



# VGGish transfer learning model for the efficient detection of payload weight of drones using Mel-spectrogram analysis

Eman I. Abd El-Latif<sup>1,7</sup> · Noha Emad El-Sayad<sup>2,7</sup> · Kamel K. Mohammed<sup>3,7</sup> · Ashraf Darwish<sup>4,7</sup> · Aboul Ella Hassanien<sup>5,6,7</sup>

Received: 16 August 2023 / Accepted: 25 March 2024  
© The Author(s) 2024

## Abstract

This paper presents an accurate model for predicting different payload weights from 3DR SOLO drone acoustic emission. The dataset consists of eleven different payload weights, ranging from 0 to 500 g with a 50 g increment. Initially, the dataset's drone sounds are broken up into 34 frames, each frame was about 5 s. Then, Mel-spectrogram and VGGish model are employed for feature extraction from these sound signals. CNN network is utilized for classification, and during the training phase, the network's weights are iteratively updated using the Adam optimization algorithm. Finally, two experiments are performed to evaluate the model. The first experiment is performed utilizing the original data (before augmentation), while the second used the augmented data. Different payload weights are identified with a potential accuracy of 99.98%, sensitivity of 99.98%, and specificity of 100% based on experimental results. Moreover, a comprehensive comparison with prior works that utilized the same dataset validates the superiority of the proposed model.

**Keywords** Mel-spectrogram · Payload weight detection · CNN · VGGish model · Audio segmentation

## 1 Introduction

Unmanned aerial vehicles (UAVs), usually referred to as drones, have become very popular in recent years because of their ability to capture high-quality images, access difficult locations, and provide stability [1, 2]. Drones are

being extensively utilized across various industries, including military and security, agriculture, construction, weather forecasting, mapping, monitoring, and surveillance. The potential for drones to revolutionize package delivery is also being explored. Furthermore, efforts have

---

✉ Eman I. Abd El-Latif  
eman.mohamed@fsc.bu.edu.eg  
https://www.egyptscience-srge.com/  
Noha Emad El-Sayad  
nelsayad@horus.edu.eg  
https://www.egyptscience-srge.com/  
Kamel K. Mohammed  
vrsc@azhar.edu.eg  
https://www.egyptscience-srge.com/  
Ashraf Darwish  
ashraf.darwish.eg@ieeeg.org  
https://www.egyptscience-srge.com/  
Aboul Ella Hassanien  
abo.aly@ku.edu.kw; aboitcairo@cu.edu.eg;  
https://www.egyptscience-srge.com/

<sup>2</sup> Electronics and Communications Department, Faculty of Engineering, Horus University-Egypt (HUE), Damietta, Egypt

<sup>3</sup> Centre for Virus Research and Studies, Al-Azhar University, Cairo, Egypt

<sup>4</sup> Faculty of Science, Helwan University, Cairo, Egypt

<sup>5</sup> College of Business Administration (CBA), Kuwait University, Kuwait City, Kuwait

<sup>6</sup> Faculty of Computers and Artificial Intelligence, Cairo University, Cairo, Egypt

<sup>7</sup> Scientific Research School of Egypt (SRSEG), Cairo, Egypt

<sup>1</sup> Faculty of Science, Benha University, Benha, Egypt

been made to enhance food production management and agricultural monitoring using drones [3].

Alongside their positive applications, drones can also be employed maliciously. An alarming incident exemplifying this misuse involved a group of drones launching a military attack on oil pumping stations in Saudi Arabia [4]. Furthermore, drones have been utilized for transporting illicit or hazardous substances, posing a significant threat, particularly when used to transport explosive payloads. These challenges necessitate the development of technologies capable of identifying and preventing drone attacks [5, 6].

However, the deployment of anti-drone systems in urban areas can negatively impact the applications of drones that pose no risk. Therefore, the development of smart systems capable of anticipating a drone's intentions is crucial. Various approaches have been explored to address the challenge of detecting approaching drones, including radar [7], radio frequency (RF) analysis [8], and sound analysis [9].

Radar has been a popular technique for drone detection, but traditional radars struggle to provide satisfactory results due to the small radar cross section of UAVs and the heavy and expensive detection radar required [10]. To overcome these limitations, some radar technologies and cutting-edge techniques, such as artificial intelligence, have been proposed for drone detection [11, 12]. However, radar-based detection is an active method that emits electromagnetic signals, making it possible for adversaries to detect and locate the radar.

Radio frequency-based approaches leverage the link between the drone and the remote control to locate the drone. Nevertheless, these techniques become ineffective in the case of autonomous drones. Visual methods rely on analyzing images captured by cameras to determine the presence of drones. This analysis can be based on appearance or movement, but it faces challenges in low visibility conditions and distinguishing drones from birds [13].

Recently, drone sound detection has garnered attention. Determining whether a drone carrying a payload or not is a crucial step to understanding its intentions. A loaded drone may indicate a potentially hostile situation. Therefore, the primary objective of this paper is to explore the possibility of identifying whether a drone is loaded or unloaded and estimate the weight of its payload by analyzing the audio signals produced by the drone. Knowledge of a drone's payload can offer numerous benefits, particularly in terms of operational effectiveness, safety, and adherence to legal regulations.

To address this objective, this paper proposes a deep learning-based technique for payload identification using drone sound, combining the Mel-spectrogram and VGGish model. The dataset employed in the paper comprises sound

recordings of a 3DR SOLO drone carrying payloads of 11 various weights, ranging from 0 to 500 g with a step size of 50 g. The model involves segmenting the drone sound signal dataset into 34 segments, each lasting 5 s. Subsequently, a convolutional neural network (CNN) model is trained using the Mel-spectrogram and VGGish model. Mel-spectrograms capture frequency and intensity patterns unique to each drone and are commonly employed in audio signal analysis. In the end, the model is assessed through two experiments. While the augmented data were used in the second experiment, the original data (before augmentation) is used in the first experiment. The results demonstrate significant changes in the pitch of the drone's sound, induced by varying motor speeds with different payload weights. By exploring and developing this deep learning-based payload identification technique for drone sound. This paper aims to contribute to the field of drone detection and enhance operational effectiveness, safety, and compliance in drone-related activities. This paper makes several significant contributions in the field of drone payload identification and weight estimation:

1. **Reliable and Accurate Approach:** The paper presents a robust and precise methodology for distinguishing between different drone payloads and accurately calculating their specific weights by analyzing the audio characteristics of drone sound. The proposed approach offers a reliable means of payload identification.
2. **Mel-Spectrogram-based Sound Analysis:** The paper leverages the Mel-spectrogram, a representation of audio signals that closely aligns with human perception, to extract crucial audio characteristic data for drone identification. This technique captures the frequencies that are most relevant to human auditory perception, enabling effective discrimination between drone payloads.
3. **Transfer Learning with VGGish Neural Network:** The paper employs transfer learning, specifically utilizing the VGGish neural network for feature extraction, to achieve exceptional performance in categorizing and differentiating drone weights. By leveraging pretrained models and fine-tuning them with the drone sound dataset. The VGGish network proves highly effective in accurately estimating the weight of a given drone payload.
4. **Impressive Accuracy:** Extensive testing and experimentation demonstrate the robustness of the proposed approach. When distinguishing between 11 distinct weights, the developed model achieves an outstanding accuracy level of 99.75%. This high level of accuracy signifies the effectiveness and reliability of the proposed methodology.

Overall, this paper provides a valuable contribution to the field of drone payload identification and weight estimation. The combination of Mel-spectrogram-based sound analysis and transfer learning with the VGGish neural network offers a reliable and accurate approach for distinguishing between different drone payloads and calculating their specific weights. The achieved accuracy levels underscore the practical viability of the proposed methodology, with potential applications in various domains where drone payload identification is crucial for operational efficiency, safety, and compliance.

The remainder of this paper is structured as follows. The paper's materials and methods are given in Sect. 2. The results of the experiment are assessed in Sect. 3. Section 4 brings the model to a conclusion and future work.

## 2 Materials and methods

### 2.1 Materials

Accurately estimating the weight of a drone's payload is essential for optimizing flight operations, ensuring safety, and maximizing efficiency. This section provides a comprehensive review of the existing related work that addresses the challenging issue of estimating the weight of drone payloads.

To our knowledge, the issue of estimating the weight of a drone's payload has been primarily addressed by [14, 15]. Seidaliyeva et al. [14] present a recent framework that employs visual data to detect the presence of a payload on a UAV. Their approach utilizes the YOLOv2 algorithm applied to a dataset comprising images of a DJI Phantom 2 drone, captured with and without various payloads. By leveraging visual cues, the framework determines whether

the UAV is carrying a payload or not. However, it is important to note that this methodology relies on visual data and may not be applicable in scenarios where the sensing system is solely acoustic.

Similarly, Nguyen et al. [15] propose a technique that does not rely on acoustic signals but instead utilizes an RF sensing system to capture drone vibrations. From the recorded data, they extract Mel-frequency cepstral coefficient (MFCC) features and employ an SVM classifier to estimate the drone's payload weight. The model reports an accuracy of 96.27% in payload estimation. While this methodology offers promising results, it is worth mentioning that it is not designed specifically for acoustic sensing systems.

Although the methodologies discussed in [14] and [15] may not directly apply to acoustic sensing systems, they provide valuable insights and alternative approaches for estimating drone payload weight using different data sources. As such, these techniques complement the focus of our explore and contribute to the broader understanding of payload estimation in various sensing environments.

In addition, there are also several notable commercial solutions in the field of estimating drone payload. One such commercial solution is the Discover G2 system [16], which serves as a counter-UAS solution. Advanced digital signal processing techniques are used by this system to determine the azimuth and elevation of the target in real-time utilizing a network of 128 linked microphones. Although the primary focus of this system is not payload estimation, its use of advanced acoustic sensing technologies demonstrates the potential for acoustic-based solutions in drone-related applications.

Another noteworthy commercial solution is DroneShield [17], which incorporates acoustic sensors as part of a multi-sensor framework designed to create an anti-drone

**Table 1** Summary of drone payload estimation methods

| Reference                | Methodology                                      | Data source            | Accuracy      | Focus                                    |
|--------------------------|--|------------------------|---------------|--|
| Seidaliyeva et al. [14]  | Visual data & YOLOv2 algorithm                   | Images of drone        | Not specified | Detecting payload visually               |
| Nguyen et al. [15]       | RF sensing, MFCC features, SVM classifier        | Drone vibrations       | 96.27%        | Estimating payload weight                |
| Discover G2 [16]         | Acoustic sensing & and digital signal processing | Microphone array       | Not specified | Counter-UAS solution                     |
| DroneShield [17]         | Acoustic sensors & and multi-sensor framework    | Integrated sensors     | Not specified | Anti-drone platform                      |
| Ctrl + Sky Model [18]    | Multi-sensor system & counter-drone solution     | Combination of sensors | Not specified | Countering unauthorized drone activities |
| Adel Ibrahim et al. [19] | MFCC components & and SVM classifiers            | Audio signal           | 98%           | Identifying specific payload class       |

platform. By integrating acoustic sensors with other sensors, DroneShield aims to provide comprehensive drone detection and mitigation capabilities.

Ctrl + Sky [18] is another multi-sensor counter-drone system that deserves mention. This solution offers the ability to detect, track, and neutralize drones using a combination of different sensors. While its primary focus is on countering unauthorized drone activities, it highlights the importance of multi-sensor approaches in accurately identifying and monitoring drones.

These commercial solutions, including the Discover G2, DroneShield, and Ctrl + Sky contribute to the practical application of drone payload estimation and demonstrate the integration of acoustic sensors within multi-sensor frameworks. While they may not directly address the specific challenge of estimating payload weight, they provide valuable insights into the broader field of drone sensing and counter-drone technologies.

Adel Ibrahim et al. [19] identified the specific payload class that the drone was carrying with a minimum classification accuracy of 98% by utilizing the Mel-frequency cepstral coefficients (MFCC) components of the audio signal and using various support vector machine (SVM) classifiers. It is noteworthy that these results were obtained with an acquisition period of just 0.25 s. Table 1 highlights various approaches and methodologies employed to tackle drone payload algorithms.

## 2.2 Mel-spectrogram

Spectrograms play a vital role in sound analysis as they provide crucial insights into the frequency of energy distribution over time. Serving as a commonly used representation for audio data [20, 21], the spectrogram characterizes the energy distribution across different frequency bands. However, to better align with human auditory perception, the Mel-scale was introduced by Newmann et al., offering a nonlinear transformation from the conventional linear frequency scale.

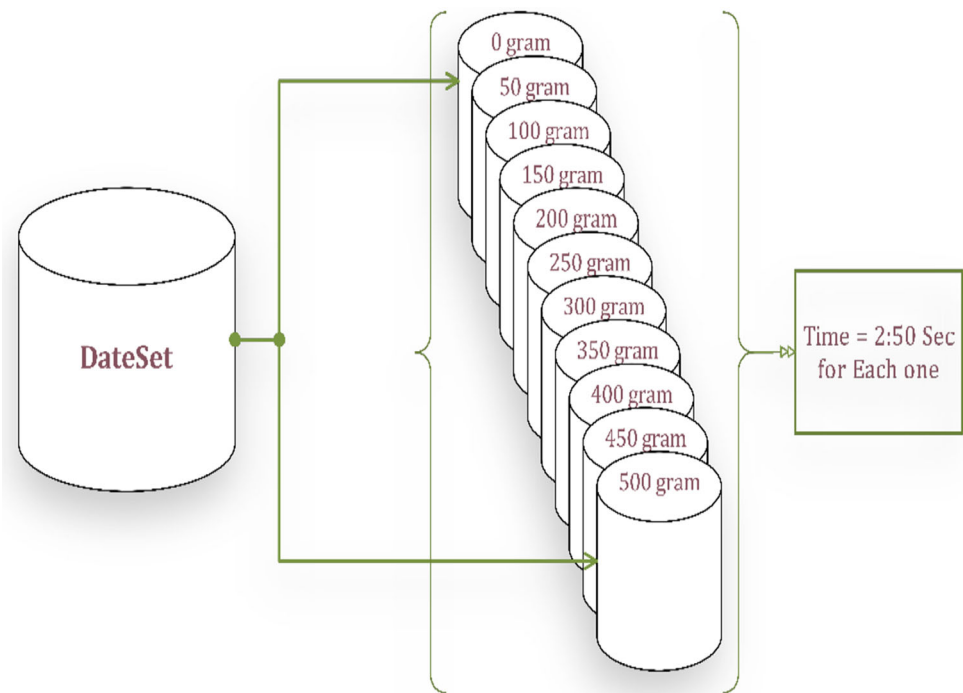
Mathematically, the Mel-scale frequency ( $f_{\text{Mel}}$ ) of a given frequency ( $f_{\text{Hz}}$ ) is computed using Eq. (1). This transformation allows the Mel-spectrum to capture acoustic features more effectively, emphasizing perceptually relevant information in the audio signal.

$$f_{\text{Mel}} = 2595 \log_{10} \left( 1 + \frac{f_{\text{Hz}}}{700} \right) \quad (1)$$

where  $f_{\text{Mel}}$  is the Mel-scale frequency of frequency  $f_{\text{Hz}}$ .

Creating the Mel-spectrum involves applying the Mel-filter bank on the frequency domain of the signal, as described by Eq. (2). The Mel-filter bank consists of triangular window functions ( $H_k$ ), and each filter is centered at a specific point in the frequency domain. These filters partition the signal into different frequency bands, emulating the frequency selectivity of the human ear.

**Fig. 1** Dataset components and characteristics



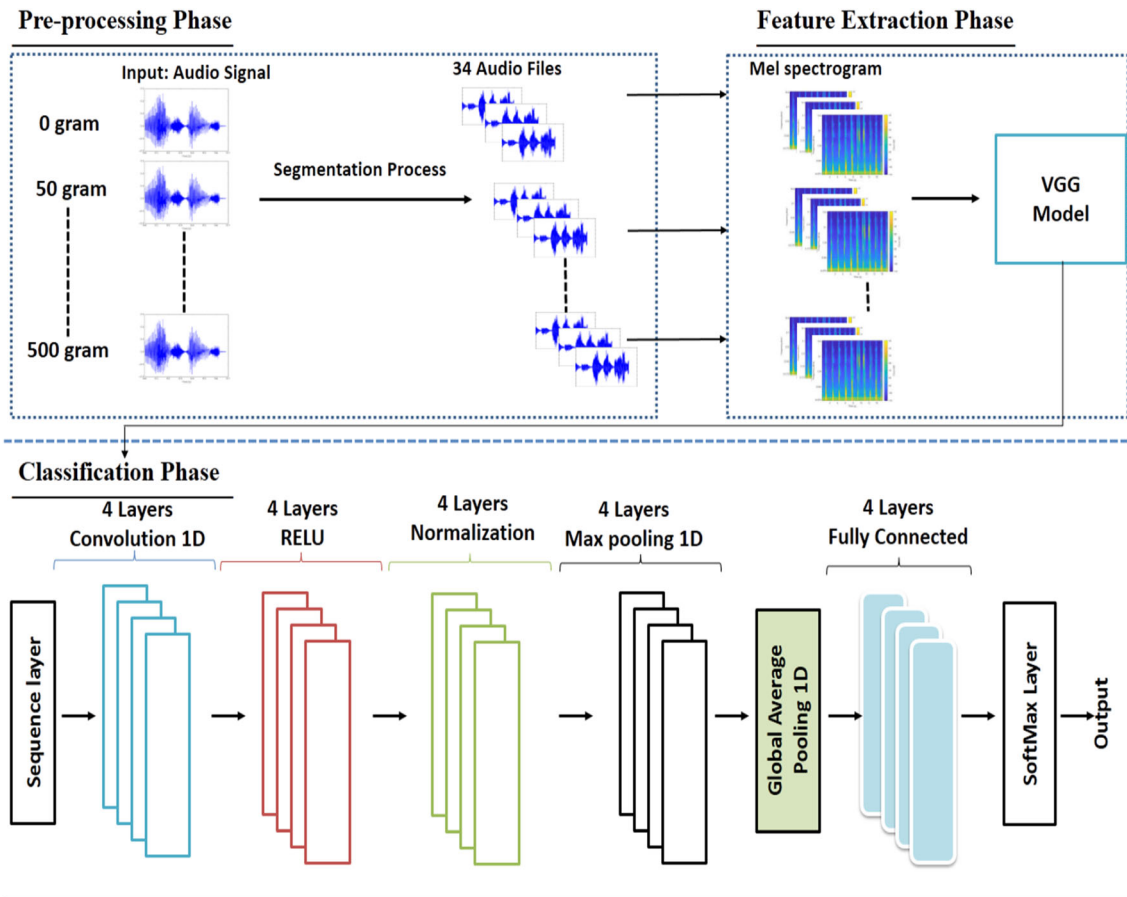


Fig. 2 The architecture of the proposed model

$$H_k(m) = \begin{cases} 0 & \text{for } m < f(k-1) \\ \frac{m-f(k-1)}{f(k)-f(k-1)} & \text{for } f(k-1) \leq m \leq f(k) \\ \frac{f(k+1)-m}{f(k+1)-f(k)} & \text{for } f(k) < m \leq f(k+1) \\ 0 & \text{for } m > f(k+1) \end{cases} \quad (2)$$

where  $H_k(m)$  is the triangular window function,  $m$  is the number of filters and  $k$  ranges from 0 to  $m-1$ . The Mel-spectrum is then obtained by calculating the energy in each frequency band using Eq. (3). This involves taking the squared magnitude of the discrete Fourier transform (DFT) coefficients and applying the corresponding triangular window functions from the Mel-filter bank.

$$X_{\text{Mel}}(x) = \sum_{m=0}^{N-1} [|X(m)|^2 H_k(m)] \quad (3)$$

In summary, the Mel-spectrum, derived using the Mel-scale and Mel-filter bank, offers a perceptually meaningful representation of audio data. Its application in sound analysis, speech recognition, and other audio-related tasks

allows for a more accurate and human-centered interpretation of the frequency energy distribution, enhancing our understanding of acoustic characteristics in diverse audio signals.

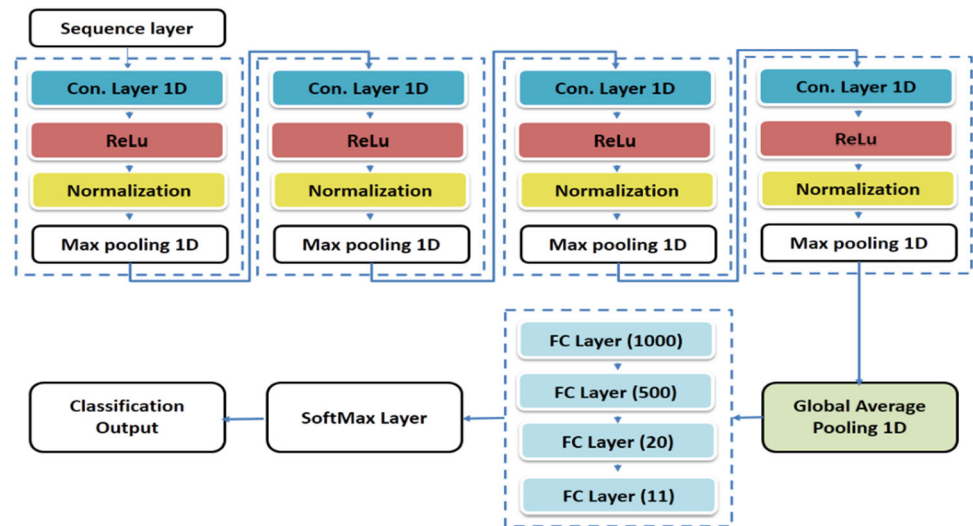
### 2.3 VGGish neural network model

The VGGish model, also known as VGGish Net, is a widely used convolutional neural network (CNN) architecture, specifically referred to as VGGish 16 due to its 16-layer configuration [22]. It is a pretrained network that offers multiple significant improvements that differentiate it from previous models, leading to enhanced performance, ease of use, and shorter training time.

One notable feature of VGGish is its use of smaller convolutional filters, typically  $3 \times 3$ , which enables it to achieve better generalization and reduces the risk of overfitting during the training process. Additionally, VGGISH's hierarchical structure allows each layer to build upon the features extracted from the previous layer, progressively learning more complex representations of the input image.



**Fig. 3** The architecture of the proposed CNN



The first layer of VGGish receives the input image and typically scales it to a fixed dimension of  $224 \times 224$ . The model comprises 13 convolutional layers, each responsible for extracting various features from the input image using multiple filters. After each convolutional layer, VGGish incorporates 5 max-pooling layers to down sampling the feature maps, further reducing the spatial dimensions and enhancing the network's robustness.

The VGGish architecture can be represented as follows:

- Input:  $224 \times 224$  RGB image
- Convolution Layer 1: Convolve input with multiple  $3 \times 3$  filters, producing feature maps.
- Max-Pooling Layer 1: Reduce spatial dimensions of feature maps.
- Convolution Layer 2: Convolve feature maps from the previous layer with additional  $3 \times 3$  filters.
- Max-Pooling Layer 2: Further down sample feature maps.
- (Convolution and Max-Pooling layers continue up to Convolution Layer 13 and Max-Pooling Layer 5, respectively.)
- Fully Connected Layers: Flatten the final feature maps and connect them to one or more fully connected layers for classification.
- Output: Class probabilities or predictions.

The VGGish model's versatility and remarkable performance have led to its application in various domains, such as audio signal classification [23–25] and drone recognition [26–28]. Its robustness and adaptability make it a powerful tool for image recognition and classification tasks in different medical and computer vision applications.

## 2.4 Convolution neural network (CNN)

CNNs are a particular kind of deep neural network that is commonly applied for analysis the of visual images. CNNs can detect and categorize important features from images. Their usages include drone and bird detection [29], image examination used in medical [30, 31], and video recognition [32]. The CONV, activation, pooling, and batch normalization layers are the stacked layers that make up a CNN model. By using the CNN feature extraction methodology, the number of features in a dataset decreases. The two most important layers in CNN are the CONV and FCN layers.

Convolution is the initial layer that is used to extract the various characteristics from the input images. The feature map, which is the result of this layer, gives information on the image's corners and edges. Later, further layers will get this feature map to receive additional characteristics from the input image. The mathematical equation of each feature map in 1D-CNN is calculated by Eq. (4):

$$y = \mu \left( \sum_{i=1}^n \text{conv1D}(To_i + b_i) \right) \quad (4)$$

where  $y$  is the input,  $\mu$  the activation function,  $n$  denotes the number of feature maps in the layer,  $x_i$  denotes the  $i$ th feature map,  $T$  denotes the trainable one-dimensional convolutional kernel,  $o_i$  is the output of the  $i$ th neuron,  $b_i$  denotes the bias of the  $i$ th feature map.

One of the most significant elements of the CNN model is the activation function. They are used to identify and approximate any kind of complex connection among network variables. The ReLU is often utilized in the activation

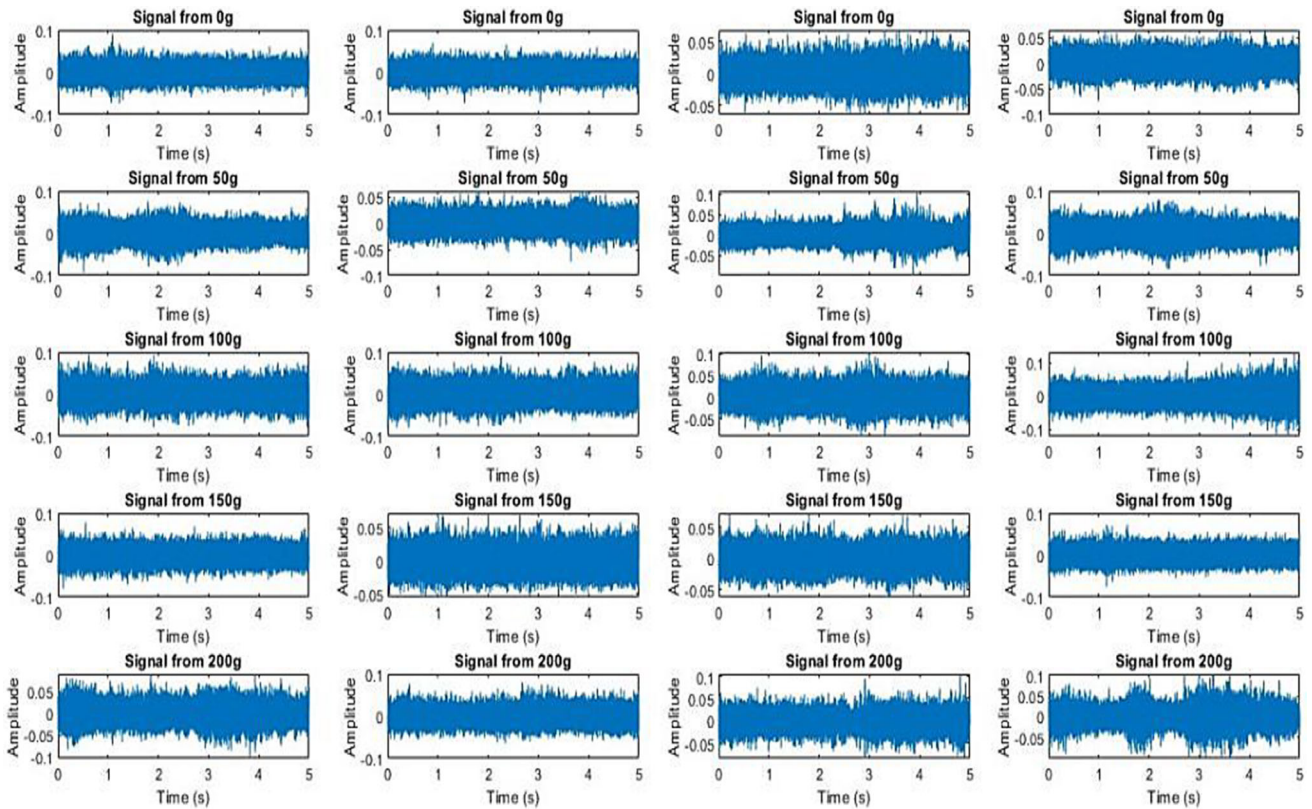


Fig. 4 Preprocessing results

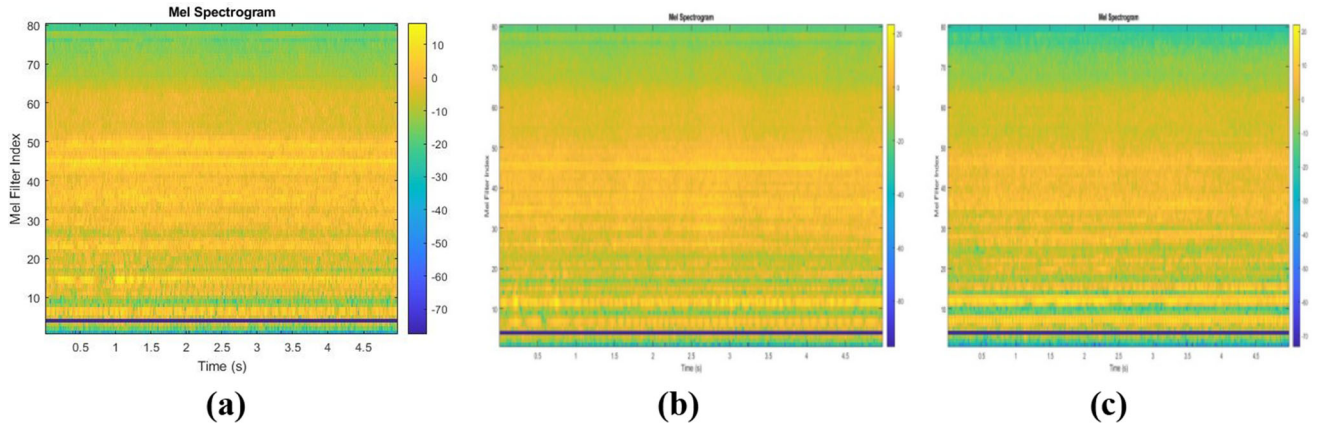


Fig. 5 Mel-spectrum of drone audio signal **a** 0 g, **b** 250 g, and **c** 500 g

function. Following a convolutional layer, the pooling layer is frequently used. The primary objective of this layer is to decrease the size of the convoluted feature map to reduce computational expenses. Average and max pooling are two examples of pooling processes. Another important part of CNN is the dropout layer. The Dropout layer works as a mask, reducing some neurons' contributions to the subsequent layer while keeping all other neurons

functioning. When the dropout layer is applied to an input vector, some of its characteristics are reduced and some hidden neurons are removed. Because they avoid the overfitting of the training data, dropout layers are crucial in CNN training. Finally, a fully connected (FC) layer is fed the output features from the dropout layer, which bases its classification decision on the weights given to each feature [33].

**Table 2** The details of suggested CNN architecture

| Layer       | Input size                |
|-------------|---------------------------|
| Conv1d_1    | $32 \times 3 \times 4096$ |
| Layernorm_1 | 32 channels               |
| Conv1d_2    | $64 \times 3 \times 32$   |
| Layernorm_2 | 64 channels               |
| Conv1d_3    | $128 \times 3 \times 64$  |
| Layernorm_3 | 128 channels              |
| Conv1d_4    | $256 \times 3 \times 128$ |
| Fc_1        | 1000                      |
| Fc_2        | 500                       |
| Fc_3        | 20                        |
| Fc_4        | 11                        |

**Table 3** Parameter value used in optimizer

| Parameter     | Value    |
|---------------|----------|
| Optimizer     | Adam     |
| Learning rate | 0.0001   |
| metric        | Accuracy |
| Batch size    | 15       |
| Epochs        | 100      |

## 2.5 Dataset description and characteristics

The dataset utilized in this paper focuses on the 3DR SOLO drone, a medium-sized commercial UAV weighing approximately 1.5 kg, inclusive of a 0.5 kg battery. The drone boasts a maximum payload capacity of 700 g, accounting for various accessories such as the 3DR Gimbal weighing approximately 390 g. The dataset used in this manuscript was collected by Savio et al. [19] and it is available for download at [34], at <https://github.com/crilab-hbku/Drone-Payload>. For data collection, a high-quality microphone was positioned at 7 m from the flying drone, capturing the drone's sound in an open outdoor setting at a park in Doha, Qatar.

To investigate the impact of different payload weights on the drone's acoustic signature, a series of carefully conducted experiments were carried out. Eleven different payload weights were explored, ranging from 0 to 500 g, with increments of 50 g as shown in Fig. 1. Each experimental recording commenced when the drone achieved a hovering state and concluded just before landing.

The dataset curated through these experiments constitutes a valuable resource for analyzing the acoustic properties of the 3DR SOLO drone under various payload configurations. The comprehensive range of payload weights enables detailed exploration of the drone's sound

characteristics, offering valuable insights for further development in the domain of drone acoustics and payload weight estimation.

## 2.6 The proposed detection of payload weight of drones model

Three crucial construction phases are included in the proposed model for drone payload identification based on the Mel-spectrogram and CNN network as shown in Fig. 2. The following are these phases:

1. **Preprocessing Phase:** A set of processing processes is carried out in the first stage of this paper to segment the entire audio. Each drone sound in the dataset is broken up into 34 frames, each frame was about 5s.
2. **Feature Extraction Phase:** In the second stage, drone features are extracted using spectrogram data and the VGG model. The spectrogram is a method for extracting significant features from audio. It adjusts the frequencies to make them more like what people can hear.
3. **Classification Phase:** The drone payload categorization process is the last step using the proposed CNN contains 23 layers. The proposed CNN's convolutional layer, which is the first layer, is used to extract important characteristics. The ReLU activation function is used after the convolutional layer to stop the network's computational complexity from increasing exponentially. After each ReLU layer, batch normalization layers are used to accelerate the network's learning.

### 2.6.1 Data preprocessing phase

The data preprocessing phase involves segmenting the entire audio dataset of drone sounds into 34 frames, each lasting around 5 s. This process is carried out to facilitate further analysis and feature extraction for the development of an accurate drone sound classification algorithm. By dividing the continuous audio recording into non-overlapping frames and generating individual audio files, the algorithm ensures that the classifier can focus on specific temporal characteristics and learn from diverse patterns in the drone sounds, ultimately enhancing the model's robustness and generalization capability on unseen data. Algorithm 1 describes the main procedure to generate 34 audio files.



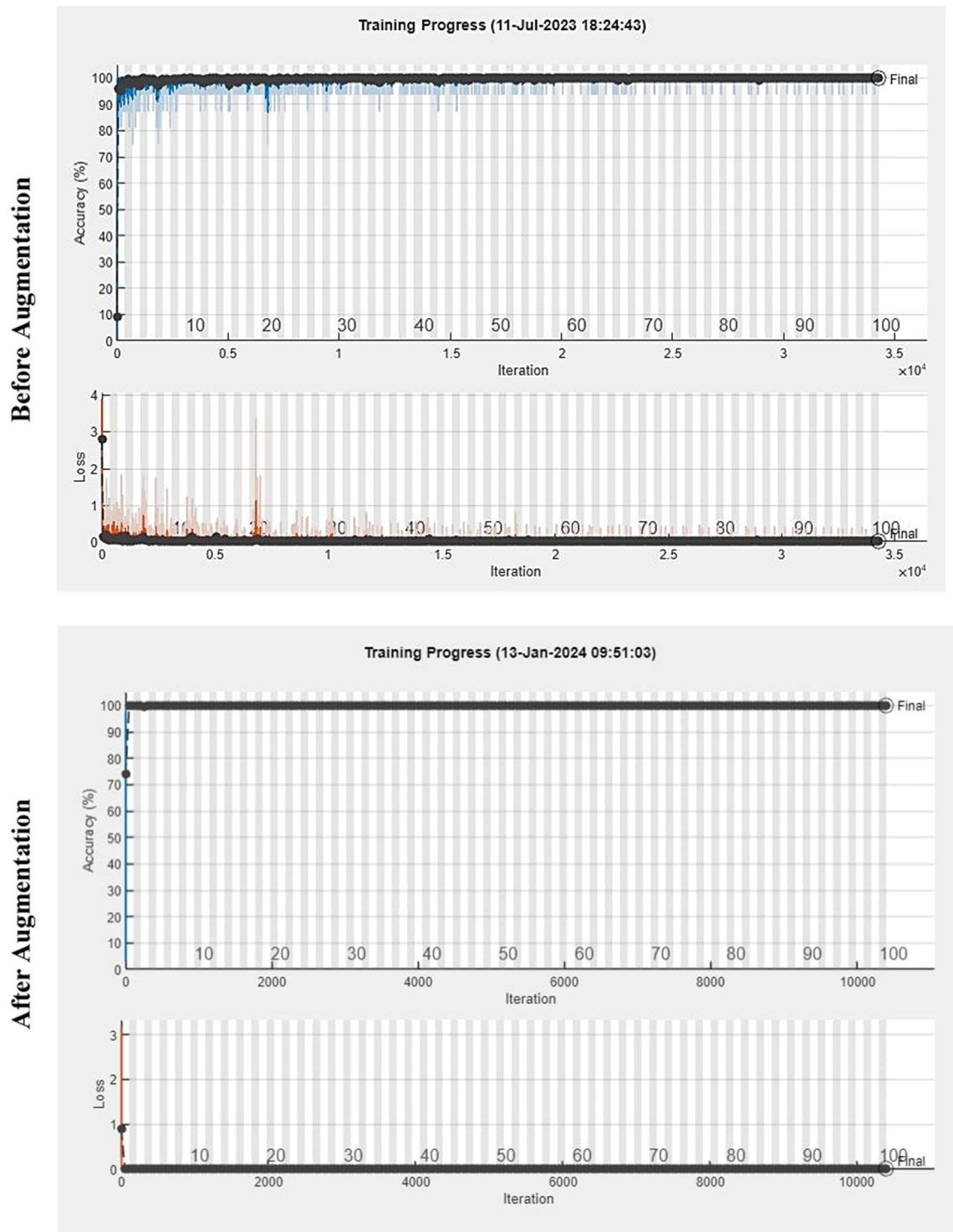


Fig. 6 Training progress for 11 classes

**Algorithm 1** Preprocessing Algorithm**Input:** Audio files**Output:** 34 audio**For each** audio in the dataset folders **Do:**

1. Partition the signal by a 25 ms Hann window with 75% overlap
2. Applied Short-Time Fourier Transform (STFT) to segment the whole audio using the following equation:

$$X_{\text{STFT}}(n,k) = \sum_{m=0}^{L-1} x(m)w(m-n)e^{-j2\pi km/L}, \quad k, n = 0, 1, \dots, L-1 \quad (5)$$

Where  $x(m)$  represents the signal,  $w(m)$  indicates *the* window function (hamming window) and  $m$  is the amount of shift.

3. Computing Discrete Fourier transform (DFT) for each segment using parent

$$y_i = \sum_{t=1}^N x_i(t) \exp\left(\frac{-j2\pi f(t-1)}{f}\right) \quad (6)$$

**Where**  $x_i$  is the  $i^{\text{th}}$  audio segment,  $y_i$  is the  $i^{\text{th}}$  segments' ranges,  $t$  and  $f$  are the time and frequency domain,  $N$  is the audio segment's overall amount of time samples.

4. Save the 34 segmented audio where each audio is 5s in the new folder

**End****Table 4** Training parameters

| Parameters          | Before augmentation | After augmentation |
|---------------------|---------------------|--------------------|
| Epoch               | 100                 | 100                |
| Maximum iteration   | 34,300              | 10,400             |
| Iteration per Epoch | 343                 | 104                |
| Training time (s)   | 3633                | 945                |

**2.6.2 Feature extraction phase**

Feature extraction is a critical step in our system model for avoiding model training errors, reducing computation time,

and improving model prediction accuracy. As a result, in this paper, Mel-spectrogram and VGG learning model are performed, as a features extractor to learn features from input data as shown in Algorithm 2.

**Table 5** Augmentation parameters

| Augmentation parameters    | Value    | Meaning  |
|----------------------------|----------|--|
| Time stretch probability   | 0.8      | This determines the probability of time-stamping an audio sample. When the value is 0.8, there is an 80% possibility that time stretching will be used   |
| Speedup factor range       | 1.3, 1.4 | This establishes the range of speed factors that can be used when extending time. When the factor is 1.3, the audio will be accelerated by 30%, and when it is 1.4, it will be increased by 40%          |
| Pitch shift probability    | 1        | This establishes the probability of using pitch shifting on a sample of audio. Two values are required: 0 or 1. When the value is 0, pitch shifting is not used  |
| Volume control probability | 0.8      | This determines the probability that an audio sample will be subjected to volume control. When the value is 0.8, the likelihood of volume control being used is 80%                                      |
| Volume gain range          | [- 5, 5] | This indicates the range of volume gain (measured in decibels) that can be used when controlling volume. A value of - 5 indicates a 5 dB drop in volume, and a value of 5 indicates a 5 dB raised volume |
| Add noise probability      | 1        | This determines the probability of introducing noise into a sample of audio. If the value is 0, then no noise will be added  |
| Time shift probability     | 0.8      | This determines the probability of time-shifting an audio sample. A value of 0.8 indicates that time shifting is applied with an 80% probability   |

**Algorithm 2** Feature Extraction Algorithm

**Input:** 34 Audio

**Output:** 4892 Extracted Features

**For each** audio in the folder **Do**

1. Compute the Mel scale filter bank using the following equation:

$$H_k(m) = \begin{cases} 0 & \text{for } m < f(k-1) \\ \frac{m - f(k-1)}{f(k) - f(k-1)} & \text{for } f(k-1) \leq m \leq f(k) \\ \frac{f(k+1) - m}{f(k+1) - f(k)} & \text{for } f(k) < m \leq f(k+1) \\ 0 & \text{for } m > f(k+1) \end{cases}$$

2. Generation the spectrogram images for each audio using the following equation:

$$X_{Mel}(x) = \sum_{m=0}^{N-1} [|X(m)|^2 H_k(m)]$$

3. Separated the spectrogram images into small windows.

**For each** spectrogram image **Do**

1. Import VGGish model.
2. Resizing image to 96×64 pixels
3. Passing images to VGGish
4. Collect the features that are the result of passing images to VGGish

**End**

**End**

**Table 6** Performance metrics of proposed model for before and after augmentation

| Parameters  | Before augmentation (%) | After augmentation (%) |
|-------------|-------------------------|------------------------|
| Accuracy    | 99.83                   | 99.98                  |
| Sensitivity | 99.82                   | 99.98                  |
| Specificity | 99.98                   | 100                    |
| Precision   | 99.81                   | 99.98                  |
| F1_score    | 99.82                   | 99.98                  |

### 2.6.3 Convolution neural network structure

The feature extraction process involves transforming the extracted features from the sound signals using a convolutional neural network (CNN) to accurately classify and detect various payload weights, as depicted in Fig. 3. The CNN architecture consists of 5 convolution layers, 3 Max-pooling layers, 1 average pooling layer, and 3 FC layers. Following each convolutional layer, the ReLU is employed as a nonlinear activation function. The softmax activation function is utilized to define the output size of the final FC layer. For optimization, the Adam optimizer is utilized, and the loss function employed is the cross entropy. The

**Table 7** Performance metrics of the proposed model on different payloads at 15 batch size

| Payloads | Accuracy | Sensitivity | Specificity | Precision | F1_score | Kappa  | J-index |
|----------|----------|-------------|-------------|-----------|----------|--------|---------|
| 0        | 99.09    | 99.09       | 100         | 100       | 99.543   | 0.8143 | 0.9909  |
| 50       | 100      | 100         | 100         | 100       | 100      | 0.8304 | 1       |
| 100      | 100      | 100         | 100         | 100       | 100      | 0.81   | 1       |
| 150      | 98.92    | 98.93       | 99.908      | 98.925    | 98.925   | 0.8433 | 0.9883  |
| 200      | 100      | 100         | 99.907      | 99.038    | 99.517   | 0.8237 | 0.9907  |
| 250      | 100      | 100         | 100         | 100       | 100      | 0.8134 | 1       |
| 300      | 100      | 100         | 100         | 100       | 100      | 0.8236 | 1       |
| 350      | 100      | 100         | 100         | 100       | 100      | 0.8304 | 1       |
| 400      | 100      | 100         | 100         | 100       | 100      | 0.8066 | 1       |
| 450      | 100      | 100         | 100         | 100       | 100      | 0.8134 | 1       |
| 500      | 100      | 100         | 100         | 100       | 100      | 0.7913 | 1       |

**Table 8** Performance metrics of the proposed model on different payloads at 50 batch size

| Payloads | Accuracy | Sensitivity | Specificity | Precision | F1_score | Kappa | J-index |
|----------|----------|-------------|-------------|-----------|----------|-------|---------|
| 0        | 99.091   | 99.09       | 100         | 100       | 99.543   | 0.81  | 0.99    |
| 50       | 100      | 100         | 100         | 100       | 100      | 0.83  | 1       |
| 100      | 100      | 100         | 100         | 100       | 100      | 0.81  | 1       |
| 150      | 97.849   | 97.849      | 99.908      | 98.913    | 98.378   | 0.84  | 0.9774  |
| 200      | 100      | 100         | 99.814      | 98.095    | 99.038   | 0.83  | 0.998   |
| 250      | 100      | 100         | 100         | 100       | 100      | 0.81  | 1       |
| 300      | 99.038   | 99.038      | 100         | 100       | 99.517   | 0.82  | 0.99    |
| 350      | 100      | 100         | 100         | 100       | 100      | 0.83  | 1       |
| 400      | 100      | 100         | 99.906      | 99.13     | 99.563   | 0.806 | 0.999   |
| 450      | 100      | 100         | 100         | 100       | 100      | 0.813 | 1       |
| 500      | 100      | 100         | 100         | 100       | 100      | 0.791 | 1       |

**Table 9** The average metrics of the proposed model on different batch size

| Batch size | Accuracy | Sensitivity | Specificity | Precision | F1_score |
|------------|----------|-------------|-------------|-----------|----------|
| 15         | 99.83    | 99.82       | 99.98       | 99.81     | 99.82    |
| 50         | 99.66    | 99.63       | 99.97       | 99.65     | 99.64    |
| 200        | 99.75    | 99.72       | 99.97       | 99.73     | 99.72    |



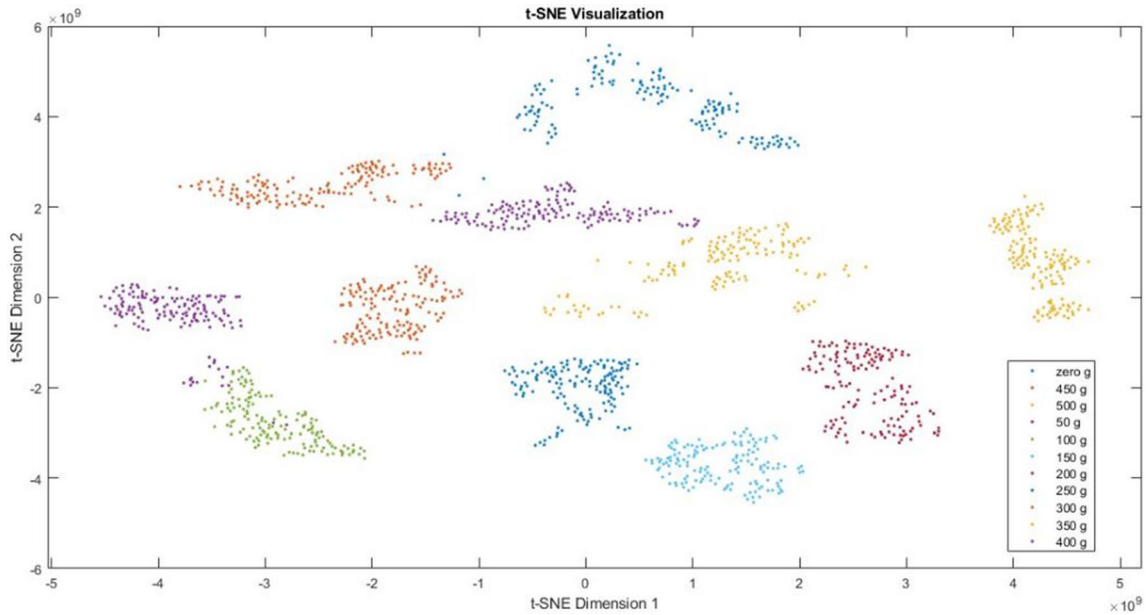


Fig. 7 The t-SNE illustration of feature for 11-class in test class

training and verification of the models are performed for 50 epochs with a batch size of 100.

### 3 Results and discussion

The metrics for examination and the experiment’s findings are given in this section. Every experiment is run on a computer with an 8th generation Intel® Core™ i7, 16 GB RAM, a 64-bit of Windows 11, and MATLAB 2022b. 70% of the dataset in the entire experiment was used for the training phase, 15 for validation and the remaining 15% was used for testing.

#### 3.1 Performance metrics

The effectiveness of experimental results is evaluated using a variety of measures, including recall, F-measure, accuracy, and precision. The following are the definitions of the metrics:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

$$\text{Recall} = \text{sensitivity} = \frac{TP}{TP + FN} \tag{8}$$

$$\text{F1 - Score} = \frac{2TP}{2TP + FP + FN} \tag{9}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \tag{10}$$

$$\text{Specificity} = \text{TNR} = \frac{TN}{TN + FP} \tag{11}$$

$$\text{sub-index index} = \text{Specificity} + \text{sensitivity} - 1 \tag{12}$$

$$\text{kappa} = \frac{\text{Accuracy} - \text{random Accuracy}}{1 - \text{random Accuracy}} \tag{13}$$

where

$$\begin{aligned} \text{RandomAccuracy} \\ = \frac{(TN + FP)(TN + FN) + (FN + TP)(FP + TP)}{(TP + TN + FP + FN)^2} \end{aligned} \tag{14}$$

where FP, FN, TP and TN is the False Positive, False Negative, True Positive, and True Negative, respectively.

Another parameter known as the area under curve (AUC) is also measured. AUC runs from [0, 1], and the closer it is to one, the better the suggested model is at recognizing drone weight.

#### 3.2 Evaluation

The dataset undergoes several preprocessing steps, as shown in Fig. 4. Start by dividing the audio data into 34 audio files, each of which is 5 s, using a periodic Hann window of 25ms with a 75% window length overlap.

Figure 5 displays the Mel spectrum of the drone’s audio signals when the drone carrying a 0g payload, 250 g and finally 500g payload. The time domain of the signal is shown on the x-axis, and the frequency domain is represented by the y-axis.

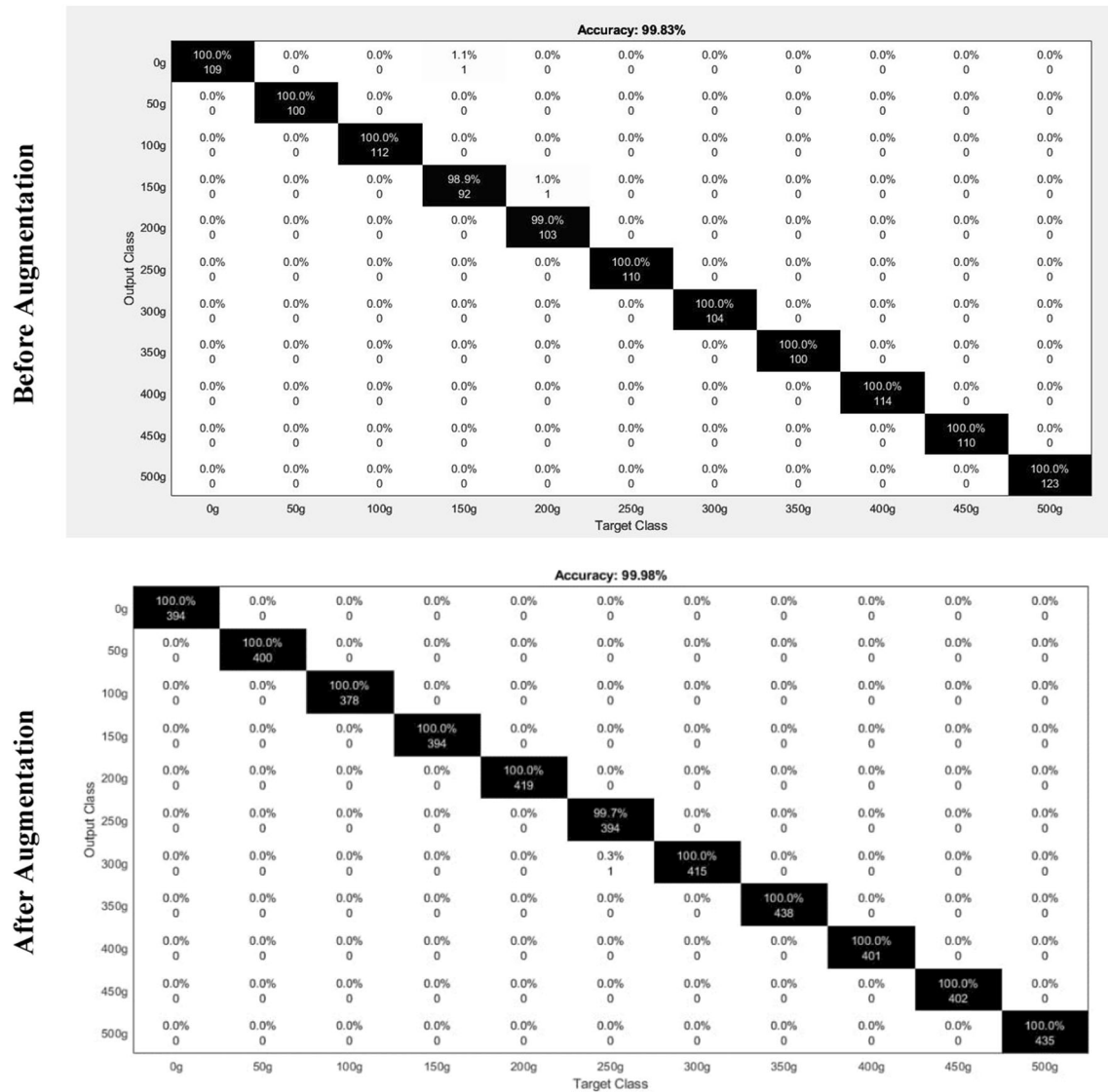


Fig. 8 The confusion matrix of the proposed model in the test class

According to the details presented in Table 2, the CNN architecture proposed in this paper comprises a total of 5 convolutional layers, 3 normalization layers, and 4 FC layers. For each convolutional layer, a 1 × 1 stride and causal padding are applied to the input. The max pooling layer, on the other hand, utilizes a 2 × 2 stride and zero padding. To execute the classification task, the output size of the final FC layer is followed by the application of the softmax activation function. This design aims to achieve effective feature extraction and enable the network to make accurate predictions for the classification task at hand.

Adam is an optimization algorithm that can be used instead of the traditional methods to update the weights of the network iterative during the training data. As shown in Table 3, different parameters are utilized for training our model. The learning rate, also known as alpha, is 0.0001

and represents the proportion at which weights are modified. The number of samples we utilize in one epoch to train a neural network is defined by the batch size, which is 15.

Figure 6 displays the training progress for 11 classes. The first row displays the data before augmentation, and the second row shows the data after augmentation. The upper subplot in this figure displays the accuracy on the training and validation set, while the lower subplot represents the training and validation loss. It is observed that the accuracy appears steadily, and there is no overfitting detected.

Table 4 displays the training parameters before and after the augmentation, as well as the maximum iteration, number of epochs, number of iterations, and training time.

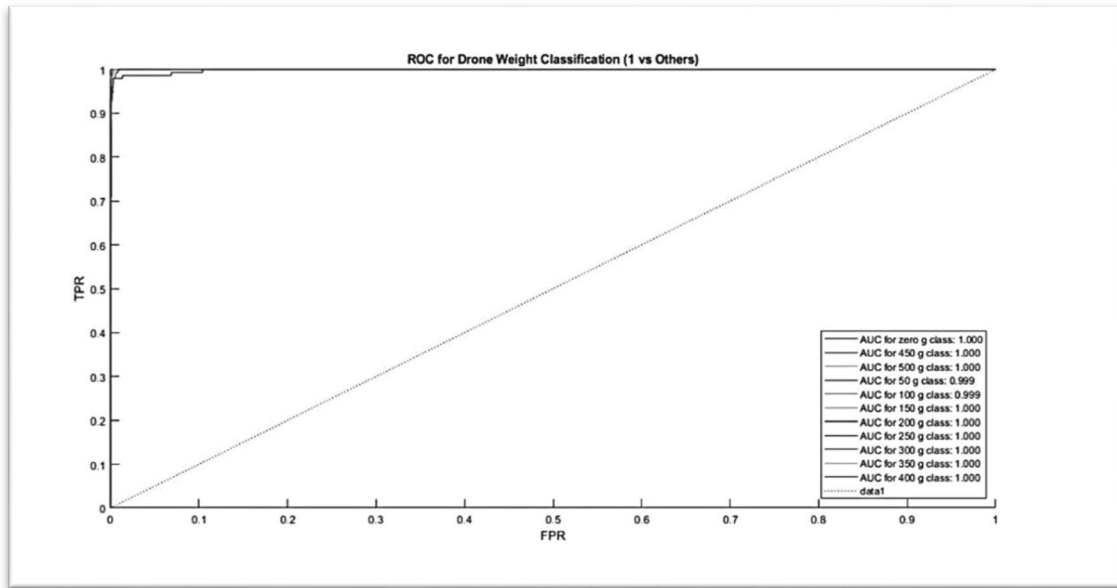


Fig. 9 AUC curve for different payload weight

Table 10 Comparison of obtained average accuracy with different models

| References            | Algorithm             | Accuracy (%) |
|-----------------------|-----------------------|--------------|
| Phuc et al. [15]      | MFCC & GMM SVM        | 96.27        |
| Traboulsi et al. [35] | LSTM & MFCC           | 87.5         |
| Savio et al. [19]     | MFCC & Cubic SVM      | 98           |
| Proposed model        | Mel-spectrogram & CNN | 99.98        |

Preprocessing step takes 65.15 s, feature extraction step takes 45.72 s, and predict load takes 0.937 s.

On the input audio signal, a variety of audio augmentation techniques are used, including pitch shifting, time stretching, volume control, time shifting, and noise probability. Each audio sample is sequentially subjected to augmentation techniques. Table 5 displays the augmentation parameters along with each augmentation’s value and significance.

Table 6 displays the performance metrics of the suggested model for both before and after augmentation. Performance metrics have increased compared to before augmentation, as this table indicates

Table 7 displays the results for each of the 11 classes at 15 batch sizes based on various performance criteria on testing. All payloads attain 100% accuracy and sensitivity; except for payloads 0 and 150 g achieve 99.09 and 98.92% accuracy, respectively. Additionally, all payloads, except for payloads of 150 g and 200 g, achieve 100% specificity and precision. The potential scores for kappa are  $-1$  (lowest performance) to  $1$  (highest performance). The

J-index is the vertical distance between the ROC curve and the equal line. The maximum value of the J-index is  $1$  (perfect test). The value of  $0$  indicates that there is no agreement between the actual and classified classes. Kappa ranges from  $-1$  to  $+1$ . A score of  $1$  denotes complete agreement between the classes predicted by the model and the actual classes

Based on a variety of performance metrics, Table 8 shows the results for each of the 11 classes at a batch size of 50. Except for payloads 0, 150, and 300 g, all payloads reach 100% for accuracy and sensitivity; these payloads achieve 99.091, 97.849, and 99.038%, respectively.

Table 9 displays the proposed model’s average metrics for various batch sizes. The lowest metrics are attained at batch sizes of 50, while the best metrics are at batch sizes of 15. The results achieve 99.83% accuracy, 99.82% sensitivity, 99.98% specificity, and 99.82% F1-score at 15 batch sizes. Furthermore, as shown in Fig. 7, the collected features are analyzed using t-SNE. The eleven classes are addressed by the variation type 11 colors. It highlights that our model yields higher evaluation metrics

Figure 8 depicts the proposed model's confusion matrix. The first row displays the data before augmentation, and the second row shows the data after augmentation. The graphic shows that the proposed model effectively classified the 11 payloads. Upon analysis of the provided data, most of the few samples that came from recordings with nearby payload weight classes are classified incorrectly. As can be seen in Fig. 9, the AUC values for all payloads are high overall, showing that the model can discern between various weights with good accuracy.

As indicated in Table 10, to assess our model, the suggested model is compared with other models. In [15], the RF method for detecting and estimating drone load is presented. The system tests 5 distinct drone models called the Parrot, Phantom, Mavic, the Parrot Bebop 2, and the Autel EVO. Each drone is equipped with weights ranging between 0 and 400g. The drone is up to 200 m away from the receiver when it is in flight. Mel-frequency cepstral coefficient (MFCC) for feature extraction and Gaussian mixture model SVM (GMM SVM) for classification allow DroneScale to estimate drone load with an average accuracy of 96.27%. The model in [35] investigates the classification of a drone carrying a payload, obtaining 87.5% accuracy using MFCC and a long short-term memory (LSTM). Savio et al. [19] use the MFCC components of the audio signal and several SVM classifiers to achieve an approximate classification accuracy of 98% in the identification of the specific payload class transported by the drone.

## 4 Conclusion and future work

This paper presents a novel and highly effective model for accurately detecting various payload weights carried by a 3DR SOLO drone using sound data. Initially, the dataset is segmented into 34 audio signals and employs the Mel-spectrogram and VGGish model for feature extraction. Then, the CNN network is used for classification, and the Adam optimization algorithm is used to iteratively update the network's weights during the training phase. In the end, two experiments are conducted to assess the model. While the second experiment used the augmented data, the first experiment used the original data (before augmentation). Different payload weights are identified with a potential accuracy of 99.98%, sensitivity of 99.98%, and specificity of 100% based on experimental results. Furthermore, a thorough comparison with previous research that made use of the same dataset confirms the superiority of our suggested model.

Overall, our paper contributes significantly to the field of drone payload weight detection, providing an innovative and reliable solution for this essential task. As the use of

drones continues to grow across various industries, the ability to accurately detect payload weights becomes increasingly vital for safety, regulatory compliance, and operational efficiency. We hope that our findings will pave the way for further advancements in this area, and that our proposed model will serve as a solid foundation for future investigation and practical applications.

Future work will explore the model's efficacy in handling heavier payloads across various drone models using deep learning techniques. Diverse and extensive datasets will be employed to enhance generalization capabilities. Additionally, investigating alternative feature extraction methods and network architectures will be pursued for improved results. Our model significantly advances payload weight identification from drone sound, providing a stable and reliable solution for critical applications in multiple industries. We anticipate our proposed model to have a positive impact in real-world settings, and our future investigations will further strengthen its applicability and performance. In addition, we plan to incorporate additional datasets from diverse drone models in our future work to validate the model's performance across a broader range of scenarios.

**Funding** Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

**Data availability** The data will be available at, <https://github.com/crilab-hbku/Drone-Payload>, access 22/2/2024.

## Declarations

**Conflict of interest** The corresponding author certifies that there is no conflict of interest on behalf of all authors.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Macrina G, Di Puglia Pugliese L, Guerriero F, Laporte G (2020) Drone-aided routing: a literature review. *Transp Res Part C Emerg Technol* 120:102762. <https://doi.org/10.1016/j.trc.2020.102762>



2. Mohsan SAH, Khan MA, Noor F, Ullah I, Alsharif MH (2022) Towards the unmanned aerial vehicles (UAVs): a comprehensive review. *Drones* 6(6):147. <https://doi.org/10.3390/drones6060147>
3. Murugan D, Garg A, Singh D (2017) Development of an adaptive approach for precision agriculture monitoring with drone and satellite data. *IEEE J Sel Topics Appl Earth Obs Remote Sens* 10(12):5322–5328
4. Raghavan S (2019) Saudis say oil pipeline was attacked by drones, possibly from Yemen
5. Yaacoub J-P, Noura H, Salman O, Chehab A (2020) Security analysis of drones systems: attacks, limitations, and recommendations. *Internet Things* 11:100218. <https://doi.org/10.1016/j.iot.2020.100218>
6. Pyrgies J (2019) The UAVs threat to airport security: risk analysis and mitigation. *J Airl Airpt manag* 9(2):63. <https://doi.org/10.3926/jairm.127>
7. Semkin V et al (2021) Drone detection and classification based on radar cross section signatures. In: 2020 international symposium on antennas and propagation (ISAP). IEEE
8. Sazdić-Jotić B et al (2022) Single and multiple drones detection and identification using RF based deep learning algorithm. *Expert Syst Appl* 187:115928
9. Utebayeva D, Ilipbayeva L, Matson ET (2022) Practical study of recurrent neural networks for efficient real-time drone sound detection: a review. *Drones* 7(1):26
10. Park H et al (2022) Method for improving range resolution of indoor FMCW radar systems using DNN. *Sensors* 22:8461
11. Park J, Jung DH, Bae KB, Park SO (2020) Range-Doppler map improvement in FMCW radar for small moving drone detection using the stationary point concentration technique. *IEEE Trans Microw Theory Tech* 68(5):1858–1871
12. Rahman S, Robertson DA (2019) Classification of drones and birds using convolutional neural networks applied to radar micro-Doppler spectrogram images. *IET Radar Sonar Navig* 14(5):653–661
13. Raina K et al (2022) Detecting UAV presence using convolution feature vectors in light gradient boosting machine. *IEEE Trans Veh Technol* 72:4332–4341
14. Seidaliyeva U, Alduraibi M, Ilipbayeva L, Almagambetov A (2020) Detection of loaded and unloaded UAV using deep neural network. In: IEEE international conference on robotic computing (IRC). IEEE, pp 490–494
15. Nguyen P, Kakaraparthi V, Bui N, Umamahesh N, Pham N, Truong H, Guddeti Y, Bharadia D, Han R, Frew E (2020) DroneScale: drone load estimation via remote passive RF sensing. In: Proceedings of the 18th conference on embedded networked sensor systems, pp 326–339
16. SquareHead Technology, Discovair G2 (2022) Available at <https://tinyurl.com/y2s4c2wl>. Accessed 30 Mar 2022
17. DroneShield Company, DroneShield (2022) Available at <https://www.droneshield.com/>. Accessed 30 Mar 2022
18. Advanced Protection Systems (aps), Ctrl+Sky (2022) Available at <https://tinyurl.com/yy3q9a5f>. Accessed 30 Mar 2022
19. Ibrahim OA, Sciancalepore S, Di Pietro R (2022) Noise2Weight: on detecting payload weight from drones acoustic emissions. *Future Gener Comput Syst* 134:319–333
20. Maity A, Pathak A, Saha G (2023) Transfer learning based heart valve disease classification from Phonocardiogram signal. *Biomed Signal Process Control* 85:104805
21. Kılıç R, Kumbasar N, Oral EA, Ozbek IY (2022) Drone classification using RF signal based spectral features. *Eng Sci Technol Int J* 28:101028. <https://doi.org/10.1016/j.jestch.2021.06.008>
22. Tammina S (2019) Transfer learning using vgg-16 with deep convolutional neural network for classifying images. *Int J Sci Res Publ (IJSRP)* 9(10):143–150
23. Diwakar M, Gupta B (2023) The robust feature extraction of the audio signal by using VGGish model
24. Grollmisch S et al (2021) Analyzing the potential of pre-trained embeddings for audio classification tasks. In: 2020 28th European signal processing conference (EUSIPCO). IEEE
25. Tsaleri E, Papadakis A, Samarakou M (2021) Comparison of pre-trained CNNs for audio classification using transfer learning. *J Sens Actuator Netw* 10(4):72
26. Saqib M et al (2017) A study on detecting drones using deep convolutional neural networks. In: 2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS). IEEE
27. Samma H et al (2021) Evolving pre-trained CNN using two-layers optimizer for road damage detection from drone images. *IEEE Access* 9:158215–158226
28. Zhang Z (2023) Drone-YOLO: an efficient neural network method for target detection in drone images. *Drones* 7(8):526
29. Oh HM, Lee H, Kim MY (2019) Comparing convolutional neural network (CNN) models for machine learning-based drone and bird classification of anti-drone system. In: 2019 19th international conference on control, automation and systems (ICCAS). IEEE
30. El-Sayed F, El-Shafai W, Taha TE (2021) Efficient fusion of medical images based on CNN. *Menoufia J Electron Eng Res* 30:79–83
31. He Q, Yang Q, Xie M (2023) HCTNet: A hybrid CNN-transformer network for breast ultrasound image segmentation. *Comput Biol Med* 155:106629
32. Singh R et al (2023) Facial expression recognition in videos using hybrid CNN & ConvLSTM. *Int J Inf Technol* 15:1819–1830
33. Alzubaidi L et al (2021) Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J Big Data* 8:1–74
34. Drone-Payload. Available at, <https://github.com/cr-lab-hbku/Drone-Payload>. Accessed 14 July 2023
35. Traboulsi A, Barbeau M (2021) Identification of drone payload using Mel-frequency cepstral coefficients and LSTM Neural networks. In: Proceedings of the future technologies conference (FTC) 2020, vol 1. Springer International Publishing

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.