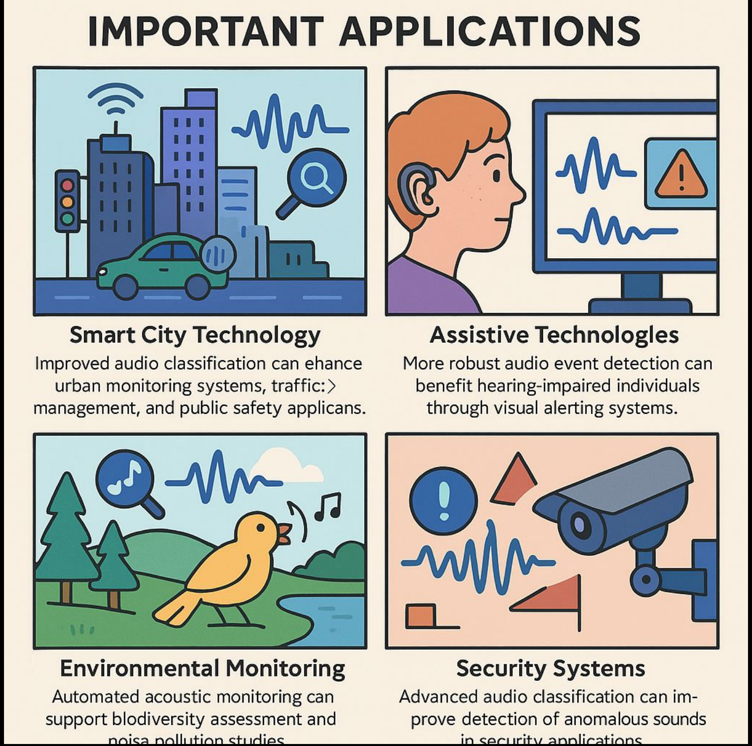
Our research has several important implications in terms of potential applications:

- Smart City Technology: Improved audio classification can enhance urban monitoring systems, traffic management, and public safety applications.

- Assistive Technologies: More robust audio event detection can benefit hearing-impaired individuals through visual alerting systems.

- Environmental Monitoring: Automated acoustic monitoring can support biodiversity assessment and noise pollution studies.

- Security Systems: Advanced audio classification can improve detection of anomalous sounds in security applications



IMPORTANT APPLICATIONS

Smart City Technology

Improved audio classification can ehance urban monitoring systems, traffic:> management, and public safety applicans.

Assistive Technologles

More robust audio event detection can benefit hearing-impaired individuals through visual alerting systems.

Environmental Monitoring

Automated acoustic monitoring can support blodiversity assessment and noisa pollution studies

Security Systems

Advanced audio classification can improve detection of anomalous sounds in security applications

TIME FOR A LIVE DEMO
38
WHAT COULD GO WRONG?
memegenerator.net

Click Here

[1] I. Moummad, N. Farrugia and R. Serizel, "Self-Supervised Learning for Few-Shot Bird Sound Classification," 2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW), Seoul, Korea, Republic of, 2024, pp. 600-604, doi: 10.1109/ICASSPW62465.2024.10627576. keywords: {Costs;Conferences;Neural networks;Self-supervised learning;Speech enhancement;Signal processing;Birds;Self-supervised learning;data augmentation;few-shot learning;bird sounds},

[2] Tripathi, A. M., & Mishra, A. (2021). Self-supervised learning for Environmental Sound Classification. Applied Acoustics, 182, 108183. https://doi.org/10.1016/j.apacoust.2021.108183

[3] K. Wilkinghoff, "Self-Supervised Learning for Anomalous Sound Detection," ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Korea, Republic of, 2024, pp. 276-280, doi: 10.1109/ICASSP48485.2024.10447156. keywords: {Training;Self-supervised learning;Signal processing;Acoustics;Task analysis;Speech processing;self-supervised learning;anomalous sound detection;domain generalization;machine listening},

[4] Abdoli, S., Cardinal, P., & Lameiras Koerich, A. (2019). End-to-End Environmental Sound Classification using a 1D Convolutional Neural Network. ArXiv, abs/1904.08990.

[5] A. Guzhov, F. Raue, J. Hees and A. Dengel, "Audioclip: Extending Clip to Image, Text and Audio," ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, Singapore, 2022, pp. 976-980, doi: 10.1109/ICASSP43922.2022.9747631.

[6] Bae, S., Kim, J., Cho, W., Baek, H., Son, S., Lee, B.H., Ha, C.W., Tae, K., Kim, S., & Yun, S. (2023). Patch-Mix Contrastive Learning with Audio Spectrogram Transformer on Respiratory Sound Classification. Interspeech.

[7] Chen, F., Zhu, Z., Sun, C., & Xia, L. (2025). Evaluating metric and contrastive learning in pretrained models for environmental sound classification. Applied Acoustics, 232, 110593. https://doi.org/10.1016/j.apacoust.2025.110593

[8] Chen, X., Wang, M., Kan, R., & Qiu, H. (2024). Improved Patch-Mix Transformer and Contrastive Learning Method for Sound Classification in Noisy Environments. Applied Sciences, 14(21), 9711. https://doi.org/10.3390/app14219711

[9] Venkatesh, S., Moffat, D., & Miranda, E.R. (2021). You Only Hear Once: A YOLO-like Algorithm for Audio Segmentation and Sound Event Detection. ArXiv, abs/2109.00962.

[10] Vu, L., Tran, T., Lim, W., & Phan, R. (2024). Toward end-to-end interpretable convolutional neural networks for waveform signals. ArXiv, abs/2405.01815.

[11] Kadandale, V.S., Montesinos, J.F., Haro, G., & G'omez, E. (2020). Multi-task U-Net for Music Source Separation. ArXiv, abs/2003.10414.

[12] A. Nasiri, Y. Cui, Z. Liu, J. Jin, Y. Zhao and J. Hu, "AudioMask: Robust Sound Event Detection Using Mask R-CNN and Frame-Level Classifier," 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), Portland, OR, USA, 2019, pp. 485-492, doi: 10.1109/ICTAI.2019.00074. keywords: {Sound Event Detection;Mask R-CNN;Audio Analysis;Audio Classifier},

[13] Ntalampiras, S., Potamitis, H.: Acoustic detection of unknown bird species and individuals. CAAI Trans. Intell. Technol. 6(3), 291–300 (2021). https://doi.org/10.1049/cit2.12007

[14] Zhao, J., Liu, X., Zhao, J., Yuan, Y., Kong, Q., Plumbley, M.D., & Wang, W. (2024). Universal Sound Separation with Self-Supervised Audio Masked Autoencoder. 2024 32nd European Signal Processing Conference (EUSIPCO), 1-5.

Q&A

THANK YOU

QUACLRS