

# A Study of Machine Learning on Car Accident Detection through Sound Recognition

Ja-Hao Chen,<sup>1\*</sup> Yu-Ju Lin,<sup>2</sup> and Pei-Hua Tsai<sup>3</sup>

<sup>1</sup>Department of Communication Engineering, Feng Chia University,  
No. 100, Wenhwa Rd., Seatwen, Taichung 407102, Taiwan (R.O.C.)

<sup>2</sup>Department of Industrial Engineering and Enterprise Information, Tung Hai University  
No. 1727, Sec. 4, Taiwan Boulevard, Taichung 407224, Taiwan (R.O.C.)

<sup>3</sup>In-service Degree Program of Master of Information and Electrical Engineering, Feng Chia University  
No. 100, Wenhwa Rd., Seatwen, Taichung 407102, Taiwan (R.O.C.)

(Received August 15, 2024; accepted February 20, 2025)

**Keywords:** machine learning, car accident detection, short-time Fourier transform

In this paper, we propose a practical and reliable car accident sound recognition model using deep learning techniques. In this study, 400 car accident sound files were collected, segmented, and classified into 1312 sound files for training the model and 327 sound files for testing the model. The sound files were transformed into spectrograms using a short-time Fourier transform. YOLOv7 was utilized to train the model to detect the sounds of vehicle skidding and collisions. During the model training, image augmentation parameters need to be turned off so that the trained overall model can achieve an average accuracy of 0.875 for vehicle skidding and collision sounds in car accidents during testing. The threshold for average precision was set to 0.8, and the misdetection rate for common vehicle horn sounds was kept below 22.5%. The verification results of this car accident sound detection model demonstrate its practical application capability.

## 1. Introduction

According to a World Health Organization report,<sup>(1)</sup> more than 1 million people die in car accidents every year, making this a public health issue that must be taken seriously. It is worth noting that in some areas and during certain times, owing to the small number of vehicles passing by, pedestrians who accidentally fall on the road may be unable to report the accident themselves, which may lead to secondary accidents due to the slow reaction of vehicles behind them. If the accident can be monitored and reported immediately when it happens, the time for relevant units to arrive at the scene can be shortened, the accident scene can be handled efficiently, and the injured can be protected, reducing the risk of accident escalation.

Current AI technology is rapidly developing, significantly enhancing the efficiency of intelligent monitoring and management in industries, healthcare, and urban management. Recent research has applied AI technology to traffic monitoring. In a previous study,<sup>(2)</sup> cameras were

---

\*Corresponding author: e-mail: [chiahaoc@fcu.edu.tw](mailto:chiahaoc@fcu.edu.tw)  
<https://doi.org/10.18494/SAM5311>

used to capture traffic videos and generate images, which were then used for deep learning model training and monitoring. In this paper, the authors revealed that decomposing videos into a large number of images and learning from them were the biggest challenges. Rezaei and Ebrahimpour-Komleh<sup>(3)</sup> used traffic accident videos for model training and testing, employing Faster R-CNN technology to improve training speed without affecting accuracy. They mentioned that about 25% of accident images were difficult to detect, including those from accidents occurring in different directions, under varying weather conditions, and those caused by special vehicles. Choi *et al.*<sup>(4)</sup> used in-car sensors, including cameras, microphones, GPS, and accelerometers, to build models using data collected from multiple sensors. The multimodal data included images, voice, and vehicle motion status during driving. This data was used to analyze and detect whether a car collision has occurred, improving collision detection accuracy. Ghahremannezhad *et al.*<sup>(5)</sup> used traffic surveillance images to build deep learning models, tracking target motion patterns based on target appearance, size, and location to determine in real time if a traffic accident has occurred. Recent studies<sup>(2–6)</sup> all used image-based monitoring technology, utilizing models trained on traffic accident images and monitored using real-time images. However, the large volume of image data, the complexity of learning calculations required for image detection, and high resource demands lead to delayed detection and high overall costs. Additionally, image capture is easily affected by weather and variable road conditions, causing detection interference and increasing the likelihood of misdetection.

Foggia *et al.*<sup>(6)</sup> mentioned that traditional video analysis systems have been widely used in detecting road accidents, but relying solely on visual information is unreliable under conditions such as nighttime, rainy days, or foggy days. Therefore, we installed a set of microphones at fixed distances and heights on one side of the road. The system extracts a set of features from the low-level audio stream that can capture event characteristics, and the high-level audio stream uses a bag-of-words model to detect short-term and continuous events, identifying abnormal sounds on the road such as tire skidding and car crash sounds for road accident monitoring. Ahmed *et al.*<sup>(7)</sup> used in-car microphone sensors with algorithms based on acoustic signal processing. They statistically analyzed ten signal parameters that are very helpful for identifying vehicle accidents and recognizing vehicle collision sounds and human fatigue distress calls. Fatimah *et al.*<sup>(8)</sup> proposed an automatic alarm sound detection algorithm that distinguishes different types of sound from noisy traffic environments. They used Fourier decomposition and Mel-frequency cepstral coefficients (MFCCs) to calculate audio feature vectors, identifying the approach of emergency vehicles and alerting drivers to make way, thereby reducing the time emergency vehicles (such as ambulances and fire trucks) take to reach their destinations. Sathruhan *et al.*<sup>(9)</sup> used deep learning technology to detect the sound models of emergency vehicles, aiming to improve the accuracy of existing vehicle accident detection technologies. This model uses convolutional neural networks (CNNs) to process short audio signals and MFCC feature extraction techniques to convert sounds into images. The study results showed that this model can achieve an accuracy rate of 93%.

Compared with image detection, the use of sound for detecting traffic accidents offers certain advantages. We employed deep learning techniques to develop a reliable and accurate sound detection model for traffic accidents, which is applicable in real-world scenarios. In Sect. 2, we

introduce the methods employed for training the traffic accident sound recognition model. In Sect. 3, we discuss the verification of the model and the techniques used to enhance its detection accuracy. Test results demonstrated that the proposed traffic accident sound detection model is highly capable of accurately detecting accidents while maintaining a low rate of misdetection.

## 2. Training Model

The process of training a traffic accident sound detection model consists of eight steps, as shown in Fig. 1. The first step is to collect a dataset of road accident sound recordings. In this study, we obtained the audio files from the Mivia Lab at the University of Salerno, which provided us with 400 recordings.<sup>(10)</sup> These audio files were recorded using the Axis P8221 Audio Module and Axis T83 omnidirectional microphones at a sampling rate of 32000 Hz, with each pulse code modulation (PCM) sample quantized to 16 bits. The recordings, saved in WAV audio format, included background sounds from various real traffic accident scenarios, such as vehicle skidding and collisions.

In the second step of our analysis of car accident sounds, we used the digital audio editing software Wave Editor<sup>(11)</sup> to perform an initial screening and classification of the sounds, as depicted in Fig. 2. This process involved the exclusion of redundant and irrelevant data, followed by the extraction of braking skid and collision sounds. These sounds were then segmented into

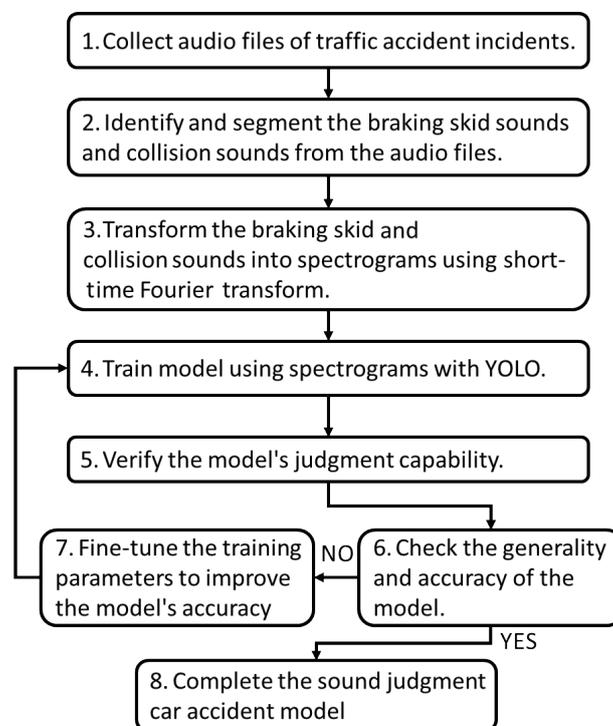


Fig. 1. Process flow for training the car accident sound detection model used in this study.

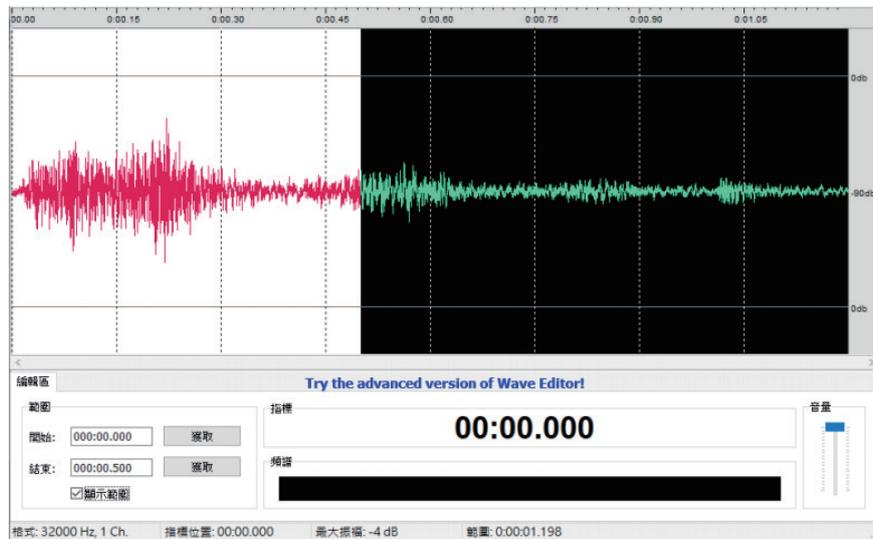


Fig. 2. (Color online) The audio editing tool is utilized for initially screening and categorizing sounds.

1639 separate files, each with a duration of 0.5 s. Finally, these files were annotated and stored for further use.

Visual acoustic significant feature extraction (VASFE) methods were proposed in previous studies.<sup>(8,9,12)</sup> These methods employ short-time Fourier transform (STFT) to transform digital audio into spectrograms for analysis, a transformation that is crucial in sound analysis techniques across various fields. In our third step, we utilized the audio analysis Python package Librosa<sup>(13)</sup> to transform the classified 1639 audio files into spectrograms using STFT as shown in Fig. 3.

In the fourth step, we utilized the spectrograms, which contain audio information, to train the car accident detection model. The primary hardware specifications employed for this training are detailed in Table 1. We used YOLOv7, a mainstream object detection algorithm renowned for its superior speed and accuracy compared with other known object detectors.<sup>(14)</sup> YOLOv7 takes advantage of graphics processing unit (GPU) acceleration to boost computational speed, capture details and contextual information in images more effectively, and enhance the efficiency of model construction. It provides a variety of parameters and commands that can be adjusted or added on the basis of development requirements. These modifications can result in different training model configurations (MODEL\_TYPE), contingent on changes in advanced environment variables such as WEIGHT, HYP\_YAML, and CFG\_YAML. For the purpose of applying the system to road accident monitoring, we opted for the YOLOv7-tiny setting, which is capable of edge computing for model training.<sup>(15)</sup>

As mentioned earlier, the 1639 spectrograms were categorized into two sound types: collision sounds (labeled: Accident) and braking skid sounds (labeled: Skidding). Of these spectrograms, 80% (1312 images) were used for training the model, while the remaining 20% (327 images) were used for testing the model's detection ability. Before training the model, several preliminary

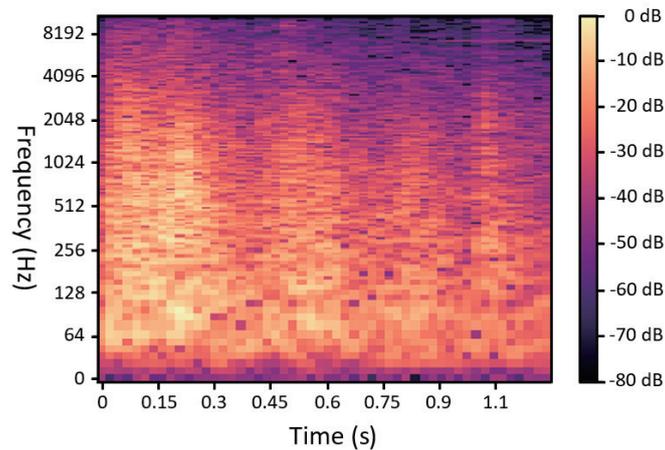


Fig. 3. (Color online) The audio file is transformed into spectrograms using STFT.

Table 1

Primary hardware specifications employed for this training model.

CPU	Intel Core i5, 3.7 GHz, 6 core
GPU	NDIVIA GeForce RTX 3070 8GB
RAM	16 GB (8 GB DDR4 2666 MHz × 2)
SDD	1 TB, SATA, read 550 MB/s, write 520 MB/s

tasks had to be completed. These included creating a new YOLOv7 virtual environment using Anaconda Navigator,<sup>(16)</sup> installing the PyTorch<sup>(17)</sup> machine learning library, and utilizing the torchvision image training tool.<sup>(18)</sup> After these preparations were completed, the training script was executed with a specification to use 1312 spectrograms for training. The training process window is shown in Fig. 4. The parameter “epochs” refers to the number of training iterations for each spectrogram. Generally, setting this parameter to 50 yields good training results.

After the YOLOv7 training was completed, the model was tested using 327 spectrograms to evaluate its detection accuracy, corresponding to step 5 as shown in Fig. 1. This testing process generates various evaluation metrics, including the Confusion Matrix.<sup>(19)</sup> Figure 5 shows the Confusion Matrix for the Accident and Skidding detections made by this car accident detection model. The values for True Positives ( $TP$ ) and True Negatives ( $TN$ ) are 0.98 and 0.99, whereas those for False Positives ( $FP$ ) and False Negatives ( $FN$ ) are 0.01 and 0.02, respectively. Additionally, the common metrics used for evaluating model performance include *precision*, *recall*, and the *F-score*.<sup>(20)</sup>

*Precision* is defined as the ratio of the number of correctly identified targets ( $TP$ ) to the total number of detected targets (the sum of  $TP$  and  $FN$ ). This can be expressed as

$$precision = \frac{TP}{TP + FN} . \quad (1)$$

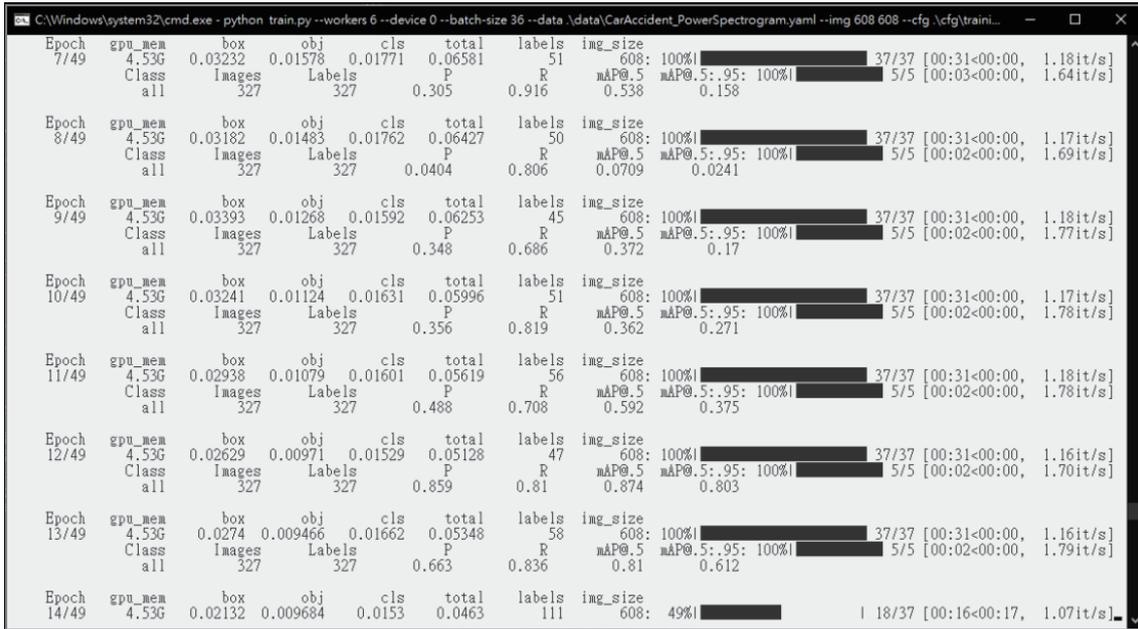


Fig. 4. Description window during model training.

Predicted	Accident	0.98	0.01
	Skidding	0.02	0.99
	True	Accident	Skidding

Fig. 5. Reflection coefficient characteristic of a patch antenna.

Figure 6 shows the precision curve of the trained model. This curve is used to evaluate the model’s precision at vary levels of confidence. When assessing the model’s performance, points that are closer to the top-left corner of the graph indicate higher performance. This implies higher accuracy and fewer misdetections.

*Recall* is defined as the ratio of the number of correctly identified target objects (*TP*) to the total number of actual target objects (the sum of *TP* and *FN*). This can be expressed as

$$recall = \frac{TP}{TP + FN} . \tag{2}$$

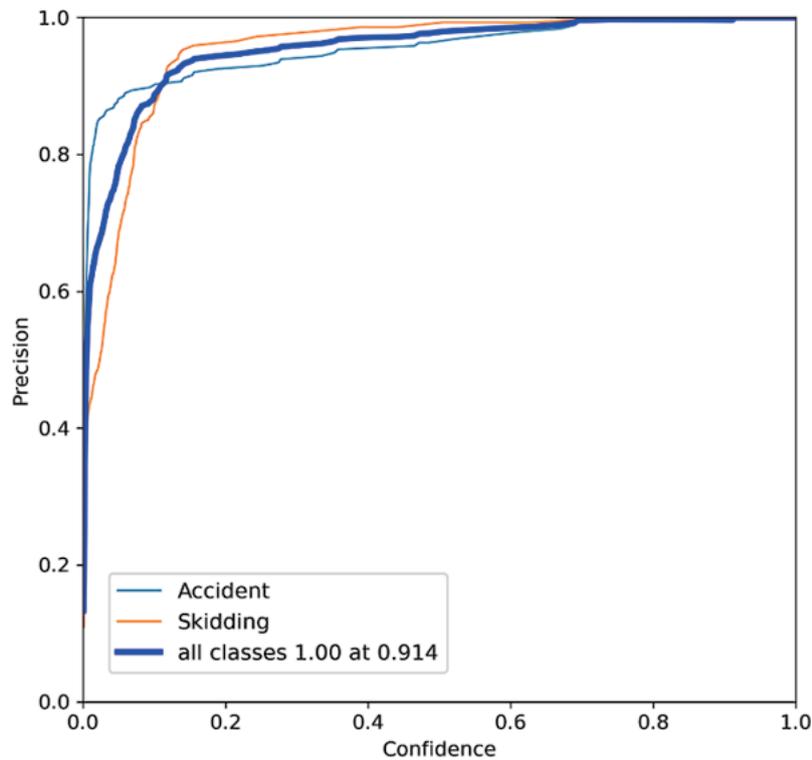


Fig. 6. (Color online) *Precision* curve of the trained model.

Figure 7 shows the recall curve of the trained model. This curve is used to evaluate the model's detection accuracy at various levels of confidence. The closer the points on the graph are to the upper right corner, the higher the performance of the model.

A PR curve is a characteristic curve that plots *recall* on the *x*-axis against the corresponding *precision* on the *y*-axis. The PR curve intuitively displays the model's precision performance at various recall rates. This indicates the model's ability to distinguish whether an object is present in the image. Figure 8 shows the PR curve of this model. As *recall* increases to 0.83, *precision* slightly decreases. However, after reaching 0.98, it drops sharply. This indicates that the model can achieve the highest recall rate while maintaining the highest precision. The blue line in the figure represents the model's mean average precision (*mAP*). When *recall* is 0.5, *mAP* can reach 0.995, which demonstrates the model's excellent performance in detection.

*F*-score simultaneously considers the *precision* and *recall* of the model. Its equation is represented as

$$F_{\beta} = \frac{2}{\text{precision}^{-1} + \text{recall}^{-1}} = (1 + \beta) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}. \quad (3)$$

When  $\beta$  equals 1, it implies that equal weight is given to both *precision* and *recall*. In this case, the *F*-score is also known as the *F1*-score. An *F*-score of 1 (or 100%) indicates that the

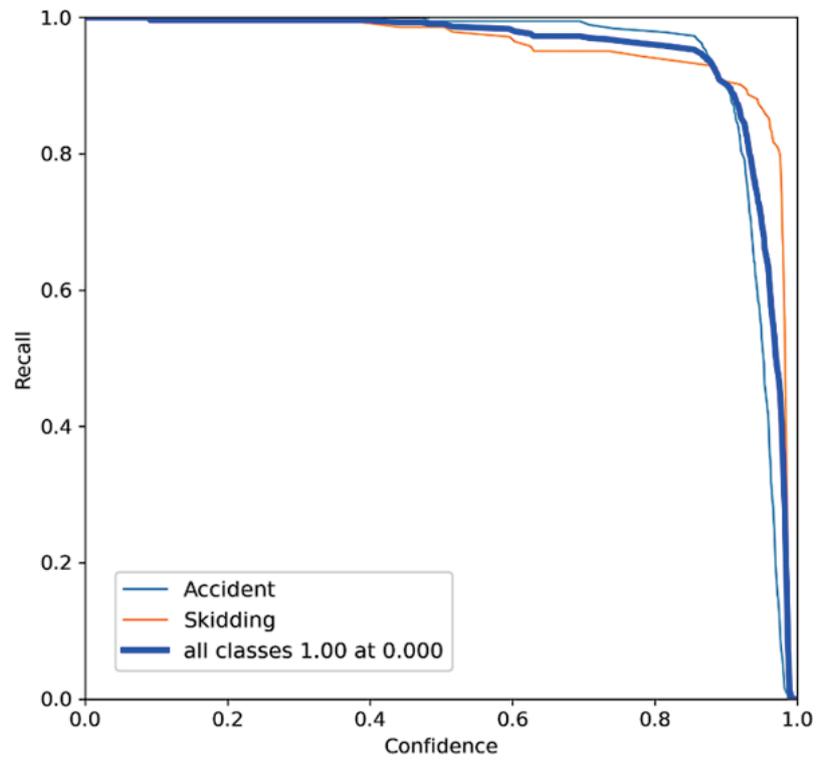


Fig. 7. (Color online) Recall curve of the trained model.

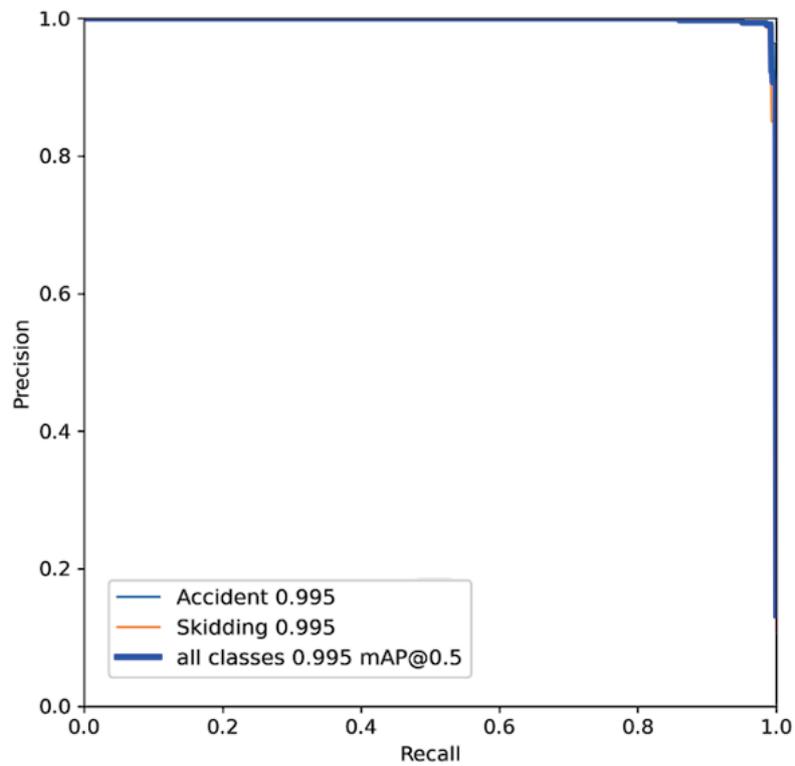


Fig. 8. (Color online) PR curve of the trained model.

algorithm has achieved optimal accuracy. Figure 9 shows the  $F1$  curve of the model after training. The curve's convex shape indicates an optimal threshold that achieves the best balance between precision and recall. Observations from Figs. 5 to 7 suggest that the trained model performs well.

### 3. Model Verification

In the previous section, we trained a model with 1312 spectrograms and tested its accuracy using another set of 312 spectrograms. To ensure the model's accuracy and generalizability, we implemented step 6 of the procedure in Fig. 1, which involved the use of actual car accident videos.<sup>(21)</sup> These videos were recorded using in-car equipment from police vehicles. The audio files we used to train the model in the previous section typically had long and clear skidding sounds. Even though the collision sounds were brief, they were distinct and clearly characteristic of car accidents. In contrast, the skidding sounds in the car accident videos<sup>(21)</sup> were relatively quieter and more blurred, and the collision sounds were also shorter and less distinct, making the car accident sound features less prominent and more similar to typical car accident sounds heard in real life. Therefore, the audio from these videos was used to conduct a more stringent test of the model.

After the car accident video files were converted into audio files, we analyzed and segmented them. We extracted the first 16 s of the car accident audio and divided it into 32 files, each

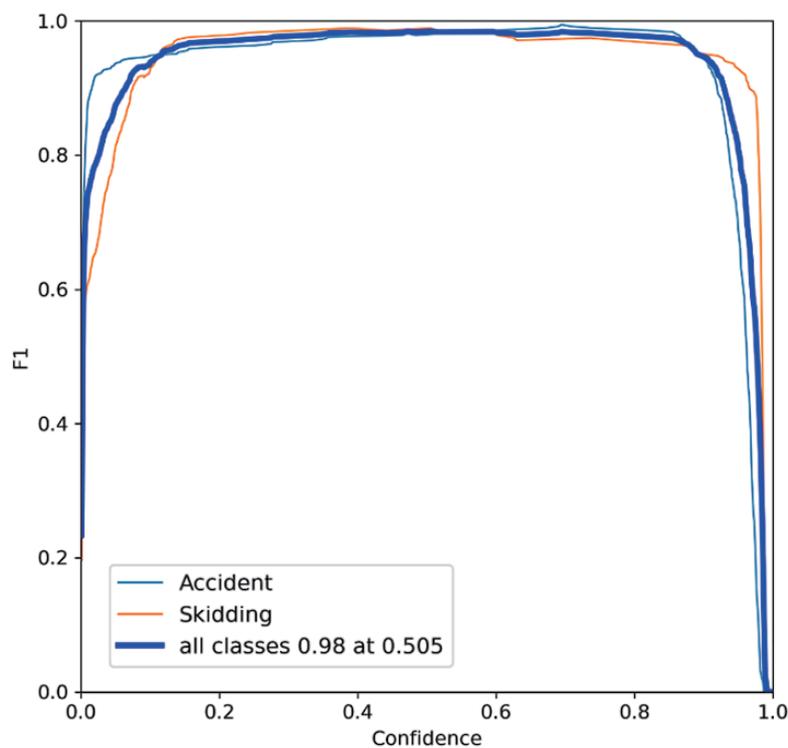


Fig. 9. (Color online)  $F1$  curve of the trained model.

lasting 0.5 s, and numbered them accordingly. The audio from the 12th to 13th seconds contains braking sounds, while that from the 13th to 15th seconds contains collision sounds. The rest of the audio is background noise. Among the 32 files, the braking sounds correspond to files numbered 25 and 26, and the collision sounds correspond to files numbered 27 to 30. We then converted these 32 files into spectrograms using STFT and labeled them. Subsequently, these spectrograms were used to test the model trained in the previous section, with the test results presented in Fig. 10. Figure 10 shows the PR curve for the model trained with default parameters. The precision for the braking sounds was 1, but for the collision sounds, it was only 0.341. This suggests that the model has a relatively low ability to accurately identify collision sounds. Considering that the test car accident audio was less clear, this result is not surprising. However, to improve its accuracy, further training of the model is necessary.

An in-depth understanding of the YOLOv7 model training parameters, which include image augmentation parameters, reveals that they can effectively improve the model's performance.<sup>(22)</sup> The image augmentation parameters encompass image scaling, image flipping, image mixing, image copy-pasting, image shifting, image hue adjustment, and image mosaicking. These image feature augmentation techniques have been demonstrated to enhance model training for general object photos. However, their effectiveness for spectrograms of car accident audio requires further verification.

Therefore, as shown in Fig. 1, we planned step 7, in which the image augmentation parameters are assigned numbers as depicted in Table 2. We systematically disabled each of the seven image

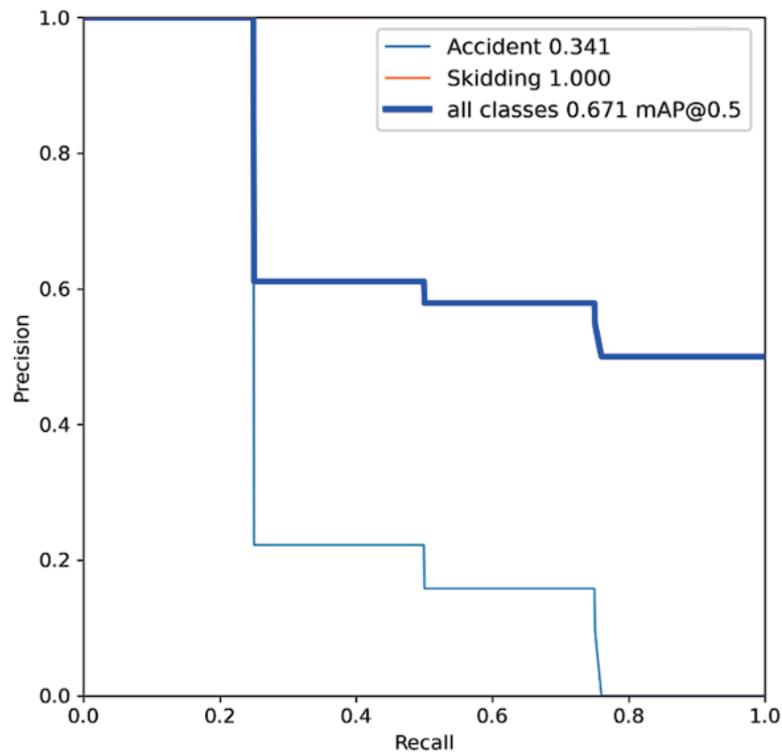


Fig. 10. (Color online) PR curve of the trained model.

Table 2

*APs* of the training and test models after turning off the image augmentation parameters.

Item	Off augmentation (parameter)	<i>AP</i> by Ref. 10 skidding spectrograms	<i>AP</i> by Ref. 10 collision spectrograms	<i>AP</i> by Ref. 21 skidding spectrograms	<i>AP</i> by Ref. 21 collision spectrograms
a	Scale (scale)	0.995	0.95	0.998	0.519
b	Horizontal/vertical flip (fliplr/flipud)	0.982	0.984	0.998	0.094
c	Mix (mixup)	0.993	0.993	0.997	0.111
d	Copy/paste (copy_paste)	0.986	0.985	0.997	0.094
e	Image shift (translate)	0.99	0.992	0.664	0.023
f	Image hue (hsv_h, hsv_s, hsv_v)	0.995	0.994	0.747	0.074
g	Image mosaic (mosaic)	0.75	0.906	0	0

augmentation parameters one by one. Returning to step 4, we used 1312 spectrograms to train seven different models. Next, in step 5, we tested these seven models using 312 spectrograms. The average precision (*AP*) results from the tests are shown in the third and fourth rows of Table 2. Following that, we utilized the spectrograms from the audio of video<sup>(21)</sup> and performed step 6. In this step, we once again disabled each of the seven image augmentation parameters one by one, trained seven different models, and individually tested them. The *AP* results from these tests are listed in the fifth and sixth rows of Table 2.

From the test results, it was observed that the *AP* values in the third and fourth rows of Table 2 remain high owing to the clarity of the audio files used in step 5. However, when tested with quieter and more blurred audio files from the video, the *AP* values in the fifth and sixth rows significantly decrease. This is particularly evident in the sixth row, representing the test results for short and blurred collision sounds, where each *AP* value is extremely low. Additionally, models trained with the image mosaic parameter disabled demonstrated inferior *AP* performance. Notably, when tested with the audio files from the video,<sup>(21)</sup> all the resulting *AP* values were zero.

In an effort to continuously enhance the model's accuracy, we adjusted the training strategy by sequentially disabling seven distinct image augmentation parameters, as outlined in Table 3. For example, Item A solely disables the image scaling parameter corresponding to item a in Table 2. Item B concurrently disables the image augmentation parameters of items a and b, and so forth. Ultimately, Item G disables all image augmentation parameters. In accordance with the configurations in Table 3, we carried out the model's training and testing, as depicted in Table 2, repeatedly. The *AP* values derived from the tests are enumerated in the third to sixth rows of Table 3. The outcomes indicate that the best results are achieved when all image augmentation parameters are disabled, particularly when testing with the audio file from video.<sup>(21)</sup> The *AP* values of the skidding and collision sound models are 0.996 and 0.755, respectively, and the overall *AP* value of the two sound models is 0.875. This *AP* value signifies an enhancement of more than 44% for the collision sound model compared with other configurations. These results clearly demonstrate that it is not appropriate to utilize image augmentation parameters when

Table 3

*APs* of the training and test models after turning off the image augmentation parameter combination.

Item	Off Item	<i>AP</i> by Ref. 10 skidding spectrograms	<i>AP</i> by Ref. 10 collision spectrograms	<i>AP</i> by Ref. 21 skidding spectrograms	<i>AP</i> by Ref. 21 collision spectrograms
A	a	0.995	0.995	0.998	0.519
B	a + b	0.996	0.996	0.998	0.187
C	a + b + c	0.995	0.995	0.998	0.464
D	a + b + c + d	0.996	0.996	0.998	0.359
E	a + b + c + d + e	0.995	0.996	0.998	0.358
F	a + b + c + d + e + f	0.995	0.99	0.998	0.426
G	a + b + c + d + e + f + g	0.99	0.986	0.996	0.755

Table 4

Using models to identify vehicle horn sounds.

Item	Number of sound clips	Number of collision sounds identified	Number of skidding sounds identified
Car horns	16	0	2
Ambulance sirens	14	0	0
Police car sirens	40	0	9

training models using spectrograms. Consequently, we opted to disable the image augmentation parameters for model training and conducted tests, necessitating annotation only when the *AP* value surpassed 0.65.

For reliability testing, we gathered common vehicle horn sounds, which included car horns, ambulance sirens, and police car sirens. These horn sounds were segmented, classified into 0.5 s intervals, and subsequently stored. Utilizing the STFT transformation spectrogram and training the model with Item G, we established the *AP* threshold at 0.8 for detection. The outcomes are presented in Table 4. We observed that the misidentification rate for car horns and police sirens was below 22.5%, with zero misidentifications for ambulance sirens. The validation results suggest that the accident recognition model developed in this study is both practical and dependable.

#### 4. Conclusions

In this study, we collected 400 accident sound files, which were segmented and classified into 1312 training sound files and 327 test sound files. A total of 1639 audio files were transformed into spectrograms by the STFT method. We utilized the YOLOv7-tiny deep learning technique for training and testing the model. It was found during the testing that the image augmentation parameters must be deactivated for the training model. By setting the *AP* threshold to 0.8, the trained model enhanced the recognition capability of vehicle skidding and collision sounds in car accidents by more than 44%, while the misjudgment rate for common

traffic horn sounds remained below 22.5%. The model verification results suggest that the car accident sound recognition model trained in this study is both practical and reliable.

## References

- 1 WTO Report: <https://data.who.int/en/indicators/i/B9D9E6A/D6176E2> (accessed July 2024).
- 2 D. K. Yadav, Renu, Ankita, and I. Anjum: 2020 2nd Int. Conf. Advances in Computing, Communication Control and Networking (ICACCCN, 2020) 232. <https://doi.org/10.1109/ICACCCN51052.2020.9362808>
- 3 Z. Rezaei and H. Ebrahimpour-Komleh: 2021 5th Int. Conf. Pattern Recognition and Image Analysis (IPRIA, 2021) 1. <https://doi.org/10.1109/IPRIA53572.2021.9483506>
- 4 J. G. Choi, C. W. Kong, G. Kim, and S. Lim: Expert Syst. Appl. **183** (2021) 115400. <https://doi.org/10.1016/j.eswa.2021.115400>
- 5 H. Ghahremannezhad, H. Shi, and C. Liu: 2022 IEEE Int. Conf Imaging Systems and Techniques (IST, 2022) 1. <https://doi.org/10.1109/IST55454.2022.9827736>
- 6 P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento: IEEE Trans. Intell. Transp. Syst. **17** (2016) 279. <https://doi.org/10.1109/TITS.2015.2470216>
- 7 M. I. Basheer Ahmed, R. Zaghoud, M. S. Ahmed, R. Sendi, S. Alsharif, J. Alabdulkarim, B. A. Albin Saad, R. Alsabt, A. Rahman, and G. Krishnasamy: Big Data and Cognitive Computing **7** (2023) 22. <https://doi.org/10.3390/bdcc7010022>
- 8 B. Fatimah, A. Preethi, V. Hrushikesh, A. Singh B. and H. R. Kotion: 2020 11th Int. Conf. Computing, Communication and Networking Technologies (ICCCNT, 2020) 1. <https://doi.org/10.1109/ICCCNT49239.2020.9225414>
- 9 S. Sathruhan, O. K. Herath, T. Sivakumar, and A. Thibbotuwawa: 2022 6th SLAAI Int. Conf. Artificial Intelligence (SLAAI-ICAI, 2022) 1. <https://doi.org/10.1109/SLAAI-ICAI56923.2022.10002605>
- 10 Mivia Laboratory web page of the Salerno University: <https://mivia.unisa.it/datasets/audio-analysis/mivia-road-audio-events-data-set/> (accessed July 2024).
- 11 Wave Editor digital audio editing software: <https://www.wave-editor.com/> (accessed July 2024).
- 12 J. Wang, K. Zhang, K. Madani and C. Sabourin: 2013 IEEE Int. Conf. IEEE Region 10 (TENC, 2013) 1. <https://doi.org/10.1109/TENCON.2013.671891>
- 13 Librosa audio analysis python package: <https://librosa.org/doc/latest/index.html> (accessed July 2024).
- 14 J. M. Lopes, L. P. Mota, S. M. Mota, J. M. Torres, R. S. Moreira, C. Soares, I. Pereira, F. R. Gouveia, and P. Sobral: Future Internet **16** (2024) 179. <https://doi.org/10.3390/fi16060179>
- 15 OneAI Documentation: <https://docs.oneai.twcc.ai/s/user-guide-en> (accessed July 2024).
- 16 Anaconda Navigator website: <https://docs.anaconda.com/navigator/> (accessed July 2024).
- 17 PyTorch website: <https://pytorch.org/> (accessed July 2024).
- 18 Torchvision library of the PyTorch project website: <https://pytorch.org/vision/stable/index.html> (accessed July 2024).
- 19 K. M. Ting: Encyclopedia of Machine Learning and Data Mining, Claude Sammut, Geoffrey I. Webb Eds. (Springer, Boston, MA., 2017) p. 260. [https://doi.org/10.1007/978-1-4899-7687-1\\_50](https://doi.org/10.1007/978-1-4899-7687-1_50)
- 20 C. Goutte and E. Gaussier: Advances in Information Retrieval (ECIR, 2005) 345. [https://doi.org/10.1007/978-3-540-31865-1\\_25](https://doi.org/10.1007/978-3-540-31865-1_25)
- 21 Car crash video URL: <https://www.youtube.com/watch?v=PtbcqXYfaoY> (accessed July 2024).
- 22 S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe: 2019 IEEE/CVF Int. Conf. Computer Vision (ICCV, 2019) 6022. <https://doi.org/10.1109/ICCV.2019.00612>

## About the Authors



**Ja-Hao Chen** received his B.S. degree in electronics engineering from Fu-Jen University, Taipei, Taiwan, in 1997 and his Ph.D. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 2002. From 2002 to 2005, he was with ALi Corporation, Taipei, Taiwan, where he worked on on-chip analog and digital I/O pads and ESD protection circuit design. From 2005 to 2007, he was with RichWave, Taipei, Taiwan, where he worked on on-chip RF power amplifier and RF switch circuit design. From 2007 to 2012, he was with the Department of Electrical Engineering, Tunghai University, Taichung, Taiwan, as an assistant professor. He is currently an associate professor in the Department of Communication Engineering, Feng-Chia University, Taichung, Taiwan. His research interests include on-chip RF circuit design and antennas for sensor and wireless power transfer applications. ([chiahaoc@fcu.edu.tw](mailto:chiahaoc@fcu.edu.tw))



**Yu-Ju Lin** received both his M.S. and Ph.D. degrees in electrical engineering from National Cheng Kung University (Taiwan) and his B.S. degree in electrical engineering from Fu Jen Catholic University (Taiwan). In 2008, he began working at National Cheng Kung University as a doctoral researcher in the Department of Electrical Engineering, where he led a team of researchers studying organic semiconductors and flexible electronics. In 2011, he joined United Microelectronics Corporation (UMC) at their Advanced Technology Development (ATD) Department as the chief engineer and research scientist for their semiconductor process. In 2017, he joined Tung Hai University (Taiwan) as an assistant professor in the Department of Industrial Engineering and Enterprise Information and conducted research on IoT, semiconductor manufacturing process, and mechatronics.



**Pei-Hua Tsai** received her B.Eng. degree in computer science and information engineering from National Taichung University of Science and Technology, Taichung, Taiwan, in 2016 and her M.S. degree in information and electrical engineering from Feng Chia University, Taichung, Taiwan, in 2023. From 2014 to 2018, she was with Shen Hsiang Tang Co., Ltd. Taichung, Taiwan, where she worked as an information technology engineer. From 2019 to 2024, she was with RITAI Technology Inc. Taichung, Taiwan, where she worked on system development. ([peihua\\_tsai@outlook.com](mailto:peihua_tsai@outlook.com))