

RESEARCH ARTICLE

A Stethoscope for Drones: Transformers-Based Methods for UAVs Acoustic Anomaly Detection

OR HAIM ANIDJAR^{1,2,3,4}, ALON BARAK^{1,2}, BOAZ BEN-MOSHE^{1,2,3},
EYAL HAGAI^{1,2}, AND SAHAR TUVYAHU^{1,2}

¹School of Computer Science, Ariel University, Ariel 4070000, Israel

²Kinematics and Computational Geometry Laboratory (K&CG), Ariel University, Ariel 4070000, Israel

³Ariel Cyber Innovation Center (ACIC), Ariel University, Ariel 4070000, Israel

⁴Data Science and Artificial Intelligence Research Center, Ariel University, Ariel 4070000, Israel


Corresponding author: Or Haim Anidjar (orhaim@ariel.ac.il)

ABSTRACT Unmanned Aerial Vehicles and the increasing variety of their applications are raising in popularity. The growing number of UAVs, emphasizes the significance of drones' reliability and robustness. Thus, there is a need for an efficient self-observing sensing mechanism to detect real-time anomalies in drone behavior. Previous works suggested prediction models from control theory, yet, they are complex by nature and hard to implement, while Deep Learning solutions are of great utility. In this paper, we propose a real-time framework to detect anomalies in drones by analyzing the sound emitted from them. For this purpose, we construct a hybrid Deep Learning based Transformer and a Convolutional Neural Network inspired by the well-known VGG architecture. Our approach is examined over a dataset that is collected from a single microphone set located on a micro drone in real-time. Our approach achieves an F1-score of 88.4% in detecting anomalies and outperforms the VGG-16 architecture. Moreover, the framework presented in this paper reduces the number of parameters of the well-known VGG-16 from 138M, into a shrunk version with 3.6M parameters only. Additionally, our real-time approach, results in a smaller number of parameters in the neural network, and yet yields high accuracy in anomaly detection in drones with an average inference time of 0.2 seconds per second. Moreover, with an earphone that weighs less than 100 grams on top of the UAV, our method is shown to be beneficial, even in extreme conditions such as a micro-size dataset that is composed of three hours of flight recordings. The presented self-observing method can be implemented by simply adding a microphone to drones and transmitting the captured audio for analysis to the remote control or performing it onboard the drone using a dedicated microcontroller.

INDEX TERMS Acoustics, UAVs, anomaly detection, CNN, deep learning, transformers, Wav2Vec2.

I. INTRODUCTION

Unmanned Air Vehicles (UAVs) are nowadays used in many industries, such as the food industry [1], retail [2], health-care [3], etc. In addition, They can be used in cinematography to follow stunt doubles during outdoor filming [4], and can even help with agriculture by doing redundant tasks like seeding, planting, or spraying [5]. Moreover, when combined with Artificial Intelligence, UAVs get incredible abilities like 3D modeling in the aftermath of an area where a disaster occurred for analysis, and even doing tasks mentioned above

The associate editor coordinating the review of this manuscript and approving it for publication was Guillermo Valencia-Palomo .

autonomously [6]. With all of these UAV features, UAVs are fragile and can be damaged or suffer from malfunctions, especially when they are autonomous [7]. While executing such tasks, UAVs can be damaged by insects or birds, which can cause damage to the UAV's blades or rotors. This can ruin the UAV's stabilization and render it incapable of flying straight. The problem may go unnoticed and the drone acts as if it is following properly on the predefined path, while in reality, it is diverging, causing the drone to miss the crops. As such, these kinds of anomalies should be detected in real-time by the UAV and should be reported immediately in order to prevent long-term problems preemptively [8]. There are cases where anomalies can be detected from the software



FIGURE 1. A standard BLE (Bluetooth Low Energy) earphone is used as a “stethoscope” for sensing the drone’s “well-being”. The middle (main) image shows a Tello micro drone (sub 100 grams) with an earphone located a few centimeters above it (see also the lower left image). A minor anomaly in the propeller “tip” is shown in the upper left image. Another (more visible) anomaly is demonstrated by the blue propeller (which has slightly different parameters than the black ones). On the right side - two examples of the experiments are presented in outdoor scenarios; the upper image shows a relatively high altitude flight (about 8 meters above the ground, to avoid “ground effects”) and the lower left image shows the drone flying in a low altitude (about 1 meter above the ground).

by monitoring the sensors and moving parts’ actions and observations against certain thresholds, but the real world is much more complex and external influences are much harder to detect. In addition, UAV’s blades can be damaged [9] by an unexpected object hitting the UAV, or the wind that can push it into a sturdy object. So the question is, how can we detect these anomalies in real-time? One can note in Figure 1 the overall suggested solution - A Bluetooth earphone is used as a “stethoscope” sensor for the drone.

Most of the UAVs today ‘know’ how to stabilize themselves, but the situations mentioned above disrupt this mechanism. When attempting to modify the UAV’s blades, there is one thing that can be easily noticed, which is the noise that the UAV makes is different from its normal state. When a UAV gets hit, it immediately tries to re-stabilize itself, and this re-stabilization causes the rotors to spin unevenly, some faster than the others, to get the UAV back to its normal state. This process produces a sound that is different from the usual sound emitted from the UAV in its normal state. Moreover, a damaged [10] blade also emits a different sound than when the UAV is in a normal state [11]. The method proposed in this work for detecting the anomalies in the emitted sound from the UAV, uses a lightweight microphone mounted on the UAV, that can be connected to an external computer through Bluetooth or any other wireless connection. The audio stream is then passed to the external computer on which the proposed algorithm is executed. The algorithm uses the power of Deep Learning (DL) [12], [13] [14] to classify sound clips into anomalous or regular ones.

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans, which is learning by example. This technique can be used for many different tasks including prediction of future events [15], classification of data to groups [16], generation of new data [17], Anomaly Detection (AD) [18], [19] and more. A Convolutional Neural Network (CNN) is a sub-class of

DL architectures, most commonly applied to analyze visual imagery [20]. CNNs utilize the convolution operation, using kernels or filters that slide along input features and provide feature maps. CNNs can also be used for audio analysis [21], by turning audio into a 2D representation called a ‘spectrogram’. Similar to an image, the data can be processed by a CNN, as a spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time.

The use of DL for AD is not a new thing [22] since DL is useful when working with data that has patterns. DL models are able to learn patterns from the data that was given to them as an example, which makes DL good at AD. An anomaly can be defined as the occurrence of an unusual event or a deviation from a common rule. Using DL, it is possible to learn to classify the activity of a system over time as anomalous or regular. AD in UAVs [8] is no different; the goal is to identify unusual activities or patterns, by monitoring a stream of sensor-based information coming from the UAV. A DL model can identify if the current input is anomalous or not by learning from examples [23]. The use of DL for AD in UAVs has many advantages [24], where one of them is the ease of use and implementation. The first methods proposed for AD in UAVs were based on models from control theory [25], which require a good understanding of linear and non-linear systems, and knowing how to do the required math to implement these methods, while DL only needs sufficient data. Moreover, today there are many frameworks and libraries that make it quick and easy to implement and train DL models.

In this paper we use a Transformer-based model [26], [27]. A Transformer is a DL model that adopts the mechanism of self-attention [28], differently weighting the significance of each part of the input data. Transformers are also used for AD in many different fields such as, aerial videos from a UAV’s [29], system logs [30] and brain-scan images [31]. The self-attention mechanism of Transformers is very useful for AD, making it easier for DL models to recognize irregular activities in the input. The Transformer used in this paper takes an array that represents the sound frequencies of a second-long recording and outputs a matrix on which the self-attention mechanism is applied, then the CNN architecture processes it and outputs the probability that an anomaly has occurred. That is, we use two models from the Wav2Vec2 [26] group of transformers, which are the Wav2Vec2-Base and Wav2Vec2-ASR-960h, and compare their performance on the task of AD in UAVs. The Wav2Vec2-Base is a transformer-based model created for speech recognition and has been pre-trained on 960 hours of unlabeled raw speech data for speech recognition tasks. Wav2Vec2-ASR-960H is another transformer-based model, that has been trained and fine-tuned to identify English letters in raw sound. Generally, we denote our framework by **Wav2BC+** which stands for the exploitation of the **Wav2Vec2-Base** and the VGG-based CNN. Wav2Vec2 has been shown to be highly effective with relatively low training data [32], [33]. This model takes raw audio as input and outputs an image-like representation of the input data, which can then be passed to a classifier

TABLE 1. List of abbreviations.

Abbreviation	Meaning
AD	Anomaly Detection
ASR	Automatic Speech Recognition
BCE	Binary Cross Entropy
CNN	Convolutional Neural Network
DL	Deep Learning
FTC	Fault-Tolerant Control
KNN	K Nearest Neighbors
LSTM	Long Short Term Memory
NB	Naive Bayes
NN	Neural Network
OOD	Out Of Distribution
RNN	Recurrent Neural Networks
SSL	Self Supervised Learning
UAV	Unmanned Aerial Vehicle

that can differentiate between an anomalous and regular sound.

Finally, The contributions of this paper are highlighted as follows in Section I-A;

A. OUR CONTRIBUTION

- 1) Development process of a compressed version of the well-known VGG-16 framework that is extremely smaller in terms of number of parameters in the neural network, that is capable of yielding high accuracy in anomaly detection in UAVs.
- 2) Our real-time approach achieves State-of-the-art performance compared to two baseline approaches, and high accuracy at detecting anomalies in non-ideal environments, which results in a working implementation of a system for real-time anomaly detection in UAV's. Namely, we outperform the traditional VGG-based CNN architecture by introducing a Transformer as feature extraction and acoustics embedding.
- 3) Quick training process over only 3 hours of recorded data, which leads to a fast convergence due to the use of Transfer Learning, from an acoustic pre-trained model. That is, we tackle a problem in acoustics using technologies used for speech recognition.
- 4) Exploitation of an earphone and minimal hardware so that it can be used on any size UAV without affecting its functionality.

The remainder of this paper is structured as follows: Section II surveys related work on anomaly detection in UAV's, as well as DL approaches; Section III describes the data collection procedure as well as the organization and preparation for training the system; Section IV describes the proposed method in detail; Section V presents the results between the different approaches compared in this work; Finally, Section VI Summarizes this paper. For ease of reading, Table 1 provides a list of abbreviations that are commonly used in this paper.

II. RELATED WORK

The topic of AD in UAVs has been covered by quite a few works [34], [35] [36]. One of the first categories of

approaches to this diagnostic problem utilizes model-based fault diagnosis with sophisticated [37] methods to evaluate model residuals and conclude on the fault's occurrence. In [25], the method is based on a nonlinear observer [38], which is an extension of linear observer [39] design techniques using transformations related to linear observability matrices. The work in [40] later boosts the method above by adding an adaptive observer, making it more dynamic, and then in [41], the method before is implemented for real-time application on multirotor UAV's. However, the studies above only address the consequence of the rotor's impairment, since the analyzed type of anomaly is simulated Loss of Effectiveness [25] in thrust generation. Some other works follow the same approach with various methods of model-based fault estimation algorithms and following control strategies such as sliding mode control [42], Model Predictive Control [43], and Kalman filters [44].

The challenge of AD is often best solved using data-driven fault detection methods [29], [45], [46], [47]. They are based on statistical modeling and classification algorithms that use sensor-based data as input and output the probability of the occurrence of an anomaly. In [48], the proposed method included a hybrid Recurrent Neural Network [49] (RNN) with Long Short Term Memory (LSTM) [48] and CNN [20] architecture and reached approximately 92% accuracy in detecting actuator faults, using the state information from the UAV, like pitch, roll, pitch rate, roll rate, yaw rate and the input commands sent to the motors as input to the hybrid network. However, the dataset that was used in that work was recorded in ideal lab environments in a predefined setup platform. Most of these works are focusing on two types of sensor-based information, vibrations, and sound. In addition, the work in [48] copes with faults in UAVs, however, it excepts from the scope of this work for one main reason, which is the usage of LSTM; [48] presents an LSTM-based architecture, on top of a CNN, which means that the required memory amount is way too big for tiny UAVs, and real-time applications as well, which are the main interest of this scope. As a result, we train a small architecture that receives as input acoustic segments of size 1-second, and any greater segment length might result in a failure, in the case of tiny UAVs. That is, whenever the acoustic signals get longer (up to an anomaly), LSTMs are hard to converge into a detection. The reason is that the probability of preserving the vocal context from an acoustic segment that is extremely far from a segment that is currently being processed diminishes exponentially with the distance from it. That is, whenever 'normal' i.e. acoustic segments are long, the model might forget the content of positions that are distant in the acoustic segment. Another problem with LSTMs, is their parallelization inability of the acoustic segments processing, since they are required to be processed sequentially (frame by frame). In conclusion, LSTMs based methods are suffering from the following problems:

- Sequential computation restricts parallel data processing.

- LSTMs have no strict modeling ability of short and long-range dependencies.
- The distance between the frames' position in an acoustic signal is linear.

The existence of AI in low memory devices such as UAVs and drones is not new; that is, in [50] for instance, was introduced a survey of methodologies that combines deep learning and data science algorithms (e.g., statistics, linear regression, Bayesian methods). Another review was introduced in [51] and discusses new and emerging forms of data and technologies which seems to be a new field for future developments on AI, as well as in [52] that presented a method for the conceptualization of healthcare system that is supported by autonomous AI devices (such as drones or UAVs) that can use edge health devices with real-time data. As the AI field progresses more and more from a complex architecture standpoint, in this paper we use a Transformer-based architecture (explained thoroughly next in this section), in order to avoid the main LSTM three drawbacks. A handful of works [53], [54] show how the use of vibrations as the main source of information helped achieve highly accurate detections, by extracting the vibration data using sensors and then use this data as input to a model that then classifies it as anomalous or normal. An instance of a work [55] that used this approach, achieved more than 94% accuracy in detecting faults in UAV components, by using vibration data as input to a fuzzy ART neural network model [56] that then outputs the probability of the anomaly has occurred.

A few works describe fault detection in UAVs by analyzing sound [57], [58]. The work in [59] used sound as a source of input data to a Feed Forward Neural Network model that outputs the probability that one of the blades is partially or fully broken. This method achieved 98% accuracy in detecting broken or partially broken blades. Those results show great potential in using sound emitted from the UAV as the source of data for AD with neural networks. However, the experiments were performed with a stationary, ground-fixed UAV and an external high-class microphone, which is ideal. Moreover, their training methods were based on the assumption that an imbalance in the blade is equivalent to a partial loss of the blade. In another work, [60], a similar neural-based algorithm with physically impaired rotors and data collected in a real flight scenario resulted in 92% accuracy at detecting broken blades, although they can only detect whether a blade is broken or not. As such, the work in [57] takes into account a wider range of fault classes including broken rotors, bearing failure, and eccentric shaft faults. Their algorithm is based on classical machine learning methods such as k-Nearest Neighbors (KNN) [61], and Support Vector Machine [62], but the dataset was recorded in a noise-free lab, from a mobile-phone positioned about 1 meter from the UAV to make the recordings as clear as possible due to the indoor environment.

Our proposed method is meant to work in non-ideal environments, and the dataset that was created for our method was recorded in a variety of environments with noise. The

work in [58], has exploited two kinds of DL models, RNN and CNN, to see which of them achieves better accuracy in AD. The task was to identify which rotor is malfunctioning and classify the malfunction under two fault classes, fractured tip, and edge distortion. The proposed method in the aforementioned work reached an accuracy of up to 98% by using an array of microphones which enables us to identify which rotor is malfunctioning but the use of an array of microphones makes the system complicated and might not be applicable to smaller UAVs. Moreover, the dataset that was used to train the model was in an indoor environment and without noise. However, in this paper, we propose an improvement by using a Transformer to weigh parts of the input differently to accentuate important features, and a system using a single microphone that weighs significantly less as a consequence.

Acoustic data as captured by a microphone has a single dimension by nature (a stream of samples in time). Yet, it can be presented as a two-dimensional matrix (Y-axis as its frequency dimension, and the X-axis as samples-in-time). Thus such a matrix can be presented as an image. The approach proposed in this paper is inspired by a CNN model called VGG [63], which was already used in the domain of acoustic analytics [64], [65]. The VGG is a CNN model used in the domain of computer vision, for tasks like image classification [66] and object detection [67]. The VGG architecture managed to achieve 8.0% top 5 error [63] on the test set at the task of image classification with the ILSVRC dataset, which consists of 1.3M images for training and 100K for testing and has 1000 different classes. VGG has scored the second highest among the tested models, right behind GoogLeNet [68] with a margin of 0.1%, but it has fewer layers and is much less complex.

Recall that in this paper we apply a Transformer architecture before the CNN, from the domain of speech recognition called Wav2Vec2 [26], that used in many tasks of speech recognition [69], and that has proved to be efficient whenever the dataset is small [32]. The Wav2Vec2 model takes raw sound data as input and outputs a more informative representation of that data [26]. Although the Wav2Vec2 is meant for tasks involving speech recognition, in this work we use the same Transformer to solve an acoustic problem. Our intuition is that an anomaly in acoustics can be considered an anomaly in patterns of speech, like speech impediments, as can be seen in [70] that shows how the Wav2Vec2 can be used for the task of identifying speech sound disorders. The Wav2Vec2 [26] is a group of transformers, that are mainly used in the context of speech recognition. The Wav2Vec2 family uses representation-learning to transform sound into a matrix representing that sound, while also accentuating important features in the sound. Representation-Learning [71], is a set of techniques that enables a system to automatically learn the representation of raw data for tasks like classification and detection [72]. This set of techniques replaces manual feature extraction and feature engineering. Transformers are a type of representation-learning that uses Self Supervised Learning (SSL) [73] to learn the best representation of raw data for a

given task. SSL is a technique in machine learning that is used to train a model with unlabeled data, usually before training it again later with labeled data for fine-tuning [26], [74], [75]. There are several approaches for solving the problem at hand, though part of them are only capable of detecting very specific anomalies. While other methods use machine learning to classify a wide range of classes, yet, such methods usually require a massive dataset for the learning (training) process. The proposed method in our work uses a more compact and realistic dataset and uses a Transformer to handle the noise in challenging environments.

III. DATA COLLECTION

The following section describes the data collection phase, namely; (i) the manner in which the dataset was generated; (ii) what features were taken into account and their influence; and (iii), the differences between particular labels and their distribution.

The dataset¹ used in this paper were recorded manually by the authors. The micro UAV used for recording is the DJI's Tello drone. This quadrotor is a cost-effective micro-drone and is popular among beginners, intermediates, and even professional drone developers. In addition, this quadrotor (see Figure 1) is very easy to manipulate, since there is an official SDK for Android and iOS smartphones for controlling the quadrotor. Also, a user-friendly interface Tello SDK written in Python is included. This allows the owners to connect and send commands to it through WiFi and run self-made scripts to control it from a computer. Moreover, the Tello drone is particularly small and has multiple flight modes that make it very agile while flying. These advantages couldn't be ignored, and as a result, the Tello was chosen for the data generation task. The data generation process was not that simple, due to two major disadvantages; (i) the Tello has a short flight time, which is about 5-10 minutes; (ii) The Tello's WiFi communication uses the UDP protocol. Since this protocol works like a stream, the quadrotor might miss some commands, and whenever this situation occurs, it may land or even crash due to the loss of communication. Undesired crashes and landings forced us to scratch the current recording and start over.

For the recording procedure, it was a deliberate choice to test the significance of the microphone's weight on the balance of the quadrotor. The recording setup was as follows: a small piece of Tin (about 5 cm long) was taped pointing upwards on top of the quadrotor at its center of mass using Duck-tape. At the tip of the Tin piece, we placed a small JBL Tune 225 TWS Bluetooth earphone acting as a microphone, and the quadrotor was controlled by a computer that also received the audio stream from the earphone over Bluetooth.

The recording procedure took place in different environments: closed rooms and open spaces, all with and without noisy environments [76] (mostly human speech), since these constraints may affect the sound waves that the microphone is

recording. These constraints were taken into account so that the dataset would be as diverse as possible, and to get better performance in non-ideal environments. The recordings were done both manually and automatically using two different scripts to control the quadrotor. The scripts also produced a log file in which information was written about the quadrotor every tenth of a second. The scripts logged information such as recording time, flying status (if an anomaly has occurred or not), barometric sensor data, yaw, pitch, and roll angles, height, and battery percentage. Next, the script saved the recorded audio in a WAV (Waveform Audio) file format and the log file corresponding to that recording in CSV (Comma-separated values) file format. For the automatic recordings, the script included numerous movement patterns that were pre-defined for variety, such as square orbits and turns in mutable altitudes.

Different types of anomalies were recorded, including partially broken and defective blades, undesired movements, or destabilization and hits from an external source. To create even more diversity, all the recordings were done with the Tello's original blades as well as third-party blades (slightly lighter than the original ones). Furthermore, actually broken blades were used in the recordings as well. Each recording lasts 2 minutes long. In order to understand how and when the small UAV experienced an anomaly while recording, we combined the movement commands sent to the quadrotor with the data received from its sensors. For instance, whenever the quadrotor moved and the command it received was to stay still, but an unwanted movement occurred. Another instance of an anomaly is whenever a hit is recognized. A hit can be characterized as a drastic unwanted change in the quadrotor's accelerometer sensor, hence, the same approach can be modified to similarly identify hits from external sources.

The audio recordings consist of 3 hours long of recordings, separated into two minutes for each audio recording (as mentioned before), each with a corresponding log file describing it. After the recording phase, and in order to use it for training the DL model, data engineering was needed. The audio recordings were split into 1-second long soundtracks and were saved into a directory with a unique name. Using the log files, for each recording a corresponding label was also written and saved as a text document) file format in a different directory with the same name as the recording. The result was two directories, one with WAV files of 1-second long sound bits, and the other contained the labels for each second-long sample.

The total number of samples is 11,040. In order to verify that the collected data is as diverse and non-trivial as we wished, we manually looked up for different types of *Normal* records and *Anomaly* records. To do that, we visualized the data. A natural challenge we faced while exploring the dataset was the class imbalance; a quick statistical analysis showed that 85% of the samples were labeled as *Normal*. In contrast, only 15% of the samples were labeled as *Anomaly*. This is not a surprise, since in most of the flight time the UAV doesn't

¹Available upon request from the authors, as well as the code.



FIGURE 2. An example of an emission output matrix, of the Wav2Vec2 - *Normal* sample.

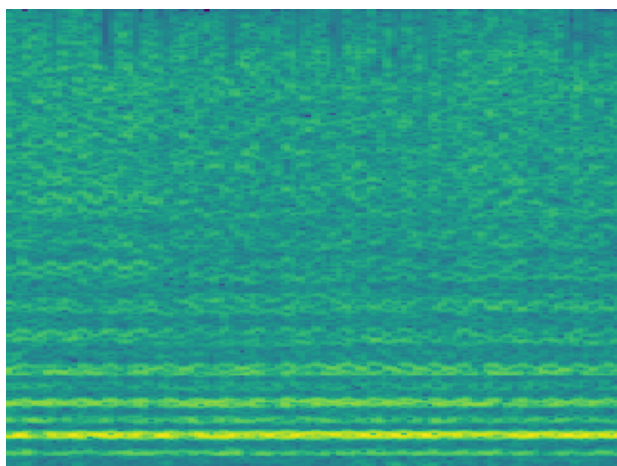


FIGURE 3. An example of a graphical representation of a spectrogram - *Normal* sample.

have any anomalies. In Section V, we discuss two methods in which the data is processed into visual images: spectrograms and emissions. One can see in Figures 2 and 3, a particular sample from an audio recording, labeled as “normal”, and visualized as an emission from the Wav2Vec2 [26] Transformer (Figure 2) and as a spectrogram (Figure 3)

We found out that flights with partially broken and defective blades are much harder to classify as anomalies, and are visually very similar to normal recordings. Hence, we speculated that our CNN (discussed in Section IV) would have a hard time catching these anomalies. Figures 4 and 6 shows two different samples, labeled as “anomaly”, and visualized as an emission from the Wav2Vec2, and as a spectrogram in Figures 5 and 7.

The first sample visualizes part of the quadrotor’s stabilization process. One can easily distinguish between the two samples. On the other hand, the second sample is much harder to classify as *Anomaly*. This sample visualizes one second from a flight where the quadrotor had a defected or partially broken blades. The sound emitted from the rotors was almost identical to the sound of proper rotors and will be explained in Sections IV and V. Finally, one can note in Figure 8 the Data-Collection process described in this Section; Namely, the *Main-Computer* component runs the process that sends

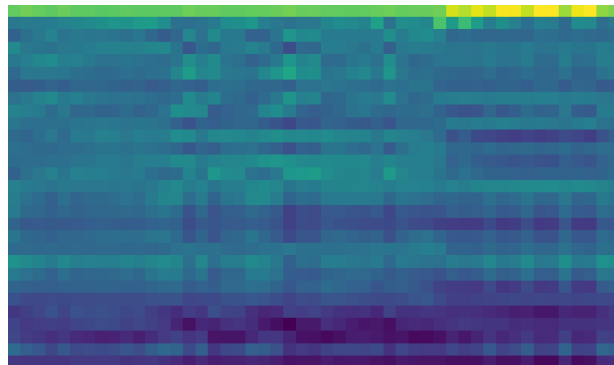


FIGURE 4. An example of an emission output matrix, of the Wav2Vec2 - *Anomaly* sample.

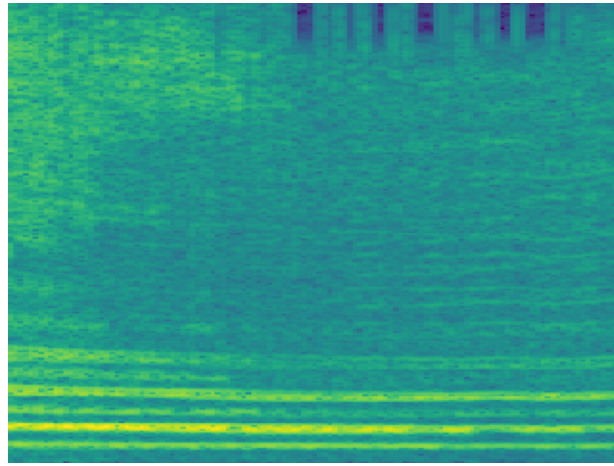


FIGURE 5. An example of a graphical representation of a spectrogram - *Anomaly* sample.

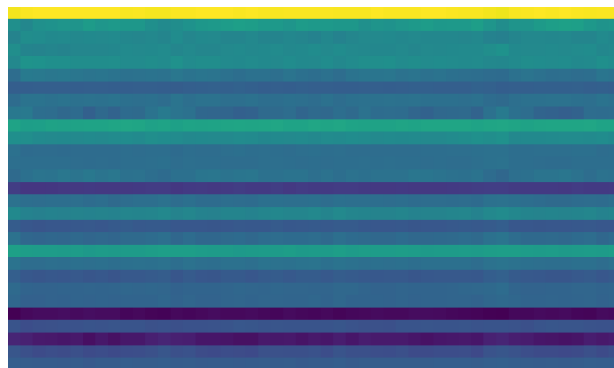


FIGURE 6. An example of a graphical representation of an abnormal emission.

the commands to the UAV, then the UAV creates sound waves during its flight time. Next, two threads are working in parallel, which is the (i) transmission of information from the microphone to the computer, and (ii) transmission of yaw, pitch, and roll states from the UAV’s Accelerometer to the computer. Finally, this process outputs whether the UAV is in *Anomaly* or *Normal* state.

IV. OUR APPROACH

The following section describes our proposed method to detect anomalies, of types: (i) unplanned stabilization, and

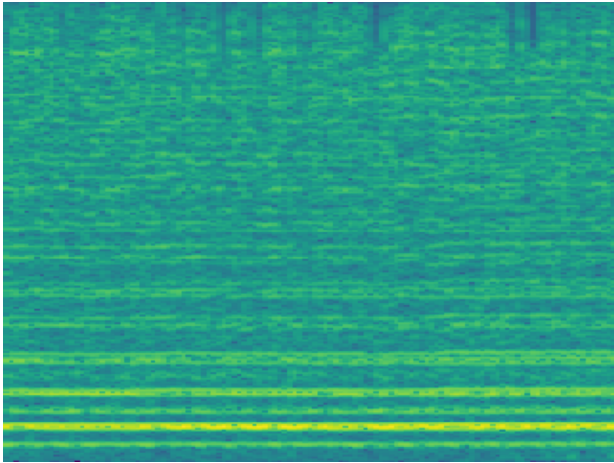


FIGURE 7. An example of a graphical representation of an abnormal spectrogram.

(ii) malfunction/tackled propeller in UAVs using its emission sound.

The remainder of this section is structured as follows; Section IV-A describes in detail the idea of a Transformer and discusses the specific Transformer based architecture used in this work. Next, Section IV-B introduces the VGG architecture and discusses its use both as a separate model and as part of the proposed hybrid model in this work (Figure 9). Section IV-C presents the Transfer Learning technique and its implementation using the Wav2Vec2 over the CNN model. Finally, Section IV-D describes the final algorithm.

A. Wav2Vec2 AND Wav2Vec2-ASR-960h

Wav2Vec2 [26] is a group of transformers, that are mainly used in the context of speech recognition. The Wav2Vec2 family uses representation-learning to transform sound into a matrix representing that sound, while also accentuating important features in the sound. Representation Learning [71], is a set of techniques that enables a system to automatically learn the representation of raw data for tasks like classification and detection [72]. This set of techniques replaces manual feature extraction and feature engineering. Transformers are a type of representation-learning that uses SSL [73] to learn the best representation of raw data for a given task. SSL is a technique in machine learning that is used to train a model with unlabeled data, usually before training it again later with labeled data for fine-tuning [26], [74], [75]. Note that in the scope of this paper, the self-supervised learning is a process already learned in the pre-trained Wav2Vec2 model, from which we perform the transfer learning. That is, the Wav2Vec2 model only was fine-tuned with our datasets and was not trained or re-trained using SSL.

As aforementioned, the Wav2Vec2 takes raw audio as input and outputs an image-like representation of the input data, which can then be passed to a classifier that can differentiate between an anomalous and regular sound. Namely, both the Wav2Vec2 models are composed of a multi-layer based convolutional feature encoder, and receive as input a

raw audio matrix, and their output are latent speech representations for each time-step among T time-steps. Next, the speech representations are fed into a Transformer, that creates T representations, extracting information from the whole sequence. Finally, the feature encoder output is discretized, in order to represent the targets (outputs) as a self-supervised-based objective function. The feature encoder contains a temporal convolution followed by a normalization layer and a GELU [77] activation function. Then, the encoder's total stride computes the amount of the T time steps, which serves as the Transformer's input. In this manner, it is possible to distinguish between the sound emitted from the rotors, which was almost identical to the sound of proper rotors. Next, the Transformer produces contextualized speech representations; that is, the feature encoder output is fed into a context network that follows the Transformer architecture as in [78]. The main change is that instead of fixed positional embeddings [78] which encode absolute positional information, the Wav2Vec2 exploits a convolutional layer that behaves as if it was a relative positional embedding. The convolution's output is being added to the inputs, followed by a GELU [77] activation function, and then apply the layer normalization process. One can note in Figure 10, the Wav2Vec2 architecture.

The Wav2Vec2 has been originally designed for human-speech recognition. Yet, it is possible to exploit it for the general acoustic problem as in UAVs. Wav2Vec2 leverages self-supervised training in a continuous framework from raw audio data. It builds context representation over continuous speech representation and self-attention capture dependencies over the entire sequence of latent representation end-to-end [79]. Speech representations can be used for several downstream tasks [80], such as AD in UAVs using sound. Similar to human speech, UAVs produce continual raw audio, when in a normal state that can be considered as a representation of silence in human speech, its anomalies are reflected as notable shifts in the acoustic signals (as in human speech), which can be recognized clearly. Therefore, exploitation of the Wav2Vec2 over an AD in UAVs sound can improve the model's ability to detect patterns in the data, which will eventually increase the model's accuracy.

B. CNN (VGG-16)

In this research, we have used an altered version of a popular CNN architecture, which is the VGG-16 [63]. The VGG-16 model is designed for image classification and object localization and won first place in ILSVR (Imagenet Large Scale Visual Recognition) competition in 2014. Although the VGG-16 model is mainly used for image processing tasks, it can be used for speech processing and phoneme recognition by converting a sound segment into a spectrogram or other visual forms that can be represented as an image. As a result, the model can classify segments or extract features from them. In order to train a VGG-based model, the input consists of a fixed-size 224×224 RGB image, where the only pre-processing being made is subtraction of the mean RGB value

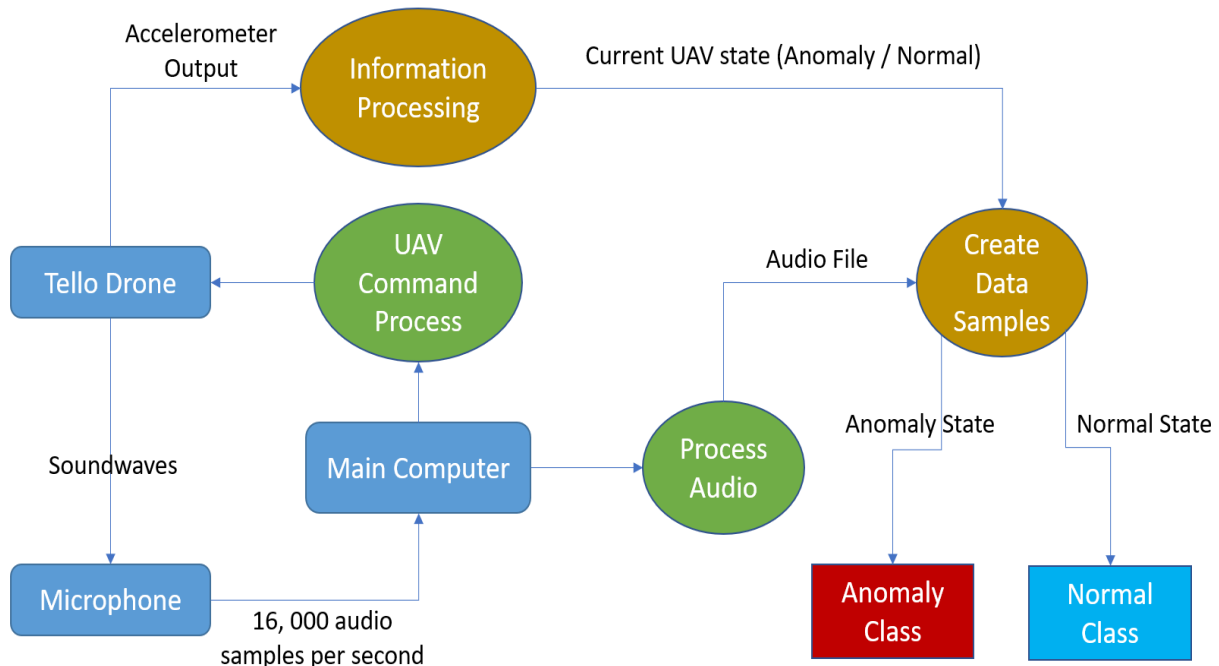


FIGURE 8. The Data-Collection process described in this section.

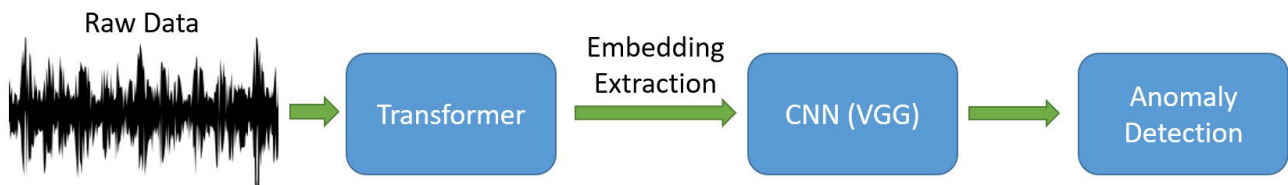


FIGURE 9. The pipeline of the anomaly detection framework proposed in this work.

from each pixel, in any image of the training set. Next, the image is being passed through a chain of convolutional layers, with extremely small filters of size 3×3 . The stride of the convolutions is set to 1 pixel, which is the spatial padding of the convolution operation. The layer's input is processed so that the spatial resolution is preserved after the convolution operation. Next, a spatial pooling operation is performed, followed by five max-pooling layers. The max-pooling computation on top of the convolutions is performed over a pixels window of size 2×2 , with a stride of size 2. Next, a chain of convolutional layers is followed by 3 fully-connected layers, where the third performs a classification of over 1000 classes from the ILSVR [81] competition. Finally, a softmax layer outputs the probability for each class. Throughout the neural-network architecture flow, all of the hidden layers are transforming the mathematical operations with a ReLU activation function [82]. We used the VGG-16 model in two methods: (i) an altered version of the VGG-16 CNN architecture as a standalone model; (ii) as the classifier in a two-layer model. In the second method, we use the Wav2Vec2 to extract features from sound segments and visualize them as images that are used as input to the CNN. In the next subsection, we go into detail about the second method that uses the Transfer Learning technique.

C. Wav2Vec2 OVER CNN

Our main approach is based on the combination of both the Wav2Vec2 and the VGG-based CNN, on which a Transfer Learning [83], [84] is applied. In DL, Transfer Learning is the application of knowledge gathered from a model that was trained for a specific task, that can later be reused as the backbone for a more advanced task. This approach is popular in DL, by applying pre-trained models that are used as the starting point on Computer Vision and Natural Language Processing tasks given the vast computing and time resources required to develop neural network models for these problems [85]. Using the Transfer Learning method can improve the chances of solving the AD in UAVs by using acoustic signals, having only 3 hours of recorded data, by using pre-trained Transformers [86]. The Wav2Vec2 is a pre-trained model with over 960 hours of data to its training set so that it can perform well on tasks that are similar in nature to the tackled task in this paper. The exploitation of the Transfer Learning technique with a pre-trained Transformer over a CNN model (VGG), contributes to faster convergence of the model while improving its accuracy [87]. In this approach, the input of the model corresponds with the Wav2Vec2's input as presented in Section IV-A, and its output is an image-like input which is the raw audio of the sound emitted from

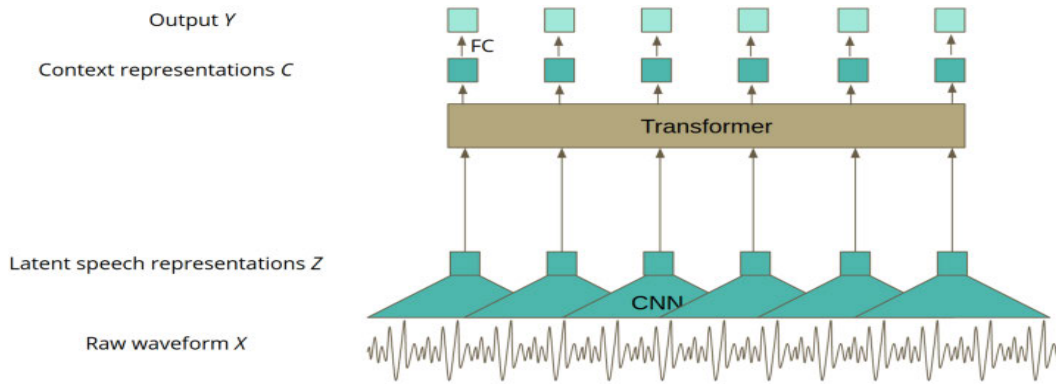


FIGURE 10. Wav2Vec2 Architecture.

TABLE 2. A summarization of the VGG-16 layers modifications, that are manifested in the CNN of the Wav2BC+ architecture.

Layers Considered / Architecture	Wav2BC+	VGG-16
Convolutional Layers	4	13
Pooling Layers	2	5
Dense Layers	2	3

the UAV, that serves as input to the VGG-based architecture employed in this paper (Figure 12). As for the output of the VGG-16 model (Section IV-B), our CNN model that is based on the VGG-16 architecture, and only outputs 2 classes, instead of 1000 as mentioned in Section IV-B, as the challenge in this work is the detection of anomalies and normal states in UAVs flight.

D. PROPOSED METHOD

In this paper we propose a method for AD in UAVs, using only the sound emitted from them. The method is based on a transformer-based model for binary classification, so that the model gets as input the raw sound that is emitted from the UAV, and outputs the probability that an anomaly has occurred. In this paper the model is built from two components, a transformer-based architecture for feature extraction called Wav2Vec2 [26] and a classifier model that is inspired by the VGG CNN architecture [63]. Figure 11 shows the architecture of VGG-16, while Figure 12 presents ours, which is the mini-VGG version of VGG-16.

Observing Figures 11-12, one can note that the main differences between the VGG-16 architectures, and the one presented by the CNN of the Wav2BC+, are as follows in Table 2. The result of our modified CNN architecture of the Wav2BC+ is a decrease in the number of parameters of the well-known VGG from 138M, to a shrunk version of the VGG with only 3.6M parameters.

Our approach demonstrates that it is possible to use tools from the domain of speech recognition and analysis in audio-analysis problems. It starts with the usage of the Wav2Vec2 in order to extract features from the raw audio data to get a representation of the audio. The new representation is then passed to the VGG-based CNN model, which serves as the classifier that computes whether the sound is anomalous or

regular. In the training process, we used 1-second long sound samples of sound. To make training faster and create a more reliable model, we used transfer learning to fine-tune the transformer and train the classifier model. One can note in Figure 9 an illustration of the pipeline proposed in this paper.

V. EXPERIMENTAL EVALUATION

The following section is dedicated to the validation of our hypothesis that using Transformer-based techniques for UAVs AD using sound can outperform classical CNN techniques. Our evaluation was conducted on an HP Omen computer, Windows 11 64Bit OS with 3.20GHz AMD Ryzen 7-5800H CPU, 32GB of RAM, and NVIDIA GeForce RTX 3070 GPU, using PyTorch (v1.14.0) and scikit-learn (v1.1.3).

A. DATASETS FOR EXPERIMENTAL EVALUATION

This section presents 3 different datasets, each of them with its purpose and uniqueness for an appropriate experiment. Table 3 summarizes the datasets used for each of the experiments.

B. ANOMALY DETECTION USING CLASS-WEIGHTS

Recall that in our dataset, 85% of the samples were labeled as Normal. In contrast, only 15% of the samples were labeled as Anomaly (Section III), which leads to an imbalanced dataset situation. The problem that arises from an imbalanced dataset, i.e. the example ratio from the Anomaly and Normal classes, should be addressed before any further progress. Suppose that our classifier would always produce “normal-state” as an answer for all the test examples, i.e. always predict Normal. Even though it would obtain ~ 85% of Accuracy, over our dataset (as ~ 85% of the dataset contains Normal examples), it would still perform poorly [88] when examining the Precision and Recall measures which indicate how successful the model is (the accuracy, precision, and recall are discussed thoroughly in the continuation of this section). As a result, we used a class-weighted cross-entropy loss function which was introduced with each class’s weight, as the inverse ratio of the number of examples of each class in the dataset, i.e. $|Anomaly|^{-1}$ for class Anomaly, and $|Normal|^{-1}$ for

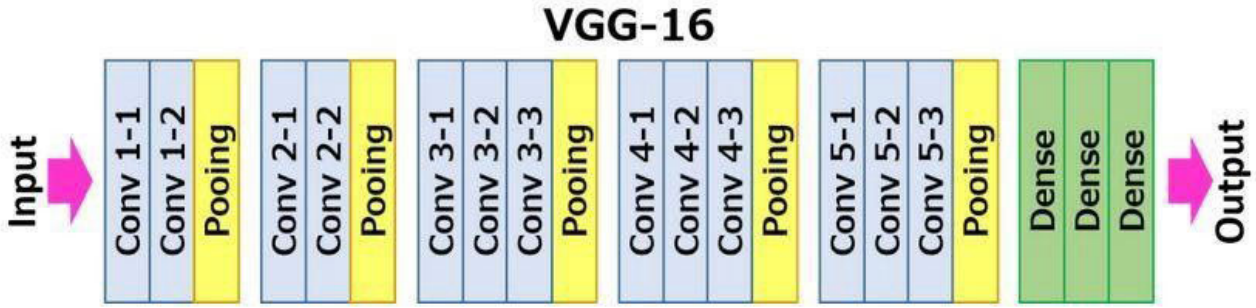


FIGURE 11. A graphical illustration of the VGG-16 architecture. It consists of 5 blocks composed of 2 or 3 convolutional layers followed by a pooling layer, and 3 dense layers before the output layer.

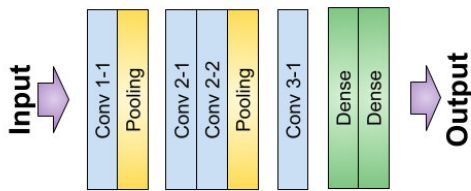


FIGURE 12. A graphical illustration of the architecture proposed in this work.

class *Normal*. Since our aim is to detect anomalies, the ratio mentioned above is normal. Therefore, we also tested the use of non-weighted binary-cross-entropy [89] loss function on the Transformer.

C. CROSS-ENTROPY & BINARY-CROSS-ENTROPY

Next, we present the loss functions considered for the models' construction in this work. We used both the Binary Cross Entropy (BCE), and the regular Cross Entropy (CE) one, as follows:

The BCE compares a target y with a prediction p in a logarithmic and hence exponential fashion. In neural network implementations, the value for y is either 0 or 1, while p can take any value between 0 and 1. The formula of the BCE loss is presented in Eq.(1):

$$BCE = -(\log(p)y + (1 - y)\log(1 - p)) \quad (1)$$

When visualizing BCE loss for a target value of 1, the loss increases exponentially whenever the prediction approaches the opposite - 0. This suggests that small deviations are punished albeit lightly, whereas big prediction errors are punished significantly. This fact makes the BCE loss as a good candidate for binary classification problems, whenever a classifier has two output classes. The Sigmoid activation function receives the last layer output (logits) as an input and outputs a single value between 0 and 1 which represents the probability of class 1 being the target class (while the probability of class 0 = 1 - P(class 1)). The BCE loss function except for a single input feature between 0 and 1. Therefore, the Sigmoid activation function is commonly used for binary classification problems as it can ensure the output of a neural

network fits the BCE loss function's input expectations. The formula of the Sigmoid is presented in Eq.(2):

$$Sigmoid = \frac{1}{1 + \exp(-x)} \quad (2)$$

The CE loss [90] on the other hand, compares a hot-dot target 1-dimensional vector y with a 1-dimensional probability vector p , both of them in a logarithmic and exponential fashion. In neural network implementation, the target vector consists of $i = 1, 2, \dots, M$ entries such that exactly $M - 1$ entries are equal to 0, and the entry representing the correct class is equal to 1, while the prediction vector consists of M entries with values between 0 to 1. The CE loss is given as follows in Eq.(3):

$$CE = - \sum_{i=1}^M y \log(p) \quad (3)$$

As for the output layer, we used the Softmax activation function that receives a 1-dimensional vector (logits) and outputs a 1-dimensional probability vector that contains the probability of each class in the vector. Therefore, using a Softmax activation function over the last layer output will ensure that the output of the model will fit the CE loss function. This fact, makes the CE a good candidate for Multi-class classification problems, whenever a classifier has more than 2 classes. Yet, it is possible to use the CE for binary classification problems, as it is a private case. The equation of the Softmax activation function is given by Eq.(4):

$$Softmax = \frac{\exp(x)}{\sum \exp(x)} \quad (4)$$

where x is a specific element in the 1-dimensional output vector of the Softmax activation function. Next, we discuss in Sections V-D and V-E the models constructed in this experimental evaluation.

D. SPECTROGRAMS AND CNN

A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies over time. Spectrograms are widely used in speech processing [91] and phoneme recognition. It is because when using spectrograms, it is possible to

TABLE 3. Datasets statistics for each of the experiments in this section.

Entry #	Experiment	Train Samples	Test Samples	Sampling Rate	Sample Duration	Anomaly-to-Normal Ratio
(1)	Section V-C	8,832	2,208	16KHz	1 second	1:10
(2)	Section V-D	None	300	16KHz	1 second	1:10
(3)	Section V-E	None	300	16KHz	1 second	1:10

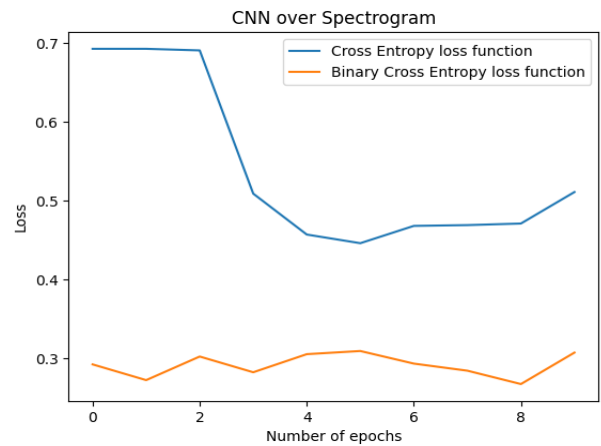
distinguish a specific frequency and its decibels over time. This can be very useful for recognizing the vocal anomalies of a flying UAV.

The first thing that comes to mind is how to adjust the dataset, so the CNN can be trained and evaluated on it. Since a spectrogram can be saved as an image, we could insert it into the CNN as input. Hence, for each recorded sample, we generated a spectrogram. The spectrograms are generated by NFFT (Non-equispaced Fast Fourier Transform) [92], with a sample rate of 16,000Hz. Each frame of audio is windowed using the Hann function [93] into a window of length 512, and the number of points of overlap between frames is 384. Such as, each spectrogram is represented as an image of size 320×240 pixels.

The “new” dataset contains the spectrograms and their labels. Each spectrogram was loaded into the dataset as grey-scale images since RGB images are not 2-dimensional. The data set was then shuffled and split into 3 subsets: training set (80% of the entire dataset which contains 8832 samples), validation set (10% of the entire dataset which contains 1104 samples), and test set (10% of the entire dataset which consists 1104 samples). It is important to note that the proportion between the output classes is kept. Next, the CNN model uses the Adam optimizer [48] with a learning rate of 0.0001, with the BCE loss function. The CNN was trained with our training set for 10 full epochs, with a mini-batch size of 16. At the end of each epoch, we evaluated the CNN’s performance by validating its stats with our validation set. The mini-batch size of the validation epoch is also 16. Before starting a new epoch, we took the measurements of loss and accuracy for the training and validation sets. After the training session ended, we evaluated the model using the testing set. Next, Figure 13 demonstrates that the non-weighted BCE loss function outperforms the weighted-class cross-entropy loss function. We provided the training process both for the CE and BCE loss functions for 8 epochs. One can note from Figure 13 that after 1 epoch only, the BCE loss already converges into minimal value, yet for comparison purpose its loss is presented up to 8 epochs as the CE loss function.

E. Wav2Vec2 OVER CNN

The Wav2Vec2 group of Transformers provides a set of pre-trained Transformers, such as Wav2Vec2-Base [32], that was pre-trained on 960 hours of unlabeled audio from LibriSpeech dataset, and Wav2Vec2-ASR-Base-960H that was pre-trained on the same 960-hours dataset, and was fine-tuned for an Automatic Speech Recognition (ASR) task. The usage of pre-trained transformers allows us to fine-tune the Transformer with a small dataset (our dataset consists of

**FIGURE 13.** Comparison between non-weighted BCE loss function and weighted Cross-Entropy loss function over 8 epochs.

~ 3 hours of labeled audio). When using pre-trained models to perform a task, in addition to instantiating the model with pre-trained weights, one also needs to build pipelines for feature extraction and post-processing, in the same manner, they were done during the training. To build this pipeline, we used the `torchaudio.pipelines` module which contains prepared pipelines for each of the Wav2Vec2 models. The idea of Transfer Learning is widely used in this section via the Transformer which is designed to transfer its input sequence to another one with the help of two parts (Encoder and Decoder [94]) and the CNN model.

In order to implement the idea of transfer learning on our models (Transformer and CNN), the pipeline includes both models. The following sections describe the fine-tuning process while emphasizing the transfer learning idea in it as well. That is, Section V-E1 describes the fine-tuning process using an already fine-tuned (to a different problem) Transformer; Next, Section V-E2 presents the idea of fine-tuning a pre-trained Transformer for AD in UAVs acoustic problems without it ever been introduced to a similar problem before.

1) FINE-TUNING Wav2Vec2+ WITH Wav2Vec2-ASR-960H

The following model consists of a pre-trained Transformer (Wav2Vec2-ASR-Base-960H) which has been fine-tuned to ASR problems and a CNN. The input for the Transformer is a 1-second waveform with a shape of a 1-dimensional array and a sample rate of 16000Hz. The waveform is transferred to a tensor with a shape of [1], [29], and [49] (Transformer’s output) and inserted as an input to the first layer of the CNN model.

Next, the fine-tuning process begins as the CNN model starts its training loop over the outputs from the Transformer. The CNN was trained for 10 full epochs, with a mini-batch

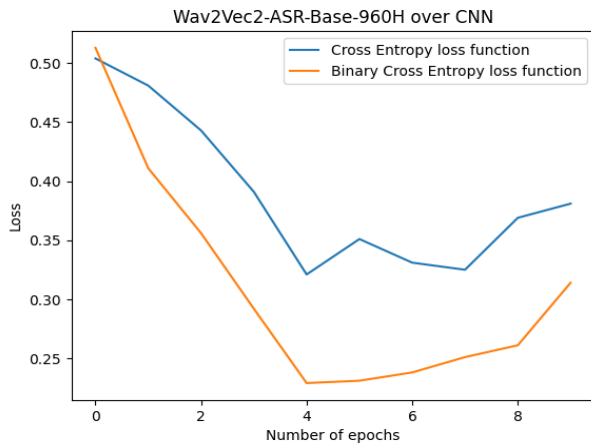


FIGURE 14. Comparison between non-weighted BCE loss function and weighted Cross-Entropy loss function over 8 epochs.

size of 16. By the end of each epoch, we evaluated the CNN's performance by validating its stats using the validation set. One can note from Figure 14 that the non-weighted BCE loss function outperforms the weighted-class CE loss function.

2) FINE-TUNING OF Wav2BC+

The following model consists of a pre-trained Transformer (Wav2Vec2-Base) and a CNN. The input for the Transformer is a 1-second waveform with a shape of a 1-dimensional array and a sample rate of 16000Hz. The waveform is transferred to a tensor with a shape of [1, 49, 768] (Transformer's output) and inserted as an input to the first layer of the CNN model. Notice that the output of Wav2Vec2-Base and the output of ASR-960H-Wav2BC+ has different shapes, which is a direct result of the last Transformer being already fine-tuned to a specific case (as ASR).

The CNN was trained for 10 full epochs, with a mini-batch size of 16. By the end of each epoch, we evaluated the CNN's performance by validating its stats using the validation set. After the training session ended, we evaluated the model using the testing set. One can note from Figure 15 that the non-weighted BCE loss function outperforms the weighted-class CE loss function.

F. RESULTS & MODELS COMPARISON

In order to measure how good a model is, there are many different metrics that can indicate the quality of a model. For classification problems with a balanced ratio of the classes present in the training dataset, accuracy is good enough and can indicate quite well how good a model is at a certain task. Accuracy aims to answer the question of how close a given set of measurements (observations or readings) are to their true value. The formula for computing the Accuracy is presented in Eq.(5):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

where True Positive (TP) is the number of inputs true and the model is classified as true, True Negative (TN) is the number

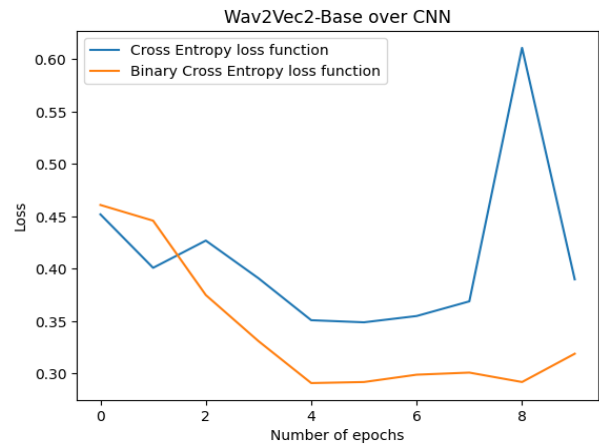


FIGURE 15. Comparison between non-weighted BCE loss function and weighted Cross-Entropy loss function over 8 epochs.

of inputs that are false and the model is classified as false. False Positive (FP) is the number of inputs that are false but the model classified as true and False Negative (FN) is the number of inputs that are true but the model classified as false.

However, accuracy is not a good enough indicator whenever the data is imbalanced, meaning that there are much more occurrences of a class relative to other classes. In this paper, the dataset is highly imbalanced (see Section III), i.e. the anomalies are by definition uncommon, hence naturally they occur infrequently in our dataset. Therefore, we use a different metric, called F1-score, which is a better metric for classification performance measurements in imbalanced datasets [95]. The F1-score uses two other metrics called Precision and Recall, to present a more precise reflection of the models' performance.

The Precision, is a measure of how many of the positive predictions made are correct (TP), and its formula is presented in Eq.(6), as follows:

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

The Recall, (or Sensitivity) is a measure of how many of the positive cases the classifier correctly predicted, over all the positive cases in the data, and its formula is presented in Eq.(7), as follows:

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

Finally, the F1-score is a metric that combines both Precision and Recall. It is generally described as the Harmonic-Mean [96] of these two. A harmonic mean is a way to calculate an average of values, generally described as more suitable for ratios (such as precision and recall) than the traditional arithmetic mean. The idea is to provide a single metric that weights the two ratios (precision and recall) in a balanced way, requiring both to have higher values for the F1-score to rise. The formula of the F1-Score is presented in Eq.(8), as follows:

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (8)$$

In order to create a fair and precise comparison between the 3 Models tested in this paper, each Model was measured using the following metrics: (i) Accuracy; (ii) Precision; (iii) Recall; and (iv) F1-Score. Table 4 presents each model's performance, based on each of these four metrics:

At first glance, the results of these 3 different models trained as part of this study, it is not very clear which one produces better results in the test set. After a deeper understanding of the results, it is possible to evaluate which model yields better results for different types of tasks. The decision to train the models over 10 epochs is a result of the nature of the models, they start to divergence after 5-8 epochs, as presented in Figures 13, 14, and 15.

Since our dataset is imbalanced, the accuracy is inappropriate enough performance measure for the problem we study. The main reason is that the overwhelming number of examples from the majority class (Normal) will overwhelm the number of examples in the minority class (Anomaly), meaning that even poor and untrained models can achieve accuracy scores of 90 percent and above. Therefore, comparing the accuracy of the models is effective, compared to other measurement metrics, as mentioned in Section V-F.

1) TRANSFORMER BASED MODEL VS CNN BASED MODEL

A comparison between the models from Entries (1) - (2) which implements the idea of Transfer learning in Table 4, and the model presented in Ent.(3) which implements the classic idea of CNN (VGG) based model, is sufficient to this study as it tests our thesis regarding the importance of using Transformers in order to detect anomalies in acoustic emitted from UAVs. Considering the Precision of each model in the Table, it can be clearly seen that the Transfer learning idea presented by Ent.(1) and Ent.(2) in Table 4 yields better performance, compared to the classic technique mentioned in Ent.(3). One can conclude from this that these models are almost never wrong when they detect an *Anomaly*.

In terms of Recall, Ent.(2) yields a value that is very close to the performance in Ent.(3). Yet, The model from Ent.(3) yields better Recall values than Entries (1) - (2). Despite of the tiny gap between the Recall of Ent.(3) and Ent.(2), it is possible to say that the Transfer Learning technique yields a model that manages to identify a fairly high number of *Anomalies*, which is more important to the problem presented in this work.

Next, and as can be seen from Ent.(2), a Transformer over CNN is a better model for tasks that focus on minimizing false positives, while the model from Ent.(3) is better for tasks that focus on minimizing false negatives. In imbalanced datasets, the goal is to improve Recall without hurting the Precision. Based on that, we might conclude that the Transformer over CNN model did better in the test compared to the classic model. However, neither Precision nor Recall tells the whole story, i.e. a model might have excellent Precision with terrible Recall and vice-versa. Thus, the F1-Score provides a manner to express both concerns with a single score. One can note from Table 4 in Entries (2) - (3) that the model that performed

better is the Transformer over CNN, since its F1-Score is higher than the CNN Model.

These results support the thesis of this paper since it claims that Transformer based model would perform better than a CNN Model. The reason for this difference between the results is based on the structure of the models. The Transformer based Model is using self-attention layers, which helps the Model identify important features in the input and emphasize them. On the other hand, CNNs in their nature are not searching for important features in their input, but search for patterns over the entire input instead, which makes the detection process harder since there is no attention to the important details. Thus, the exploitation of Transfer Learning with the Transformer and the CNN allows the CNN to train and look for patterns over the important features and thus improve its performance.

2) Wav2Vec2-ASR-BASE-960H VS Wav2Vec2-BASE

The performance comparison of Entries (1) - (2) in Table 4 provides us with a deeper understanding of the selection of the Transformer. In addition, it shed light on whether using a fine-tuned (to a different problem such as ASR) model, that might result in underperformance compared to a regular pre-trained Model.

According to Table 4, in terms of accuracy, Ent.(2) yields a more accurate model (0.92) compared to Ent.(1) which yields 0.90. Since our dataset is imbalanced, the accuracy measurement method is not a good enough metric for this case. Therefore, the other metrics such as the Precision, Recall, and F1-Score are more accurate metrics.

As for the Precision, Ent.(1) yields a precision of 0.85, while Ent.(2) yields a Precision of 0.87. Similarly, the Recall value is higher in Ent.(2) which is 0.89, while Ent.(1) ends up with a Recall of 0.82. As a result, the F1-Score in Ent.(1) is 0.84 which is lower than the F1-Score in Ent.(2) which is 0.88.

Considering the results of the models that correspond with Entries (1) - (2), it is possible to say that the regular pre-trained Transformer (Ent.(2)) performed better than the already fine-tuned Transformer (Ent.(1)). The main reason for this gap in results between these 2 models is the fact that fine-tuning a Transformer to a specific problem (ASR) reduces the number of output features that the Transformer feeds the CNN with. As a result, the CNN receives a small number of features to search for patterns on, which leads the model to under-performance, compared to the model with the regular Transformer. Therefore, one can conclude that the Wav2Vec2-Base Transformer over a CNN (VGG) model is the best, out of these two.

G. RESULTS - OUT OF DISTRIBUTION EXPERIMENT

A vital criterion for deploying a powerful classifier in many real-world AI-based applications is the ability to detect test instances that are considered sufficiently far away from the training-set distribution. Many classification problems, such as speech recognition, visual object detection, and Anomaly

TABLE 4. Results table for the comparison of the Transformer Wav2Vec2-ASR-Base-960H + CNN (VGG-based), Transformer Wav2BC+ (VGG-based), and the CNN (VGG-based) on spectrograms. Ent.(3) represents an ablation study [97] with respect to Entries (1)-(2), that examines the performance of the model, whenever the removal of the Wav2Vec2 component occurs. It is important to note that Entries (1)-(2) are based on the Wav2Vec2 model which is based on self-supervised learning, while in Ent.(3) is only exploited a CNN-based model, which corresponds to supervised learning.

Entry #	Model	Accuracy	Precision	Recall	F1-Score
(1)	ASR-960H-Wav2BC+	0.90	0.85	0.82	0.84
(2)	Wav2BC+	0.92	0.87	0.89	0.88
(3)	CNN on top of Spectrograms	0.93	0.83	0.91	0.87

Detection in general, have gained great accuracy metrics by using neural networks. However, determining the uncertainty of a specific prediction is still a difficult problem. Predictive uncertainty ability that is well-calibrated, is crucial since it can be used in a variety of AI-based applications.

Neural networks employing the Softmax (Eq.(4)) activation layer that is exploited for AD problems as in this work, are known to produce results that are relative to the training and test sets distribution. Yet, whenever it is possible, an effective AI framework has to be able to generalize in front of Out Of Distribution (OOD) [98], [99] cases, by flagging the ones that are beyond their capacity, as well as request human intervention. In the world of Anomaly Detection, the concept of OOD can be manifested in problems such as binary classification, or even one-class classifier [100]. One of the acceptable approaches to transforming IC into an OOD detection problem is adding an ‘unknown’ class to a classification model. However, this procedure requires apriori tagged OOD data for training, which is an unbound amount of data in theory - a difficult problem whenever the dataset to train is (i) limited, and (ii) bounded by the data collection process time-frame, i.e. the time and conditions of the data collection. Thus, when designing an architecture for a classification problem, one of the penetration-test that should be considered before detecting anomalies has to test OOD cases that might have been recorded in different time-frames and conditions.

As such, and in order to prove the robustness of our Transformer-based approach, we have recorded additional dataset by using the drone, and the same recording set, except for a different environment from the one described in Section III. That is, the new dataset has been recorded when musical songs are being played from a microphone, very close to the drone whenever it flies. Clearly, it is an OOD situation, since the initial dataset did not consider such a scenario at all. These audio recordings contain additional 300 test samples of size 1-seconds, such that 11% of the test-samples are representing the *Anomaly* class and 89% of the audio recordings are considered as *Normal*. The ratio between the Anomaly class samples and the Normal class samples is approximately 1:10, which simulates the real-world AD problem where Anomalies appear rarely. Finally, both the Transformer-based approach that was suggested in this paper, as well as the CNN-only-based one (VGG-16) were tested, by the same data pre-processing and inference processes as presented in Sections III and IV-D, with a Softmax threshold of 0.5, for the OOD computation.

Both the models yield slightly lower results when tested over the new samples of the experiment, as a result of the

samples being recorded in a new and different environment than the ones on which the models were trained on. One can note from Table 5 that the Transformer-based model yields better results compared to the CNN-based model. These results support our thesis and prove that the Transformer based model is more robust and accurate than the CNN-based model.

H. REAL-TIME & EMBEDDED EXPERIMENT

To determine the feasibility of the proposed model in real-time scenarios, an inference of the model was deployed on a Raspberry Pi single-board computer. As aforementioned (in Section IV), an earbud was placed on top of the Tello quadrotor. The main idea of this experiment is to test the Wav2Vec2 model capabilities in real-time mode, on mini-computers that are equipped with basic hardware. An earbud was connected via Bluetooth to the Raspberry Pi, and thus it generated real-time audio samples. Next, each audio sample of length 1-second has been converted into a spectrogram, and fed to the input layer of the Wav2Vec2 model, to get a classification of the audio sample as an *Anomaly* or as a *Normal* sample.

The entire experiment consisted of running the feed-forward function of the Wav2Vec2 and the CNN models, for exactly 5 minutes were recorded in real-time and generated 300 audio samples online. Later, and to test the real-time results, these audio samples were manually tagged sample-by-sample, i.e. second-by-second, such that each audio sample was tagged either as an *Anomaly* or as a *Normal* sample. These real-time audio samples contain exactly 300 test samples of length 1-seconds, such that 33 of them are of class *Anomaly*, and the remaining 267 are of class *Normal*. Again, we encounter the anomalous situation, in which the ratio between the Anomaly samples and the Normal ones is again $\approx 1:10$.

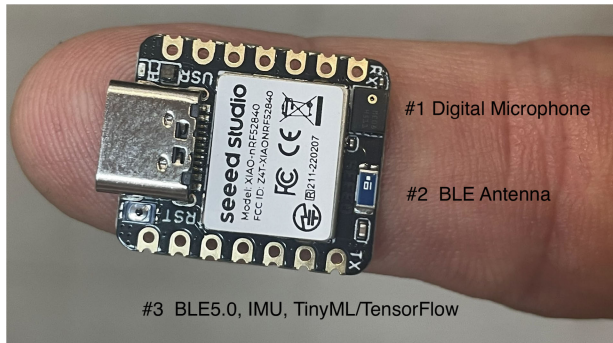
The duration of the whole real-time experiment (without any optimizations) was 60 seconds; i.e., 0.2 seconds on average per one-second audio sample. That is, after recording each audio sample of length 1-second as a WAV file, it took 0.2 seconds on average to (i) turn the audio samples into spectrograms; (ii) turn the spectrograms into an input-matrix to feed the input layer of the Wav2Vec2 model; and (iii) apply the feed-forward function of both the Wav2Vec2 model and the CNN models, and get a classification for the audio-sample (*Anomaly*, or *Normal*). In order to allow a fully real-time solution, the overall processing time should be lower than the sampling time. The major runtime component in the suggested system is the use of Transformers.

TABLE 5. Result for the OOD experiment.

Entry #	Model	Accuracy	Precision	Recall	F1-Score
(1)	Wav2BC+	0.90	0.77	0.84	0.80
(2)	CNN on top of Spectrograms	0.78	0.61	0.68	0.63

TABLE 6. Results of the real-time & embedded experiment on both Raspberry Pi device and mobile device.

Entry	Model	Platform	RAM Weight	Accuracy	Precision	Recall	F1-Score	Avg. Audio-Sample Processing
(1)	Wav2BC+	PC	360MB	0.91	0.924	0.91	0.915	0.2 seconds
(2)	Wav2BC+	Mobile	80MB	0.86	0.86	0.86	0.86	0.2 seconds

**FIGURE 16.** Seed Studio XIAO nRF52840 Sense: a $\approx 2\text{cm} \times 2\text{cm}$, and less than 2 grams micro-controller, equipped with an IMU, a digital microphone, and a Bluetooth 5.0 communication module. Such a micro-controller can be used to run TinyML or TensorFlow Lite, and thus is a suitable candidate for implementing our sound analysis method.

In order to use the Wav2Vec2-Base pre-trained transformer in a mobile environment, performing quantization on it might be a necessary step. Thus, the model has been converted to a qint8 dynamic (i.e. weights-only) quantized model, a common solution for heavy models requiring significant RAM allocation - which is inapplicable for mobile devices. This operation shrunk the Wav2Vec2-Base model into a lighter version, from 360MB to 80MB, making it a more tailored model for edge-mobile usages. Next, we tested our approach on two embedded platforms, designed for real-time scenarios. As can be seen in 6, Ent.(1) presents the real-time experiment where the Wav2Vec2 model in the Wav2BC+ framework is the Wav2Vec2-Base transformer, on top of a CPU-based computation. Since such a model is too heavy for mobile devices, we repeated the same experiment as presented in this section (Ent.(2)), on top of the embedded device. As the lighter and quantized Wav2Vec2 model is smaller than the Base version regarding RAM allocation, we could expect a slight degradation in the accuracy metrics, while the average processing time per audio sample remains the same, thus making it applicable for mobile devices as well.

Finally, one can note from Table 6 that the Transformer-based model preserved the level of results from the last two experiments (Sections V-F and V-G). Moreover, the Wav2Vec2 method is suitable for real-time edge computing platforms such as Raspberry Pi and can be adapted to run on even smaller System on Chip (SoC) platforms.

VI. CONCLUSION AND FUTURE WORK

In this paper, we presented Wav2BC+, a framework to detect anomalies in sound waves emitted from a UAV using deep-learning methods, and focused on the benefits of transfer-learning to construct an improved model for the anomaly detection problem in UAVs. We have shown that by using a Transformer based model, followed by a CNN, one could achieve better results in detecting anomalies in UAVs using sound waves, compared to the well-known VGG (CNN-based) over spectrogram approach. That is, we have developed a real-time approach that outperformed two baselines, so that our suggested compressed version of the well-known VGG-16 framework, is extremely smaller in terms of the number of parameters in the neural network, and is capable of yielding high accuracy in anomaly detection in UAVs as well. In terms of performance metrics, the Wav2BC+ maintains high accuracy metrics in all of the experiments, and reduces the number of parameters of the well-known VGG from 138M, into a shrunk version of the VGG with only 3.6M parameters. Moreover, we employed our technique over an extremely small dataset, which is a problem on its own due to a lack of information. In addition, the compressed version for CNN suggested in our approach, enables us to apply it on top of tiny devices that cannot cope with high-consuming applications. For industrial purposes, one can assimilate our transfer-learning framework on top of any kind of drone or UAV that is able to run such architectures.

Even though obtaining better results, the AD problem is still not entirely addressed. Hence, one possible direction for future research would be addressing the AD problem from an external sound source as well; i.e., creating a dataset of sound waves emitted by a UAV from different distances and not only from its top. Another possible direction for future research would be to classify anomalies per type by training the model with a larger dataset containing different examples of anomalies, labeled by their different types. From the architectural standpoint, another future work could be the construction of an actual acoustic sensor and analyzers for drones. Such devices may be implemented using a tiny micro-controller capable of running TinyML (as shown in Figure 16). The availability of such devices may help the community to construct a comprehensive dataset for a wide range of UAVs [18].

In addition, in terms of the neural networks' performance, more sophisticated deep-learning techniques can be of great utility, especially for the real-time scenario. Among such techniques, one can find depth-wise separable convolution [101], atrous spatial pyramid pooling [102], and attention mechanisms [103], [104], as well as improvement in the transformers themselves.

REFERENCES

- [1] N. I. Jasim, H. Kasim, and M. A. Mahmoud, "Towards the adoption of drones in foodservice industry: A generic model development," *J. Theor. Appl. Inf. Technol.*, vol. 99, no. 21, pp. 1–14, 2021.
- [2] B. Alkouz, B. Shahzaad, and A. Bouguettaya, "Service-based drone delivery," in *Proc. IEEE 7th Int. Conf. Collaboration Internet Comput. (CIC)*, Dec. 2021, pp. 68–76.
- [3] S. Wulfovich, H. Rivas, and P. Matabuena, "Drones in healthcare," in *Digital Health: Scaling Healthcare to the World*, 2018, pp. 159–168.
- [4] I. Mademlis, V. Mygdalis, N. Nikolaidis, M. Montagnuolo, F. Negro, A. Messina, and I. Pitas, "High-level multiple-UAV cinematography tools for covering outdoor events," *IEEE Trans. Broadcast.*, vol. 65, no. 3, pp. 627–635, Sep. 2019.
- [5] Y. Lan and S. Chen, "Current status and trends of plant protection UAV and its spraying technology in China," *Int. J. Precis. Agricult. Aviation*, vol. 1, no. 1, pp. 1–9, 2018.
- [6] S. Khan, M. Tufail, M. T. Khan, Z. A. Khan, J. Iqbal, and A. Wasim, "Real-time recognition of spraying area for UAV sprayers using a deep learning approach," *PLoS ONE*, vol. 16, no. 4, Apr. 2021, Art. no. e0249436.
- [7] B. T. Clough, "Unmanned aerial vehicles: Autonomous control challenges, a researcher's perspective," *J. Aerosp. Computing, Inf., Commun.*, vol. 2, no. 8, pp. 327–347, Aug. 2005.
- [8] C. Titouna, F. Nait-Abdesselam, and H. Moungla, "An online anomaly detection approach for unmanned aerial vehicles," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2020, pp. 469–474.
- [9] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyükköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 9, pp. 731–747, Sep. 2018.
- [10] Y.-J. Cha, W. Choi, and O. Büyükköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 5, pp. 361–378, Feb. 2017.
- [11] T. Zhou and R. Fattah, "Tonal noise acoustic interaction characteristics of multi-rotor vehicles," in *Proc. 23rd AIAA/CEAS Aeroacoustics Conf.*, Jun. 2017, p. 4054.
- [12] X. Ma, J. Wu, S. Xue, J. Yang, C. Zhou, Q. Z. Sheng, H. Xiong, and L. Akoglu, "A comprehensive survey on graph anomaly detection with deep learning," *IEEE Trans. Knowl. Data Eng.*, early access, Oct. 8, 2021, doi: 10.1109/TKDE.2021.3118815.
- [13] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, Mar. 2021.
- [14] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*.
- [15] W. Zhu, L. Xie, J. Han, and X. Guo, "The application of deep learning in cancer prognosis prediction," *Cancers*, vol. 12, no. 3, p. 603, Mar. 2020.
- [16] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 778–782, May 2017.
- [17] J. Islam and Y. Zhang, "GAN-based synthetic brain PET image generation," *Brain Informat.*, vol. 7, no. 1, pp. 1–12, Dec. 2020.
- [18] C. Cavallaro and E. Ronchieri, "Identifying anomaly detection patterns from log files: A dynamic approach," in *Proc. 21st Int. Conf. Comput. Sci. Appl. Cagliari*, Italy: Springer, Sep. 2021, pp. 517–532.
- [19] T. Fernando, H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, "Deep learning for medical anomaly detection—A survey," *ACM Comput. Surveys*, vol. 54, no. 7, pp. 1–37, 2021.
- [20] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015, *arXiv:1511.08458*.
- [21] J. J. Huang and J. J. A. Leanos, "AcNet: Efficient end-to-end audio classification CNN," 2018, *arXiv:1811.06669*.
- [22] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly detection and diagnosis from system logs through deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1285–1298.
- [23] J. Galvan, A. Raja, Y. Li, and J. Yuan, "Sensor data-driven UAV anomaly detection using deep learning approach," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2021, pp. 589–594.
- [24] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Comput.*, vol. 22, pp. 949–961, Jan. 2019.
- [25] Z. Cen, H. Noura, and Y. A. Younes, "Robust fault estimation on a real quadrotor UAV using optimized adaptive Thau observer," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, May 2013, pp. 550–556.
- [26] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, H. Lin, Eds. Red Hook, NY, USA: Curran Associates, 2020, pp. 12449–12460.
- [27] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, vol. 3, pp. 111–132, 2022.
- [28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. U. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 1–11.
- [29] P. Jin, L. Mou, G.-S. Xia, and X. X. Zhu, "Anomaly detection in aerial videos with transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, 2022, Art. no. 5628213.
- [30] S. Huang, Y. Liu, C. Fung, R. He, Y. Zhao, H. Yang, and Z. Luan, "HitAnomaly: Hierarchical transformers for anomaly detection in system log," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 4, pp. 2064–2076, Dec. 2020.
- [31] W. H. L. Pinaya, P.-D. Tudosiu, R. Gray, G. Rees, P. Nachev, S. Ourselin, and M. J. Cardoso, "Unsupervised brain anomaly detection and segmentation with transformers," 2021, *arXiv:2102.11650*.
- [32] C. Yi, J. Wang, N. Cheng, S. Zhou, and B. Xu, "Applying Wav2vec2.0 to speech recognition in various low-resource languages," 2020, *arXiv:2012.12121*.
- [33] C. Yi, J. Wang, N. Cheng, S. Zhou, and B. Xu, "Transfer ability of monolingual Wav2vec2.0 for low-resource speech recognition," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–6.
- [34] E. Khalastchi, G. A. Kaminka, M. Kalech, and R. Lin, "Online anomaly detection in unmanned vehicles," in *Proc. 10th Int. Conf. Auton. Agents Multiagent Syst.*, vol. 1, Sep. 2011, pp. 115–122.
- [35] Y. He, Y. Peng, S. Wang, and D. Liu, "ADMOST: UAV flight data anomaly detection and mitigation via online subspace tracking," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 4, pp. 1035–1044, Apr. 2019.
- [36] B. Wang, Y. Chen, D. Liu, and X. Peng, "An embedded intelligent system for on-line anomaly detection of unmanned aerial vehicle," *J. Intell. Fuzzy Syst.*, vol. 34, no. 6, pp. 3535–3545, Jan. 2018.
- [37] E. L. Diget, A. Hasan, and P. Manoonpong, "Fault-tolerant model predictive control for multirotor UAVs," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2022, pp. 4305–4310.
- [38] N. Kazantzis and C. Kravaris, "Nonlinear observer design using Lyapunov's auxiliary theorem," *Syst. Control Lett.*, vol. 34, no. 5, pp. 241–247, 1998.
- [39] B. D. Gallas and H. H. Barrett, "Validating the use of channels to estimate the ideal linear observer," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 20, no. 9, pp. 1725–1738, 2003.
- [40] Z. Cen and H. Noura, "An adaptive Thau observer for estimating the time-varying LOE fault of quadrotor actuators," in *Proc. Conf. Control Fault-Tolerant Syst. (SysTol)*, Oct. 2013, pp. 468–473.
- [41] C. Zhaohui, H. Noura, T. B. Susilo, and Y. A. Younes, "Engineering implementation on fault diagnosis for quadrotors based on nonlinear observer," in *Proc. 25th Chin. Control Decis. Conf. (CCDC)*, May 2013, pp. 2971–2975.
- [42] F. Sharifi, M. Mirzaei, B. W. Gordon, and Y. Zhang, "Fault tolerant control of a quadrotor UAV using sliding mode control," in *Proc. Conf. Control Fault-Tolerant Syst. (SysTol)*, Oct. 2010, pp. 239–244.
- [43] A. S. Candido, R. K. Harrop Galvao, and T. Yoneyama, "Actuator fault diagnosis and control of a quadrotor," in *Proc. 12th IEEE Int. Conf. Ind. Informat. (INDIN)*, Jul. 2014, pp. 310–315.

- [44] M. Moghadam and F. Caliskan, "Actuator and sensor fault detection and diagnosis of quadrotor based on two-stage Kalman filter," in *Proc. 5th Austral. Control Conf. (AUCC)*, 2015, pp. 182–187.
- [45] M. H. M. Ghazali and W. Rahiman, "Vibration-based fault detection in drone using artificial intelligence," *IEEE Sensors J.*, vol. 22, no. 9, pp. 8439–8448, May 2022.
- [46] X. Lv and H. Ni, "Smart fault detection and monitoring of power line by drones," in *Proc. 4th Int. Conf. Electron. Inf. Technol. Comput. Eng.*, Nov. 2020, pp. 501–505.
- [47] H. Ahn, "Deep learning based anomaly detection for a vehicle in swarm drone system," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Sep. 2020, pp. 557–561.
- [48] J. Fu, C. Sun, Z. Yu, and L. Liu, "A hybrid CNN-LSTM model based actuator fault diagnosis for six-rotor UAVs," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2019, pp. 410–414.
- [49] W. De Mulder, S. Bethard, and M.-F. Moens, "A survey on the application of recurrent neural networks to statistical language modeling," *Comput. Speech Lang.*, vol. 30, no. 1, pp. 61–98, 2015.
- [50] P. Radanliev and D. de Roure, "Review of algorithms for artificial intelligence on low memory devices," *IEEE Access*, vol. 9, pp. 109986–109993, 2021.
- [51] P. Radanliev and D. De Roure, "New and emerging forms of data and technologies: Literature and bibliometric review," *Multimedia Tools Appl.*, vol. 82, no. 2, pp. 2887–2911, Jan. 2023.
- [52] P. Radanliev and D. De Roure, "Advancing the cybersecurity of the health-care system with self-optimising and self-adaptative artificial intelligence (part 2)," *Health Technol.*, vol. 12, no. 5, pp. 923–929, Sep. 2022.
- [53] O. Bektash and A. L. Cour-Harbo, "Vibration analysis for anomaly detection in unmanned aircraft," in *Proc. Annu. Conf. Prognostics Health Management Soc.*, 2020, pp. 1–11.
- [54] M. H. M. Ghazali and W. Rahiman, "An investigation of the reliability of different types of sensors in the real-time vibration-based anomaly inspection in drone," *Sensors*, vol. 22, no. 16, p. 6015, Aug. 2022.
- [55] F. Pourpanah, B. Zhang, R. Ma, and Q. Hao, "Anomaly detection and condition monitoring of UAV motors and propellers," in *Proc. IEEE SENSORS*, Oct. 2018, pp. 1–4.
- [56] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Netw.*, vol. 4, no. 6, pp. 759–771, 1991.
- [57] A. Altinors, F. Yol, and O. Yaman, "A sound based method for fault detection with statistical feature extraction in UAV motors," *Appl. Acoust.*, vol. 183, Dec. 2021, Art. no. 108325.
- [58] A. Bondyra, M. Kołodziejczak, R. Kulikowski, and W. Giernacki, "An acoustic fault detection and isolation system for multirotor UAV," *Energies*, vol. 15, no. 11, p. 3955, May 2022.
- [59] G. Iannace, G. Ciaburro, and A. Trematerra, "Fault diagnosis for UAV blades using artificial neural network," *Robotics*, vol. 8, no. 3, p. 59, Jul. 2019.
- [60] W. Liu, Z. Chen, and M. Zheng, "An audio-based fault diagnosis method for quadrotors using convolutional neural network and transfer learning," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2020, pp. 1367–1372.
- [61] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN model-based approach in classification," in *Proc. OTM Confederated Int. Conf. Move Meaningful Internet Syst.* Cham, Switzerland: Springer, 2003, pp. 986–996.
- [62] T. Evgeniou and M. Pontil, "Support vector machines: Theory and applications," in *Machine Learning and Its Applications: Advanced Lectures*. Springer, 2001, pp. 249–257.
- [63] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
- [64] P. Beckmann, M. Kehler, and M. Cernak, "Word-level embeddings for cross-task transfer learning in speech processing," 2019, *arXiv:1910.09909*.
- [65] M. Heck, M. Suzuki, T. Fukuda, G. Kurata, and S. Nakamura, "Ensembles of multi-scale VGG acoustic models," in *Proc. Interspeech*, Aug. 2017, pp. 1616–1620.
- [66] T. Kaur and T. K. Gandhi, "Automated brain image classification based on VGG-16 and transfer learning," in *Proc. Int. Conf. Inf. Technol. (ICIT)*, Dec. 2019, pp. 94–98.
- [67] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.
- [68] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
- [69] O. Mohamed and S. A. Aly, "Arabic speech emotion recognition employing Wav2vec2.0 and Hubert based on BAVED dataset," 2021, *arXiv:2110.04425*.
- [70] Y. Getman, R. Al-Ghezi, K. Voskoboinik, T. Grósz, M. Kurimo, G. Salvi, T. Svendsen, and S. Strömbergsson, "Wav2vec2-based speech rating system for children with speech sound disorder," in *Proc. Interspeech*, Sep. 2022.
- [71] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
- [72] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Unsupervised representation learning of structured radio communication signals," in *Proc. 1st Int. Workshop Sens., Process. Learn. Intell. Mach. (SPLINE)*, Jul. 2016, pp. 1–5.
- [73] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4L: Self-supervised semi-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1476–1485.
- [74] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.
- [75] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," 2019, *arXiv:1909.11942*.
- [76] Y.-J. Cha, A. Mostafavi, and S. S. Benipal, "DNoiseNet: Deep learning-based feedback active noise control in various noisy environments," *Eng. Appl. Artif. Intell.*, vol. 121, May 2023, Art. no. 105971.
- [77] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*.
- [78] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
- [79] Z. Fan, M. Li, S. Zhou, and B. Xu, "Exploring wav2vec 2.0 on speaker verification and language identification," 2020, *arXiv:2012.06185*.
- [80] A. T. Liu, S.-W. Yang, P.-H. Chi, P.-C. Hsu, and H.-Y. Lee, "Mockingjay: Unsupervised speech representation with deep bidirectional transformer encoders," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6419–6423.
- [81] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3828–3836.
- [82] A. Fred Agarap, "Deep learning using rectified linear units (ReLU)," 2018, *arXiv:1803.08375*.
- [83] R. Ribani and M. Marengoni, "A survey of transfer learning for convolutional neural networks," in *Proc. 32nd SIBGRAPI Conf. Graph., Patterns Images Tuts. (SIBGRAPI-T)*, Oct. 2019, pp. 47–57.
- [84] A. Kensert, P. J. Harrison, and O. Spjuth, "Transfer learning with deep convolutional neural networks for classifying cellular morphological changes," *SLAS Discovery, Advancing Life Sci. RD*, vol. 24, no. 4, pp. 466–475, 2019.
- [85] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in NLP," 2019, *arXiv:1906.02243*.
- [86] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *Sci. China Technol. Sci.*, vol. 63, no. 10, pp. 1872–1897, 2020.
- [87] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Jan. 2009.
- [88] C. Seiffert, T. M. Khoshgoftaar, J. V. Hulse, and A. Napolitano, "Building useful models from imbalanced data with sampling and boosting," in *Proc. FLAIRS Conf.*, 2008, pp. 306–311.
- [89] U. Ruby and V. Yendapalli, "Binary cross entropy with deep learning technique for image classification," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 4, pp. 5393–5397, Aug. 2020.
- [90] M. Martinez and R. Stiefelhagen, "Taming the cross entropy loss," in *Proc. German Conf. Pattern Recognit.* Cham, Switzerland: Springer, 2018, pp. 628–637.
- [91] B. E. D. Kingsbury, N. Morgan, and S. Greenberg, "Robust speech recognition using the modulation spectrogram," *Speech Commun.*, vol. 25, nos. 1–3, pp. 117–132, Aug. 1998.
- [92] R. Lakshmanan, A. Pichler, and D. Potts, "Nonequispaced fast Fourier transform boost for the Sinkhorn algorithm," 2022, *arXiv:2201.07524*.

- [93] P. Kahlig, "Some aspects of Julius von Hann's contribution to modern climatology," in *Interactions Between Global Climate Subsystems: The Legacy of Hann*, vol. 75, 1993, pp. 1–7.
- [94] T. He, X. Tan, Y. Xia, D. He, T. Qin, Z. Chen, and T.-Y. Liu, "Layer-wise coordination between encoder and decoder for neural machine translation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.
- [95] L. A. Jeni, J. F. Cohn, and F. De La Torre, "Facing imbalanced data-recommendations for the use of performance metrics," in *Proc. Humaine Assoc. Conf. Affect. Comput. Intell. Interact.*, Sep. 2013, pp. 245–251.
- [96] W. F. Ferger, "The nature and use of the harmonic mean," *J. Amer. Stat. Assoc.*, vol. 26, no. 173, pp. 36–40, Mar. 1931.
- [97] S. Montaha, S. Azam, A. K. M. R. H. Rafid, M. Z. Hasan, A. Karim, and A. Islam, "TimeDistributed-CNN-LSTM: A hybrid approach combining CNN and LSTM to classify brain tumor on 3D MRI scans performing ablation study," *IEEE Access*, vol. 10, pp. 60039–60059, 2022.
- [98] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.
- [99] A. Manolache, F. Brad, and E. Burceanu, "DATE: Detecting anomalies in text via self-supervision of transformers," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2021, pp. 267–277.
- [100] Z. Wang and Y.-J. Cha, "Unsupervised deep learning approach using a deep auto-encoder with a one-class support vector machine to detect damage," *Struct. Health Monitor.*, vol. 20, pp. 406–425, Jan. 2021.
- [101] R. Zhang, F. Zhu, J. Liu, and G. Liu, "Depth-wise separable convolutions and multi-level pooling for an efficient spatial CNN-based steganalysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1138–1150, 2020.
- [102] Y. Wang, B. Liang, M. Ding, and J. Li, "Dense semantic labeling with atrous spatial pyramid pooling and decoder for high-resolution remote sensing imagery," *Remote Sens.*, vol. 11, no. 1, p. 20, 2019.
- [103] D. H. Kang and Y.-J. Cha, "Efficient attention-based deep encoder and decoder for automatic crack segmentation," *Structural Health Monitor.*, vol. 21, no. 5, pp. 2190–2205, Sep. 2022.
- [104] W. Choi and Y.-J. Cha, "SDDNet: Real-time crack segmentation," *IEEE Trans. Ind. Electron.*, vol. 67, no. 9, pp. 8016–8025, Sep. 2020.



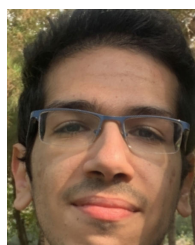
ALON BARAK is currently pursuing the B.Sc. degree in computer science and mathematics with the School of Computer Science, Ariel University, Israel. Under the supervision of Prof. Boaz Ben-Moshe and Dr. Or Haim Anidjar, he is a Lab Member of the Kinematics and Computational Geometry (K&CG) Laboratory, Ariel University. His research interests include machine-learning and deep-learning algorithms, especially in acoustics and computer vision.



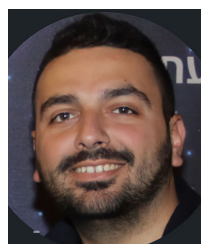
BOAZ BEN-MOSHE received the Ph.D. degree in computer science from Ben Gurion University, in 2004. From 2004 to 2005, he was a Post-doctoral Researcher with SFU (Vancouver). Since 2006, he has been a Faculty Member with the Department of Computer Science, Ariel University. In 2008, he and Prof. Nir Shvalb founded the Kinematics and Computational Geometry Laboratory. He is currently the Head of the Aerospace and Nano Satellite Research Center (the owner of the SATLLA pico-satellites series). He is also the Founder of Momentis Surgical (<https://www.momentissurgical.com>) and SimHawk (<https://www.simhawk.net>). His research interests include computational geometry, navigation algorithms, new space, autonomous, and bioinspired robotics. Since 2018, he has been the Chair of the Computer Science Department, Ariel University.



EYAL HAGAI is currently pursuing the B.Sc. degree in computer science and mathematics with the School of Computer Science, Ariel University, Israel. Under the supervision of Prof. Boaz Ben-Moshe and Dr. Or Haim Anidjar, he is a Lab Member of the Kinematics and Computational Geometry (K&CG) Laboratory, Ariel University. His research interests include machine-learning and deep-learning algorithms, especially in acoustics and computer vision.



SAHAR TUVYAHU is currently pursuing the B.Sc. degree in computer science and mathematics with the School of Computer Science, Ariel University, Israel. Under the supervision of Prof. Boaz Ben-Moshe and Dr. Or Haim Anidjar, he is a Lab Member of the Kinematics and Computational Geometry (K&CG) Laboratory, Ariel University. His research interests include machine-learning and deep-learning algorithms, especially in acoustics data analysis and visualization.



OR HAIM ANIDJAR received the B.Sc. degree in computer science from Bar Illan University, Israel, and the M.Sc. and Ph.D. degrees in computer science and applied mathematics from Ariel University, Israel, in 2015, 2019, and 2021, respectively. He was a Lab Member of the Kinematics and Computational Geometry (K&CG) Laboratory, the Ariel Cyber Innovation Center (ACIC), the Data Science and Artificial Intelligence Research Center (DSAIRC), and the Architectural AI Research Laboratory (AAIRL), Ariel University. In 2022, he was a Postdoctoral Fellow with the ACIC, Ariel University. Formerly, he was the Founder, the CEO, and the Chief Data Scientist of Libonea.AI. He is currently a Lecturer and a Senior Faculty Member with the School of Computer Science, Ariel University. His research interests include using deep-learning methods at the intersection between natural language processing (NLP) and speech recognition (SR), especially in multidisciplinary methods, such as language models, speaker change detection and diarization, speaker identification, and anomaly detection in acoustic and textual datasets.

...