Indoor Environment Sound Classification

PRESENTED BY:

PRANIT DAS(24CS4512)

STUDENT 2

STUDENT 3

STUDENT 4

Objective

Build a model to correctly classify indoor sounds into predefined categories .

Establish a real-time system that can detect and classify sounds in real time, thus facilitating prompt response to possible emergencies.

Give priorities to sounds that may signal potential risks, including falls or accidents involving walkers and crutches.

Dataset Description

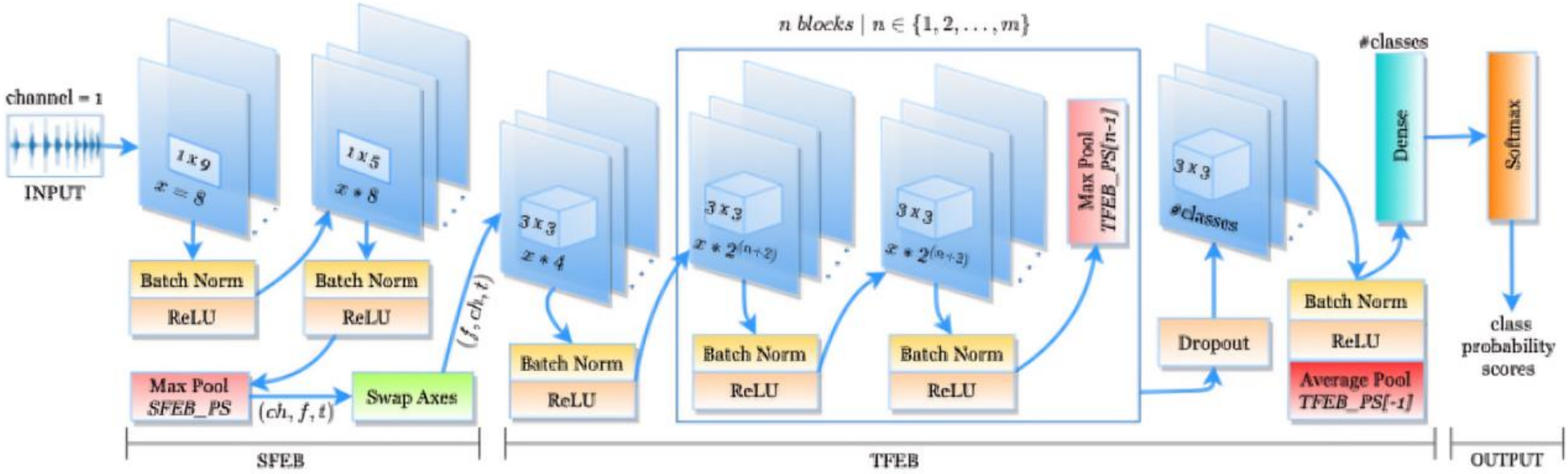
Sound	Bathroom_Tap: Sound of a bathroom faucet being turned on or off.
Sound	Walker_Crutch: Sound of a walker or crutch being used.
Silence or background	No_Class: Silence or background noise without any relevant sound event.
Flush	Flush: Sound of a toilet flushing.
Shower	Shower: Sound of a shower running.
Switch	Switch: Sound of a light switch being flipped.
Sound	Basin_Tap: Sound of a sink faucet being turned on or off.
Sound	Door: Sound of a door opening or closing.

Used Architecture: ACDNet

ACDNet (Audio Classification Deep Network) is a specialized deep learning architecture developed for efficient **environmental sound classification** on resource-constrained edge devices, like microcontrollers. It is smaller, flexible, compression-friendly and more efficient than other pre-existing models

How does ACDNet work?

- Audio Input: ACDNet operates directly on **raw audio waveforms** . The raw audio signal is usually preprocessed (e.g., resampled, normalized) and represented in 1D representation
- Convolutional Layers: ACDNet uses 1**D convolutional layers** in the **Spectral Feature Extraction Block (SFEB)** to process raw audio. These layers extract spectral features from the audio signal, while later layers in the **Temporal Feature Extraction Block (TFEB)** capture higher-level temporal patterns.
- Feature Extraction: Feature extraction in ACDNet is achieved through a combination of spectral and temporal layers. The **SFEB** handles lower-level spectral (frequency) features, while the **TFEB** focuses on higher-level temporal patterns in the audio.
- Classification: The classification stage indeed uses **fully connected (dense) layers**, but it occurs after extensive feature extraction through convolutional layers, with **global average pooling** commonly used to reduce dimensionality before the dense layers.
  - **Dense Layers**: These layers map the extracted features to a final set of class scores.
  - **Softmax Activation**: The final output layer typically uses the **softmax activation function**, which converts the network's output into a probability distribution over the possible classes.



ACDNet is designed for raw audio input, bypassing the need for hand-crafted features (e.g., spectrograms).

Input Layer:

- The model processes the waveform directly, eliminating the need to convert the audio into a spectrogram or any other time-frequency representation.

Spectral Feature Extraction Block (SFEB):

- **Purpose**: This block is responsible for extracting low-level spectral features directly from the raw audio waveform.
- **Layers**:
  - Uses 2**D convolutional layers** with **stride** and **kernel size** configurations optimized to capture low-level frequency patterns.
  - Each 2D convolution is followed by **batch normalization** (to stabilize training) and **ReLU activation** (to introduce non-linearity).
  - **Max pooling layers** follow certain convolutions to downsample the data and focus on essential features.

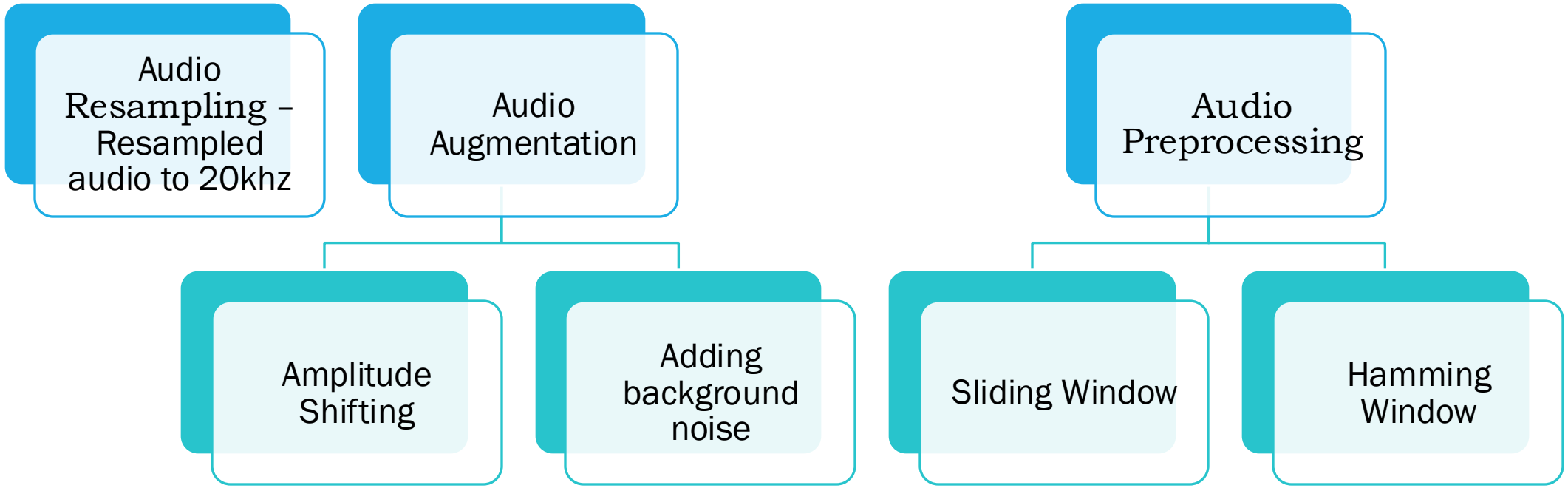
Temporal Feature Extraction Block (TFEB):

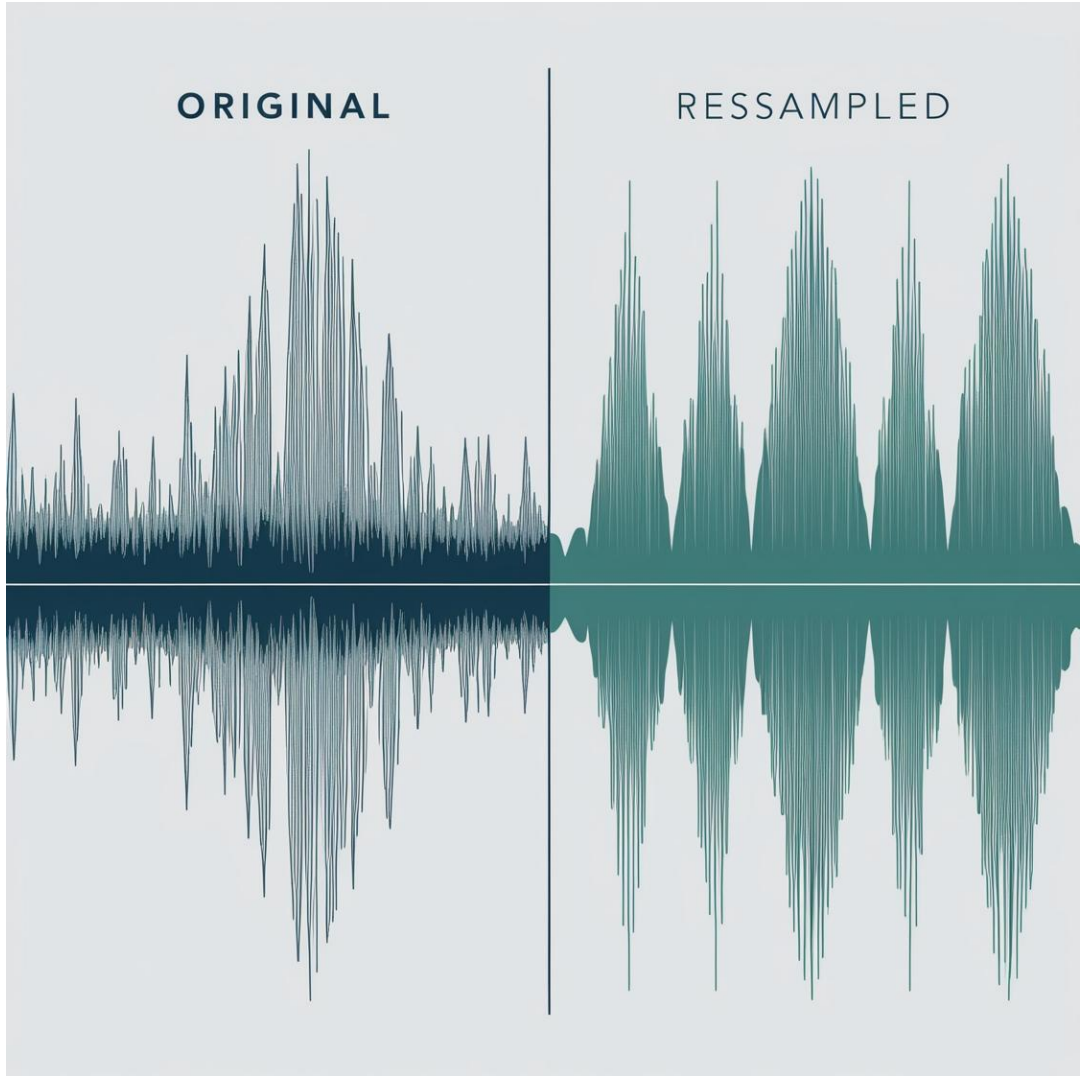
- **Purpose**: This block extracts higher-level temporal (time-based) features that capture patterns over longer durations in the audio signal.
- **Layers**:
  - Composed of additional 2**D convolutions** arranged in a VGG-style architecture, where **convolutional layers** are grouped in pairs, each followed by a **ReLU activation** and a **max pooling layer**.
  - These convolutional and pooling layers allow the model to learn hierarchical and time-based features from the input signal.

Classification and Dense Layers:

- **Purpose**: After feature extraction, the model uses **fully connected (dense) layers** to interpret the extracted features and assign them to audio classes.
- **Layers**:
  - After the TFEB, **global average pooling** is applied to reduce the number of parameters before passing the features to the dense layers.
  - The dense layers map the high-level features to class scores. **ReLU** activations are applied in the intermediate layers to maintain non-linearity.
  - The final dense layer uses **softmax activation** to output a probability distribution over the possible classes.

Preprocessing





Audio Resampling

- Convert all audio samples to 20kHz for consistency.
- Simplifies processing and ensures consistent quality.

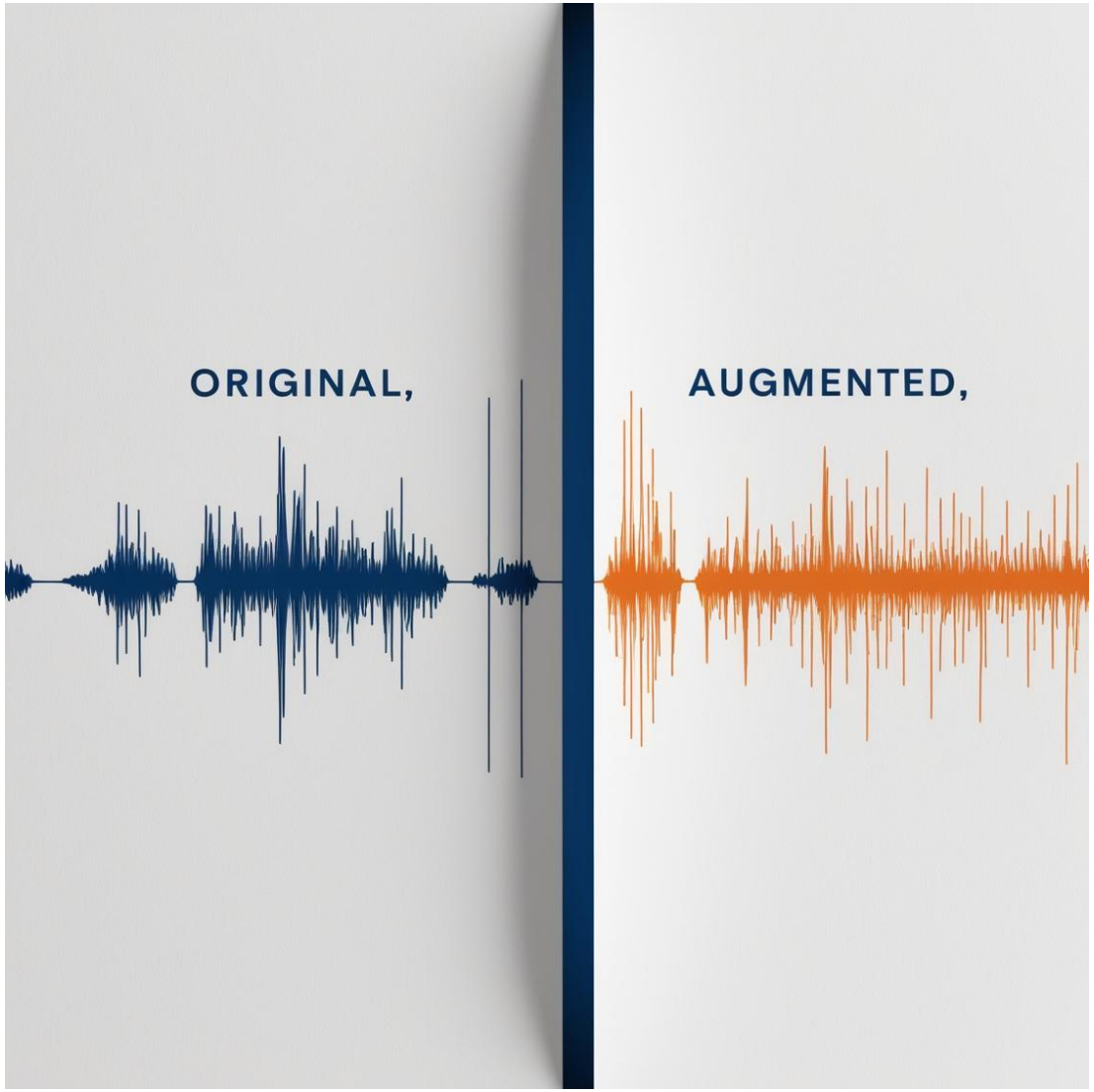
Audio Augmentation

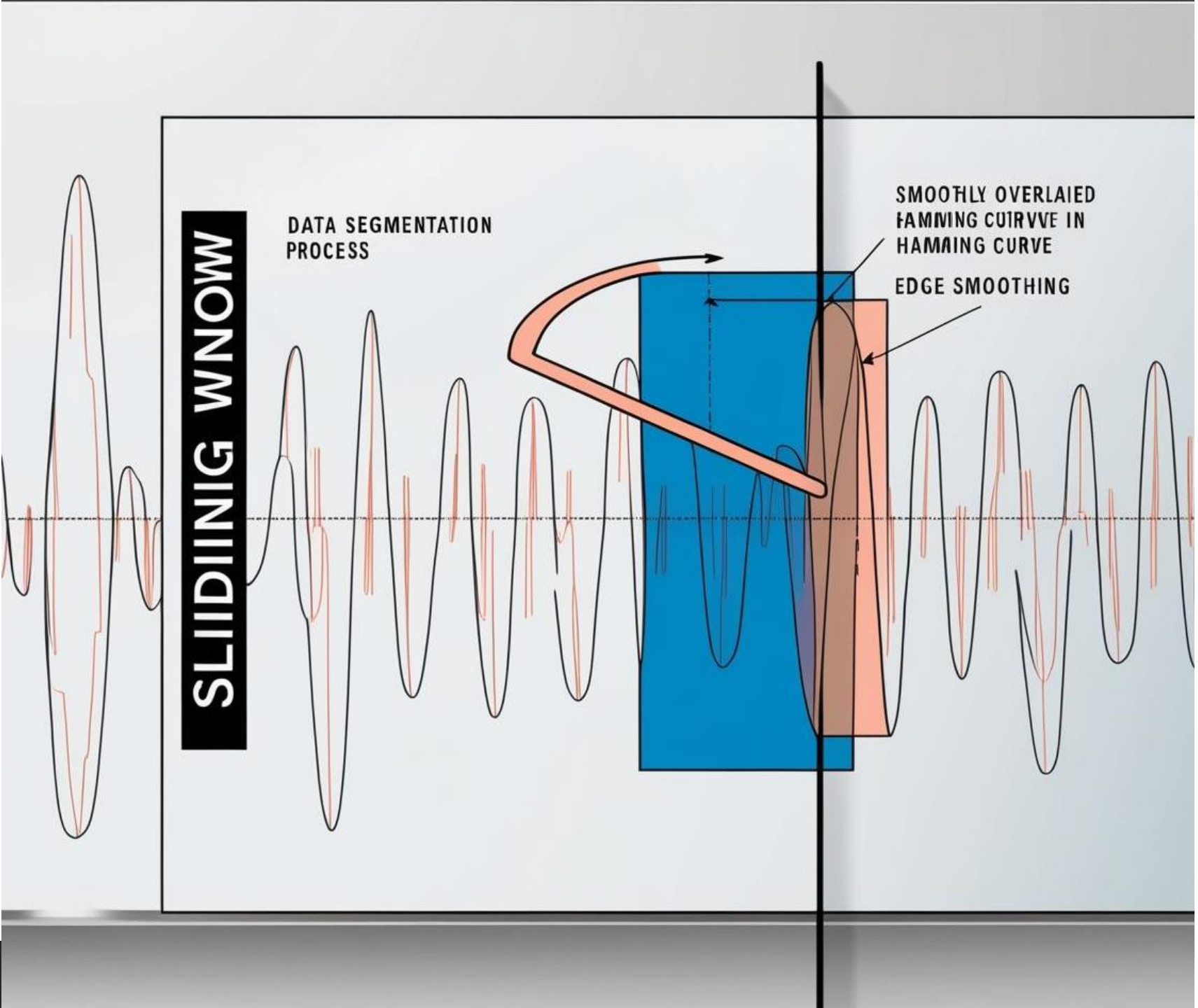
- **Amplitude Shifting:**

  Random volume adjustments.

- **Adding Noise:**

  Adds background noise to simulate real-world conditions.





Audio Preprocessing

- **Sliding Window**: Breaks audio into overlapping segments.
- **Hamming Window**: Smooths edges in each segment to improve feature quality.

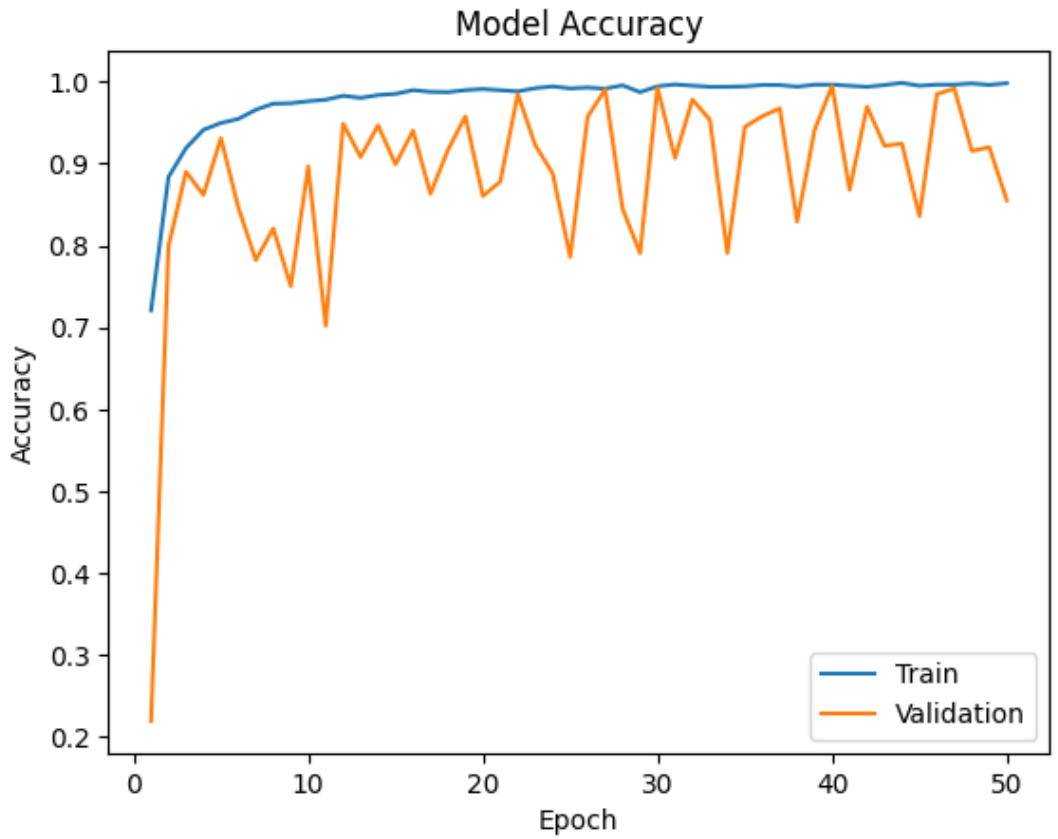
ACDNET
Model description

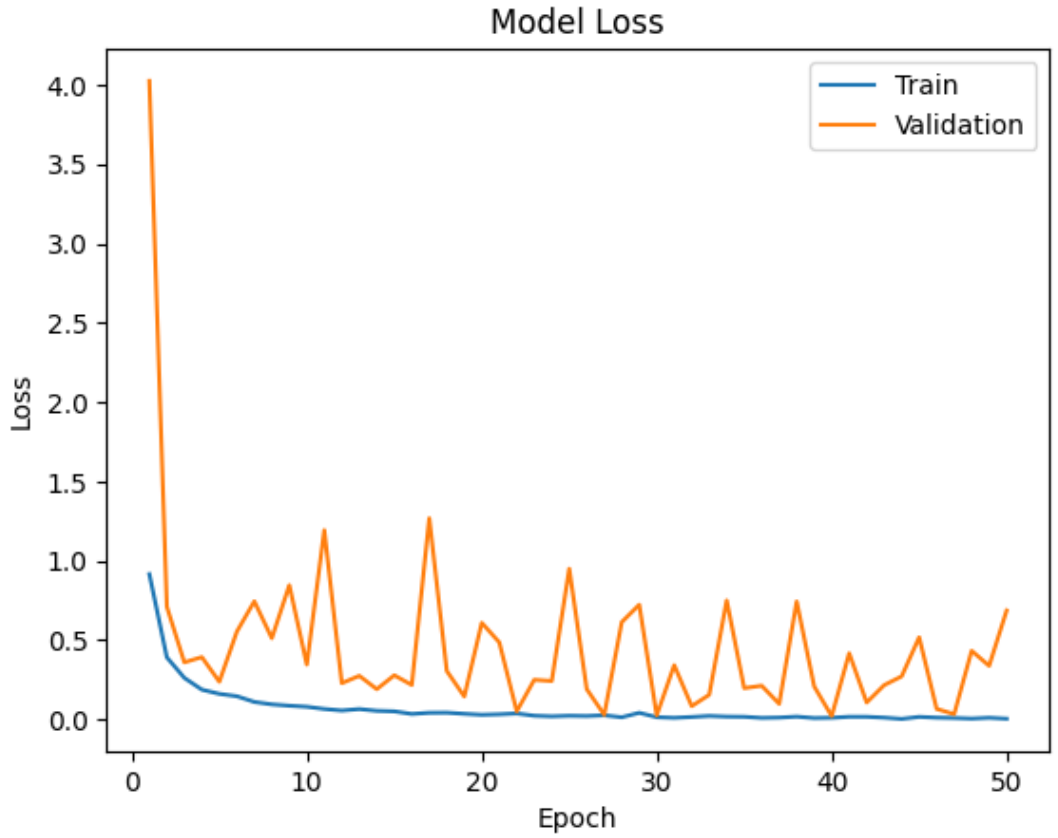
Model size = 17.99MB

Target Model size = <1MB

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 1, 15109, 8)	80
batch_normalization (BatchNormalization)	(None, 1, 15109, 8)	32
conv2d_1 (Conv2D)	(None, 1, 7553, 64)	2,624
batch_normalization_1 (BatchNormalization)	(None, 1, 7553, 64)	256
max_pooling2d (MaxPooling2D)	(None, 1, 151, 64)	0
permute (Permute)	(None, 64, 151, 1)	0
conv2d_2 (Conv2D)	(None, 64, 151, 32)	320
batch_normalization_2 (BatchNormalization)	(None, 64, 151, 32)	128
max_pooling2d_1 (MaxPooling2D)	(None, 32, 75, 32)	0
conv2d_3 (Conv2D)	(None, 32, 75, 64)	18,496
conv2d_4 (Conv2D)	(None, 32, 75, 64)	36,928
batch_normalization_3 (BatchNormalization)	(None, 32, 75, 64)	256
max_pooling2d_2 (MaxPooling2D)	(None, 16, 37, 64)	0
conv2d_5 (Conv2D)	(None, 16, 37, 128)	73,856
conv2d_6 (Conv2D)	(None, 16, 37, 128)	147,584
batch_normalization_4 (BatchNormalization)	(None, 16, 37, 128)	512
max_pooling2d_3 (MaxPooling2D)	(None, 8, 18, 128)	0
conv2d_7 (Conv2D)	(None, 8, 18, 256)	295,168
conv2d_8 (Conv2D)	(None, 8, 18, 256)	590,080
batch_normalization_5 (BatchNormalization)	(None, 8, 18, 256)	1,024
max_pooling2d_4 (MaxPooling2D)	(None, 4, 9, 256)	0
conv2d_9 (Conv2D)	(None, 4, 9, 512)	1,180,160
batch_normalization_6 (BatchNormalization)	(None, 4, 9, 512)	2,048
conv2d_10 (Conv2D)	(None, 4, 9, 512)	2,359,808
batch_normalization_7 (BatchNormalization)	(None, 4, 9, 512)	2,048
max_pooling2d_5 (MaxPooling2D)	(None, 2, 4, 512)	0
dropout (Dropout)	(None, 2, 4, 512)	0
conv2d_11 (Conv2D)	(None, 2, 4, 8)	4,104
batch_normalization_8 (BatchNormalization)	(None, 2, 4, 8)	32
average_pooling2d (AveragePooling2D)	(None, 2, 1, 8)	0
flatten (Flatten)	(None, 16)	0
dense (Dense)	(None, 8)	136
dense_1 (Dense)	(None, 8)	72

Model Performance ACDNET (Validation: Accuracy 85.52%;Loss 0.68)





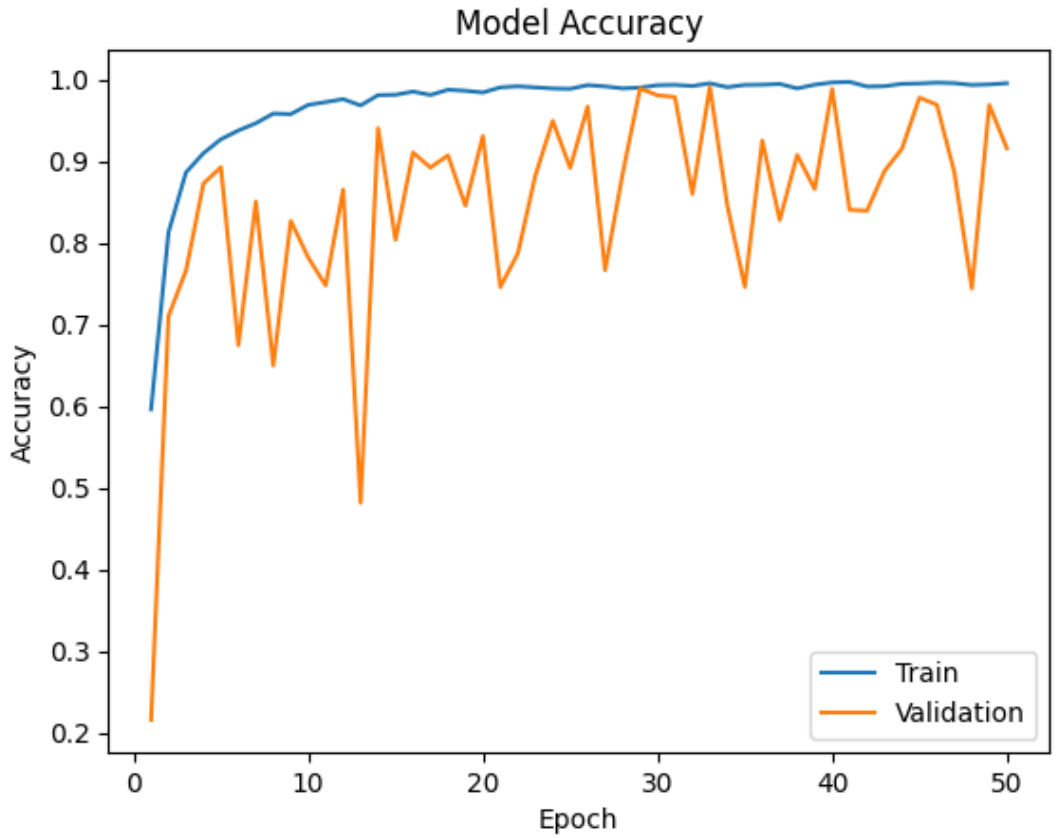
ACDNET + LSTM Model description

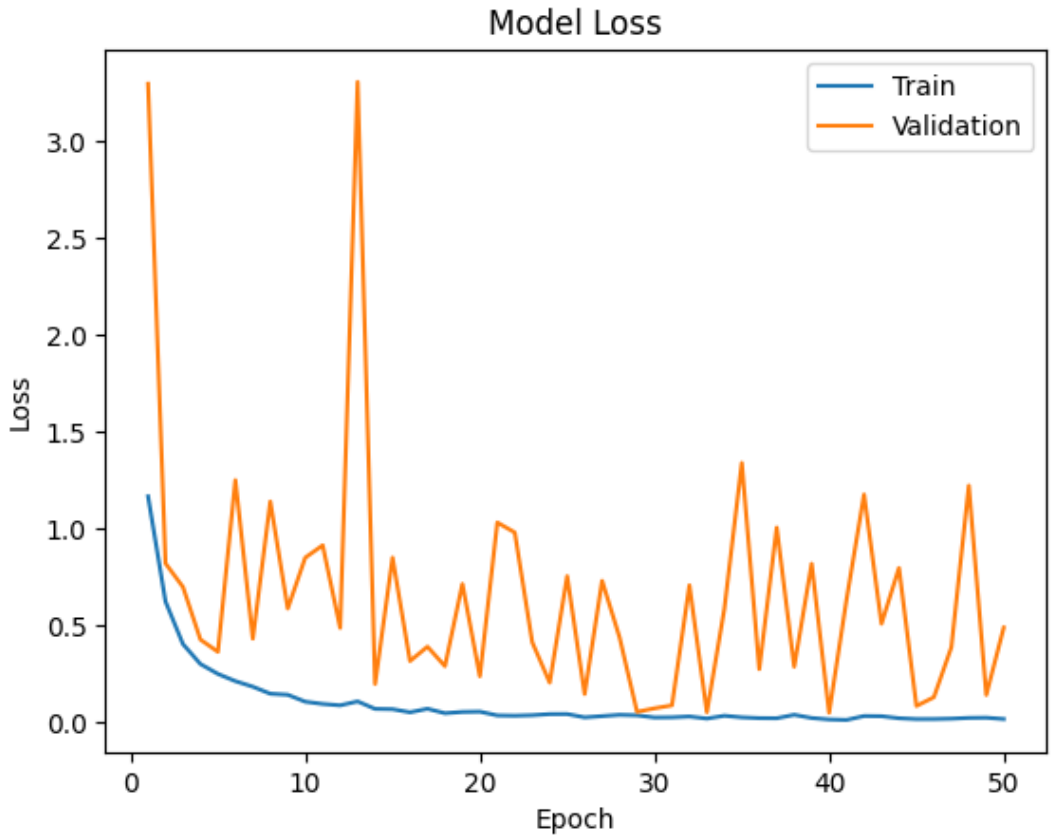
Model size = 18.04MB

Target Model size = <1MB

Layer (type)	Output Shape	Param #
conv2d_144 (Conv2D)	(None, 1, 15109, 8)	80
batch_normalization_128 (BatchNormalization)	(None, 1, 15109, 8)	32
conv2d_145 (Conv2D)	(None, 1, 7553, 64)	2,624
batch_normalization_129 (BatchNormalization)	(None, 1, 7553, 64)	256
max_pooling2d_72 (MaxPooling2D)	(None, 1, 151, 64)	0
permute_12 (Permute)	(None, 64, 151, 1)	0
conv2d_146 (Conv2D)	(None, 64, 151, 32)	320
batch_normalization_130 (BatchNormalization)	(None, 64, 151, 32)	128
max_pooling2d_73 (MaxPooling2D)	(None, 32, 75, 32)	0
conv2d_147 (Conv2D)	(None, 32, 75, 64)	18,496
conv2d_148 (Conv2D)	(None, 32, 75, 64)	36,928
batch_normalization_131 (BatchNormalization)	(None, 32, 75, 64)	256
max_pooling2d_74 (MaxPooling2D)	(None, 16, 37, 64)	0
conv2d_149 (Conv2D)	(None, 16, 37, 128)	73,856
conv2d_150 (Conv2D)	(None, 16, 37, 128)	147,584
batch_normalization_132 (BatchNormalization)	(None, 16, 37, 128)	512
max_pooling2d_75 (MaxPooling2D)	(None, 8, 18, 128)	0
conv2d_151 (Conv2D)	(None, 8, 18, 256)	295,168
conv2d_152 (Conv2D)	(None, 8, 18, 256)	590,080
batch_normalization_133 (BatchNormalization)	(None, 8, 18, 256)	1,024
max_pooling2d_76 (MaxPooling2D)	(None, 4, 9, 256)	0
conv2d_153 (Conv2D)	(None, 4, 9, 512)	1,180,160
batch_normalization_134 (BatchNormalization)	(None, 4, 9, 512)	2,048
conv2d_154 (Conv2D)	(None, 4, 9, 512)	2,359,808
batch_normalization_135 (BatchNormalization)	(None, 4, 9, 512)	2,048
max_pooling2d_77 (MaxPooling2D)	(None, 2, 4, 512)	0
dropout_12 (Dropout)	(None, 2, 4, 512)	0
conv2d_155 (Conv2D)	(None, 2, 4, 32)	16,416
batch_normalization_136 (BatchNormalization)	(None, 2, 4, 32)	128
average_pooling2d_12 (AveragePooling2D)	(None, 2, 1, 32)	0
flatten_12 (Flatten)	(None, 64)	0
reshape_10 (Reshape)	(None, 2, 32)	0
lstm_16 (LSTM)	(None, 2, 8)	1,312
batch_normalization_137 (BatchNormalization)	(None, 2, 8)	32
global_average_pooling1d_20 (GlobalAveragePooling1D)	(None, 8)	0
dense_24 (Dense)	(None, 8)	72
dense_25 (Dense)	(None, 8)	72

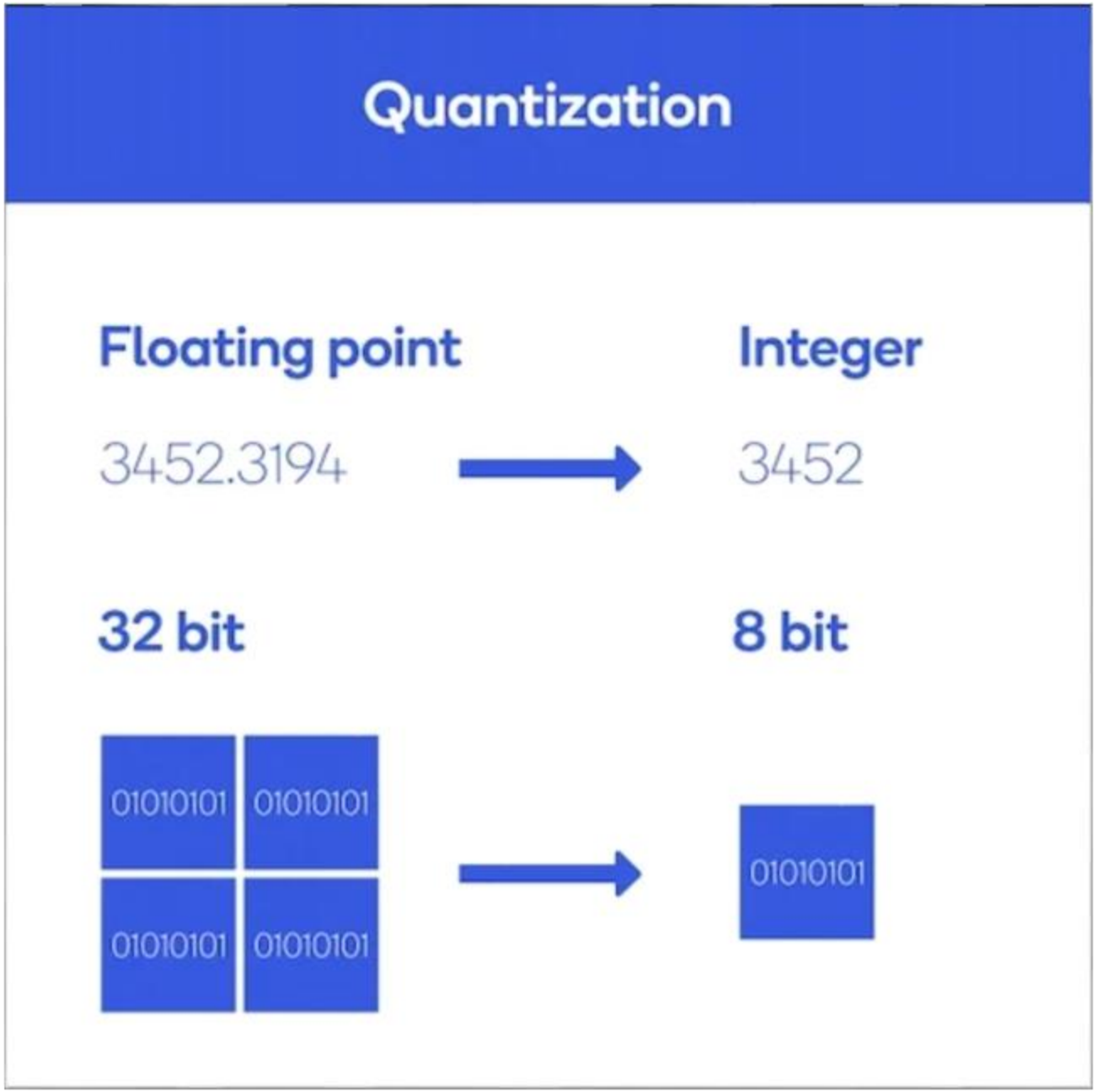
Model Performance ACDNET + LSTM (Validation: Accuracy 91.61%;Loss 0.48)





Pruning

- Quantization
  - converting the model's parameters to lower precision (e.g., INT8 or FP16).



Pruning

Reduce Number of Channels or Filters

- Reduce the number of filters in convolutional layers.
- Smaller hidden dimensions in dense layers.

Model Pruning

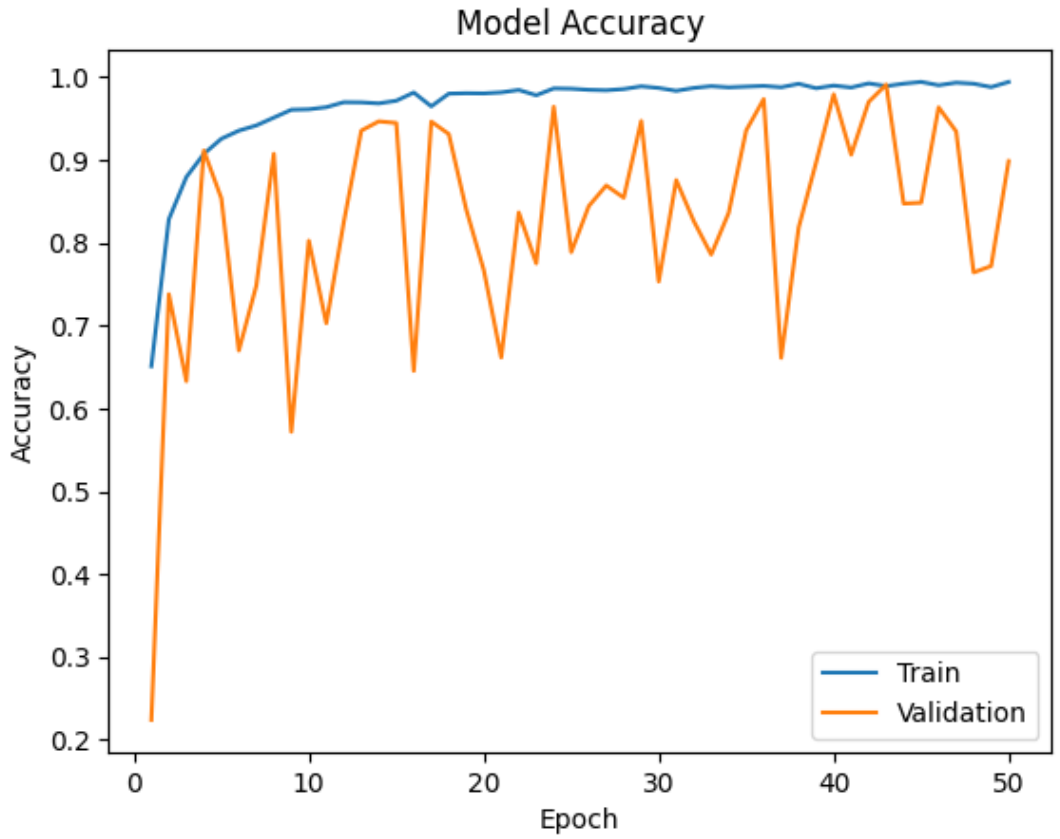
- Remove insignificant layers that contribute little to the final output.

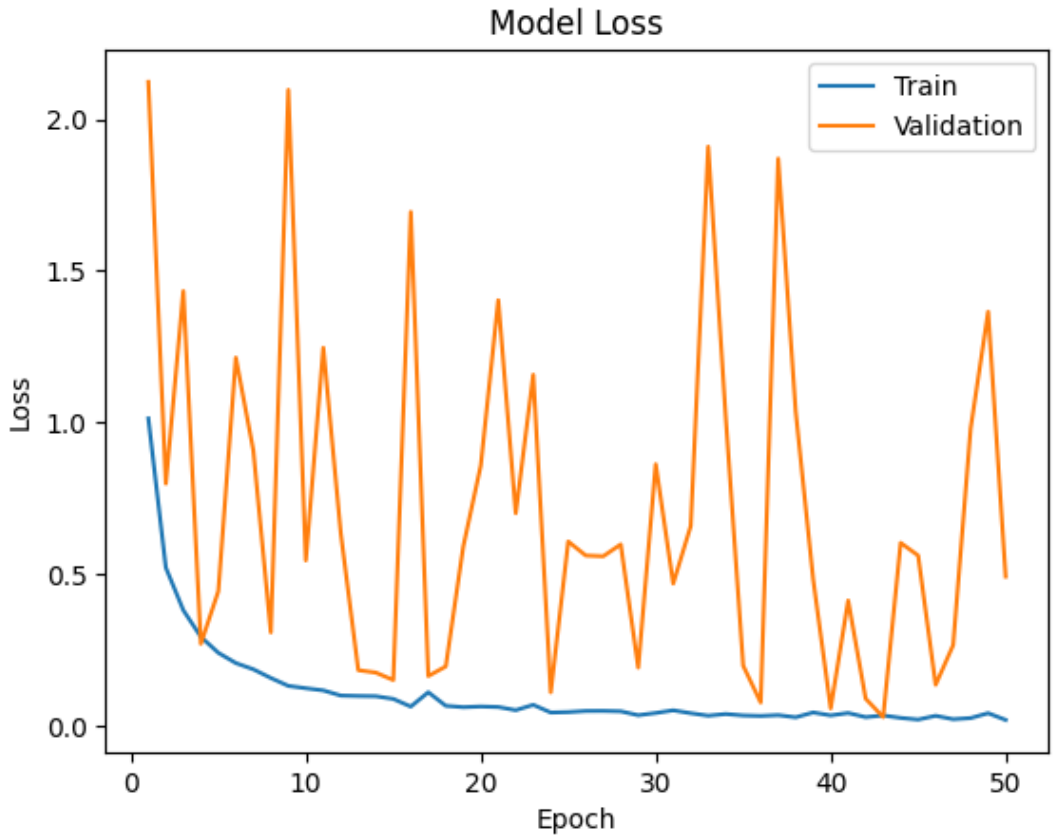
ACDNET Pruned Model description

MODEL SIZE = 238.83 KB

Layer (type)	Output Shape	Param #
conv2d_236 (Conv2D)	(None, 1, 15109, 7)	70
batch_normalization_176 (BatchNormalization)	(None, 1, 15109, 7)	28
conv2d_237 (Conv2D)	(None, 1, 7553, 20)	720
batch_normalization_177 (BatchNormalization)	(None, 1, 7553, 20)	80
max_pooling2d_119 (MaxPooling2D)	(None, 1, 151, 20)	0
permute_20 (Permute)	(None, 20, 151, 1)	0
conv2d_238 (Conv2D)	(None, 20, 151, 10)	100
batch_normalization_178 (BatchNormalization)	(None, 20, 151, 10)	40
max_pooling2d_120 (MaxPooling2D)	(None, 10, 75, 10)	0
conv2d_239 (Conv2D)	(None, 10, 75, 14)	1,274
conv2d_240 (Conv2D)	(None, 10, 75, 22)	2,794
batch_normalization_179 (BatchNormalization)	(None, 10, 75, 22)	88
max_pooling2d_121 (MaxPooling2D)	(None, 5, 37, 22)	0
conv2d_241 (Conv2D)	(None, 5, 37, 31)	6,169
conv2d_242 (Conv2D)	(None, 5, 37, 35)	9,800
batch_normalization_180 (BatchNormalization)	(None, 5, 37, 35)	140
max_pooling2d_122 (MaxPooling2D)	(None, 2, 18, 35)	0
conv2d_243 (Conv2D)	(None, 2, 18, 41)	12,956
conv2d_244 (Conv2D)	(None, 2, 18, 69)	25,530
batch_normalization_181 (BatchNormalization)	(None, 2, 18, 69)	276
max_pooling2d_123 (MaxPooling2D)	(None, 1, 9, 69)	0
batch_normalization_182 (BatchNormalization)	(None, 1, 9, 69)	276
dropout_20 (Dropout)	(None, 1, 9, 69)	0
conv2d_245 (Conv2D)	(None, 1, 9, 8)	560
batch_normalization_183 (BatchNormalization)	(None, 1, 9, 8)	32
average_pooling2d_20 (AveragePooling2D)	(None, 1, 2, 8)	0
flatten_20 (Flatten)	(None, 16)	0
dense_40 (Dense)	(None, 8)	136
dense_41 (Dense)	(None, 8)	72

Model Performance ACDNET Pruned (Validation: Accuracy 89.9%;Loss 0.49)





Real Time Accuracy

Model	Accuracy
ACDNet	16%
ACDNet + LSTM	15%
Quantized ACDNet	14.51%
Pruned ACDNet	9%