

Desain Sistem Deteksi Kecelakaan Lalu Lintas Berbasis Suara dengan CNN pada platform Embedded Artificial Intelligence

Ahmada Haiz Zakiyil Ilahi¹, Naufal Mukhfi Robbani², Arif Irwansyah³, Budi Nur Iman⁴

haizahmada2001@pasca.student.pens.ac.id¹, noufalmukhfi@gmail.com², arif@pens.ac.id³,

alfaruqi@pens.ac.id⁴

^{1,2,3,4} Politeknik Elektronika Negeri Surabaya

Informasi Artikel

Diterima : 8 Agu 2024
Direvisi : 13 Agu 2024
Disetujui : 3 Sep 2024

Kata Kunci

Sistem Deteksi
Kecelakaan, Klasifikasi
Audio, Mel Spectrogram,
Convolutional Neural
Network, Raspberry Pi 4,
Telegram

Abstrak

Kecelakaan lalu lintas sering kali mengakibatkan kerugian besar, termasuk kehilangan nyawa. Banyak korban jatuh karena penanganan kecelakaan yang tidak memadai, seperti keterlambatan dalam memberikan informasi dan lokasi kecelakaan yang sulit dijangkau sehingga membuat korban menjadi tidak tertolong. Penelitian ini bertujuan mengatasi permasalahan yang disebabkan oleh kecelakaan lalu lintas dengan merancang dan menciptakan alat deteksi berbasis suara menggunakan metode Convolutional Neural Network (CNN) pada platform embedded artificial intelligence yang memanfaatkan Raspberry Pi 4. Alat ini memanfaatkan usb microphone dan GPS Ublox Neo-m8n untuk menangkap suara serta pengiriman data koordinat secara real-time. Dataset dibagi menjadi dua kelas yaitu crash dan normal, kemudian dataset diolah melalui augmentasi dan divisualisasikan sebagai Mel Spectrogram. Model yang didapat mencapai akurasi 98,63% dan loss 1,37% pada prediksi data testing, sementara implementasi real-time pada Raspberry Pi 4 dengan usb microphone menghasilkan akurasi 82% dari 50 sampel file audio. Sistem ini beroperasi dengan FPS rata-rata sebesar 13.14 untuk proses streaming dan 6.89 untuk proses prediksi serta dihasilkan konsumsi daya rata-rata sebesar 10.069 Watt. Output yang dihasilkan berupa pesan peringatan melalui aplikasi Telegram berupa informasi lokasi dan waktu kecelakaan lalu lintas. Penelitian ini diharapkan dapat meningkatkan respons dalam penanganan kecelakaan, mengurangi kerugian, dan menyelamatkan nyawa.

Keywords

Accident Detection System, Audio Classification, Mel Spectrogram, Convolutional Neural Network, Raspberry Pi 4, Telegram

Abstract

Traffic accidents often resulted in significant losses, including loss of life. Many victims perished due to inadequate accident response, such as delays in providing information and hard-to-reach accident locations, leaving the victims without help. This study aimed to address the issues caused by traffic accidents by designing and creating a sound-based detection tool using Convolutional Neural Network (CNN) on a Raspberry Pi 4. The tool utilized a USB microphone and GPS Ublox Neo-m8n to capture sound and send coordinate data in real time. The dataset was divided into two classes: crash and normal, then processed through augmentation and visualized as Mel Spectrogram. The resulting model achieved an accuracy of 98.63% and a loss of 1.37% on test data predictions, while real-time implementation on the Raspberry Pi 4 with a USB microphone achieved an accuracy of 82% from 50 audio sample files. The system operated with an average FPS of 13.14 for streaming and 6.89 for prediction, and it had an average power consumption of 10.069 Watts. The output consisted of warning messages sent via the Telegram application, providing information on the location and time of traffic accidents. This study was expected to improve response times in handling accidents, reduce losses, and save lives.

A. Pendahuluan

Transportasi merupakan salah satu kebutuhan utama dalam kehidupan sehari-hari, namun meningkatnya jumlah alat transportasi juga berdampak pada tingginya kecelakaan lalu lintas. Data Kementerian Perhubungan mencatat peningkatan kasus kecelakaan di Indonesia, mencapai 103.645 kasus dengan 25.266 korban jiwa pada tahun 2021 [1]. Banyaknya korban diakibatkan buruknya penanganan kecelakaan seperti keterlambatan informasi serta tempat yang jauh dan sulit dijangkau sehingga sering membuat korban menjadi tidak tertolong. Respons cepat terhadap kecelakaan menjadi kunci utama dalam menyelamatkan korban, dan teknologi memiliki peranan vital dalam meningkatkan respons tersebut [2]. Dengan adanya permasalahan tersebut, diperlukan adanya sebuah alat yang dapat dipasang pada tempat tertentu untuk mendeteksi kecelakaan lalu lintas atau kejadian darurat di tempat tersebut secara *real-time* agar korban yang tidak tertolong dapat diminimalisir.

Metode klasifikasi suara sangat bergantung pada teknik pemrosesan sinyal untuk mengekstraksi fitur dan menggunakan model deep learning seperti convolutional neural network (CNN) [3], [4] untuk mempelajari pola kompleks atau fitur relevan dari data suara. Namun, kebisingan sering menjadi masalah saat mengelompokkan jenis suara tertentu, seperti suara tabrakan di lingkungan bising seperti area perkotaan atau lalu lintas padat [5]. Oleh karena itu, teknik manipulasi data dapat diterapkan untuk mengatasi tantangan ini dan menghasilkan data yang lebih beragam [6], [7].

Dalam beberapa tahun terakhir, teknologi telah berperan penting dalam peningkatan respons kecelakaan lalu lintas. Berbagai solusi muncul dalam mendeteksi kecelakaan lalu lintas berdasarkan analisis suara serta pendekatan deep learning. Misalnya, studi [8] memperkenalkan model Deep CNN yang menggunakan data audio seperti MFCC dan teknik augmentasi data. Hasilnya, model ini mencapai akurasi 94%. Studi [9] membandingkan CNN, RNN, dan DNN dengan dataset audio yang mengandung suara bising perkotaan. Arsitektur CNN terbaik memiliki durasi segmen 200ms, rasio tumpang tindih 0,5, frekuensi cut-off bandpass 100Hz hingga 12000Hz, dan ambang batas 0,9. Arsitektur ini mencapai akurasi 93,337%. Studi [10] menunjukkan kinerja deteksi secara real-time pada perangkat NVIDIA Jetson Nano. Akurasi deteksi mencapai 99,84% dengan error 0,3%, meskipun diuji di lingkungan dengan sedikit noise, sehingga kurang mewakili kondisi jalan raya sesungguhnya.

Sistem yang diusulkan dalam penelitian ini merupakan perancangan alat deteksi kecelakaan lalu lintas berbasis suara menggunakan CNN yang dilengkapi dengan teknik segmentasi dan augmentasi audio pada dataset. Dengan adanya alat ini, diharapkan dapat meminimalisir jumlah korban yang tidak tertolong dengan memberikan notifikasi secara real-time melalui aplikasi Telegram serta menyertakan informasi dari lokasi kecelakaan. Penelitian ini juga melibatkan implementasi pada Raspberry Pi 4 dengan usb microphone dan GPS Ublox Neo-M8N untuk pengujian secara real-time dalam kondisi indoor maupun outdoor. Dengan demikian, penelitian ini membuka potensi untuk memberikan respons cepat dan efektif terhadap kecelakaan lalu lintas, mengurangi kerugian, dan meningkatkan keselamatan pengguna jalan.

B. Metode Penelitian

Metode penelitian ini mencakup beberapa tahapan dalam membuat sistem deteksi suara kecelakaan mobil yang terintegrasi pada suatu perangkat mikroprosesor. Berikut merupakan penjelasan detail mengenai tahapan dalam pembuatan sistem pada penelitian ini.

1. Pembuatan Dataset

Tahapan awal dalam perancangan sistem melibatkan pembuatan dataset yang terdiri dari suara kecelakaan dan suara normal. Data audio yang digunakan dalam penelitian ini merupakan kumpulan dari berbagai rekaman dalam format .wav yang diperoleh dari berbagai sumber referensi [11]-[15]. Data audio dipecah menjadi potongan dengan durasi 3 detik dan diubah sampling rate-nya menjadi 22050 Hz. Teknik augmentasi data digunakan untuk meningkatkan jumlah dan variasi data pada kelas tabrakan agar terhindar dari masalah overfitting saat proses pelatihan. Data augmentasi tersebut dihasilkan melalui bantuan library dari Audiomentations [16] dengan beberapa pengaturan yang telah ditetapkan, sebagai berikut:

- *Background Noise*

Pada dataset, ditambahkan beberapa suara latar belakang seperti kebisingan Gaussian, suara hujan, kerumunan, petir, sirene, klakson mobil, angin, dan teriakan. Kebisingan Gaussian diberlakukan dengan amplitudo minimal dan maksimal 0.001 hingga 0.015, sedangkan suara latar belakang lainnya diberlakukan dengan rasio sinyal-ke-kebisingan (snr) minimal dan maksimal masing-masing 3 dB hingga 30 dB.

- *Pitch Shift*

Proses yang mengubah nada atau pitch data audio dengan menggesernya ke atas atau ke bawah tanpa mengubah tempo. Nilai semitone minimum dan maksimum diatur secara acak pada rentang -4 hingga 4.

- *Shift*

Proses yang memindahkan atau menggeser sampel audio ke depan atau ke belakang pada sumbu waktu. Proses ini memungkinkan pengaturan nilai pergeseran minimum dan maksimum, yaitu -5 dan 5, atau setengah panjang audio.

- *Time Stretch*

Fitur ini memungkinkan penggunaan tempo yang lebih cepat atau lebih lambat pada data audio tanpa mempengaruhi nada atau pitch. Nilai minimum dan maksimum dari rate telah diatur secara acak pada rentang 0,8 hingga 1,25.

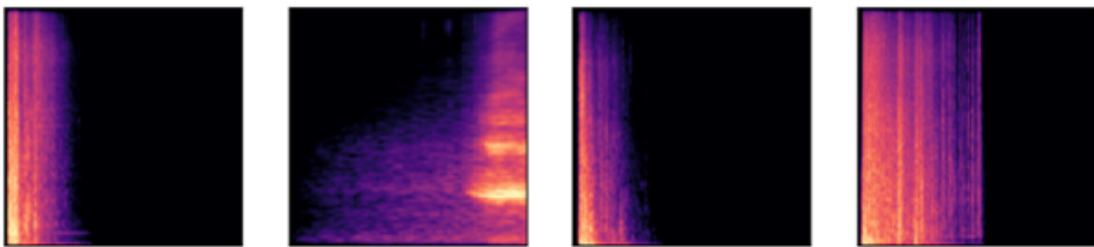
Tabel 1 Augmentasi Dataset

Kelas	Total Sampel	Hasil Augmentasi	Pembagian Data		
			Train	Validation	Test
Crash	225	3060	2142	612	306
Normal	3537	3537	2476	708	353

Proses augmentasi data memainkan peran kunci dalam menjaga proporsi data yang seimbang pada setiap label. Sebelum dimasukkan ke dalam model, seluruh data audio melewati tahap preprocessing terlebih dahulu. Data dibagi menjadi subset latihan, validasi, dan pengujian dengan rasio 7:2:1. Pembagian data ini menggunakan metode stratifikasi dan dilakukan secara acak. Rincian lebih lanjut mengenai pembagian dataset dapat ditemukan pada Tabel 1.

2. Preprocessing Data

Kemudian langkah selanjutnya dilakukan proses preprocessing data yang melibatkan transformasi Short Time Fourier (STFT) untuk menghasilkan mel spektrogram sebagai fitur utama yang dinormalisasi. Proses transformasi tersebut diimplementasikan melalui library Librosa dari Python [17]. Hasil transformasi tersebut disimpan dalam suatu file gambar PNG, kemudian input gambar tersebut dilakukan *resize* dengan lebar dan tinggi sebesar 224x224 dengan format RGB. Selain itu, data juga dinormalisasi untuk mengatasi masalah numerik dan mempercepat konvergensi model. Teknik normalisasi data dilakukan dengan membagi nilai piksel dengan 255. Hasil preprocessing data ditunjukkan pada gambar 1.

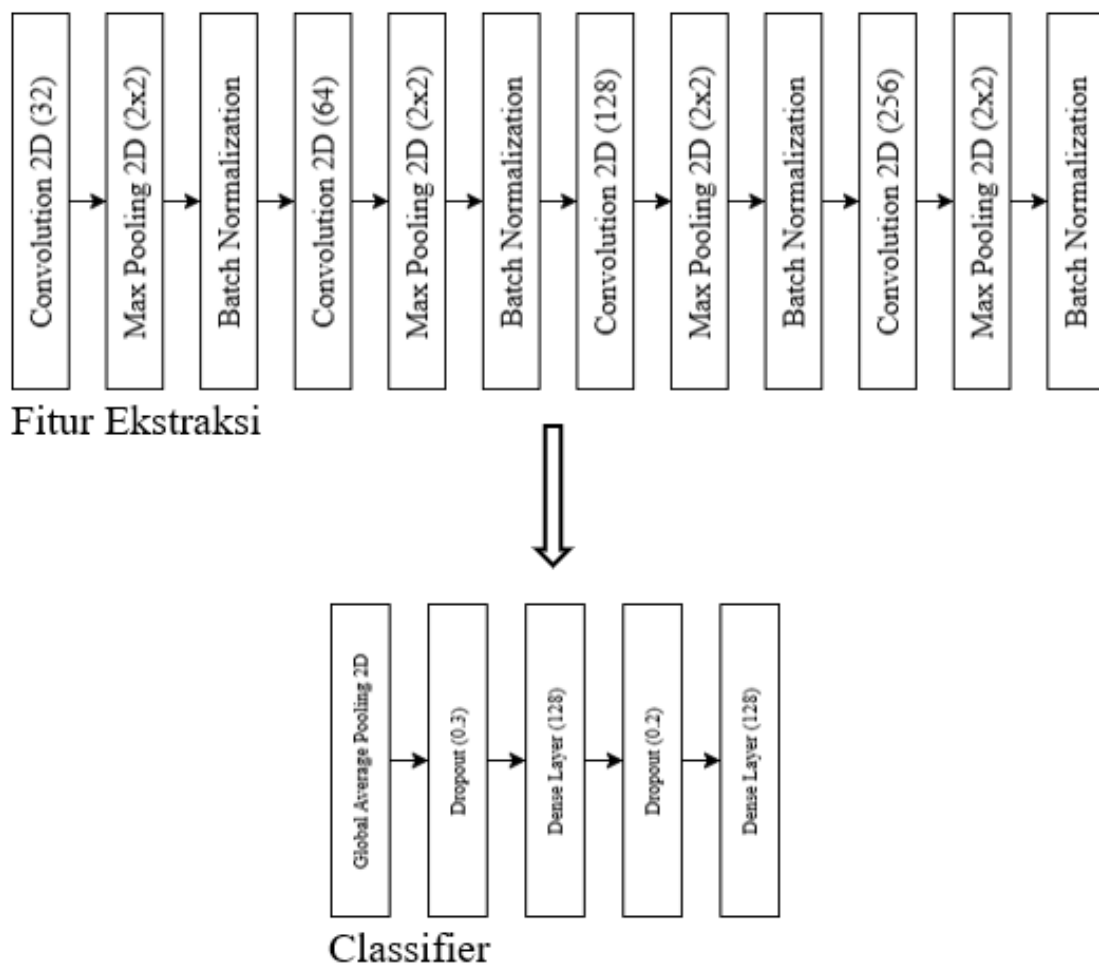


Gambar 1 Sampel Gambar Mel-Spektrogram

3. Pembuatan Model

CNN adalah salah satu metode yang termasuk dalam jaringan syaraf tiruan yang digunakan dalam pengolahan citra untuk tujuan pengenalan dan klasifikasi objek pada gambar. Pendekatan ini melibatkan penggunaan operasi konvolusi dengan menggunakan sebuah kernel atau filter dengan ukuran dan nilai tertentu yang diterapkan pada gambar. Kernel ini berisi bobot (weight), bias, dan lapisan aktivasi yang dilatih menggunakan jaringan syaraf tiruan. Kernel berfungsi sebagai filter yang nilai-nilainya disesuaikan melalui proses pelatihan sehingga output dari operasi konvolusi dapat digunakan untuk mendeteksi dan mengklasifikasi objek.

Untuk arsitektur dari model penelitian ini, model CNN dirancang dengan empat lapisan konvolusi dengan filter 32, 64, 128, dan 256, dengan kernel berukuran 3x3, strides 1x1, padding same, dan fungsi aktivasi relu. Pada setiap lapisan konvolusi, dilakukan max pooling dengan ukuran 2x2 dan penerapan batch normalization untuk meningkatkan kinerja dan stabilitas selama pelatihan. Kemudian digunakan Global Average Pooling 2D untuk mengambil nilai rata-rata dari saluran konvolusi sebelumnya, dan menerapkan dropout sebesar 0,3 untuk mengurangi jumlah parameter pada model dan mencegah overfitting. Selanjutnya, kami menambahkan lapisan dense dengan 64 neuron dan fungsi aktivasi relu, serta dropout sebesar 0,2. Pada bagian classifier, model ini memiliki 2 lapisan output dengan fungsi aktivasi softmax. Total parameter pada model kustom ini adalah 404.994 yang semuanya dapat di-train. Bentuk arsitektur CNN tersebut dapat dilihat pada Gambar 2



Gambar 2 Arsitektur CNN untuk Model

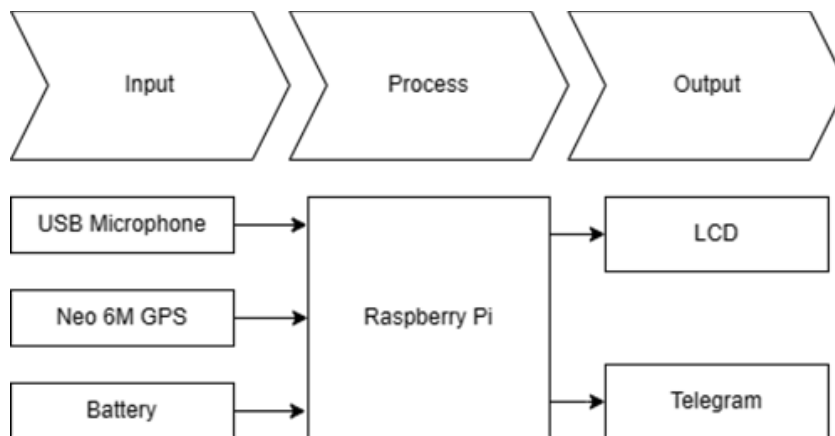
```
...
Total params: 404,994
Trainable params: 404,994
Non-trainable params: 0
```

Gambar 3 Parameter pada Model

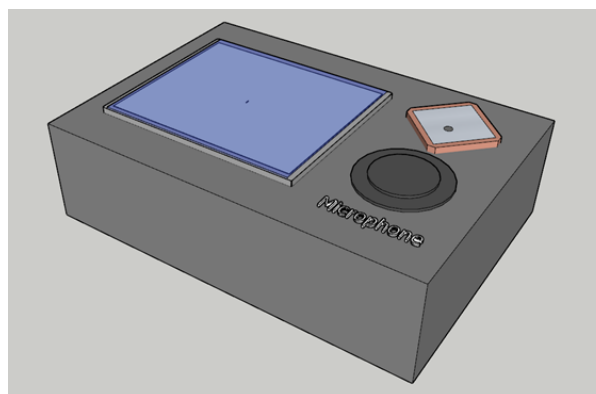
Pada gambar 3 menunjukkan total parameter pada model mencapai 404,994. Parameter dalam konteks ini merujuk pada bobot dan bias yang ada dalam jaringan saraf yang harus dipelajari selama proses pelatihan. Jumlah total parameter ini mencakup semua parameter yang dapat diubah dan disesuaikan selama pelatihan. Model ini dilatih dengan beberapa pengaturan dari hyperparameter diantaranya optimizer Adam, learning rate 0.001, epoch sebanyak 100, dan batch size sebesar 10. Selama proses pelatihan, bobot dengan hasil evaluasi terbaik disimpan dalam suatu file model (.h) untuk dideploy pada perangkat eksperimen.

4. Perancangan Keseluruhan Sistem

Gambaran umum keseluruhan sistem diilustrasikan pada gambar 4, terlihat bahwa pada hardware terdiri dari bagian input, bagian proses, serta bagian output. Adapun untuk input sendiri menggunakan sensor usb microphone dan GPS Ublox NEO M8N. Kemudian pada bagian kontroler utama, sistem menggunakan mikrokontroler Raspberry Pi 4. Jika terdeteksi adanya kecelakaan yang terjadi, hasil pembacaan dari sensor GPS yang terpasang pada alat dikirimkan ke kontroler utama dengan komunikasi serial. Selanjutnya data yang diterima oleh kontroler utama akan dikirim melalui notifikasi Telegram. Hardware kemudian dirakit dan disusun didalam box agar dapat menjalankan fungsi sesuai dengan bagiannya masing-masing. Bentuk yang seminimalis mungkin dengan fungsi yang sesuai menjadi hal yang sangat penting desain keseluruhan sistem, seperti yang ditunjukkan pada gambar 5.



Gambar 4 Blok Diagram Sistem



Gambar 5 Desain Alat yang Dibuat

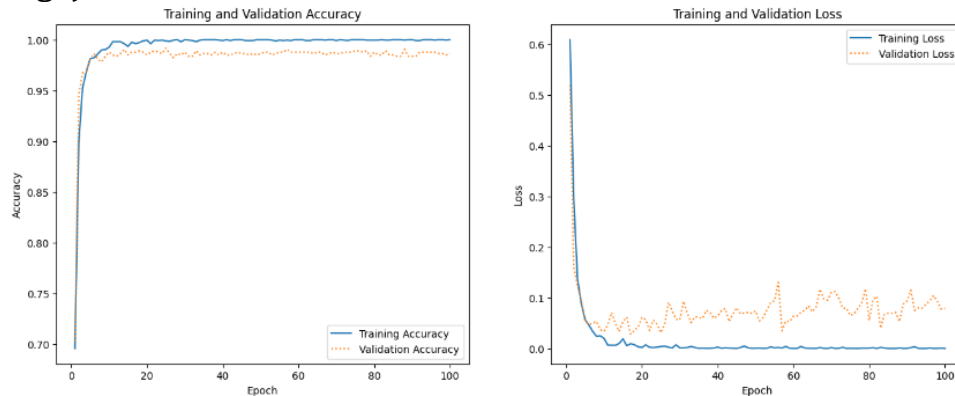
Dalam eksperimen ini, digunakan jenis alat perekaman suara berupa mikrofon omnidirectional dengan respons frekuensi 100Hz-16kHz. Proses perekaman suara dan plot data dalam eksperimen ini menggunakan sampling rate sebesar 22050 Hz. Proses deteksi pada sistem ini melibatkan dua tugas yang beroperasi secara serial, yaitu *streaming* dan prediksi. Proses *streaming* tersebut terjadi pengambilan input suara melalui mikrofon yang menghasilkan data analog untuk dikonversi ke bentuk

gambar *mel-spectrogram* serta dilakukan preprocessing data, seperti *resize* dan normalisasi. Setelah, dilanjutkan proses prediksi yaitu data yang telah diolah sebelumnya diprediksi dengan model CNN yang telah dilatih. Hasil prediksi tersebut menghasilkan sebuah nilai yang menunjukkan probabilitas dari masing-masing kelas, kemudian nilai tersebut disimpulkan berdasarkan nilai terbesar diantaranya. Output dari deteksi ini menghasilkan informasi adanya tabrakan ataupun normal. Jika sistem mendeteksi adanya suara tabrakan maka mengambil data *latitude*, *longitude* melalui GPS untuk dikirimkan menuju pesan *telegram*. Namun, jika output masih terdeteksi kondisi normal maka proses *streaming* dan prediksi terus berulang.

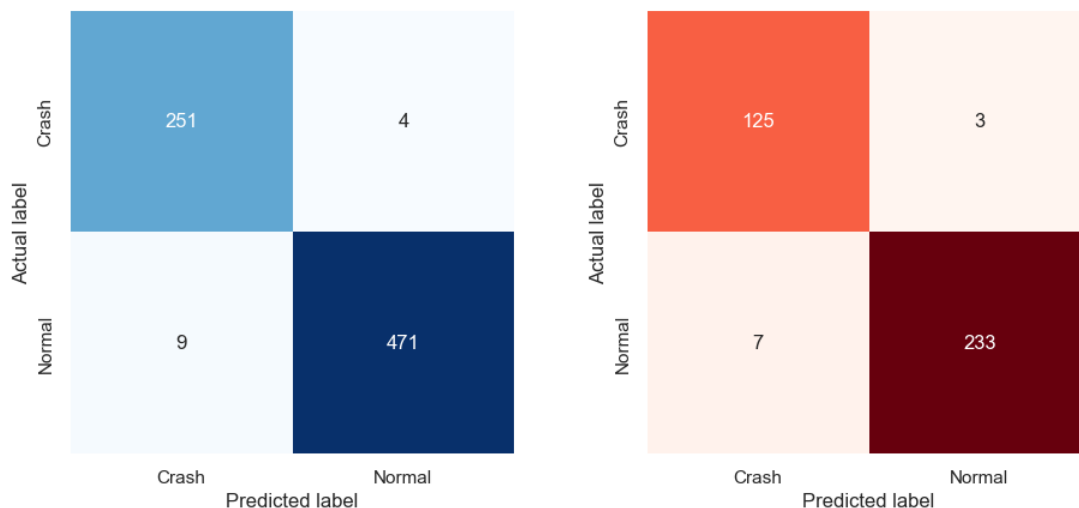
C. Hasil dan Pembahasan

Pengujian dilakukan pada Raspberry Pi 4 dengan usb microphone dan GPS Ublox NEO M8N untuk pengambilan data koordinat secara real-time.

1. Pengujian Model



Gambar 6 Hasil Training Model berdasarkan Akurasi dan Loss



Gambar 7 Confusion Matrix berdasarkan Data Training (Biru) dan Testing (Oranye)

Gambar 7 menunjukkan confusion matrix dari hasil dari pelatihan model berdasarkan data latihan dan pengujian. Dari matriks tersebut, beberapa nilai dalam kinerja matriks dapat dihitung, termasuk akurasi, kerugian (loss), presisi, recall, dan F1-Score.

Tabel 2 Hasil Akurasi dan Loss berdasarkan Dataset

<i>Dataset</i>	<i>Akurasi</i>	<i>Loss</i>
Data Training	99,16%	0,84%
Data Testing	98,63%	1,37%

Tabel 3 Hasil Parameter Matriks dari Data Testing

<i>Kelas</i>	<i>Presisi</i>	<i>Recall</i>	<i>F1-Score</i>
<i>Crash</i>	0.987	0.97	0.978
<i>Normal</i>	0.976	0.946	0.96

Seperti yang terlihat dari data yang telah disajikan pada tabel 2 dan 3 menunjukkan performa model yang cukup baik dengan tingkat akurasi yang tinggi dan loss yang kecil. Begitu juga pada nilai presisi, recall dan f1-score dari data testing menunjukkan keakuratan dalam mengklasifikasikan antara suara kecelakaan dan normal meskipun banyak variasi pada dataset akibat teknik augmentasi data.

2. Pengujian Pengukuran Daya pada Raspberry Pi

Dalam pengujian ini, dilakukan pengukuran nilai daya yang diperlukan oleh Jetson Nano saat berada dalam kondisi idle atau saat menjalankan program deteksi. Nilai daya ini dihitung dengan mengalikan tegangan dengan arus. Pengujian dilakukan menggunakan adaptor daya berkapasitas 5V/3A dengan arus yang terdeteksi oleh sensor ACS712.

Tabel 4 Pengukuran Arus pada Raspberry Pi

<i>Uji Ke-</i>	<i>Kondisi</i>	
	<i>Idle (A)</i>	<i>Running (A)</i>
1	1.023	1.969
2	1.092	1.943
3	1.005	1.994
4	1.078	1.958
5	0.976	1.971
6	1.039	1.988
7	1.102	1.945
8	0.964	2.001
9	1.001	1.978
10	1.067	1.952
Rata-rata:	1.034	1.974

Hasil data yang tercantum dalam Tabel 5 mengindikasikan bahwa dalam kondisi idle, Raspberry Pi memerlukan arus rata-rata sebesar 1.034 A. Oleh karena itu, konsumsi daya yang dibutuhkan oleh perangkat tersebut mencapai 5.274 watt. Namun, ketika program deteksi berjalan, nilai arus rata-rata meningkat menjadi 1.974 A, sehingga konsumsi daya juga mengalami peningkatan menjadi 10.069 watt.

3. Pengujian Kecepatan FPS Proses Streaming dan Prediksi

Pada pengujian ini, waktu pemrosesan yang diperlukan oleh Raspberry Pi 4 dan laptop diukur dan dibandingkan. Setelah diperoleh nilai rata-rata, kecepatan FPS dapat dihitung untuk masing-masing perangkat seperti yang terlihat pada Tabel 6.

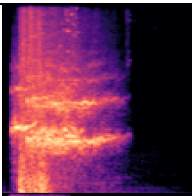
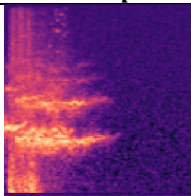
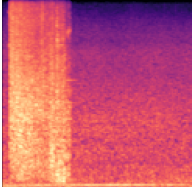
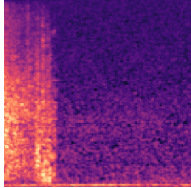
Tabel 5 Pengukuran Waktu dan FPS pada Program Deteksi

Uji Ke-	Raspberry Pi 4		Laptop	
	Streaming	Prediksi	Streaming	Prediksi
1	0.074	0.141	0.065	0.136
2	0.079	0.145	0.062	0.132
3	0.076	0.143	0.067	0.136
4	0.080	0.144	0.064	0.131
5	0.073	0.147	0.061	0.138
6	0.075	0.149	0.068	0.133
7	0.078	0.146	0.069	0.135
8	0.077	0.145	0.066	0.135
9	0.075	0.142	0.064	0.139
10	0.077	0.146	0.060	0.134
11	0.074	0.149	0.068	0.138
12	0.076	0.146	0.067	0.137
13	0.078	0.149	0.063	0.136
14	0.079	0.149	0.062	0.130
15	0.074	0.147	0.065	0.130
16	0.076	0.140	0.061	0.130
17	0.075	0.147	0.066	0.133
18	0.077	0.141	0.063	0.139
19	0.076	0.142	0.069	0.139
20	0.073	0.143	0.067	0.134
Rata-Rata				
Waktu	76.1 ms	145.05 ms	64.85 ms	134.75 ms
FPS	13.14	6.89	15.42	7.42

Hasil data menunjukkan bahwa waktu pemrosesan yang dibutuhkan oleh Raspberry Pi 4 sedikit lebih lama dibandingkan dengan laptop, meskipun perbedaannya tidak signifikan. Begitu pula dengan kecepatan FPS pada masing-masing perangkat, Raspberry Pi 4 memiliki nilai FPS sekitar 13.14 untuk proses streaming dan 6.89 untuk proses prediksi, sementara laptop memiliki nilai FPS sekitar 15.42 untuk proses streaming dan 7.42 untuk proses prediksi.

4. Pengujian perekaman oleh usb microphone berdasarkan bentuk mel-spectrogram

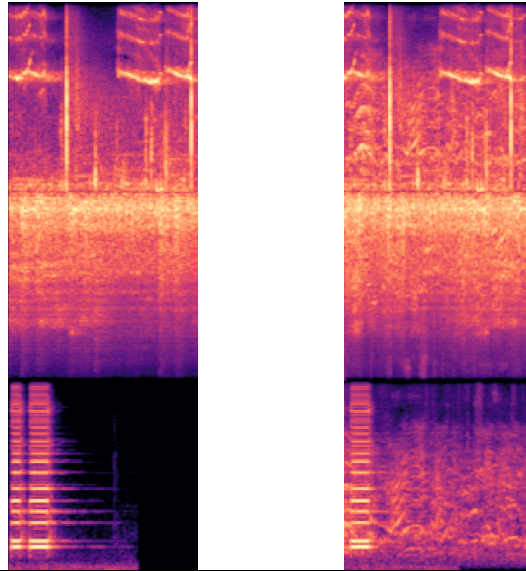
Tabel 6 Perbandingan Uji USB Microphone dengan Dataset

No.	Jenis Suara	Tampilan <i>Mel-Spektrogram</i>	
		<i>Dataset</i>	<i>USB Microphone</i>
1.	Crash (1)		
2.	Crash (2)		

3. Kembang Api

4. Kereta

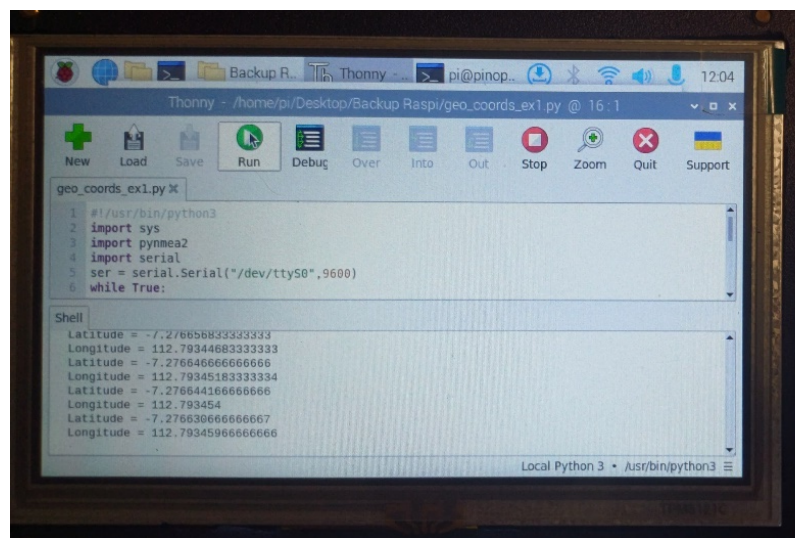
5. Klakson Mobil



Pada uji coba ini, dilakukan pengujian dengan membandingkan visualisasi hasil transformasi mel-spectrogram dari rekaman usb microphone dengan gambar-gambar dari dataset. Alat perekaman yang digunakan adalah usb microphone omnidirectional dengan respon frekuensi 16kHz. Proses perekaman suara dalam uji coba ini menggunakan sampling rate 22050 Hz. Hasil perbandingan antara gambar mel-spectrogram dari rekaman microphone dan file audio dari dataset ditampilkan dalam Tabel 5. Dari hasil tersebut, terlihat bahwa keduanya memiliki bentuk yang sangat mirip meskipun terdapat sedikit noise pada hasil rekaman microphone akibat proses perekaman.

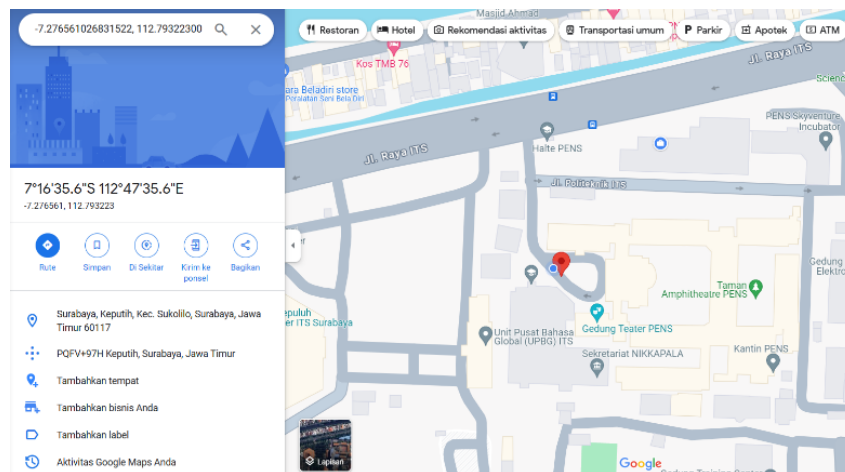
5. Pengujian data koordinat yang dihasilkan oleh GPS

Pengujian ini dilakukan secara streaming untuk mengambil data latitude dan longitude. Pada proses streaming, dilakukan parsing data untuk mengambil data yang diinginkan sesuai program yang telah dirancang. Hasil pengujian dapat dilihat pada gambar.



Gambar 8 Pengujian Koordinat GPS

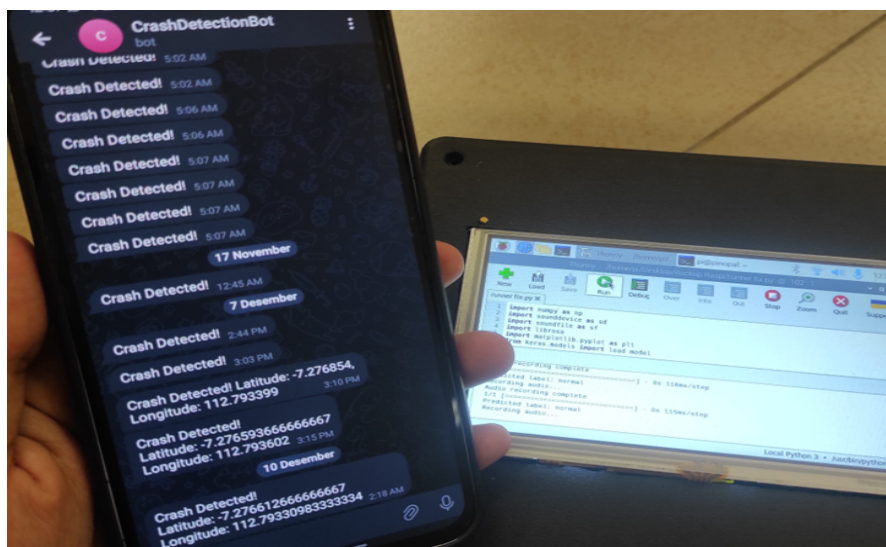
Pengambilan data GPS berlokasi dikawasan kampus PENS, data yang didapat kemudian dicek melalui google maps dari latitude dan longitude yang didapat. Hasil percobaan yang didapatkan menghasilkan data lokasi GPS yang cukup akurat dengan error berkisar 1 hingga 3 meter. Data tersebut sudah cukup membantu dalam pengambilan titik lokasi dari alat yang dipasang untuk mendeteksi adanya kecelakaan yang terjadi.



Gambar 9 Pengecekan Koordinat GPS Menggunakan Google Maps

6. Pengujian implementasi alat deteksi secara real-time menggunakan dataset dan audio baru

Dalam uji coba ini, alat yang dirancang diaktifkan dengan menghubungkannya ke adaptor 5V. Pengujian ini bertujuan untuk mengevaluasi kinerja dan ketepatan deteksi alat dalam mengidentifikasi suara kecelakaan dan suara normal. Selain itu, pengujian ini juga dimaksudkan untuk menguji kemampuan alat dalam mengklasifikasi suara, serta keakuratannya dalam mengirimkan pesan peringatan melalui platform Telegram.



Gambar 10 Pengujian secara Langsung

Tabel 7 Hasil Akurasi Deteksi Menggunakan Dataset

Kelas	Total Sampel	Total Prediksi	Akurasi
<i>Crash</i>	25	22	0.88
Normal	25	23	0.92
Total Akurasi:			90%

Tabel 8 Hasil Akurasi Deteksi Menggunakan Audio Baru

Kelas	Total Sampel	Total Prediksi	Akurasi
<i>Crash</i>	25	19	0.76
Normal	25	22	0.88
Total Akurasi:			82%

Data yang tercantum pada Tabel 8 dan tabel 9 menunjukkan tingkat akurasi alat dalam mengenali dan mengklasifikasikan suara kecelakaan dan suara normal sebesar 90% pada dataset dan 82% pada audio baru. Sampel yang digunakan sebanyak 50 audio, terdiri dari 25 audio kecelakaan dan 25 audio kondisi normal. Setiap kali sistem mengidentifikasi suara tabrakan, pesan peringatan berhasil dikirimkan secara langsung ke Telegram, sesuai dengan yang terlihat pada Gambar 10.

7. Pengujian Jarak Optimal Deteksi dari Alat di Dalam dan Luar Ruangan

Pengujian ini bertujuan untuk mengevaluasi jarak optimal oleh sistem dalam mendeteksi suara tabrakan di suatu kondisi lingkungan, baik di dalam ruangan yang terkendali maupun di luar ruangan yang mungkin memiliki gangguan suara sehingga mempengaruhi kemampuan deteksi.

Tabel 9 Pengujian Jarak Deteksi dari Alat di Dalam Ruangan

Jarak (meter)	Uji Ke-				
	1	2	3	4	5
1	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓
3	✓	✓	✓	✓	✓
4	✓	✓	✓	✓	✓
5	✓	✓	X	✓	✓
6	✓	✓	✓	X	✓
7	✓	X	X	✓	✓
8	X	X	✓	X	X
9	X	✓	X	X	X
10	X	X	X	X	X

Pengujian jarak optimal deteksi dilakukan di dalam ruangan dengan variasi jarak antara 1 hingga 10 meter. Hasil pengujian menunjukkan bahwa alat mencapai tingkat deteksi yang optimal pada jarak sekitar 4-5 meter, di mana responsnya paling akurat dan responsif terhadap suara kecelakaan. Pada jarak kurang dari 4 meter, alat memberikan respons yang cepat namun dengan tingkat kepekaan yang

berlebihan, mengakibatkan potensi terjadinya false positive. Sementara itu, pada jarak lebih dari 4 meter, respons alat menunjukkan penurunan keakuratan.

Tabel 10 Pengujian Jarak Deteksi dari Alat di Luar Ruangan

Jarak (meter)	Uji Ke-				
	1	2	3	4	5
0.5	✓	✓	✓	✓	✓
1	✓	✓	✓	✓	✓
1.5	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	X
2.5	✓	X	X	✓	✓
3	✓	X	✓	X	✓
3.5	X	X	X	X	X
4	X	X	X	X	X
4.5	X	X	X	X	X
5	X	X	X	X	X

Kemudian untuk pengujian di luar ruangan, diberikan variasi jarak antara 0.5 hingga 5 meter. Hasil pengujian menunjukkan bahwa alat mencapai tingkat deteksi yang optimal pada jarak sekitar 1.5 hingga 2 meter, di mana responsnya paling akurat dan responsif terhadap suara kecelakaan. Sementara itu, pada jarak lebih dari 3.5 meter, respons alat menunjukkan penurunan keakuratan.

D. Simpulan

Penelitian ini menghasilkan sistem deteksi kecelakaan lalu lintas berbasis suara dengan Convolutional Neural Network (CNN) menggunakan teknik segmentasi dan augmentasi audio serta melalui pemrosesan sinyal dalam bentuk Mel-Spektogram. Sistem tersebut bertujuan untuk mengurangi korban yang tidak tertolong dengan memberikan notifikasi real-time melalui Telegram serta informasi lokasi kecelakaan menggunakan GPS Ublox Neo-M8N, dan diimplementasikan pada Raspberry Pi 4 dengan usb microphone dan GPS untuk diuji dalam kondisi indoor dan outdoor. Model yang dilatih menunjukkan hasil yang sangat baik dengan akurasi pada data testing sebesar 98.63% dan loss sebesar 1.37%. Pengujian kinerja sistem dalam identifikasi kecelakaan secara real-time mencapai tingkat akurasi deteksi sebesar 90%. Pemrosesan komputasi pada sistem ini diperoleh FPS rata-rata sebesar 13.14 untuk proses streaming dan 6.89 untuk proses prediksi dengan konsumsi daya rata-rata sebesar 10.069 Watt. Sistem ini mampu mendeteksi kecelakaan mencapai tingkat optimal pada jarak 4 meter untuk didalam ruangan dan 2 meter untuk diluar ruangan. Selain itu, sistem ini mampu mengirimkan notifikasi melalui Telegram dengan informasi lokasi kecelakaan, memiliki deviasi lokasi sebesar 1m hingga 3m. Dengan performa ini, sistem deteksi kecelakaan lalu lintas berbasis suara ini menunjukkan kehandalan dan efektivitas dalam meningkatkan respons dan keselamatan di jalan raya.

E. Ucapan Terima Kasih

Kami ucapkan terima kasih kepada pihak-pihak yang telah memberikan dukungan terhadap penelitian.

F. Referensi

- [1] K. G. Rimate, L. Siar, and Cobi E. M. Mamahit, "Pertanggungjawaban penyelenggara jalan atas kerusakan jalan yang mengakibatkan kecelakaan lalu lintas," *LEX Priv.*, vol. 11, no. 2, 2023.
- [2] M. B. A. Rabbani, M. A. Musarat, W. S. Alaloul, S. Ayub, H. Bukhari, and M. Altaf, "Road Accident Data Collection Systems in Developing and Developed Countries: A Review," *Int. J. Integr. Eng.*, vol. 14, no. 1, pp. 336–352, Mar. 2022, doi: 10.30880/ijie.2022.14.01.031.
- [3] A. Alsobhani, H. M. A. Alaboodi, and H. Mahdi, "Speech Recognition using Convolution Deep Neural Networks," *J. Phys. Conf. Ser.*, vol. 1973, no. 1, 2021, doi: 10.1088/1742-6596/1973/1/012166.
- [4] F. M. Ihsan and M. N. Fauzan, "Identifikasi Audio Ancaman Menggunakan Metode Convolutional Neural Network," *J. Sist. dan Teknol. Inf.*, vol. 10, no. 4, p. 446, 2022, doi: 10.26418/justin.v10i4.52433.
- [5] K. S. Harshavardhan and Mahesh, "Urban sound classification using ANN," in 2022 International Interdisciplinary Humanitarian Conference for Sustainability (IIHC), IEEE, Nov. 2022, pp. 1475–1480. doi: 10.1109/IIHC55949.2022.10060146.
- [6] Z. Mushtaq and S.-F. Su, "Environmental sound classification using a regularized deep convolutional neural network with data augmentation," *Appl. Acoust.*, vol. 167, no. July, p. 107389, Oct. 2020, doi: 10.1016/j.apacoust.2020.107389.
- [7] O. O. Abayomi-Alli, R. Damaševičius, A. Qazi, M. Adedoyin-Olowe, and S. Misra, "Data Augmentation and Deep Learning Methods in Sound Classification: A Systematic Review," *Electronics*, vol. 11, no. 22, p. 3795, Nov. 2022, doi: 10.3390/electronics11223795.
- [8] Z. Islam and M. Abdel-Aty, "Deep Convolutional Neural Network for Roadway Incident Surveillance Using Audio Data," pp. 1–8, 2022.
- [9] A. A. Pratama, S. Sukaridhoto, M. H. Purnomo, V. Lystianingrum, and R. P. N. Budiarti, "Design of Audio-Based Accident and Crime Detection and Its Optimization," *JOIV Int. J. Informatics Vis.*, vol. 7, no. 1, p. 214, Mar. 2023, doi: 10.30630/joiv.7.1.1643.
- [10] A. H. Z. Ilahi, "Rancang Bangun Sistem Deteksi Kecelakaan pada Mobil berbasis Suara," *Politeknik Elektronika Negeri Surabaya*, 2023.
- [11] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, "Audio Surveillance of Roads: A System for Detecting Anomalous Sounds," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 279–288, Jan. 2016, doi: 10.1109/TITS.2015.2470216.
- [12] P. Foggia, A. Saggese, N. Strisciuglio, and M. Vento, "Cascade classifiers trained on gammatonegrams for reliably detecting audio events," in 2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE, Aug. 2014, pp. 50–55. doi: 10.1109/AVSS.2014.6918643.
- [13] K. J. Piczak, "ESC," in *Proceedings of the 23rd ACM international conference on Multimedia*, New York, NY, USA: ACM, Oct. 2015, pp. 1015–1018. doi: 10.1145/2733373.2806390.
- [14] J. Salamon, C. Jacoby, and J. P. Bello, "A Dataset and Taxonomy for Urban Sound Research," in *Proceedings of the 22nd ACM international conference on*

- Multimedia, New York, NY, USA: ACM, Nov. 2014, pp. 1041–1044. doi: 10.1145/2647868.2655045.
- [15] I. Trowitzsch, J. Taghia, Y. Kashef, and K. Obermayer, “The NIGENS General Sound Events Database,” pp. 1–5, Feb. 2019, [Online]. Available: <http://arxiv.org/abs/1902.08314>
- [16] Iver Jordal, “iver56/audiomentations: v0.34.1,” Zenodo. [Online]. Available: <https://github.com/iver56/audiomentations>
- [17] B. McFee et al., “librosa: Audio and Music Signal Analysis in Python,” in Proceedings of the 14th Python in Science Conference, 2015, pp. 18–24. doi: 10.25080/Majora-7b98e3ed-003.