Building a Mass Shooting Detector with Crowd Noise Injection

Dimas Tri Kurniawan
dimas.tri01@ui.ac.id

***Abstract*—This is the abstract**

***Index Terms*—This, is, the, keywords**

I. Introduction

This is the introduction.

II. Related Works

These are some related works.

III. Methods

A. Generate Spectrogram

The Short-Time Fourier Transform (STFT) is a widely used signal processing technique that transforms a signal into the time-frequency domain. The purpose of STFT is to analyze how the frequency content of a signal evolves over time by breaking the signal into smaller, overlapping segments.

Consider a continuous-time signal $x(t)$, where $t \notin \mathbb{R}$ represents time. The STFT is defined as the Fourier transform of the windowed signal, which is effectively a local frequency representation of the signal. Mathematically, the STFT of $x(t)$ is given by:

$$\text{STFT}_x(t, f) = \int_{-\infty}^{\infty} x(\tau) w(\tau - t) e^{-j2\pi f \tau} d\tau \tag{1}$$

where t represents time, f represents frequency, $x(t)$ is the signal to be analyzed, $w(t)$ is a window function, which typically localizes the signal in time, ensuring that the transform captures a time-frequency representation, and $e^{-j2\pi f \tau}$ is the complex exponential kernel that represents the oscillatory behavior of the Fourier transform.

In practice, the signal $x(t)$ is sampled at discrete intervals $t_n = n\Delta t$ where Δt is the sampling period. Additionally, the signal is typically analyzed over a finite time window, with the window shifting along the time axis. This leads to the following discrete formulation of the STFT:

$$\text{STFT}_x(n, m) = \sum_{k=0}^{L-1} x[n - k] w[k] e^{-j\frac{2\pi m k}{L}} \tag{2}$$

where n is the time index corresponding to the n-th segment of the signal, m is the frequency bin index corresponding to the m-th frequency component, L is the length of the window $w[k]$, which typically has a length of N (the size of the analysis window). The signal $x[n - k]$ is a sampled signal at discrete indices n and k. The window $w[k]$ is applied to each segment to smooth the signal and reduce edge effects, The exponential term $e^{-j2\pi \frac{mk}{L}}$ is the discrete Fourier basis function. The window function $w[k]$ is typically chosen to minimize spectral leakage. Common window choices include the Hamming, Hanning, or Blackman-Harris windows. The length of the window L determines the tradeoff between time resolution and frequency resolution. A longer window provides better frequency resolution but poorer time resolution, whereas a shorter window gives higher time resolution but lower frequency resolution.

The spectrogram is the discrete version of the STFT for a real-valued signal x. It works by first dividing the signal into overlapping frames of size n, each of which is multiplied by a window function of length m. The STFT is then computed by applying the Fast Fourier Transform (FFT) to each windowed segment.

B. Convolutional Layer

A spectrogram contains both spatial and temporal information. The spatial dimension corresponds to frequency content (vertical axis), while the temporal dimension corresponds to the time evolution of the signal (horizontal axis). Convolutional layers on spectrograms exploit this structure by learning both time-dependent and frequency-dependent patterns.

To reduce the dimensionality of the feature maps and improve computational efficiency, CNNs typically use pooling layers (e.g., max pooling or average pooling). Pooling layers downsample the feature maps by summarizing local regions, which helps to make the network invariant to small translations and distortions in the input. In the case of audio spectrograms, pooling operations also allow the network to focus on the most salient features and avoid overfitting.

After passing through the convolutional and pooling layers, the resulting feature maps are flattened and fed into one or more fully connected layers. These layers serve to integrate the learned features and produce a final decision, typically in the form of class probabilities for different audio categories. In an audio classification task, the output layer could consist of a softmax activation function that produces a probability distribution over predefined categories. The network is trained end-to-end, with the weights of the convolutional filters and fully connected layers being optimized using backpropagation and gradient descent.

IV. Experiments

A. Dataset Setup

The data used for our for our model was sourced from the Kaggle: Gunshot Audio Dataset. It is a dataset containing

gunshot audios from 8 different type of guns. These were collected on YouTube using videos that are open to everyone.

It is quite challenging to obtain a dataset of live mass shooting audio recordings, where not only the sounds of gunfire are present, but also background screaming. In addition, various audio clips from YouTube will be used to supplement the background screaming sounds. Subsequently, the screaming audio will be injected into the mass shooting audio to create a recording that closely resembles the actual atmosphere.

B. Training Model

The model accepts color images and first normalizes their pixel values, ensuring a consistent scale for learning. Its backbone is a stack of convolutional layers with nonlinear activations interleaved with spatial downsampling; this design learns hierarchical features, starting from local patterns and progressing to more abstract representations. The feature maps are then flattened and passed to a fully connected projection that culminates in a final layer producing a two-class probability distribution via a normalization function.

This study employs three models, each trained using different datasets. Model 1 is trained from scratch using the provided dataset. Upon completion of the training process, the model is saved in Keras format. Model 2 is a fine-tuning of Model 1, achieved by incorporating audio data resulting from noise injection into the dataset. This is done by mixing non-gunshot audio (background audio) with gunshot audio. Model 3 is also a fine-tuning of Model 1, with modifications applied to the energy of each audio sample.

Training optimizes a cross-entropy objective using stochastic gradient descent and tracks accuracy as the primary metric. In the current configuration, the loss is set to expect unnormalized scores while the model outputs normalized probabilities; aligning the loss with the output (or vice versa) would provide more faithful gradient signals and typically yields better calibration. During each optimization step, gradients flow from the classification output back through the dense and convolutional layers, nudging filters to respond more strongly to task-relevant structures and suppress distractors.

The schedule runs for a bounded number of epochs with early stopping enabled. The stopping criterion monitors training accuracy and restores the best weights encountered during learning. While this curbs redundant updates once performance plateaus, monitoring a held-out validation signal would more directly promote generalization and reduce overfitting.

C. Evaluation Setup

For the evaluation of the model's performance, 3 primary metrics were used: accuracy and recall. Accuracy is one of the most commonly used metrics for evaluating the performance of classification models, particularly in the context of supervised learning. It provides a simple and intuitive measure of how often the model correctly predicts the class label for a given set of instances. The general definition of accuracy is the ratio of correctly predicted instances to the total number of instances in the dataset.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{3}$$

The accuracy metric assumes that the cost of false positives and false negatives is equal, as it does not differentiate between the types of errors a model makes. However, this assumption is not suitable for the problem because false negatives are more dangerous than false positives. We want the model to detect shooting incidents accurately if they indeed occur. Therefore, we will use an additional metric, which is recall.

Recall, also known as sensitivity or true positive rate, is a performance metric used to evaluate the effectiveness of a classification model, particularly in contexts where it is important to identify as many positive instances as possible. It measures the proportion of actual positive cases that the model correctly identifies. A high recall indicates that the model is good at detecting positive cases, minimizing false negatives.

$$Recall = \frac{TruePositives}{True\ Positives + False\ Negatives} \tag{4}$$

V. Results and Analysis

Interestingly, all three models exhibit comparable prediction quality. Evaluation of the models using the testing dataset yields identical accuracy and recall scores. This indicates that the developed models are highly suitable for the audio classification task in this study.

Model	Accuracy	Recall
Model 1	0.99713	1
Model 2	0.99713	1
Model 3	0.99713	1

Furthermore, the performance of all three models is not influenced by variations in audio energy levels, whether in gunshot or non-gunshot audio. This characteristic is particularly important, as in real-world scenarios, gunshot sounds are not always louder than background noise. For example, gunshot audio may be overshadowed by loud crowd noises, or it may originate from a considerable distance from the detection system.

The perfect recall achieved by the models indicates their ability to detect gunshot events without failure whenever such events occur. High recall is crucial in this study, as misclassification of gunshot sounds could lead to an increased number of casualties. With a perfect recall score, the models successfully meet the expected performance criteria and are ready for deployment.