


# An audio-based risky flight detection framework for quadrotors

Wansong Liu<sup>1</sup> | Chang Liu<sup>1</sup>  | Seyedomid Sajedi<sup>2</sup> | Hao Su<sup>3</sup> | Xiao Liang<sup>2</sup> |  
Minghui Zheng<sup>1</sup> 

<sup>1</sup>Department of Mechanical and Aerospace Engineering, University at Buffalo, Buffalo, New York, USA

<sup>2</sup>Department of Civil, Structural and Environmental Engineering, University at Buffalo, Buffalo, New York, USA

<sup>3</sup>Department of Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, North Carolina, USA

## Correspondence

Minghui Zheng and Xiao Liang.  
Email: [mhzheng@buffalo.edu](mailto:mhzheng@buffalo.edu) and [liangx@buffalo.edu](mailto:liangx@buffalo.edu)

## Funding information

National Science Foundation - U.S., Grant/Award Number: 2046481

## Abstract

Drones have increasingly collaborated with human workers in some workspaces, such as warehouses. The failure of a drone flight may bring potential risks to human beings' life safety during some aerial tasks. One of the most common flight failures is triggered by damaged propellers. To quickly detect physical damage to propellers, recognise risky flights, and provide early warnings to surrounding human workers, a new and comprehensive fault diagnosis framework is presented that uses only the audio caused by propeller rotation without accessing any flight data. The diagnosis framework includes three components: leverage convolutional neural networks, transfer learning, and Bayesian optimisation. Particularly, the audio signal from an actual flight is collected and transferred into time–frequency spectrograms. First, a convolutional neural network-based diagnosis model that utilises these spectrograms is developed to identify whether there is any broken propeller involved in a specific drone flight. Additionally, the authors employ Monte Carlo dropout sampling to obtain the inconsistency of diagnostic results and compute the mean probability score vector's entropy (uncertainty) as another factor to diagnose the drone flight. Next, to reduce data dependence on different drone types, the convolutional neural network-based diagnosis model is further augmented by transfer learning. That is, the knowledge of a well-trained diagnosis model is refined by using a small set of data from a different drone. The modified diagnosis model has the ability to detect the broken propeller of the second drone. Thirdly, to reduce the hyperparameters' tuning efforts and reinforce the robustness of the network, Bayesian optimisation takes advantage of the observed diagnosis model performances to construct a Gaussian process model that allows the acquisition function to choose the optimal network hyperparameters. The proposed diagnosis framework is validated via real experimental flight tests and has a reasonably high diagnosis accuracy.

## KEYWORDS

aerial robotics, deep learning, flying robots, propeller diagnosis

## 1 | INTRODUCTION

In recent decades, considering the low cost and high maneuverability, there has been increasing interest in applying drones in different fields, including transportation monitoring [1], infrastructure inspection [2, 3], and disaster resilience [4, 5]. Despite these applications, the sophisticated scenario accomplishment is premised on the drone's steady flight. The potential safety issue associated with an unsteady drone flight

limits the reliability of the drone application, especially considering that some drone tasks in the warehouse, including moving, carrying, and monitoring, require close proximity between drones and human workers [6]. This potential issue increases the risk of physical injury to human workers.

Therefore, it is no surprise that many researchers have paid attention to the theme of drone fault diagnosis. Avram et al. [7] proposed a dynamics-based method to detect, isolate, and estimate sensor bias faults in drone accelerometers and

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs](https://creativecommons.org/licenses/by-nc-nd/4.0/) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2024 The Authors. *IET Cyber-Systems and Robotics* published by John Wiley & Sons Ltd on behalf of Zhejiang University Press.

gyroscope measurements. Al Younes et al. [8] introduced an output estimator to detect the sensor faults. Hansen and Blanke [9] used non-linear analytical redundancies to detect and isolate the airspeed sensor and even the wind speed, but the propulsion dynamics are unknown. On the other hand, compared to the abnormal computational commands caused by the fault sensor, the physically damaged propeller may cause the flight to unbalance and fail directly, which threatens human beings' life safety. Ghalamchi and Mueller [10] used the vibration data captured by a built-in accelerometer sensor to discriminate the physically broken propeller. The collected audio data in de Jesus Rangel-Magdalenos et al. [11] were transformed into spectra that depict the distribution of energy across various frequency bands. Subsequently, a manual statistics-based discrimination process was employed to distinguish the intact and physically broken propellers. In a similar vein, Iannace et al. [12] first collected the audio data from the quadrotor fixed in a tripod, and then built an artificial neural network to process the converted spectrums and automatically detect the unbalanced propeller.

The audio signal produced from propeller rotation could be a critical criterion for analyzing the drone's abnormal behavior. Considering the time–frequency representation property, ‘spectrogram’ is a common approach for representing the audio signals of drones. Cabrera-Ponce et al. [13] utilized spectrogram images to train a network to identify whether a drone was flying nearby. Harmanny et al. [14] applied the spectrogram to discriminate birds and mini-UAVs. Spectrogram has the potential to display the characteristics of a drone flight with a broken propeller. Even if it is feasible for human beings to use their eyes to discriminate the broken propeller upon spectrogram images, the time taken for identification may not be fast enough to guarantee human safety when facing an unsteady flying drone.

To guarantee efficiency and accuracy, deep neural networks turn out to be great candidates in the fault diagnosis fields [15–17]. Among different kinds of neural network architectures, convolutional neural network (CNN) possesses outstanding decision-making capability for image-related studies [18, 19]. Therefore, CNNs have been widely incorporated into fault diagnosis studies to extract some abnormal features of images. For example, Liu et al. [20] employed infrared images to build up a CNN for rotating machinery fault diagnosis. Guo et al. [21] implemented short-time Fourier transform to transform the signal of the global positioning system, inertial measurement unit, and air data system into two-dimensional images for training and used CNN to distinguish the damaged drones.

Furthermore, due to the natural uncertainty of the network-based models, these diagnosis models cannot be fully trusted, especially when they are related to human safety [22]. The uncertainty can be used to quantify the confidence degree of a model for a diagnosis problem. For example, a model might possess high uncertainty if the softmax output is 0.49 and 0.51 in a binary classification problem. A diagnosis model can be uncertain in its decision even with a high softmax output [23]. Therefore, besides the decision made by the softmax outputs, it is necessary to explicitly measure and

consider the uncertainty of the network-based diagnosis model to achieve a more reliable and accurate diagnosis result.

The fault diagnosis of different drones requires repetitive training work since only one well-trained network is capable of diagnosis for one specific drone. It is important to efficiently generalise the CNN-based diagnosis method to other drones. In recent years, transfer learning techniques have been used to extend the adaption of a model, which takes a big step forward toward the development of a generalised fault diagnosis model. Zhong et al. [24] developed a CNN using a large-scale annotated gas turbine normal dataset and applied transfer learning to accomplish a fault diagnosis task with limited fault data. Wen et al. [25] proposed a supervised transfer learning to diagnose the data-driven fault by minimising the discrepancy penalty between source features and target features. Zhang et al. [26] employed transfer learning to guarantee the bearing fault diagnosis performance under different working conditions.

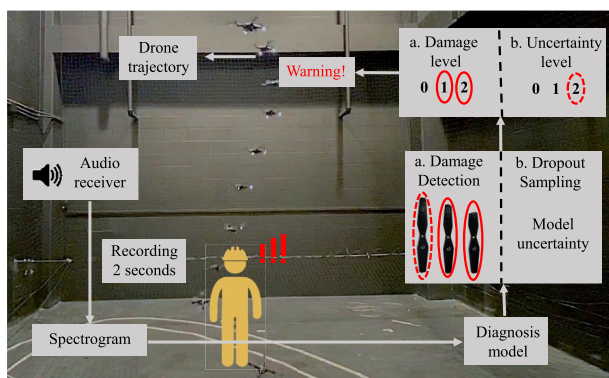
The training process of a network-based diagnosis model tunes the weights of the network layers to be optimal such that it guarantees a promising diagnosis accuracy. On the other hand, the model hyperparameters, for example, learning rate and mini-batch size, also contribute to the robustness of the fault diagnosis model. Different combinations of hyperparameter values may lead to different performances of a drone fault diagnosis model. Furthermore, the tuning of the model hyperparameters is usually heuristic and requires expert experience. Therefore, finding a suitable combination of the hyperparameter values is computationally expensive. Even though there are some hyperparameter optimisation algorithms for neural network learning like grid search or random search, the hyperparameter selection efficiency is still impeded by evaluating too many unnecessary candidates. Thus, considerable attention has been given to implementing Bayesian optimisation for network training [27–30]. Bayesian optimisation takes advantage of the evaluated objective function (e.g., validation error) to build up a probabilistic model such that it reduces unnecessary computation by selecting more reliable hyperparameters.

This paper proposes an audio-based deep learning method to detect if the flight may fail because of any physically broken drone's propeller. The audio signal was produced by actual quadrotors. Compared to traditional audio-based fault diagnosis, which directly uses audio signal data to do analysis [31, 32], this paper represents the audio signal as time–frequency spectrogram images with continuous energy variation. A CNN is built with a spectrogram dataset to diagnose the unbalanced propeller. This study was developed based on some preliminary works in Ref. [33]. Different from the works in Ref. [33], which use fixed drones on an iron tower, the data collection and experiment validation in this study were conducted with flying drones. This change improves the application potential of this fault diagnosis method. What's more, the employment of uncertainty quantification and Bayesian optimisation in this study refines the reliability and robustness of the diagnosis model.

In the scene shown in Figure 1, an unsteady flying drone is threatening the human worker's safety. The audio receiver

records the abnormal noise caused by the drone flight. First, the collected audio data are converted to a 2D spectrogram using short-time Fourier transform and colour map and sent to the network-based diagnosis model. Next, the model determines the condition of the drone propeller and estimates the damage level for the human worker's safety. Particularly, damage level '0' means the propeller is in good condition, flying steady, '1' means the propeller is slightly broken, and '2' means the propeller is seriously broken, may fall directly. The human worker will receive a warning based on the damage level and avoid potential physical injury. Furthermore, the confidence of the diagnosis model regarding the spectrogram input, that is, uncertainty, is also measured using Monte Carlo dropout sampling (MCDS), which has three levels as additional information to ensure the safety of human workers. For example, a broken propeller may be misclassified as a '0' damage level and have a '2' uncertainty level. In this case, the early warning would also be triggered to further ensure the safety of human workers due to the high uncertainty level. In addition, transfer learning and Bayesian optimisation are combined to obtain a generalised diagnosis model with selected training hyperparameters. This approach possesses great classification accuracy regarding detecting broken propellers during drone flights.

The uniqueness of this work is that the proposed diagnosis framework, which uses only the audio caused by the propeller rotation without accessing any flight data, can significantly promote wide application in many types of drones without reconfiguring the hardware or software. The main contributions of this paper are summarised as follows. (1) We explicitly evaluate the uncertainty of the diagnosis model and assess potential risks based on both the diagnosed propeller damage level and the quantified uncertainty level to ensure human



**FIGURE 1** Overview of the presented autonomous flight failure detection framework: this framework only uses the audio caused by the flight without accessing any other data from the drone and can quickly detect physical damage to propellers, quantify the confidence of the diagnosis, recognise a risky flight, and provide early warning to surrounding human workers. This framework is motivated by the fact that drones have increasingly collaborated with human workers in some workspaces, such as warehouses. The failure of drone flight may bring potential risks to human beings' life safety during some aerial tasks. This work aims to provide early and reliable warning of such failures, which can alert surrounding human workers in advance.

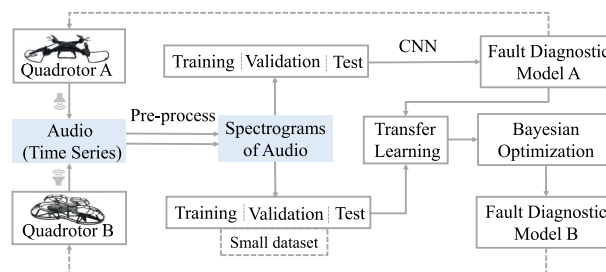
safety. (2) A generalised flying drone propeller diagnosis framework leveraging several network-based techniques is proposed. The fast and reliable decision-making process from CNN, the minor data requirement from transfer learning, and the optimised network hyperparameters from Bayesian optimisation enhance the efficiency, adaption, and robustness of this diagnosis framework, respectively. (3) The effectiveness of the proposed diagnosis framework has been experimentally validated using flying drones.

The remainder of this paper is organised as follows. Section 2 introduces the whole framework; Section 3 presents the data collection and pre-process; Section 4 presents the diagnosis method based on CNNs; Section 5 extends this method to different drones based on transfer learning; Section 6 combines Bayesian optimisation with transfer learning; and Section 7 concludes this paper.

## 2 | PROPOSED DRONE PROPELLER DIAGNOSIS SCHEME

This paper employs the audio caused by flight as the essential component of the diagnosis model. This method is easy to implement and does not require any additional sensors.

Figure 2 demonstrates the framework of the proposed diagnosis method. The propeller rotation of Quadrotor A generates a series of audio signal data that are captured in the time domain. To display more audio signal features, the time domain-based audio signals are automatically converted to a time–frequency domain image called a 'spectrogram'. The two-dimensional images are labelled based on different damage levels of propellers and separated into training, validation, and



**FIGURE 2** The architecture of the presented framework: the framework includes three components that leverage a convolutional neural network (CNN), transfer learning, and Bayesian optimisation. The audio signal from an actual flight is first collected and transferred into time–frequency spectrograms. A CNN model that utilises these spectrograms is then developed to identify whether there is any broken propeller involved in a specific drone flight. Also, to reduce data dependence on different drone types, the CNN-based diagnosis model is further augmented by transfer learning. That is, the knowledge of a well-trained diagnosis model is refined by using a small set of data from a different drone. The modified diagnosis model has the ability to detect the broken propeller of the second drone (i.e., Quadrotor B in the figure). Furthermore, to reduce the hyperparameters' tuning efforts and reinforce the robustness of the network, Bayesian optimisation takes advantage of the observed diagnosis model performances to construct a Gaussian process model, which allows the acquisition function to choose the optimal network hyperparameters.

test datasets. CNN is trained as the fault diagnosis model for Quadrotor A to distinguish the labelled spectrograms. To extend the exploitable range of the trained CNN model, transfer learning is implemented to obtain a modified network by using a small set of audio signals from Quadrotor B and a part of a well-trained network generated from Quadrotor A. To reinforce network robustness, Bayesian optimisation is employed to select the optimal network training hyperparameters by using a Gaussian process model built with the results of multiple training processes. The modified network, which corresponds to the optimal hyperparameters, has the promising capability to detect whether a broken propeller has been involved in Quadrotor B.

### 3 | AUDIO DATA COLLECTION AND ANALYSIS

#### 3.1 | Audio data processing

The audio data show the overall variation of the drone noise intensity with time. However, the flying drone with a broken propeller does not always display an apparent noise amplitude change in the audio signal. This study employs the spectrogram as the time–frequency interpretation of the audio signal collected from flying drones. The spectrogram illustrates the signal in both time and frequency domains, and highlights the temporal evolution of the audio signal spectral content [34]. To construct a spectrogram, an audio signal collected from a flying drone is first windowed into several short signals. Then, the Fourier transform is applied to compute the windowed segments that contain the frequency information. Eventually, the computed segments overlap with each other along with the time intervals according to a certain percentage. The horizontal and vertical axes of the constructed spectrograms describe the frequency and time of the drone's audio signal, respectively. The discrete-time spectrogram  $S(t, k)$  is denoted as:

$$S(t, k) = \sum_{n=0}^{N-1} w(n)a(t+n)e^{-i\frac{2\pi k}{N}n} \quad (1)$$

where  $t$  is time instance,  $k$  is frequency,  $w(n)$  is Hann window with  $N$  samples, and  $a(t)$  is the drone's audio signal. In addition, the point of the spectrogram is represented with a colour that indicates the intensity of the drone's audio signal.

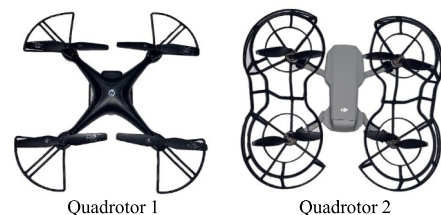
#### 3.2 | Experimental test platform and data collection

The constituents of the test platform are two types of quadrotors with different scales, several different level-damaged propellers, and an audio receiver. Figure 3 illustrates the two quadrotors used to collect audio data and verify the experimental test results. The selection of these two types of quadrotors is based on the following reasons: (1) Their physical

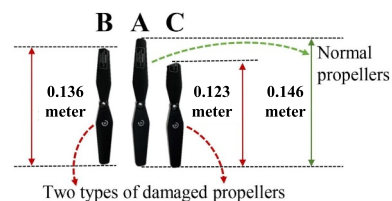
dimensions render them suitable for the execution of tasks within warehousing environments. (2) The proximity in size between these two quadrotors facilitates a reasonable application of diagnostic knowledge generalisation based on transfer learning techniques. (3) These two quadrotors possess the capability to be equipped with protective guards, thereby ensuring the safety of human operators during experimental tests, particularly in instances of broken propellers.

Notably, the potential damage conditions of propellers are theoretically limitless. Given the impracticality of exhaustively including all propeller damage conditions in our experimental tests, our approach centres on three representative damage levels of the propeller: the unbroken level, the marginally broken level, and the seriously broken level. Moreover, these three damage levels of the propeller can serve as critical thresholds, effectively covering most of the range of potential damage conditions of the propeller. Figure 4 shows three levels of damaged propellers with labels from quadrotor 1. The flawless propeller is labelled as A. The marginally damaged propeller is labelled as B. The propeller with serious damage is labelled as C. The propeller with label A has a 0.146 m diameter as length. The propeller with label B has a 0.136 m length. The length of the propeller labelled C is 0.123 m.

The audio signal data collection experiments are divided into three parts for the experimental quadrotor. Table 1 indicates the different propeller combinations used for the test configurations. The quadrotor is set to normal thrust command to make it hover when the propeller is good. When using a different combination of the broken propeller with good propellers, it cannot support continuous smooth flying and decreases altitude quickly. The audio recorder gathers each sample of audio data with a 2 s duration. Figure 5a,b show the



**FIGURE 3** Quadrotors used for the experimental tests. The first quadrotor is used to validate the CNN diagnosis model and the Bayesian optimisation algorithm, while the second quadrotor is used to validate the effectiveness of transfer learning in the framework.

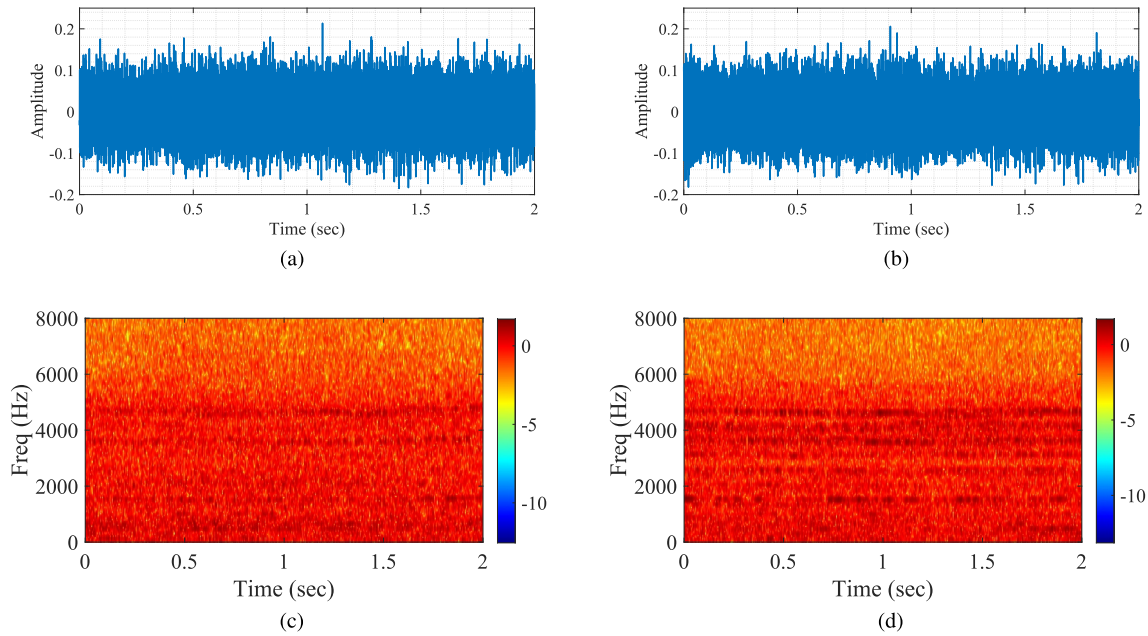


**FIGURE 4** Three types of propellers were used for experimental tests: A is an unbroken propeller, B is a slightly broken propeller, and C is a seriously broken propeller.



**TABLE 1** Test configurations.

Tests	Propeller sets	Thrust commands
Configuration 1	Four propellers in good condition	Normal
Configuration 2	Three propellers in good condition plus propeller B	Normal
Configuration 3	Three propellers in good condition plus propeller C	Normal



**FIGURE 5** Audio signals and spectrogram images generated from the drone with broken and unbroken propellers. (a) Audio signals generated from the flight with a broken propeller. (b) Audio signals generated from the flight with unbroken propellers. (c) Spectrogram generated from the flight with a broken propeller. (d) Spectrogram generated from the flight with unbroken propellers.

collected audio data samples with different damage level propellers, respectively.

The audio signal may display some specific features in the spectrogram due to propeller rotation. The audio signals are converted to spectrogram images according to Equation (1). Two spectrograms from broken and unbroken propellers are exhibited in Figure 5c,d. Unlike spectrums [12] that display the energy distribution of the audio data in the frequency domain, the spectrogram offers additional information in the time domain. Meanwhile, given the energy variation of the different frequencies and time instances, it displays diverse colors. The audio signal with a 2 s duration is distributed evenly and emerges as several straight lines with several certain frequency channels in the horizontal direction. Therefore, the spectrogram illustrates the audio signal in two dimensions both integrally and consecutively.

## 4 | IMAGE-BASED DEEP LEARNING DETECTION MODEL

### 4.1 | A CNN-based diagnosis model

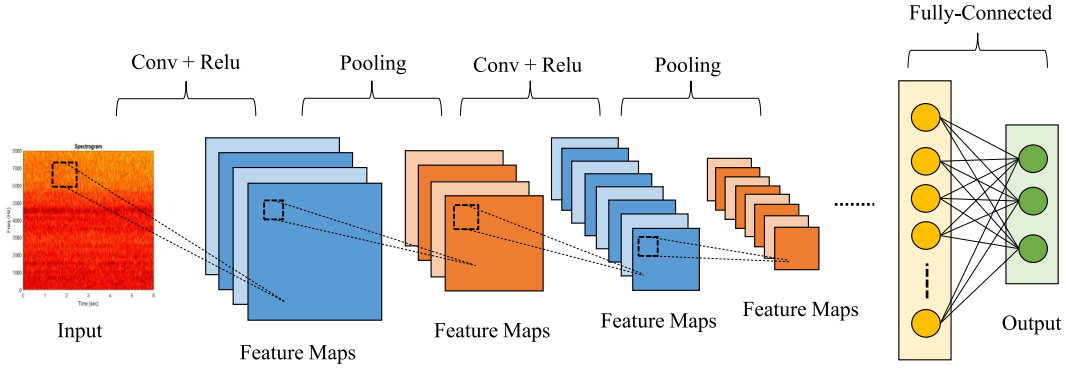
Although the properties of audio signals are demonstrated outstandingly by the spectrogram when diagnosing drone

propellers, how to efficiently distinguish spectrograms corresponding to drone flights with broken propellers is still a challenge. Many researchers have paid great attention to deep learning techniques to automatically monitor the health of different systems, including infrastructure [35–38]. Therefore, this paper proposes a CNN model to assist with decision-making.

To accurately diagnose the propeller condition, firstly, the audio features of the flying drone have to be extracted from the constructed spectrogram. Figure 6 shows the general architecture of CNN. The audio-based spectrogram image is the network input. The audio signal's power displayed in the spectrogram has red, green, and blue channels, where each channel is composed of brightness intensities from 0 to 255. A leaning kernel (filter) is employed to convolve around the audio-based spectrogram upon a certain stride such that the audio characteristics corresponding to each time and frequency can be extracted. The extracted audio feature map is obtained using the following equation:

$$M(i, j) = \sum_{u=-a}^a \sum_{v=-b}^b k(u, v) I(i + u, j + v) \quad (2)$$

where  $M$  is the extracted audio feature map,  $k$  is a  $u \times v$  size kernel, and  $I$  is the original spectrogram image. Furthermore,



**FIGURE 6** The CNN architecture: the two-dimensional spectrogram image is the input of the first convolution layer. The input image features are extracted using a filter based on Equation (2) in the convolution layer. The ReLU activation function turns the elements of the extracted feature maps to be non-negative. The pooling layer reduces the size of the convolved feature maps. The fully connected layer utilises the feature maps from the last pooling layer to predict the class of the input spectrogram image. The network output comprises three propeller conditions, that is, unbroken, slightly broken, and seriously broken propellers.

to prevent exponential growth in network training, each audio feature map  $M$  extracted from the convolutional layer passes through a Relu activation function  $R(x) = \max(0, x)$  that sets all negative values to zero.

Next, a preliminary diagnosis decision has to be made based on the extracted audio feature map. To assist the diagnosis model in making a decision, we apply the Softmax function to represent the extracted audio characteristics in a more straightforward way. The computed audio feature map in Equation (2) is converted to a probability score vector  $S$ . The diagnosis model training stochastically initialises the layer weights to learn the audio characteristics in the first propagative iteration. The network output  $\hat{Y}$  is labelled based on the different damage levels of the propeller. The evaluation of the diagnosis decision is described using the following cross-entropy loss equation:

$$L = -\frac{1}{Z} \sum_{n=1}^Z \sum_{c=1}^C B_{n,c} \{Y \neq \hat{Y}\} \log(s_{n,c}) \quad (3)$$

where  $Z$  is the observed spectrogram numbers,  $Y$  stands for ground truth propeller damage level label,  $C$  is the number of the propeller conditions,  $B_{n,c}$  is the binary indicator from  $Y$  and  $\hat{Y}$  for propeller damage level  $c$  in the observation  $n$ , and  $s_{n,c}$  is the probability score for propeller damage level  $c$  in the observation  $n$ .

Eventually, the accuracy of the propeller diagnosis decisions needs to keep improving during the training process. Equation (3) indicates the extent of the misclassification made by the diagnosis model. To minimize the misclassification, the stochastic gradient descent with momentum is implemented to modify the layer's parameters by backpropagation in each iteration using the following equation:

$$\theta_{i+1} = \theta_i - \eta \nabla L(\theta_i) + \alpha(\theta_i - \theta_{i-1}) \quad (4)$$

where  $\theta$  is the parameter vector (weights and biases) used to learn audio characteristics,  $\eta$  is the learning rate,  $\nabla L(\theta)$  is the gradient of loss calculated based on Equation (3), and  $\alpha$  is the momentum rate. By optimising  $\theta$ , the diagnosis model is capable of extracting valuable audio characteristics from spectrogram images such that the drone flight with a broken propeller can be accurately detected.

## 4.2 | Uncertainty quantification of propeller diagnosis model

The description in the previous subsection briefly illustrates the CNN-based diagnosis model. The propeller condition can be accurately identified based on a diagnosis model  $f$  with well-tuned parameters  $\Theta$ . However, such a data-driven model still leaves room for a level of uncertainty in propeller diagnosis. The natural uncertainty may result in mistakes in drone diagnosis, posing risks regarding the safety of human workers. Therefore, the propeller condition predicted by the diagnosis model should not be the only consideration for triggering early warning.

We introduce MCDS to capture the inconsistency of the diagnostic results and explicitly quantify the uncertainty of the diagnosis model as another essential element and further improve the reliability of the propeller diagnosis. Dropout in deep learning algorithms is capable of not only avoiding overfitting but also serving as a Bayesian approximation of a Gaussian process model over network parameters [23]. Given spectrogram images  $\mathbf{S}$  and identified propeller conditions  $\hat{Y}$ , the posterior over the parameters  $p(\Theta | \mathbf{S}, \hat{Y})$  is evaluated using Bayesian inference. Due to the difficulty of analytical evaluation of the posterior,  $p(\Theta | \mathbf{S}, \hat{Y})$  can be approximated with a dropout distribution  $q(\Theta)$  using variational inference [39]. In addition, the network training process is beneficial for learning  $q(\Theta)$  [40]. Therefore, using MCDS, the mean softmax

score of the propeller condition for a spectrogram image at test time is denoted as follows [41]:

$$s_c \approx \frac{1}{M} \sum_{m=1}^M \text{Softmax}(f^{\hat{\Theta}_m}(\mathbf{S})) \quad (5)$$

where  $M$  is the Monte Carlo sampling size,  $\hat{\Theta}_m$  indicates the diagnosis model parameters fitting to  $q(\Theta)$ , and  $\mathbf{S}$  is the spectrogram input. This mean softmax score encapsulates the aggregate softmax score derived from all dropout samples, with each sample contributing uniquely to this mean score. The inconsistency in softmax scores across individual samples actually reveals the lack of confidence in diagnosing propeller conditions. In addition, the entropy of the probability score vector (i.e.,  $H(S)$ ) is defined as the uncertainty of the diagnosis model and can be used to determine the lack of confidence in the diagnosis model in Ref. [41].  $H(S)$  is defined using the following equation in this study:

$$H(S) = - \sum_{c=1}^C s_c \log(s_c) \quad (6)$$

Note that the uncertainty level is identified based on some user-defined thresholds of the entropy  $H(S)$ . Additionally, besides the calculated entropy, the standard deviation of the softmax score also has the potential to be an indicator of the diagnostic model's uncertainty.

### 4.3 | Propeller diagnosis test

#### 4.3.1 | Diagnosis model training

We conducted experimental tests with the audio receiver placed randomly in proximity. The experimental propellers have three different damage levels, which are unbroken, slightly broken and seriously broken. Each propeller condition evolves its own dataset with 138 spectrogram samples. 70% (290) of the spectrogram images are used to constitute the training dataset of the neural network. The training dataset takes the most proportion of the spectrogram images and is used to learn a set of weights of a model for the diagnosis. Next, 15% (62) of the spectrogram images are used to constitute the validation dataset of the neural network, which prevents overfitting by tuning the parameters of the diagnosis model. Finally, the test dataset is composed of the remaining spectrogram samples (62 spectrogram images). The test data are treated as an unknown dataset for the model to observe the diagnosis ability of the well-trained model. We design a 2D convolutional network with three layers for the spectrogram-based diagnosis model. Besides, instead of solely converting the audio data to image-based spectrograms, we transform the audio data to sequence-based spectrums as introduced in prior studies [11, 12]. The allocation of training, validation, and test

datasets for spectrums mirrors that of the spectrograms. We develop a 1D CNN with three layers to process spectrums.

To evaluate the influence of proximity to the audio receiver, a series of experiments are conducted at various distances, that is, 2, 4, 6, and 8 m. For each distance setting, we conduct experimental tests to generate spectrogram samples corresponding to the propellers under three conditions. Each propeller condition generates a distinct dataset, including 140 spectrogram images. We partition these images, allocating 70% (294 spectrogram images) for training, 15% (63 spectrogram images) for validation, and the remaining samples (63 spectrogram images) for testing. Notably, rather than training individual networks for each distance, a single neural network is trained to comprehensively capture the diagnostic performance across all distances. Furthermore, we introduce white Gaussian noise into the collected audio data at different signal-to-noise ratios (SNRs) of 50, 30, and 10 dB to simulate audio data collected from real-world environments. Initially, an assessment of the accuracy in classifying the noise-added audio data without retraining the network is conducted. Subsequently, a retraining phase of the network is performed using the noise-added audio data, and the resulting classification accuracy is evaluated.

#### 4.3.2 | Test results

Figure 7 demonstrates the propeller diagnosis using spectrograms with uncertainty quantification at test time. The audio data generated from the propeller rotation are captured by a microphone and transferred to spectrograms as the input data of the CNN model. We obtain the computational time corresponding to different dropout sample sizes  $M$  and summarise the results in Table 2. The results show that augmenting the sample size has a notable effect in terms of prolonging the inference time. A large value of  $M$  (e.g.,  $M = 30$  or  $M = 40$ ) signifies the number of samples in quantifying the uncertainty of the diagnostic model, thereby enhancing the accuracy of uncertainty quantification. However, it is important to acknowledge that this enhancement is accompanied by an increment in computational time (e.g., 0.504 s or 0.746 s),

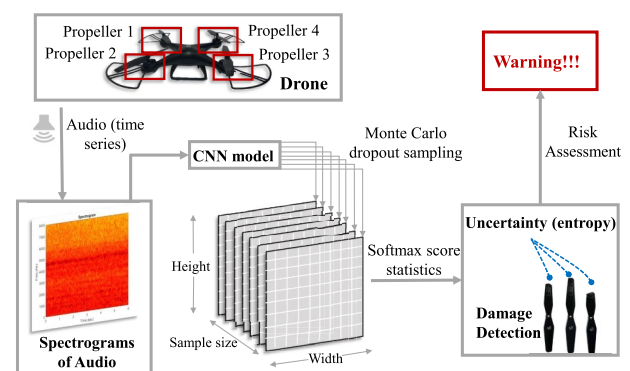


FIGURE 7 Propeller diagnosis with uncertainty quantification.

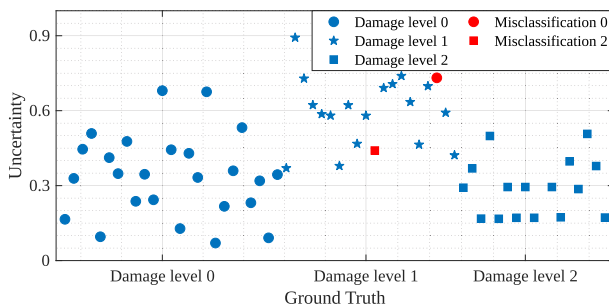
which potentially limits the efficiency of the online diagnosis. Consequently, the selection of an appropriate value for  $M$  becomes a trade-off problem. We select  $M = 20$  to balance these two factors, and the corresponding computational time of 0.257 s could be appropriately utilised in the online diagnosis. The damage level and uncertainty level are determined based on the softmax score statics from the sample bin.

Furthermore, Figure 8 illustrates the diagnosis results using the test dataset. The identification of each image for the propeller condition is labeled using different markers. The corresponding uncertainty is calculated using Equation (6). Particularly, high uncertainty means that the probability scores for each propeller condition tends to be even. In other words, for markers with high uncertainty values, the model shows less confidence in diagnosing propeller conditions. Overall, the flight with a damaged propeller can be detected with **96.78%** accuracy, surpassing the accuracy of 93.55% obtained when employing spectrum representations. The only two mistakes made by the diagnosis model are highlighted using red colors. Based on the observation of the modified figure, compared to the other two label categories, ‘damage level 1’ exhibits the highest average uncertainties. Meanwhile, it is noteworthy that ‘damage level 1’ is also the label most likely to be misclassified. We adopt a predetermined threshold of 0.7 to govern the generation of warnings. Note that even though the diagnosis model misclassifies a broken propeller to be an unbroken one, that is, the red dot, an early warning would still be triggered due to high uncertainty.

The drone with unbroken propellers would either hover or move horizontally within predefined regions. However, Figure 9 depicts the flight of the drone with a seriously broken propeller, where the drone is in unsteady flight and directly falls

**TABLE 2** Details of Monte Carlo dropout sampling (MCDS) computational time.

Classification with MCDS	Inference time (s)
$M = 10$	0.124
$M = 20$	0.257
$M = 30$	0.504
$M = 40$	0.746



**FIGURE 8** Diagnosis results: the blue markers indicate the diagnosed propeller conditions, the red markers stand for the mistakes made by the diagnosis model. The  $x$ -axis shows the ground truth of samples. Please note there is no misclassification for damage level 1.

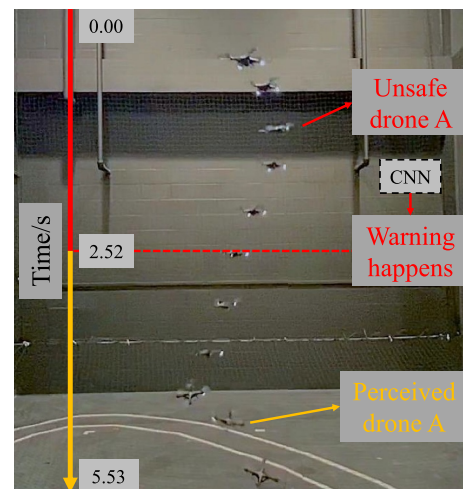
down. After 2 s of audio collection time, the whole diagnosis process requires 0.52 s to distinguish the unsafe drone based on the identified damage and uncertainty levels. Note that the execution time of the diagnostic process may slightly vary. A warning signal would be given to humans so that they could perceive the threatening drone before it crashes.

Table 3 illustrates the diagnostic accuracy based on the various distances between the audio receiver and the quadrotor as well as the different levels of SNR. Based on the obtained results, the optimal operating distance falls within the approximate range of 4–6 m. Distances shorter than this range tend to introduce wind noise, adversely affecting classification accuracy, while longer distances may marginally reduce accuracy due to a decrease in the audio signal magnitude. Meanwhile, environmental noises exert a substantial influence on classification accuracy. Specifically, when there is no network retraining, classification accuracy experiences a notable decline (i.e., from 94.35% to 49.17%) as the SNR changes from 50 dB to 10 dB. Fortunately, through the process of network retraining, it is possible to improve the classification accuracy, bringing it up from 49.17% to 94.05%.

## 5 | GENERALISED DIAGNOSTIC MODEL FOR DRONES

### 5.1 | Diagnostic model generalisation using transfer learning

The accurate and reliable CNN-based drone fault diagnosis exhibits the potential for industrial implementation. However, only one well-trained network diagnosis model probably fails to discriminate the broken propellers of different drones due to the drone's property diversity, for example, different drone frames, propeller sizes, and rotation speeds. A straightforward way to detect broken propellers for other drones would be by developing the CNN-based diagnosis model for each drone.



**FIGURE 9** The experimental test of the CNN-based diagnosis model. The experimental video is available via this [link](#).



**TABLE 3** Diagnostics accuracy comparison among cases with various distances between the audio receiver and the quadrotor, as well as the different levels of signal-to-noise ratio (SNR).

Distance from quadrotor (m)	Without noise (%)	SNR = 50 dB	SNR = 30 dB	SNR = 10 dB
		Retraining (no retraining) (%)	Retraining (no retraining) (%)	Retraining (no retraining) (%)
2	95.24	92.06 (93.10)	96.83 (74.76)	93.65 (49.52)
4	100.0	100.0 (100.0)	93.65 (81.67)	96.83 (54.05)
6	100.0	100.0 (95.48)	96.83 (80.48)	93.65 (56.43)
8	96.83	95.24 (88.81)	98.41 (75.95)	92.06 (36.67)
Overall	98.02	96.83 (94.35)	96.42 (78.21)	94.05 (49.17)

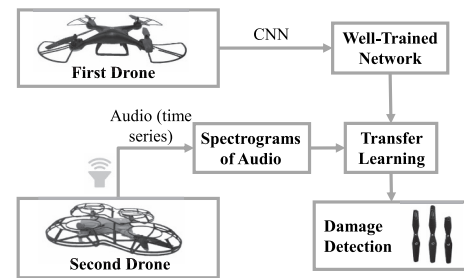
However, this approach is inefficient considering that it needs tremendous data collection and training time. Therefore, this paper applies the transfer learning technique [42] to transfer diagnosis knowledge between a well-trained drone model and a new drone model to generalise the network-based drone fault diagnosis.

Figure 10 illustrates the drone-based transfer learning framework. Firstly, the CNN employs sufficient spectrogram images based on the first drone as training data to obtain a well-trained diagnosis model. The audio-based spectrograms collected from the first drone and the layer weights of the diagnosis model used to extract valuable audio characteristics are defined as source domain  $D_S$ . The identification based on three different propeller damage levels is defined as diagnosis task  $T_S$  for the first drone. Next, transfer learning reserves most of the network layers to retain the knowledge of  $D_S$  and  $T_S$  and replaces the last few layers with some new layers to extract specific audio features of spectrograms generated from the second drone. Finally, a second drone-based diagnosis task  $T_T$  in a diagnosis target domain  $D_T$  can be improved based on the knowledge from the first drone's  $D_S$  and  $T_S$  by transfer learning, where  $D_S \neq D_T$ ,  $T_S \neq T_T$ . Note that the training process of transfer learning only takes a few spectrogram images from the second quadrotors as training data, and fine-tunes the layer weights.

## 5.2 | Experiment tests of generalised diagnostic model

As mentioned earlier, the accuracy of the CNN-based fault diagnosis model for the first quadrotor can achieve 96.78%. If this diagnosis model directly employs all the spectrogram images generated from the second quadrotor as the test dataset, the classification accuracy is 32.61%, which closely approximates the proportion of the propeller types. To generalise the fault diagnosis model, transfer learning takes the spectrogram images collected from the second quadrotor.

To see if the diagnosis model can be generalised using only a small dataset based on transfer learning, we take only five images based on each propeller damage condition to construct the training dataset (15 images). To validate the performance of

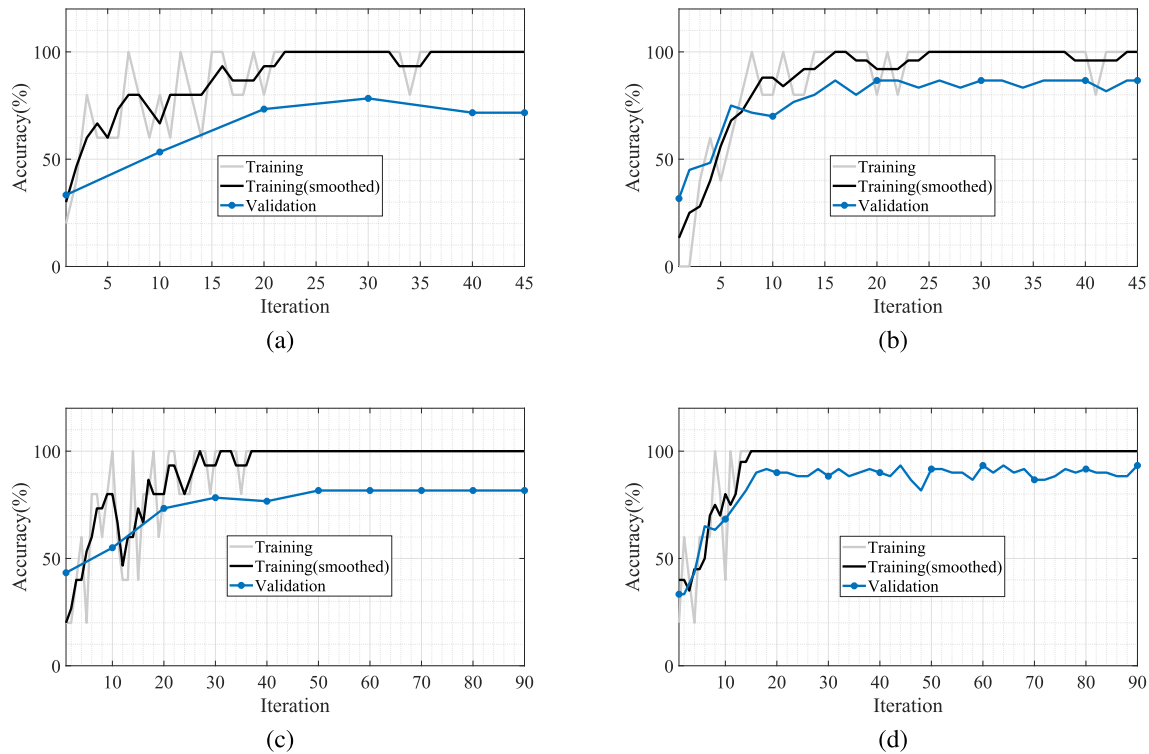


**FIGURE 10** The two drone-based transfer learning frameworks.

the generalised diagnosis model, the test dataset is composed of 339 images totally. In addition, the remaining 60 images constitute the validation dataset. Figure 11a,b is the transfer learning training processes without transfer learning and with transfer learning respectively, and the classification accuracies of the test data are 66.67% and **89.68%**, respectively.

Figure 12 indicates the transfer learning confusion matrix of the test dataset. The true and predicted classes both contain three labels: ‘unbroken’, ‘slightly broken’, and ‘seriously broken’. The confusion matrix provides the classification details. The true positive rate (TPR) and false negative rate (FNR) in the row indicate the classification result for each true class. To elaborate, consider the category ‘slightly broken’ as an example. There are a total of 113 observations that possess this true label (comprising 6 misclassified, 100 correctly classified, and 7 misclassified). Consequently, the TPR for this category is calculated to be 88.5%. On the other hand, the precision and false discovery rate (FDR) in the column indicate the classification result for each predicted class. For example, a total of 120 observations are classified as ‘slightly broken’ (including 6 misclassified, 100 correctly classified, and 14 misclassified). Hence, the precision for this predicted class stands at 83.3%.

To improve the fault detection performance, the number of spectrogram images from each propeller damage condition has been augmented from 5 to 10. The validation dataset maintains the same number of images. The test dataset has 324 spectrogram images. The user-defined configuration of the network hyperparameters is as follows: the mini-batch size is 4, the maximum number of epoches is 20, the initial learning rate



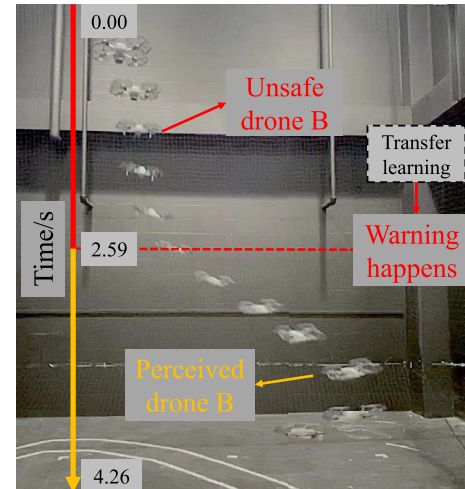
**FIGURE 11** The comparison of the training processes with and without transfer learning. (a) The training process using 15 training images without transfer learning. (b) The training process using 15 training images with transfer learning. (c) The training process using 30 training images without transfer learning. (d) The training process using 30 training images with transfer learning.

	Predicted Class			TPR	FNR
	unbroken	slightly broken	seriously broken		
True Class					
unbroken	106	6	1	93.8%	6.2%
slightly broken	6	100	7	88.5%	11.5%
seriously broken	1	14	98	86.7%	13.3%
Precision	93.8%	83.3%	92.5%		
FDR	6.2%	16.7%	7.5%		

**FIGURE 12** The network classification result: TPR indicates the true positive rate, FNR stands for the false negative rate, and FDR represents the false discovery rate.

is 0.01, the momentum is 0.9, and the L2 regularisation is  $1.0 \times 10^{-4}$ . Figure 11c,d demonstrate the training process without transfer learning and with transfer learning, respectively, and the classification results of the test data are 82.72% and 93.21%, respectively.

The detection results in Figure 11 indicate a few points. First, the detection capability of the diagnosis model would be retarded by the insufficient training data. The other point is that transfer learning allows the knowledge of the first drone-based diagnosis model to be utilised by the new diagnosis model such that the detection can be accurate, for example, 89.68% and 93.21%, even when using the same size



**FIGURE 13** The experimental test of transfer learning-based diagnosis model.

of the training data. In addition, the experimental test of the transfer learning-based diagnosis model is illustrated in Figure 13. After 2.59 s of collection and computational time, the diagnosis model modified by transfer learning identifies the second unsafe drone and sends a warning signal such that human workers can perceive the second threatening drone before it crashes.

## 6 | HYPERPARAMETERS TUNING USING BAYESIAN OPTIMISATION

The parameters of network layers are updated properly during the training cross-validation process so that the generalised fault diagnosis model has reasonable detection performance. However, the network hyperparameters such as the mini-batch size and the learning rate, as another crucial factor of the network training, are pre-defined and not trainable. Different hyperparameter selections have an impact on parameter updating during the training phase, causing the broken propeller detection to be questionable. Meanwhile, the evaluation of each hyperparameter selection takes the whole training process, which results in multiple training computations. Therefore, to obtain a robust diagnosis model and avoid expensive evaluations, this paper uses Bayesian optimisation to select the hyperparameters for transfer learning.

### 6.1 | Objective function setup

Bayesian optimisation is a useful optimisation approach for finding the global optimum for the objective function, which is expensive to be evaluated and without explicit form. In this paper, the Bayesian optimisation treats the validation error according to each hyperparameter selection as the objective function  $f(x)$ , where  $x$ , is a vector of selected hyperparameters, including the size of minibatch, the maximum epoch, the initial learning rate, the momentum and the L2 regularisation. The objective function is an essential component of the hyperparameter selection, and the framework of the objective function setup is illustrated in Figure 14. The customised layers are used to learn some specific features of the spectrogram images from the second quadrotor. Only training data (15 images) and validation data (60 images) have been involved in the objective function build-up. The validation error after each training constructs the objective function of the Bayesian optimisation. The  $\beta\{\hat{Y} = Y\}$  means the number of validation samples, for which the ground truth label is equal to the classified label.  $\beta\{Y\}$  means the total number of validation samples.

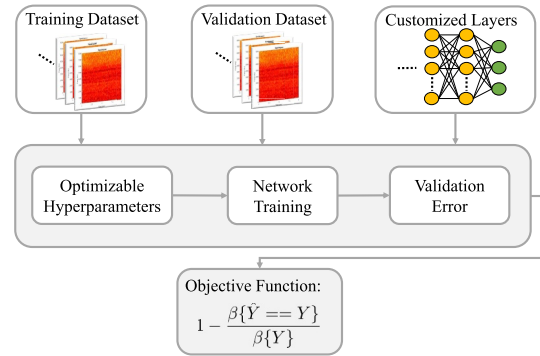


FIGURE 14 The objective function of Bayesian optimisation setup framework.

### 6.2 | Optimal diagnosis model hyperparameter selection

All evaluations of the validation error contribute to a Gaussian process model. The Gaussian process prior is denoted as follows:

$$f(x) \sim \mathcal{N}\left(\mu(x), \Sigma(x, x)\right) \quad (7)$$

where  $N$  here means normal distribution,  $\mu$  and  $\Sigma$  indicate a mean function and a covariance function, which correspond to every hyperparameter selection. A series of hyperparameter selections  $[x_1, \dots, x_n]$  is denoted as  $X$ . Based on these observed hyperparameter selections and the validation errors  $f(X)$ , the objective function value of next hyperparameter selection  $x^*$  is obtained with the following posterior process (see details in Refs. [43, 44]):

$$f(x^*) | f(X) \sim \mathcal{N}(\mu_*(x^*), \sigma_*^2(x^*)) \quad (8)$$

$$\mu_*(x^*) = \sum (X, x^*)^T \sum (X, X)^{-1} (f(X) - \mu(X)) + \mu(x^*) \quad (9)$$

#### Algorithm 1 Bayesian optimisation with neural network.

- Assign objective function with Gaussian Process.
  - Set initial evaluation hyperparameters  $x_1$  randomly based on the pre-defined range.
  - Apply the initial hyperparameters to transfer learning training process.
  - Set index of training processes  $n=1$  and the total number of complete training processes  $M$  ( $M=30$  in this study).
- while**  $n \leq M$  **do**
1. Update the posterior mean  $\mu_*(x^*)$  and posterior variance  $\sigma_*^2(x^*)$  according to Eq. (9) and Eq. (10).
  2. Obtain the expected improvement  $EI(x^*)$  in Eq. (11).
  3. Generate normal distributed  $x_c$  as sample point candidates and use Nelder-Mead algorithm to search  $x_{n+1}$  according to Eq. (12).
  4. Use selected hyperparameters  $x_{n+1}$  to evaluate the validation error  $f(x_{n+1})$  by training neural network.
  5. Augment the observed dataset by adding  $(x_{n+1}, f(x_{n+1}))$  and update Gaussian Process.
  6. Increment  $n$ .
- end**
- Return the optimal hyperparameters  $x_{opt}$  according to the minimum objective function value.

$$\sigma_*^2(x^*) = \sum(x^*, x^*) - \sum(X, x^*)^T \sum(X, X)^{-1} \sum(X, x^*) \quad (10)$$

where  $\mu(x^*)$  is the prior mean of next hyperparameter selection  $x^*$ , and the  $\mu_*(x^*)$  and  $\sigma_*^2(x^*)$  are the posterior mean and posterior variance, respectively. This paper employs the expected improvement acquisition function, which maximises the difference between the observed minimum validation error  $f$  and the posterior validation error to select the next hyperparameter configuration to be evaluated. The acquisition function is denoted as follows (see details in Refs. [43, 45]):

$$\begin{aligned} \text{EI}(x^*) &= E[\max(0, f' - f(x^*)) | X, f(X)] \\ &= \int_{-\infty}^{f'} (f' - f(x^*)) N(f(x^*); \mu_*, \sigma_*^2) df(x^*) \quad (11) \\ &= \sigma_*(x^*) \varphi\left(\frac{\Delta_*(x^*)}{\sigma_*(x^*)}\right) + \Delta_*(x^*) \Phi\left(\frac{\Delta_*(x^*)}{\sigma_*(x^*)}\right) \end{aligned}$$

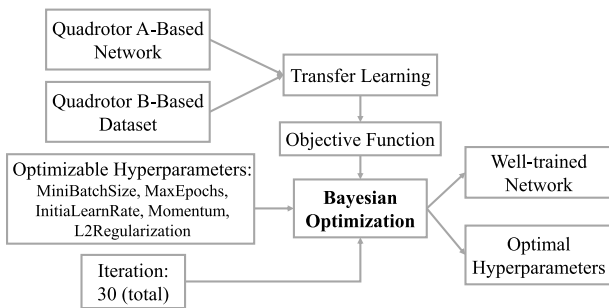
where  $\text{EI}(x^*)$  indicates the expected improvement, which is a positive value conditioned on observed validation errors and  $\Delta_*(x^*)$  is  $f' - \mu_*(x^*)$ . The  $\varphi$  and  $\Phi$  are the probability density functions of normal distribution and cumulative distribution.

To obtain the optimal hyperparameter selection  $x_{n+1}$ , a series of sample points  $x_c$  are generated as candidates according to normal distribution. By using the Nelder–Mead algorithm, the sample point with the maximum value of expected improvement is determined by local searching [46]. The equation is denoted as follows:

$$x_{n+1} = \text{argmax}(\text{EI}(x_c)) \quad (12)$$

### 6.3 | Validation with experimental tests

The framework of modified transfer learning including Bayesian optimisation is demonstrated in Figure 15. A well-trained network developed from drone A and the spectrogram images dataset from drone B are the basic components of transfer learning. In order to discover the optimal



**FIGURE 15** The architecture of transfer learning using Bayesian optimisation.

hyperparameter selection  $x_{\text{opt}}$ , a certain value range of each optimisable hyperparameter is pre-defined (e.g, the mini-batch size ranges from 2 to 8, the maximum epoch ranges from 10 to 20, the initial learning rate ranges from  $1.0 \times 10^{-5}$  to  $1.0 \times 10^{-2}$ , the momentum is from 0.8 to 0.98, and the L2 regularisation is from  $1.0 \times 10^{-8}$  to  $1.0 \times 10^{-2}$ ) and the total number of training processes is 30. The transfer learning provides the validation error for each complete training process as the objective function in Bayesian optimisation. A new fault diagnosis model is chosen with corresponding hyperparameters based on the Bayesian optimisation result. The algorithm of neural network training with Bayesian optimisation is presented in Algorithm 1.

After 30 complete training processes, the optimal configuration of network hyperparameters is that the mini-batch size is 8, the maximum epoch is 11, the initial learning rate is 0.0094, the momentum is 0.8390, and the L2 regularisation is  $4.228 \times 10^{-5}$ . The detection accuracy result of the corresponding diagnosis model with 339 test spectrogram images is shown in Figure 16. In the second row of the classification result, only one test image generated from the drone with a slightly broken propeller is classified as ‘a severely broken propeller’ image inaccurately. Compared with the detection accuracy (89.68%) without Bayesian optimization, the detection accuracy achieved is around **99.70%**.

## 7 | CONCLUSIONS AND FUTURE WORK

This paper has presented a new and comprehensive drone fault diagnosis method that can provide early warning signs for risky flights. One advantage of this method is that it only requires audio data collected from the drone flight in real-time. The features of audio data are first represented by a spectrogram and then sent to a CNN-based diagnosis model that can identify the propeller damage level and the model uncertainty level. This diagnosis model is further improved by leveraging transfer learning techniques such that it only requires a small amount of training data. Moreover, the hyperparameters of the diagnosis model are optimised to speed the convergence rate, enhance the robustness of the fault diagnosis model, and

		Predicted Class			TPR	FNR
		unbroken	slightly broken	seriously broken		
True Class	unbroken	113			100%	
	slightly broken		112	1	99.1%	0.9%
	seriously broken			113	100%	
Precision		100%	100%	99.1%		
FDR				0.9%		

**FIGURE 16** The network classification result based on Bayesian optimisation: TPR indicates the true positive rate, FNR stands for the false negative rate, and FDR represents the false discovery rate.



reduce some unnecessary computational workloads. All the data sets used in this study are collected from real flight tests, and the diagnosis model and the early warning system have been validated on real flight tests. Accuracy of around 99.70% can be achieved as discussed in this paper.

Although the diagnostic accuracy using the proposed approach is promising, it is essential to acknowledge two limitations of this approach. Firstly, there is a need to improve the utilisation of the quantified uncertainty associated with the diagnostic model. This is beneficial for eliminating unnecessary early warnings or alarms and enhancing the reliability of the warning system. The second limitation is the lack of efficient ways to make the proposed approach to be more robust and generalised, particularly when facing diverse real-world environments. Future studies will be dedicated to exploring adaptive methodologies of determining uncertainty for early warning generation. Furthermore, we intend to integrate unsupervised learning techniques into our approach such that the practical applicability can be enhanced. For example, we could employ Generative Adversarial Networks (GANs), a type of widely used unsupervised learning models, for the development of a propeller diagnostic model. The Generator of a GAN exhibits the capacity to create synthetic spectrogram images, thereby substantially alleviating the burdens associated with data collection and labelling. Meanwhile, the Discriminator of a GAN excels at learning critical features for distinguishing between real and synthetic spectrogram images. The learned feature extraction skills can be utilised to further improve the quality of the propeller diagnosis.

## ACKNOWLEDGEMENTS

This material is based upon the work that is partially supported by the National Science Foundation-USA under Grant No. 2046481. The findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

## DATA AVAILABILITY STATEMENT

Data is available on request from the authors.

## ORCID

Chang Liu  <https://orcid.org/0000-0003-1953-3731>

Minghui Zheng  <https://orcid.org/0000-0002-1460-3246>

## REFERENCES

- Barmounakis, E.N., Vlahogianni, E.I., Golias, J.C.: Unmanned aerial aircraft systems for transportation engineering: current practice and future challenges. *Int. J. Transport. Sci. Technol.* 5(3), 111–122 (2016). <https://doi.org/10.1016/j.ijtst.2017.02.001>
- Sajedi, S.O., Liang, X.: A convolutional cost-sensitive crack localization algorithm for automated and reliable RC bridge inspection. In: *Risk-Based Bridge Engineering: Proceedings of the 10th New York City Bridge Conference, New York, 26–27 August 2019*. CRC Press, p. 229 (2019)
- Liang, X.: Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization. *Comput. Aided Civ. Infrastruct. Eng.* 34(5), 415–430 (2019). <https://doi.org/10.1111/micc.12425>
- Liang, X., Zheng, M., Zhang, F.: A scalable model-based learning algorithm with application to UAVs. *IEEE Control Syst. Lett.* 2(4), 839–844 (2018). <https://doi.org/10.1109/lcsys.2018.2849576>
- Zheng, M., Chen, Z., Liang, X.: A preliminary study on a physical model oriented learning algorithm with application to UAVs. In: *ASME 2019 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers Digital Collection (2019)
- Adlakha, R., et al.: Integration of acoustic compliance and noise mitigation in path planning for drones in human–robot collaborative environments. *J. Vib. Control* 29, 4757 (2022)
- Avram, R.C., Zhang, X., Campbell, J.: *Sensor Fault Diagnosis in Quadrotors Using Nonlinear Adaptive Estimators*. Wright State University Dayton United States (2014)
- Al Younes, Y., et al.: Sensor fault diagnosis and fault tolerant control using intelligent-output-estimator applied on quadrotor UAV. In: *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1117–1123. IEEE (2016)
- Hansen, S., Blanke, M.: Diagnosis of airspeed measurement faults for unmanned aerial vehicles. *IEEE Trans. Aerospace Electron. Syst.* 50(1), 224–239 (2014)
- Ghalamchi, B., Mueller, M.: Vibration-based propeller fault diagnosis for multicopters. In: *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1041–1047. IEEE (2018)
- de Jesus Rangel-Magdaleno, J., et al.: Detection of unbalanced blade on UAV by means of audio signal. In: *2018 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, pp. 1–5. IEEE (2018)
- Iannace, G., Ciaburro, G., Trematerra, A.: Fault diagnosis for UAV blades using artificial neural network. *Robotics* 8(3), 59 (2019). <https://doi.org/10.3390/robotics8030059>
- Cabrera-Ponce, A.A., Martinez-Carranza, J., Rascon, C.: Detection of nearby UAVs using CNN and spectrograms. In: *International Micro Air Vehicle Conference and Competition (IMAV)*, Madrid (2019)
- Harmanny, R.I., de Wit, J.J., Premel-Cabic, G.: Radar micro-Doppler mini-UAV classification using spectrograms and cepstrograms. *Int. J. Microw. Wireless Technol.* 7(3–4), 469–477 (2015). <https://doi.org/10.1017/s1759078715001002>
- Zhang, W., et al.: A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mech. Syst. Signal Process.* 100, 439–453 (2018). <https://doi.org/10.1016/j.ymssp.2017.06.022>
- Chen, Z., et al.: Deep neural networks-based rolling bearing fault diagnosis. *Microelectron. Reliab.* 75, 327–333 (2017). <https://doi.org/10.1016/j.microrel.2017.03.006>
- De Bruin, T., Verbert, K., Babuška, R.: Railway track circuit fault diagnosis using recurrent neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* 28(3), 523–533 (2016). <https://doi.org/10.1109/tnnls.2016.2551940>
- Pereira, S., et al.: Brain tumor segmentation using convolutional neural networks in MRI images. *IEEE Trans. Med. Imaging* 35(5), 1240–1251 (2016). <https://doi.org/10.1109/tmi.2016.2538465>
- Levi, G., Hassner, T.: Age and gender classification using convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 34–42 (2015)
- Liu, Z., et al.: Infrared image combined with CNN based fault diagnosis for rotating machinery. In: *2017 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*, pp. 137–142. IEEE (2017)
- Guo, D., et al.: A hybrid feature model and deep learning based fault diagnosis for unmanned aerial vehicle sensors. *Neurocomputing* 319, 155–163 (2018). <https://doi.org/10.1016/j.neucom.2018.08.046>
- Sajedi, S., et al.: Uncertainty-assisted image-processing for human-robot close collaboration. *IEEE Rob. Autom. Lett.* 7(2), 4236–4243 (2022). <https://doi.org/10.1109/lra.2022.3150487>

23. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In: International Conference on Machine Learning, pp. 1050–1059. PMLR (2016)
24. Zhong, S.S., Fu, S., Lin, L.: A novel gas turbine fault diagnosis method based on transfer learning with CNN. *Measurement* 137, 435–453 (2019). <https://doi.org/10.1016/j.measurement.2019.01.022>
25. Wen, L., Gao, L., Li, X.: A new deep transfer learning based on sparse auto-encoder for fault diagnosis. *IEEE Trans. Syst. Man Cybern. Syst.* 49(1), 136–144 (2017). <https://doi.org/10.1109/tsmc.2017.2754287>
26. Zhang, R., et al.: Transfer learning with neural networks for bearing fault diagnosis in changing working conditions. *IEEE Access* 5, 14347–14357 (2017). <https://doi.org/10.1109/access.2017.2720965>
27. Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian optimization of machine learning algorithms. In: *Advances in Neural Information Processing Systems*, pp. 2951–2959 (2012)
28. Shahriari, B., et al.: Taking the human out of the loop: a review of Bayesian optimization. *Proc. IEEE* 104(1), 148–175 (2015). <https://doi.org/10.1109/jproc.2015.2494218>
29. Salama, A., Hassani, A.E., Fahmy, A.: Sheep identification using a hybrid deep learning and Bayesian optimization approach. *IEEE Access* 7, 31681–31687 (2019). <https://doi.org/10.1109/access.2019.2902724>
30. Wu, J., et al.: Hyperparameter optimization for machine learning models based on Bayesian optimization. *J. Electron. Sci. Technol.* 17(1), 26–40 (2019)
31. Glowacz, A., et al.: Early fault diagnosis of bearing and stator faults of the single-phase induction motor using acoustic signals. *Measurement* 113, 1–9 (2018). <https://doi.org/10.1016/j.measurement.2017.08.036>
32. Kemalkar, A.K., Bairagi, V.K.: Engine fault diagnosis using sound analysis. In: 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), pp. 943–946. IEEE (2016)
33. Liu, W., Chen, Z., Zheng, M.: An audio-based fault diagnosis method for quadrotors using convolutional neural network and transfer learning. In: 2020 American Control Conference (ACC), pp. 1367–1372. IEEE (2020)
34. Hory, C., Martin, N.: Mixture densities formulation of a spectrogram segmentation task. In: 2002 11th European Signal Processing Conference, pp. 1–4. IEEE (2002)
35. Sajedi, S.O., Liang, X.: Intensity-based feature selection for near real-time damage diagnosis of building structures. arXiv preprint arXiv:191011240 (2019)
36. Sajedi, S.O., Liang, X.: Vibration-based semantic damage segmentation for large-scale structural health monitoring. *Comput.-Aided Civil Infrastruct. Eng.* 35, 579 (2020)
37. Liang, X., Mosalam, K., Muin, S.: Simulation-based data-driven damage detection for highway bridge systems. In: Proc. 11th Nat. Conf. Earthquake Eng. (NCEE) (2018)
38. Sajedi, S.O., Liang, X.: A data-driven framework for near real-time and robust damage diagnosis of building structures. *Struct. Control Health Monit.* 27(3), e2488 (2020). <https://doi.org/10.1002/stc.2488>
39. Graves, A.: Practical variational inference for neural networks. *Adv. Neural Inf. Process. Syst.* 24, 2348–2356 (2011)
40. Gal, Y., Ghahramani, Z.: Bayesian convolutional neural networks with Bernoulli approximate variational inference. arXiv preprint arXiv:150602158 (2015)
41. Kendall, A., Gal, Y.: What uncertainties do we need in Bayesian deep learning for computer vision? *Adv. Neural Inf. Process. Syst.* 30 (2017)
42. Pan, S.J., Yang, Q.: A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* 22(10), 1345–1359 (2009). <https://doi.org/10.1109/tkde.2009.191>
43. Frazier, P.I.: A tutorial on Bayesian optimization. arXiv preprint arXiv:180702811 (2018)
44. Rasmussen, C.E.: Gaussian processes in machine learning. In: *Summer School on Machine Learning*, pp. 63–71. Springer (2003)
45. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. *J. Global Optim.* 13(4), 455–492 (1998). <https://doi.org/10.1023/a:1008306431147>
46. Lagarias, J.C., et al.: Convergence properties of the Nelder–Mead simplex method in low dimensions. *SIAM J. Optim.* 9(1), 112–147 (1998). <https://doi.org/10.1137/s1052623496303470>

**How to cite this article:** Liu, W., et al.: An audio-based risky flight detection framework for quadrotors. *IET Cyber-Syst. Robot.* e12105 (2024). <https://doi.org/10.1049/csy2.12105>