Model-Based Data Augmentation Applied to Deep Learning Networks for Classification of Micro-Doppler Signatures Using FMCW Radar

Neda Rojhani, *Member, IEEE*, Marco Passafiume[®], *Member, IEEE*, Mehdi Sadeghibakhi, Giovanni Collodi[®], *Member, IEEE*, and Alessandro Cidronali[®], *Senior Member, IEEE*

Abstract-Deep neural networks (DNNs) have become a relevant subject in the classification of radio frequency signals and remote sensing data. A primary challenge is a tradeoff between obtaining data that are suitable for DNN training and the effort that making experimental measurements requires. Hence, the quality and quantity of data used for the training and testing of models are crucial for effective classifier development. The training dataset should cover a wide range of cases that synthesize the actual scenarios being classified. This work proposes a novel data augmentation method based on a deterministic model to generate a simulated dataset of radar micro-Doppler signatures suitable for unmanned aerial vehicle (UAV) target classification, without requiring measurement data. It is shown that the DNN trained using the properly generated model-based data offers improved classification accuracy performance. Results are presented for a two-class classification of the number of UAV motors using a 77-GHz frequency-modulated continuouswave (FMCW) automotive radar system. The effectiveness of the proposed methodology is proven: a classification accuracy of 78.68% is achieved using a convolutional neural network (CNN) trained using the synthetic dataset, while an accuracy of 66.18% is achieved by using a typical signal processing data augmentation method on a limited measured dataset.

Index Terms—Data augmentation, deep convolutional neural network (CNN), micro-Doppler signatures, radar classification, remote sensing imagery.

I. INTRODUCTION

DESPITE the widespread attention given to unmanned aerial vehicles (UAVs) in various civil and commercial [1],

Manuscript received 19 October 2022; accepted 17 November 2022. Date of publication 2 January 2023; date of current version 5 May 2023. This work was supported in part by the National Italian Research Project (PRIN-2015) "Millimetre-Wave GaN-Based Radar Systems for Early Detection of Micro Unmanned Aerial Vehicles" under Grant 2015CPC2MA. (*Corresponding author: Alessandro Cidronali.*)

Neda Rojhani is with the Department of Electrical and Computer Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada (e-mail: neda.rojhani@ucalgary.ca).

Marco Passafiume, Giovanni Collodi, and Alessandro Cidronali are with the Department of Information Engineering, University of Florence, 50139 Florence, Italy (e-mail: marco.passafiume@unifi.it; giovanni.collodi@unifi.it; alessandro.cidronali@unifi.it).

Mehdi Sadeghibakhi is with the Department of Computer Engineering, University of Ferdowsi Mashhad, Mashad 9177948974, Iran (e-mail: mehdi.sadeghibakhi@mail.um.ac.ir).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMTT.2023.3231371.

Digital Object Identifier 10.1109/TMTT.2023.3231371

agricultural [2], industrial [3], and defense applications [4], they are undoubtedly a serious threat to the public and to flight safety [5]. Hence, it is necessary to identify drones in order to decrease such risks [6], [7]. However, due to their small size [8], low flight speed, and low flight altitude, UAVs are easily hidden by buildings or misclassified as birds [9]; thus, the identification and classification of drones is a challenging task [10].

One of the foremost features of a UAV's radar signal is the micro-Doppler signature [11]. Contrary to a stationary object, objects in motion generate modulated Doppler components referred to as the micro-Doppler signature, which is provided as additional components of the Doppler signature of the drone's fuselage. Furthermore, micro-Doppler signatures rely on the number of motors, motor speed, and orientation of the drone; therefore, by analyzing the micro-Doppler signatures, information pertaining to the drone can be obtained [12], [13].

In the present literature about radar target classification using micro-Doppler signatures, it has been demonstrated that these techniques are capable of providing a high classification accuracy. The most relevant are based on empirical mode decomposition [14], feature extraction [15], and log-Gabor filters [16], as well as singular value decomposition [17]. Nevertheless, the major drawback of the standard approaches is their low scalability.

Recently, deep neural networks (DNNs) have gained much attention in the radar community as a means of target classification [18], [19], [20], target detection [21], static object recognition [22], and automatic target recognition [23]. Furthermore, DNNs have also been utilized in micro-Dopplerbased systems for the classification of human activities [24], hand gestures [25], and drones [26]. It should be mentioned that although the use of DNNs, such as convolutional neural networks (CNNs), for target recognition or classification was presented as having a satisfactory scalability and a high classification accuracy, it requires big databases, and one of its fundamental challenges is that the available dataset might lead to results that are quite limited.

In this study, we introduce a model-based data augmentation method to produce a simulated dataset of radar micro-Doppler signatures that is a suitable training database for CNNs.

Data augmentation is one of the most efficient techniques for preventing model overfitting during the training of a network [27]. In data augmentation, new synthetic data samples are produced from existing data by combining different data vectors and nonlinear operations [28]. Despite the effectiveness of random data augmentation [28], the training of the network requires a large number of epochs. Furthermore, the final training relies on totally hidden features, which could be related to measurement nonidealities rather than to effective class characteristics. An example of this issue is the deep learning (DL) network aging phenomenon [29] in which DL network-based classifiers trained and continuously retrained on augmented datasets show a decrease in the classification accuracy as the training time increases. Such an effect is due to the variation of hidden features in the measurement scenario that are wrongly conisdered as actual class features during training. This causes the negation of the training memory, thus causing the DL network to be unable to make accurate predictions. By replacing the physical information source with a model dependent only on actual target-class properties, it is possible to build extensive datasets that express effective features, removing the dependence on real measurement chain properties.

This work aims to prove the effectiveness of data augmentation based on a UAV micro-Doppler model, relying only on ideal parameters derived by the deterministic description of the physical scenario under consideration. We demonstrate that this dataset augmentation method, combined with CNN classification, is particularly suited for UAV classification, where this latter considers radar images acquired by well-established methods (i.e., noise reduction, focusing, and clutter removing), and because these do not represent a critical aspect of this study, they are not treated in this work.

According to the literature [26], the research on machine learning (ML)-based UAV classification by radar considers several objectives: UAV presence, UAV versus bird recognition [30], multi-UAV presence [31], and UAV characterization in terms of number of rotors classification [32]. In this work, we selected this latter classification objective as a significant case study, although it could in principle be extended to other classes and other scenarios.

This article is organized as follows. In Section II, we discuss a comprehensive treatment of the proposed model-augmenting method, while a description of the CNN is provided in Section III. Finally, in Section IV, we provide a detailed independent validation of the approach by considering two different classes of UAVs in a controllable and repeatable environment.

II. MODEL-BASED AUGMENTATION METHODOLOGY

This section discusses the data augmentation method, which is based on a mechanical model of the micro-Doppler signatures that can be applied to the classification of UAVs.

A. Preliminary Concepts

It is well known that in automatic classification, a dataset for object classification should include observations of a large number of samples for each targeted class. Automatic classification relies on a classifier, which is defined as an unknown nonlinear function that takes as an argument a vector of sample points related to the features and outputs the predicted class; such a function is determined through an optimization process in the classifier training phase. On this basis, the DNN learns about meaningful features during training by identifying the similarities between measurements related to the same class. In order to avoid an ill-posed numerical problem that would lead to overfitting issues, a comprehensive set of linearly independent equations is needed [33]. However, when one is dealing with a high-dimensional feature space, the latter can only be provided when they are associated with an exhaustive training set, so a huge training dataset must be provided.

A measurement-based training dataset can benefit from data augmentation, leading to exhaustive classifier training while reducing the amount of experimental effort. Dataset augmentation can be performed using two different approaches. Augmentation in the data space is achieved by applying random transformations to input datasets to produce new samples. It is a straightforward process but might expose one to the risk of erasing meaningful features or enhancing features that are not related to the targeted classes. Augmentation in the feature space is achieved by performing raw feature identification and then combining only measurements that appear to be similar, i.e., near each other in the feature space [34].

Data augmentation with analytical models can produce a meaningful training dataset while reducing significantly the effort required for experimental dataset acquisition. In a typical scenario, as proposed here or in [35], this approach produces a multitude of synthetic data vectors using the mathematical generator function

$$\mathbf{f}: \overline{\mathbf{x}} \to \overline{\mathbf{s}} = \mathbf{f}(\overline{\mathbf{x}}) \text{ with } \overline{\mathbf{x}} \in \mathbf{C}^{1 \times N_{\text{params}}}, \ \overline{\mathbf{s}} \in \mathbf{C}^{1 \times N_{\text{pts}}}.$$
 (1)

The generator function in (1) is defined as a nonlinear transformation between the $\mathbb{C}^{1 \times N_{params}}$ space of scenario parameter vectors, or rather the vectors comprising all the parameters that describe the physical scenario as following the model definition, and the $\mathbb{C}^{1 \times N_{pts}}$ space of measurement data vectors.

As in [35], the augmentation in the data space for the generic class "A" resembles the generation of synthetic data vectors $\overline{\mathbf{s}_{A''}}(i)_{1 \times N_{pts}}$ with N_{pts} samples according to the following equation:

$$\overline{\mathbf{s}_{\mathbf{A}}}(i) = \mathbf{d}_i \left(\mathbf{f}(\overline{\mathbf{x}_{\mathbf{A}}}) + \overline{\mathbf{n}}_{i,S} \right) + \overline{\mathbf{n}}_{i,M}$$
(2)

where $\overline{\mathbf{n}}_{i,j}$ are the perturbation vectors, $\overline{\mathbf{x}_{\mathbf{A}^{"}}}$ is a set of scenario parameters for class "*A*", and $\mathbf{d}_i(\overline{\mathbf{f}})$ is the nonlinear random transformation function applied to the data vector.

The perturbation vector introduces the effects of nonidealities into scenario $\overline{\mathbf{n}}_{i,S}$ and measurement chain $\overline{\mathbf{n}}_{i,M}$, and both can be written as

$$\overline{\mathbf{n}}_{i,j} = \mathcal{N}_{1 \times N_{\text{pts}}}(0, \sigma_j).$$
(3)

Following a feature-space augmentation approach, synthetic data vector generation for the class "A" data subset is obtained as follows:

$$\overline{\mathbf{s}_{\mathbf{A}}}(i) = \mathbf{f}(\overline{\mathbf{x}_{\mathbf{A}}} + \overline{\mathbf{n}}_{i,P}). \tag{4}$$



Fig. 1. Example of the definition of classes in the feature space.

In (4), as opposed to (2), the dataset variability is given only by the perturbation vector $\overline{\mathbf{n}}_{i,P}$ applied to the vector of scenario parameters. Random transformations or the addition of noise to generated data vectors are expressly avoided, as they could introduce new features that are totally unrelated to the meaningful characteristics of the target. Note that noise added to generated data vectors does not show impulsive auto-correlation because of its limitation in a number of samples. By this, different synthesized vectors could show unexpected correlation due to the added noise term. In (4), the parameter perturbation vector is defined as

$$\overline{\mathbf{n}}_{i,P} = \mathcal{N}_{1 \times N_{\text{params}}}(0, \sigma_P) \tag{5}$$

and it models the variability of the parameter vector $\overline{\mathbf{x}}$ within scenarios related to the same classes. In the case under investigation, this refers to the slight differences in the parameters between different UAVs in the same class.

The amount of noise variance, σ_P , applied to the vector of scenario parameters must be carefully chosen. It must be large enough to effectively model the expected physical scenario variability within a given target class. However, it must also be limited to avoid spreading dataset samples along the feature space, which can lead to misleading class specification.

The concept described above is schematized in Fig. 1 for a simple 2-D space of the scenario parameters x_i and x_j ; it relies on the theory of classifiers based on support vector machines (SVMs) [36]. In this example, target classes are related to specific relations between parameter vector terms. This leads to the definition of straight lines in 2-D spaces (or hyperplanes when dealing with higher dimensional spaces inside the feature space).

SVM classification is achieved by assigning to each sample the class described by the hyperplane with the minimum distance to the target. Assuming that each provided sample is assigned to the correct class, during the training phase, the network coefficients are optimized to achieve a feature-space clustering with a clear separation between classes. Therefore, the training dataset should provide samples with clear distinctions between classes, thus imposing an upper limit on the standard deviation of the parameters σ_P .

B. Model-Based Dataset Augmenter

Passafiume et al. [12] introduced a fully deterministic model for the micro-Doppler signatures of flying UAVs in



Fig. 2. Model-based dataset generation chain.

frequency-modulated continuous-wave (FMCW) radar echo signals (6). Such a model considers the effects of mechanical vibrations on the range distance and thus introduces a varying term due to vibrations into the range argument alongside the Doppler effects on the radar signal. By this model, the micro-Doppler effect is included in the single IF signal acquisition, given the mandatory condition that the period of range fluctuations are way lower than the time window of the single radar observation [12]. The model depends on the target's physical parameters, such as the expected radar cross section (RCS) and the mechanical and vibration frequencies of each UAV engine. In this work, we adopt such a model as a data augmenter function within a feature-space model-based dataset, as discussed in Section II-A.

When the analytical model is applied to a four-rotor UAV (i.e., a quadcopter), the FMCW radar signal facing the quadcopter on boresight is given as

$$s_{\rm IF}(t) = \sum_{k=1}^{N_{\rm eng}} \Gamma_k \cos\left\{\frac{8\pi\,\mu}{c_0} \left[R_0 + \sum_{j=1}^4 A_{kj}\cos(\omega_{kj}t + \phi_{kj})\right]t\right\}$$
(6)

where R_0 is the distance to the quadcopter, Γ_i is the expected RCS related to the specific *k*th engine, and $\overline{\mathbf{A}}_k$, $\overline{\omega}_k$, and $\overline{\phi}_k$ are the vibrational parameters related to each *i*th engine. In this model, we assume that the UAV body's RCS is negligible with respect to those associated with the rotors.

From a radar signal point of view, each engine is defined by an ensemble of 13 parameters, i.e., its RCS plus four sets of amplitude, frequency, and phase values. Each set is related to a different vibrational component carried by the engine according to the mechanical model applied in [12].

The model-based dataset augmenter should completely reproduce the data-gathering process, as shown in Fig. 2; thus, the physical model output, as defined by the $s_{IF}(t)$ function of (6), is subjected to the same vector transformations that are expected to be caused by the FMCW radar signal processing.

Therefore, defining $\overline{\mathbf{x}}$ as the scenario parameter vector containing all the parametric terms of the function model $s_{\text{IF}}(t)$, the final augmenter (generator) function $\mathbf{f}(\overline{\mathbf{x}})$ related



 $p \{x_{^{u}A^{n}i} = x\}$ $(\sigma_{P}|\overline{\mathbf{x}_{^{u}A^{n}i}}|)\sqrt{2\pi}$ $\overline{\mathbf{x}_{^{u}A^{n}i}} - \sigma_{P}|\overline{\mathbf{x}_{^{u}A^{n}i}}|$ $\overline{\mathbf{x}_{^{u}A^{n}i}} + \sigma_{P}|\overline{\mathbf{x}_{^{u}A^{n}i}}|$

Fig. 4. Expected probability density function of noise added to parameter vector terms.

each engine is rewritten as

$$\overline{\mathbf{x}}_{k} = \left\langle \Gamma_{k}, A_{k\alpha}, \omega_{k\alpha}, \phi_{k\alpha}, A_{k\beta}, \omega_{k\beta}, \phi_{k\beta}, A_{k(\alpha+\beta)}, C_{\omega_{k(\alpha+\beta)}}, \phi_{k(\alpha+\beta)}, A_{k(\alpha-\beta)}, C_{\omega_{k(\alpha-\beta)}}, \phi_{k(\alpha-\beta)} \right\rangle^{T}$$
(10)

where $C_{\omega_{k(\alpha+\beta)}}$ and $C_{\omega_{k(\alpha-\beta)}}$ are defined as

$$C_{\omega_{k(\alpha+\beta)}} = \frac{\omega_{k(\alpha+\beta)}}{\left|\omega_{k\alpha} + \omega_{k\beta}\right|}, \quad C_{\omega_{k(\alpha-\beta)}} = \frac{\omega_{k(\alpha-\beta)}}{\left|\omega_{k\alpha} - \omega_{k\beta}\right|} \quad (11)$$

thus introducing as a target class parameter the skew of the mixing terms with respect to the expected mixing term frequencies.

According to (4), data augmentation for each class "*A*", "*B*", … is provided by spreading datasets around their related model parameter vectors $\overline{\mathbf{x}_{^{\circ}\mathbf{A}^{"}}}, \overline{\mathbf{x}_{^{\circ}\mathbf{B}^{"}}}, \ldots$ Such vectors could be initialized using an initial fit of the scenario model to one sample measurement per class. Note that the parameter vectors defined in (8) and (9) are highly heterogeneous; thus, to introduce a balanced augmentation over the entire set of parameters, the standard deviation of the noise added to each vector term needs to be proportional to the value of the term itself. Hence, for each *i*th synthetic vector of the class "A" data subset, the related parameter perturbation vector defined in (5) is redefined as

$$\overline{\mathbf{n}_{\mathbf{A}_{i,P}}} = \begin{pmatrix} \mathcal{N}_{1\times 1}(0, \sigma_{P} | x_{\mathbf{A}_{1}}) \\ \mathcal{N}_{1\times 1}(0, \sigma_{P} | x_{\mathbf{A}_{1}}) \\ \vdots \\ \mathcal{N}_{1\times 1}(0, \sigma_{P} | x_{\mathbf{A}_{N}}) \end{pmatrix}, \text{ with } \sigma_{P} \in [0, 1]$$

$$(12)$$

where
$$x_{A^n} = \overline{x_{A^n}}_n$$
 is equal to the *i*th term of the class "A" model parameter vector, and σ_P is a single value representing the relative amount of spreading introduced to every parameter. Fig. 4 shows the expected probability density function for every scenario parameter.

C. Training Method

Following the discussion in Section II-B, several augmented datasets are produced for different values of σ_P , generating N different augmented vectors for each class. Each subset of data

Fig. 3. UAV rotational reference system.

to this specific data-gathering chain is defined as

$$\mathbf{f}(\overline{\mathbf{x}}) = \mathrm{FFT}\{s_{\mathrm{IF}}(t, \overline{\mathbf{x}})\}.$$
(7)

According to [12], placing the reference system as shown in Fig. 3 and considering that the FMCW radar is placed on the top or bottom of the UAV, for each engine, only the microrotations on the pitch and roll axes can be considered.

The projections of these movements on the radar line of sight lead to four different vibrational micro-Doppler terms: ω_{α} , which is related to roll axis vibrations; ω_{β} , which is related to pitch axis vibrations; and $|\omega_{\alpha} + \omega_{\beta}|$ and $|\omega_{\alpha} - \omega_{\beta}|$, which mix components of both types of vibrations. Thus, the parameter subvector for each engine is defined as

$$\overline{\mathbf{x}}_{k} = \left\langle \Gamma_{k}, A_{ka}, \omega_{ka}, \phi_{ka}, A_{k\beta}, \omega_{k\beta}, \phi_{k\beta}, A_{k(\alpha+\beta)}, |\omega_{ka} + \omega_{k\beta}|, \phi_{k(\alpha+\beta)}, A_{k(\alpha-\beta)}, |\omega_{k\alpha} - \omega_{k\beta}|, \phi_{k(\alpha-\beta)} \right\rangle^{T}$$
(8)

retaining the information related to all the UAV vibrational components. The overall vector has the following form:

$$\overline{\mathbf{x}} = \begin{pmatrix} \overline{\mathbf{x}}_1 \\ \overline{\mathbf{x}}_2 \\ \vdots \\ \overline{\mathbf{x}}_{N_{eng}} \end{pmatrix}$$
(9)

comprising $N_{eng} \times 13$ scalar values, and $\overline{\mathbf{x}}$ can be considered a short representation of each class within a first-approximation feature space.

Despite the fact that we can associate the mixing terms to the roll and pitch components, we can optimize them to consider higher order phenomena that are not considered by the basic mechanical model [12]. The parameter vector for



Fig. 5. Flowchart of the proposed training methodology.

is produced by applying the generator function in (4), where the class-related reference vector is described in (10) and the perturbation vector is defined in (12).

The optimum σ_P parameter should be estimated to avoid overfitting issues due to the possible lack of augmentation effects and to prevent the risk of class misspecification due to excess dispersion into the feature space, as shown in Fig. 1. Identifying the optimum σ_P parameter is a step related to the definition of the training dataset; thus, an appropriate approach consists of introducing the optimization of this parameter during the testing phase of the neural network.

The first fit for the reference vectors [cf. (9)] related to each class, namely, "QUAD" for the quadcopter (reference vector $\overline{\mathbf{x}_{\text{``QUAD''}}}$) and "HELI" for the single-engine UAV (reference vector $\overline{\mathbf{x}_{\text{``HELI''}}}$), is achieved over one generic measured sample for both classes. The risk of overfitting due to the choice of only one measurement for the definition of the model parameters is expected to be overcome by applying the proposed augmentation method based on the perturbation of the parameters.

A flowchart showing the training process for performing UAV classification is shown in Fig. 5. The proposed method performs the neural network training using all the different augmented datasets, which are generated with different σ_P values. After the training phase, each network is tested on the limited measure dataset and the accuracy on the test set is evaluated for each σ_P . The entire training and testing process is completed for three different ML algorithms: the SVM, gradient booster (XGBoost), and naive Bayes (NB). The



Fig. 6. Dataset acquisition chain.

dispersion parameter that achieves the highest classification accuracy is considered the best dispersion parameter. At the completion of the procedure, the best value of σ_P is given and can be used to produce wider augmented datasets, which are then used for the training of the more complex CNN.

D. Radar Data Acquisition

The entire processing flow discussed in Section II-C works on measurement data generated by the radar data acquisition block described in detail in Fig. 6. As expressed by (6), the micro-Doppler signature model introduces the effect of mechanical vibrations only with respect to the range distance, ignoring the spreading over different scattering angles. As a result, each synthetic measurement consists of a vector representing a range profile.

Dealing with a multiple-input–multiple-output (MIMO) radar system means that each radar acquisition provides a 2-D radar image I(r, ϑ) defined in the range and direction-of-signal-arrival domains. Considering an FMCW radar system with n receiving antennas, each data acquisition leads to a set of n different baseband time readings $s_{\text{IF}n}(t)$. The radar-angle image is then calculated as

$$I(r, \vartheta) = FFT_n \{ FFT\{s_{IFn}(t)\} \}$$
(13)

where the inner FFT{.} is calculated with respect to time for each antenna trace, mapping the frequency domain to range applying the transform

$$R = \frac{\mathbf{c}_0 \cdot f}{2\mu} \tag{14}$$

where c_0 is the speed of light, μ is the FMCW radar ramp coefficient, and *f* is the spectrum frequency domain. The outer FFT{.} is calculated on the orthogonal domain for the different antenna traces, thus achieving the angular domain, as described in [37].

The micro-Doppler behavior is then extracted by applying a moving-target indicator (MTI) filter [38]; thus, the radar image is calculated as

$$I_{MTI}(r,\vartheta) = I(r,\vartheta) - \frac{1}{N_{MTI}} \sum_{i=1}^{N_{MTI}} I_{-i}(r,\vartheta)$$
(15)

where N_{MTI} is the MTI frame depth and $I_{-i}(r, \vartheta)$ is the *i*th frame in the stack of the last N_{MTI} frames.

Due to the actual limitations of the micro-Doppler model, in order to obtain a measurement trace that is suitable for target classification, a single-dimension range profile must be extracted from each radar image after the application of the MTI filter. Considering $\hat{\vartheta}$ as the estimated angular direction for the UAV target, the data trace is given as

$$\overline{\mathbf{s}} = \frac{1}{\Delta\vartheta} \int_{\vartheta - \frac{\Delta\vartheta}{2}}^{\vartheta + \frac{\Delta\vartheta}{2}} \mathbf{I}_{\mathrm{MTI}}(r,\vartheta) \, d\vartheta \tag{16}$$

where $\Delta \vartheta$ is an approximation of the expected spreading of peaks around the signal peak direction. For the experiments provided hereinafter, we consider $\Delta \vartheta = 7^{\circ}$.

In order to produce data suitable as input of the UAV classification network, the range profile, after the application of the MTI algorithm, is passed through a target signature extraction block. The latter extracts, from the overall MTI-filtered range profile data vector, all the vector slices that could be related to a UAV kind of target. The extraction works taking a slice of 400 samples around each different peak throughout the range profile vector. As a result of the MTI application, the peaks result positioned in correspondence of targets showing high-frequency vibrations, as introduced in [12]. Each slice corresponds to a possible signature for a related UAV target and the length of 400 samples is dependent on the radar hardware parameters shown in Table I and [12]. These slices are the actual input for the DL classification network, and the latter is assumed to be able to identify the UAV target class based on them.

The application of MTI together with the extraction block allows to extract, from the radar image, only the signatures of interest for the actual classification application, overcoming the presence of artifacts as clutter or other scenario nonidealities. In this work, the signatures not related to the UAV targets are filtered out by the MTI algorithm, which removes targets with low vibrational frequencies [38], [39].

Note that the actual work is about UAV target classification, and thus, the proposed classification networks are supposed to deal with an input dataset that comprises only the signatures that must be classified. Considering the generality of radar systems and application scenarios, the classification is supposed to be applied at the highest level of postprocessing after the entire dataset acquisition chain shown in Fig. 6.

III. DEEP NEURAL NETWORKS

The architecture of a DNN depends significantly upon the complexity of the problem under consideration; it is related to how each layer is implemented, as well as the computational method used in each layer. The most widely used DL approaches are CNNs [40], recurrent neural networks (RNNs) [41], deep belief networks (DBNs) [42], and the recently introduced approach of generative adversarial networks (GANs) [43]. However, the most popular supervised DNNs for image processing, object recognition, image formation, and classification are based on CNN architectures [44]. The layers that make up a CNN and proposed CNN architectures will be discussed in detail in the following.

A. CNN Architecture

Generally, CNNs are made up of several essential layers for feature extraction. A convolutional layer, an activation function, a pooling layer, a fully connected (FC) layer, and a classification module are the main components of each CNN. Each component is briefly described as follows.

1) Convolutional Layer: A convolutional layer is the primary component of a CNN. It is like a filter bank that can be repeated numerous times. Suppose that the input array of the model is a tensor with a size of $(C1 \times W1 \times H1)$ in which W1 is the width, H1 is the height, and C1 is the number of input channels. In the first convolutional layer, a small square filter with a size of $(C1 \times K \times K)$ in which the number of channels is equal to the number of channels of the input tensor is applied to the input tensor, and using elementwise addition and multiplication, an output tensor with a size of $(1 \times W2 \times H2)$ is created. In each convolutional layer, there are multiple filters (M) and corresponding output tensors $(1 \times W2 \times H2)$, which are added together to create the final feature map $(M \times W2 \times H2)$ of that layer. Compared to FC layers, convolutional layers have fewer parameters to train. In these kinds of architectures, there are some unchangeable hyperparameters in the training process, namely, the size of each filter (K), the number of filters in each layer (M), and stride [44].

2) Activation Function: An activation function is a critical part of designing a neural network. It is a mathematical function applied elementwise to the outputs of convolutional layers to provide nonlinearity to the network. Note that the nonlinearity property enables the network to be made deeper. Otherwise, the network would have only a single layer, regardless of the complexity of the architecture. The rectified linear unit (ReLU), defined as $y = \max(0, x)$, is the most common activation function used in CNNs for hidden layers. In contrast, the activation function for the output layer depends on the type of prediction problem being considered [44], [45], [46].

3) Pooling Layer: A pooling layer, commonly used after convolutional and nonlinear layers, is applied to reduce the dimensions of each feature map. Pooling is accomplished by sliding a window (typically a square window) across each feature map and extracting only one value from each window. The number of channels will not be modified after pooling since downsampling is only performed on the width and height dimensions. Average pooling and max pooling are the two standard pooling methods that are most used in CNNs. Furthermore, pooling helps in making the representation nearly invariant to slight changes in the input.

4) FC Layer: The last pooling/convolutional layer's output is passed into an FC layer. An FC layer is responsible for learning nonlinear combinations of prior high-level information, and one or more FC layers can be utilized sequentially. Furthermore, all the inputs of this layer are attached to its output elements, as in a normal multilayer perceptron neural network.



Fig. 7. Proposed CNN model architecture for extracting features for UAV classification. The input data consist of a subset of 400 samples of range profile from the output of the dataset acquisition chain, centered about the identified target range.

5) Classification Module: The last unit in the CNN is the classification layer, which is applied to transform the output of the last FC layer into the probability that the input belongs to a given class. For multiclass classification, the softmax classifier [47] with cross-entropy loss [48] is extensively used.

B. Proposed CNN Architecture

The proposed CNN architecture is shown in Fig. 7. The model consists of the input data, two convolutional layers, two ReLU functions, two pooling layers, and two FC layers used as a classifier. The input data generated as described in Section II-B consist of a (400×1) frequency-domain vector. It represents 400 samples of range profile from the output of the dataset acquisition chain shown in Fig. 6.

As shown in Fig. 7, before the convolution operation, the input is reshaped to $(1 \times 20 \times 20)$; then, it is introduced into the network. Feature extraction is performed by using two convolutional layers, where the first convolutional layer was configured with K = 32 filters and the next layer was configured with K = 64 filters. Furthermore, to reduce the computational complexity, two max- and average-pooling layers with pooling sizes of (2×2) and (7×7) , respectively, were applied after the convolutional layers.

To formulate a nonlinear model, the ReLU was chosen as the activation function

$$f(x) = \begin{cases} x, & \text{if } x \ge 0\\ 0, & \text{if } x < 0 \end{cases}$$
(17)

where x is the convolutional layer output. Eventually, after the pooling layer, a flattening layer [49] is deployed to transform the 1-D input matrix into a vector. Then, all feature vectors x are given to the FC layers, which perform the following operation:

$$c_f = w_f x + b_f \tag{18}$$

where c_f is the output vector of the FC layer, which is calculated by multiplying the input feature vector x, and w_f , which is the FC layer weight matrix, and adding b_f , which is the bias of w_f . Finally, a softmax activation function and a classification module are applied to finalize the network.

The softmax function is applied to the last FC layer, thereby normalizing the output of the FC layer between 0 and 1. The softmax transfer function is

$$\hat{y} = \operatorname{softmax}(c_f) = \frac{\exp(c_f)}{\sum(\exp(c_f))}$$
(19)

where \hat{y} is the result of the softmax function. Finally, we performed class generalization by calculating the maximum output of the softmax algorithm. Furthermore, the proposed model employs an Adam optimizer [50] to make use of the adaptive learning rate's (LR) power for each parameter.

In addition, we chose a cross-entropy loss function to reduce the loss (L) of the CNN

$$L = -\frac{1}{m} \sum_{i=1}^{m} y_i \cdot \log(\hat{y}_i)$$
⁽²⁰⁾

where \hat{y}_i is the class predicted by the CNN, y_i is the label of the input data, and *m* denotes the number of classes. The parameter *m* is equal to 2 because there are two different classes in the UAV classification: "QUAD" and "HELI."

IV. EVALUATION AND RESULTS

This section discusses the experimental validation of the method described in Section III using a commercial millimeter-wave radar open platform and two classes of commercial UAVs.

A. Dataset Generation

1) Experiment-Based Dataset: The measurement campaign was accomplished in a semi-anechoic scenario (Fig. 8). The adopted radar is a compact commercial open platform [51], implementing a 77-GHz FMCW MIMO radar system, with eight receiving and four transmitting antennas. The system is



Fig. 8. Experimental setup adopted for the experimental validation.

based on a commercial chipset, allowing a certain degree of flexibility on the most relevant radar parameters (e.g., chirp parameters, carrier, transmitter power, and analog IF path parameters). The receive antennas involve serial fed patch antennas with eight elements, while the transmitting ones use two serial fed patch antennas that are fed from the differential output signal of the power amplifier; with the latter capable to provide 10 dBm of output power at 77 GHz. The reference oscillator is capable of generating a chirp signal with a maximum frequency range of 4 GHz, from 75 to 79 GHz. This carrier frequency was selected due to the potential imaging-enabling capability due to the short wavelength, and the availability of commercial platforms normally adopted for transport and traffic telematics and radio-determination applications in vehicular communications. The radar platform is controlled at a higher level by a field-programmable gate array (FPGA), which controls the switching between transmitters and the chirp, which is capable to carry on the basic signal processing operations (e.g., sample rate reduction). The highest level of platform control is actuated by a microcontroller unit, which allows a fast connection to a remote processor unit through which the baseband RX channel signals can be acquired coherently, while the postprocessing of the radar image was carried out using a MATLAB code. Each receiver path is equipped with a single mixer device, as is common in FMCW radar architectures [37], [52]. The mixer is followed by the six-channel Analog Devices AD8283 integrated radar receive path, which is equipped with a corresponding number of analog front end and a 12-bit analog-to-digital converter capable of a maximum sampling frequency of 72 MSPS [53]. The FMCW MIMO radar parameters adopted in this study are presented in Table I.

Two different kinds of UAV targets were used for measurements; they are both shown in Fig. 9. The drone targets used for the new measurements dataset belong to the same classes of reference dataset presented in [12], leading to the

TABLE I FMCW MIMO RADAR PLATFORM PARAMETERS

Ramp frequency	75–79 GHz
RF output power	10 dBm
No. of RX antennas	8
No. of TX antennas	4
RX antenna antenna gain	15.8 dBi; $\Theta_H = 76.5^\circ, \Theta_V = 12.8^\circ$
TX antenna antenna gain	17.2 dBi; $\Theta_H = 51^\circ, \Theta_V = 13.2^\circ$
Ramp time up	2.56 ms
Ramp time down	0.32 ms
Ramp coefficient μ	1.56×10^{12} Hz/s
Ramps per frame	1000
Chirp repetition interval	200 ms
Samples per chirp	504
IF sampling frequency	200 kHz
No. of FFT points	4096
FFT process gain	36.12 dB
MTI frame depth	10



Fig. 9. Target UAVs for experimental validation. (a) Single-engine UAV, RadioFLY Helicopter. (b) Quadcopter UAV, DJI Drone Mavic Mini.

model parameters shown in Table II, but they have different mechanical specifications than the ones previously involved. This aspect is of fundamental importance because the objective of actual work is to provide by augmentation a training dataset able to instruct a DNN to recognize targets belonging to the same classes of reference ones despite having totally different specifications.

For the single-engine UAV (class "HELI"), we selected a small plastic helicopter with body dimensions of $10 \times 30 \times 20$ cm, while for the quadcopter UAV (class "QUAD"), a DJI Mavic Mini 2 drone [54] with body dimensions of $24.5 \times 29 \times 5.5$ cm was adopted.

As shown in [12] and briefly discussed in Section II-D, an MTI algorithm [55] with a depth of 10 frames was used



Fig. 10. Examples of acquired radar images for both target classes. (a) Drone-DJI quadcopter, single-frame acquisition. (b) RadioFLY helicopter, single-frame acquisition. (c) Drone-DJI quadcopter, radar image after MTI algorithm. (d) RadioFLY helicopter, radar image after MTI algorithm. (e) Drone-DJI quadcopter, range profile in boresight direction.

to extract the micro-Doppler signatures and remove the clutter from the acquired measurements. For each measurement, the range domain was sampled over 4096 points, but to lighten the dataset, only 400 points (or rather, features) centered with respect to the actual target range were utilized. For each target class, 45 measurements were taken for various yaw orientations (Fig. 3). Fig. 10 shows two example radar acquisitions for each target class before and after the application of the MTI filtering algorithm and after the extraction of the range profiles; being these latter evaluated as in Section II-D.

2) Synthetic Dataset: The model introduced in (6) does not consider the angular information provided by the MIMO array. UAV classification is done by evaluating only the range profiles on the line of sight between the radar system and the target. Several datasets comprising 32000 synthetic measurements of 400 features each, equal to the number of points for each real measurement, are produced by considering various standard deviation parameters (i.e., $\sigma_P = [0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.40, 0.50, 0.60])$ for the terms $\overline{\mathbf{n}_{\cdot \mathbf{QUAD}^{\prime\prime}}}_{i,P}$ and $\overline{\mathbf{n}_{\cdot \mathbf{HELF}^{\prime}}}_{i,P}$ in (12) in the data generator function shown in (4). For each class, the model parameter vectors $\overline{\mathbf{x}_{\cdot \mathbf{HELF}^{\prime\prime}}}$ and $\overline{\mathbf{x}_{\cdot \mathbf{QUAD}^{\prime\prime}}}$, as shown in (4), are those proposed in [12] and shown in Table II. These vectors were obtained by fitting the model parameters to one sample measurement for each target class [12].

It is noteworthy that the UAV models considered in previous measurements are different from those involved in this work. The use of the model coefficients related to different UAVs, similar only within a given class, further demonstrates the methodology's effectiveness, showing that a model-based augmenter can easily cover measurement scenarios that were not considered initially.

TABLE II MODEL PARAMETER VECTORS FOR EACH CLASS

	Term	X"HELI"	X"QUAD"				
	Γ_1	0.044	0.001				
	$A_{1\alpha}$	2.05×10^{-3}	2.24×10^{-3}				
	$\omega_{1,\alpha}$	2.69×10^{3}	1.77×10^{3}				
	$\phi_{1\alpha}$	1.04	-1.70				
	$A_{1\beta}$	2.08×10^{-3}	2.19×10^{-3}				
	$\omega_{1,\beta}$	3.98×10^{3}	4.84×10^{3}				
Engine 1	$\phi_{1,\beta}$	0.81	-0.51				
	$A_1(\alpha + \beta)$	6.68×10^{-4}	1.64×10^{-3}				
	$C_{\omega_1(\dots+\omega)}$	104%	100%				
	$\phi_1(\alpha+\beta)$	1.90	3.88				
	$\frac{\varphi_1(\alpha+\beta)}{4}$	2.35×10^{-3}	3.60×10^{-3}				
	C_{α}	92%	100%				
	$\omega_{1(\alpha-\beta)}$	0.04	0.01				
	$\varphi_{1(\alpha-\beta)}$	-0.04	0.01				
	Γ_2	0	0.003				
	$A_{2\alpha}$	-	3.24×10^{-3}				
	$\omega_{2\alpha}$	-	4.79×10^{3}				
	$\phi_{2\alpha}$	-	1.06				
	$A_{2\beta}$	-	1.63×10^{-3}				
	$\omega_{2\beta}$	-	2.61×10^{3}				
Engine 2	$\phi_{2\beta}$	-	1.93				
	$A_{2(\alpha+\beta)}$	-	1.42×10^{-3}				
	$C_{\omega_{2(\alpha+\beta)}}$	-	100%				
	$\phi_{2(\alpha+\beta)}$	-	-2.64				
	$A_{2(\alpha-\beta)}$	-	2.61×10^{-3}				
	C_{ω_2}	-	99%				
	$\phi_{2(\alpha-\beta)}$	-	-1.54				
	Γ		0.001				
	13	0	1.67×10^{-3}				
	$A_{3\alpha}$	-	1.07×10^{-1}				
	$\omega_{3\alpha}$	-	2.62×10^{-1}				
	$\frac{\varphi_{3\alpha}}{1-1}$	-	1.52 2.02 × 10 ⁻³				
	$A_{3\beta}$	-	$5.02 \times 10^{-5.06} \times 10^{-3}$				
Engine 3	$\omega_{3\beta}$	-	3 35				
	$\frac{\varphi_{3\beta}}{1}$	-	1.21×10^{-3}				
	C	-	1.00%				
	$\omega_{3(\alpha+\beta)}$	-	4.21				
	$\varphi_{3(\alpha+\beta)}$	-	4.51				
	$A_{3(\alpha-\beta)}$	-	4.76×10^{-6}				
	$C_{\omega_3(\alpha-\beta)}$	-	100%				
	$\phi_{3(\alpha-\beta)}$	-	-1.70				
	Γ_4	0	0.011				
	$A_{4\alpha}$	-	1.82×10^{-3}				
	$\omega_{4\alpha}$	-	5.07×10^{3}				
	$\phi_{4\alpha}$	-	0.68				
	$A_{4\beta}$	-	1.44×10^{-3}				
	$\omega_{4\beta}$	-	2.62×10^3				
Engine 4	$\phi_{4\beta}$	-	-2.58				
	$A_{4(\alpha+\beta)}$	-	8.29×10^{-4}				
	$C_{\omega_4(\alpha+\beta)}$	-	100%				
	$\phi_{4(\alpha+\beta)}$	-	-3.92				
	$A_{4(\alpha-\beta)}$	-	3.88×10^{-3}				
	$C_{\omega_{A}}$	-	100%				
	$\phi_{4(\alpha-\beta)}$	-	-1.71				
1	$\mu = (\alpha = \rho)$	i i i i i i i i i i i i i i i i i i i	1				

Fig. 11 shows several measurements related to each class of the actual measured dataset; they are compared with one measurement from the other class aligned to the target range distance and with the synthetic data vector produced using the model parameter vectors of Table II [12]. It is clear that the synthetic model is able to reproduce features related to different classes that are expressed through different trace behaviors found on both sides of the maximum peak identifying the main body of the drone target.



Fig. 11. Measured range profile (blue line) and synthetic range profile made with model parameters (red line) for both classes. Each curve in this figure contains 4096 samples of the range profile, of which 400 samples centered around the target range are used as inputs to the CNN model of Fig. 7. (a) Single-engine UAV, RadioFLY helicopter. (b) Quadcopter UAV, DJI Drone Mavic Mini.

B. Performance Metrics for Classification Problems

Following the model implementation and the feature extraction, as well as the class estimation, the next step consists of determining the effectiveness of the model-based dataset generation method using some statistical indicator. The metrics employed to assess the DL model are important since they determine how the DL network performs. Nevertheless, various performance metrics are used to evaluate different learning algorithms. For the sake of this study, we focus on the metrics that are relevant to classification problems. Among the classification performance metrics, we employ the confusion matrix (CM), accuracy, and precision [56].

1) Confusion Matrix: A CM is composed of predicted and actual classification information that is presented in a specific table layout. During the predictive analysis, a square CM with positive and negative rates is constructed (including both true and false outcomes). Fig. 12 shows an example of a CM for a binary classification, where TP stands for true positive and indicates that the model correctly identified the class to which a sample belongs. FP stands for false positive and indicates that the model predicted that a sample belongs to a class that it does not belong to. FN stands for false negative and indicates that the model predicted that a sample does not belong to the class that it actually does belong to. Finally, TN stands for true negative and indicates that the model correctly predicted that a sample does not belong to a class that it actually does not belong to. An appropriate model will have high values for



Fig. 12. CM for a binary classification.

the principal diagonal elements, TP and TN, and low values for the off-diagonal elements.

2) Accuracy: The accuracy of the classification problem is the ratio of the number of correct predictions made by the model to the total number of predictions made by the model. The accuracy (acc) is given by

$$acc = \frac{TP + TN}{TP + FP + FN + TN}.$$
 (21)

3) *Precision:* The precision describes how many samples were correctly classified in each class, and it is calculated as

$$\text{prec} = \frac{\text{TP}}{\text{TP} + \text{FP}}.$$
 (22)

C. Experimental Results

In this section, we prove the effectiveness of the proposed methodology by presenting the classification accuracy obtained by the data augmentation method using the proposed model-based augmenter versus the classification accuracy of a typical random signal processing augmentation method.

First, the best σ_P value, leading to the highest classification accuracy, should be chosen in order to obtain the optimal model-based dataset. Since a large number of tests involving network training and testing should be carried out over several σ_P values, the ML algorithms, due to their reduced computational cost and lower computation time, are preferred over CNNs. Then, the final classification accuracy is further improved by applying the proposed CNN architecture.

1) *ML Accuracy Versus* σ_P : Several SVM, XGBoost, and NB networks were trained with synthetic datasets with $\sigma_P = [0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.40, 0.50, 0.60].$

The trained network was assessed on three distinct datasets referred as it follows. The "Old" refers to the dataset obtained in [12] that includes 30 measured signatures for the UAVs mentioned in [12], the "New" refers to the measurement dataset described in Section IV-A1, which consists of 136 measured signatures for the UAVs considered in this article, and "Mix" dataset is generated by merging the "Old" and "New" datasets. Table III shows the class distribution between measurements for each dataset.

With augmentation, trained CNNs are expected to be capable of correctly classifying new measurements not related to

TABLE III Case Distribution in Measured Datasets

Dataset	Overall Items	Dataset Weight	HELI Items	QUAD Items
Old	30	0.18	12 (40%)	18 (60%)
New	136	0.82	44 (32%)	92 (68%)
Mix	166	1.00	56 (34%)	110 (66%)



Fig. 13. Trend of classification accuracy on measured dataset with respect to σ_P parameter values.

those used to extract model parameters. Therefore, the proposed approach should provide the highest level of accuracy for the "New" dataset choosing higher values for the σ_P parameter since the objective is to augment the training dataset to provide a reliable dataset that is not related to the "Old" measurements reducing the risk of overfitting.

Table IV provides the classification accuracy results for these three test datasets, and Fig. 13 shows the accuracy trends with respect to σ_P parameter values. Due to space limitation considerations, only a limited set of σ_P values are reported in Table IV. Due to the different number of measurements considered between the "Old" and "New" datasets, as shown in Table III, the accuracy achieved for the "Mix" dataset results to be equal to the weighted averages of the accuracies for each single dataset. The relative weights for each dataset are listed in Table III.

Table IV and Fig. 13 reveal that $\sigma_P = 0.20$ leads to the highest accuracy on the dataset of new measurements compared to the other σ_P values. Therefore, we chose $\sigma_P = 0.20$ to create our best possible input dataset of 32 000 samples, each of which has 400 features (i.e., range profile sample points). In Fig. 13, it is noteworthy that the maximum accuracy is achieved within a specific interval of σ_P , i.e., between 0.1 and 0.2, and in the same interval, the accuracy is maximized for all the measurement datasets. Such behavior is expected, as discussed in Section II, because the spreading of model parameters by the perturbation vector provides a reliable augmentation until relative ratios between features for each sample are preserved. Furthermore, the provided results prove the effectiveness of the proposed methodology for achieving training datasets extended by augmentation, reliable enough to cover measurement scenarios not initially considered.

2) Proposed CNN Performance Results: Next, we initiated the training of the proposed CNN (cf. Section III-B) using the optimal input dataset produced in the previous step.

TABLE IV

CLASSIFICATION ACCURACY ON MEASURED DATASET WITH RESPECT TO σ_P PARAMETER VALUES WHERE "OLD" REFERS TO THE DATASET IN [12], "New" Refers to the Dataset in This Article, and "Mix" Refers to the Dataset Derived by Merging the Two

	$\sigma_P = 0.00$		$\sigma_P = 0.05$		$\sigma_P = 0.10$		$\sigma_P = 0.20$			$\sigma_P = 0.30$					
	Old	New	Mix	Old	New	Mix	Old	New	Mix	Old	New	Mix	Old	New	Mix
SVM	40%	35%	36%	72%	64%	65%	72%	71%	71%	72%	72%	72%	30%	41%	39%
Naive Bayes	36%	36%	36%	68%	68%	68%	74%	75%	75%	74%	68%	77%	77%	51%	56%
XGBoost	23%	33%	31%	96%	57%	64%	94%	68%	73%	94%	77%	80%	91%	49%	56%



7 HEL 5.15 % 60 Actual value 50 8 83 QUAD 30 5.88 % 61.03 % 20 HELI QUAD Predicted value

Acc = 66.18 %

Fig. 14. CM of the network trained on the model-based augmenter dataset and tested on measurement data.

In the training procedure, we first split the 32000-sample dataset into three different parts: 1) training dataset; 2) validation dataset; and 3) test dataset, which contains 80%, 10%, and 10% of the samples, respectively. The model is then trained with a backpropagation algorithm and an LR of LR = 0.001 for 50 epochs. Finally, the best model, chosen using the validation dataset, is tested on the real measurement dataset from Section IV-A1.

Fig. 14 shows the predictive performance of the proposed CNN model for UAV classification using the real measurement dataset. The columns in the matrix presented in Fig. 14 represent the predicted classes, while the rows represent the actual classes. Considering Fig. 14 and based on the precision, defined in (22), it is obvious that in the proposed network, 78% of the samples taken from the quadcopter are correctly classified as "QUAD"-class samples, while 80% of the samples taken from the helicopter are correctly classified as "HELI"class samples. Accordingly, the total network classification accuracy, as defined in (21), is equal to 78.68%.

Next, to assess the effectiveness of the proposed augmentation method, the CNN model is retrained following the procedure described previously for 100 epochs using the experimental dataset adopted in [12], which uses the random signal processing augmenter. The process of data augmentation involves transforming existing data to increase the number of samples without collecting new data. In order to accomplish this, we used the "tsaug" signal processing augmentation [57]. The "tsaug" is a Python package that augments signal data, offering a variety of augmentation methods: in this study, drift,

Fig. 15. CM of the network trained on a conventional signal processing augmenter dataset and tested on measurement data.

TABLE V CLASSIFICATION PRECISION OF THE NETWORK TRAINED ON DIFFERENT TRAINING DATASETS

Training Dataset	HELI	QUAD
Conventional Augmenter Dataset	15.55%	91.21%
Deterministc Augmenter Dataset with $\sigma_P=0.20\%$	80.00%	78.02%

pool, and reversing are used to augment the signal, as they maintain the input sample size [57].

Fig. 15 shows the predictive performance of the CNN model for UAV classification on the real measurement dataset described in Section IV-A1. According to Fig. 15 and (22), 90% of the samples taken from the quadcopter are correctly classified as "QUAD"-class samples, while 15% of the samples taken from the helicopter are correctly classified as "HELI"-class samples. Accordingly, the total network acc is equal to 66.18%; however, the classifier is highly biased toward the "QUAD" class.

Table V shows the classification precision achieved by the different networks: one trained with the conventional augmented dataset and one trained with dataset built through the proposed deterministic augmenter for two different σ_P values. It is clear that despite achieving a similar accuracy, the network trained with the deterministic augmented dataset exhibits a far lower classification bias. We can thus draw the conclusion that conventional augmenter leads to the unreliable

80 70

40

10

training dataset, even for the basic case of the two UAV classes classification considered in this work.

V. CONCLUSION

This article introduced a new data augmentation methodology based on a deterministic model of a physical scenario. This work specifically addressed the classification of UAV targets according to the number of motors using the micro-Doppler radar signatures collected by a millimeter-wave FMCW radar system.

The proposed method successfully trained a CNN for UAV classification without requiring a measurement session to generate a training dataset. The initial model parameters were obtained through a fitting procedure that took into account measurement data collected in a different scenario with UAV targets different from those considered in this work; however, these UAV targets did belong to the same classes as the targets considered in this work. A perturbation vector is introduced on vector of model parameters achieved by fitting to reduce the dependence of the augmented dataset on reference measurements, allowing the latter to cover the new targets involved in this study.

In contrast to typical random signal processing augmentation methods, the augmented dataset produced with the deterministic augmenter provides the variability required to perform successful training while avoiding the risk of overfitting and preserving physically significant features.

By the proposed methodology, a CNN trained on a synthetic dataset produced by the deterministic augmenter introduced in this work achieved a classification accuracy of 78.68% with a minimum classification precision of 78.02%. In comparison, the same network trained on a dataset built using the standard augmentation methods obtained a classification accuracy of 66.18% with a far lower minimum precision of 15.55%, thus showing a dramatic classification bias, which makes such network unusable also in the case of the fundamental classification objective considered in this work.

The proposed deterministic augmenter is based on a simple and scalable physical model of the backscattering for the expected radar targets. The achieved results have shown that despite the simplicity of the model, the latter was able to provide a meaningful of informative features for the training, as demonstrated by the lowering of the classification bias for the CNN network trained by deterministic augmenter.

Therefore, it can be expected that increasing the model complexity and introducing more parameters, i.e., extending the model to provide range profiles collected from different receiving antennas, it could be possible to produce large training datasets covering different kinds of UAVs, by specifying details about their own geometries. Following this, it could be considered to train future DL networks to provide more detailed information about the identified UAV targets, without the need of large measurement campaigns for training involving several UAV specimens.

The same approach in principle could be extended to other application scenarios, as human recognition [58], or medical imaging [59], where the collection of a large training dataset would require a significant amount of time and resources.

REFERENCES

- H. González-Jorge, J. Martínez-Sánchez, M. Bueno, and A. P. Arias, "Unmanned aerial systems for civil applications: A review," *Drones*, vol. 1, no. 1, pp. 2–19, Jul. 2017.
- [2] D. C. Tsouros, S. Bibi, and P. G. Sarigiannidis, "A review on UAV-based applications for precision agriculture," *Information*, vol. 10, no. 11, pp. 1–26, Nov. 2019.
- [3] Y. Tan, J. Wang, J. Liu, and N. Kato, "Blockchain-assisted distributed and lightweight authentication service for industrial unmanned aerial vehicles," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 16928–16940, Sep. 2022.
- [4] J. Peng, Z. Zhang, Q. Wu, and B. Zhang, "Anti-jamming communications in UAV swarms: A reinforcement learning approach," *IEEE Access*, vol. 7, pp. 180532–180543, 2019.
- [5] C. Xu, X. Liao, J. Tan, H. Ye, and H. Lu, "Recent research progress of unmanned aerial vehicle regulation policies and technologies in urban low altitude," *IEEE Access*, vol. 8, pp. 74175–74194, 2020.
- [6] C. Premachandra, D. Ueda, and K. Kato, "Speed-up automatic quadcopter position detection by sensing propeller rotation," *IEEE Sensors J.*, vol. 19, no. 7, pp. 2758–2766, Apr. 2019.
- [7] R. Opromolla, G. Fasano, and D. Accardo, "A vision-based approach to UAV detection and tracking in cooperative applications," *Sensors*, vol. 18, no. 10, pp. 1–26, Oct. 2018.
- [8] M. Pieraccini, L. Miccinesi, and N. Rojhani, "RCS measurements and ISAR images of small UAVs," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 32, no. 9, pp. 28–32, Sep. 2017.
- [9] J. Gong, J. Yan, D. Li, and R. Chen, "Using radar signatures to classify bird flight modes between flapping and gliding," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 9, pp. 1518–1522, Sep. 2020.
- [10] A. Ramachandran and A. K. Sangaiah, "A review on object detection in unmanned aerial vehicle surveillance," *Int. J. Cognit. Comput. Eng.*, vol. 2, pp. 215–228, Jun. 2021.
- [11] V. C. Chen, F. Li, S.-S. Ho, and H. Wechsler, "Micro-Doppler effect in radar: Phenomenon, model, and simulation study," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 1, pp. 2–21, Jan. 2006.
- [12] M. Passafiume, N. Rojhani, G. Collodi, and A. Cidronali, "Modeling small UAV micro-Doppler signature using millimeter-wave FMCW radar," *Electronics*, vol. 10, no. 6, pp. 1–16, Mar. 2021.
- [13] H. C. Kumawat, M. Chakraborty, and A. A. B. Raj, "DIAT-RadSATNet—A novel lightweight DCNN architecture for micro-Doppler-based small unmanned aerial vehicle (SUAV) targets' detection and classification," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022.
- [14] B.-S. Oh, X. Guo, F. Wan, K.-A. Toh, and Z. Lin, "Micro-Doppler mini-UAV classification using empirical-mode decomposition features," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 227–231, Feb. 2018.
- [15] R. I. A. Harmanny, J. J. M. de Wit, and G. P. Cabic, "Radar micro-Doppler feature extraction using the spectrogram and the cepstrogram," in *Proc. Eur. Radar Conf.*, Rome, Italy, Oct. 2014, pp. 165–168.
- [16] F. H. C. Tivive, S. L. Phung, and A. Bouzerdoum, "Classification of micro-Doppler signatures of human motions using log-Gabor filters," *IET Radar Sonar Navigat.*, vol. 9, no. 9, pp. 1188–1195, Dec. 2015.
- [17] J. J. M. de Wit, R. I. A. Harmanny, and P. Molchanov, "Radar micro-Doppler feature extraction using the singular value decomposition," in *Proc. Int. Radar Conf.*, Oct. 2014, pp. 1–6.
- [18] S. Wagner, K. Barth, and S. Brüggenwirth, "A deep learning SAR ATR system using regularization and prioritized classes," in *Proc. IEEE Radar Conf.*, May 2017, pp. 772–777.
- [19] Z. Lin, K. Ji, M. Kang, X. Leng, and H. Zou, "Deep convolutional highway unit network for SAR target classification with limited labeled training data," *IEEE Trans. Geosci. Remote. Sens. Lett.*, vol. 14, no. 7, pp. 1091–1095, Jul. 2017.
- [20] S. Chen, H. Wang, F. Xu, and Y.-Q. Jin, "Target classification using the deep convolutional networks for SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4806–4817, Apr. 2016.
- [21] M. Kang, K. Ji, X. Leng, and Z. Lin, "Contextual region-based convolutional neural network with multilayer fusion for SAR ship detection," *Remote Sens.*, vol. 9, no. 8, pp. 1–14, Aug. 2017.
- [22] J. Lombacher, M. Hahn, J. Dickmann, and C. Wöhler, "Potential of radar for static object classification using deep learning methods," in *IEEE MTT-S Int. Microw. Symp. Dig.*, San Diego, CA, USA, Apr. 2016, pp. 1–4.
- [23] J. Ding, B. Chen, H. Liu, and M. Huang, "Convolutional neural network with data augmentation for SAR target recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 3, pp. 364–368, Mar. 2016.

- [24] R. P. Trommel, R. I. A. Harmanny, L. Cifola, and J. N. Driessen, "Multi-target human gait classification using deep convolutional neural networks on micro-Doppler spectrograms," in *Proc. Eur. Radar Conf. (EuRAD)*, London, U.K., Oct. 2016, pp. 81–84.
 [25] Y. Kim and B. Toomajian, "Hand gesture recognition using micro-
- [25] Y. Kim and B. Toomajian, "Hand gesture recognition using micro-Doppler signatures with convolutional neural network," *IEEE Access*, vol. 4, pp. 7125–7130, 2016.
- [26] B. Taha and A. Shoufan, "Machine learning-based drone detection and classification: State-of-the-art in research," *IEEE Access*, vol. 7, pp. 138669–138682, 2019.
- [27] N. E. Khalifa, M. Loey, and S. Mirjalili, "A comprehensive survey of recent trends in deep learning for digital images augmentation," *Artif. Intell. Rev.*, vol. 2021, pp. 1–27, Sep. 2021.
- [28] Y. D. Dadon, S. Yamin, S. Feintuch, H. H. Permuter, I. Bilik, and J. Taberkian, "Moving target classification based on micro-Doppler signatures via deep learning," in *Proc. IEEE Radar Conf.*, Atlanta, GA, USA, May 2021, pp. 1–6.
- [29] Z. Xie, F. He, S. Fu, I. Sato, D. Tao, and M. Sugiyama, "Artificial neural variability for deep learning: On overfitting, noise memorization, and catastrophic forgetting," *Neural Comput.*, vol. 33, no. 8, pp. 2163–2192, Jul. 2021.
- [30] A. Coluccia et al., "Drone vs. bird detection: Deep learning algorithms and results from a grand challenge," *Sensors*, vol. 21, no. 8, p. 2824, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/8/2824
- [31] W. Zhang and G. Li, "Detection of multiple micro-drones via cadence velocity diagram analysis," *Electron. Lett.*, vol. 54, no. 7, pp. 441–443, 2018.
- [32] P. Zhang, L. Yang, G. Chen, and G. Li, "Classification of drones based on micro-Doppler signatures with dual-band radar sensors," in *Proc. Prog. Electromagn. Res. Symp.-Fall*, 2017, pp. 638–643.
- [33] X. Zhang, Z. Wang, D. Liu, Q. Lin, and Q. Ling, "Deep adversarial data augmentation for extremely low data regimes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 15–28, Jan. 2021.
 [34] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer,
- [34] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, no. 1, pp. 321–357, Jan. 2002.
- [35] M. Sheeny, A. Wallace, and S. Wang, "Radio: Parameterized generative radar data augmentation for small datasets," *Appl. Sci.*, vol. 10, no. 11, pp. 1–13, Jun. 2020.
- [36] D. Martens, B. Baesens, and T. V. Gestel, "Decompositional rule extraction from support vector machines by active learning," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 2, pp. 178–191, Feb. 2009.
- [37] A. Cidronali et al., "System level analysis of millimetre-wave GaNbased MIMO radar for detection of micro unmanned aerial vehicles," in *Proc. Photon. Electromagn. Res. Symp.-Spring (PIERS-Spring)*, Rome, Italy, Jun. 2019, pp. 438–450.
- [38] M. Ash, M. Ritchie, and K. Chetty, "On the application of digital moving target indication techniques to short-range FMCW radar data," *IEEE Sensors J.*, vol. 18, no. 10, pp. 4167–4175, May 2018.
 [39] M. Ispir and C. Candan, "On the design of staggered moving target
- [39] M. Ispir and C. Candan, "On the design of staggered moving target indicator filters," *IET Radar, Sonar Navigat.*, vol. 10, no. 1, pp. 205–215, Jan. 2016.
- [40] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022.
 [41] Z. Zhang, L. Zheng, J. Yu, Y. Li, and Z. Yu, "Three recurrent neural neural
- [41] Z. Zhang, L. Zheng, J. Yu, Y. Li, and Z. Yu, "Three recurrent neural networks and three numerical methods for solving a repetitive motion planning scheme of redundant robot manipulators," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 3, pp. 1423–1434, Jun. 2017.
 [42] G. Wang, J. Qiao, C. Liu, and Z. Shen, "How deep is deep enough
- [42] G. Wang, J. Qiao, C. Liu, and Z. Shen, "How deep is deep enough for deep belief network for approximating model predictive control law," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 2067–2078, Jul. 2022.
- [43] E. Ayanoglu, K. Davaslioglu, and Y. E. Sagduyu, "Machine learning in NextG networks via generative adversarial networks," *IEEE Trans. Cognit. Commun. Netw.*, vol. 8, no. 2, pp. 480–501, Jun. 2022.
 [44] A. H. Oveis, E. Giusti, S. Ghio, and M. Martorella, "A survey on
- [44] A. H. Oveis, E. Giusti, S. Ghio, and M. Martorella, "A survey on the applications of convolutional neural networks for synthetic aperture radar: Recent advances," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 37, no. 5, pp. 18–42, May 2022.
- [45] Y. Wang, Y. Li, Y. Song, and X. Rong, "The influence of the activation function in a convolution neural network model of facial expression recognition," *Appl. Sci.*, vol. 10, no. 5, pp. 1–20, Mar. 2020.
- [46] M. Varshney and P. Singh, "Optimizing nonlinear activation function for convolutional neural networks," *Signal, Image Video Process.*, vol. 15, no. 6, pp. 1323–1330, Feb. 2021.
- [47] F. Gao, B. Li, L. Chen, Z. Shang, X. Wei, and C. He, "A softmax classifier for high-precision classification of ultrasonic similar signals," *Ultrasonics*, vol. 112, pp. 1–8, Apr. 2021.

- [48] L. Li, M. Doroslovacki, and M. H. Loew, "Approximating the gradient of cross-entropy loss function," *IEEE Access*, vol. 8, pp. 111626–111635, 2020.
- [49] J. Jin, A. Dundar, and E. Culurciello, "Flattened convolutional neural networks for feedforward acceleration," 2014, *arXiv:1412.5474*.
- [50] C.-H. Lin, Y.-C. Lin, and P.-W. Tang, "ADMM-ADAM: A new inverse imaging framework blending the advantages of convex optimization and deep learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022.
- [51] INRAS Radarbook Platform. (Oct. 15, 2022). NOFFZ ComputerTechnology GmbH. [Online]. Available: https://noffz.com/en/products/RFtest-and-simulator/
- [52] A. Cidronali, L. Pagnini, G. Collodi, and M. Passafiume, "A highly linear Ka-band GaN-on-Si active balanced mixer for radar applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 11, pp. 4453–4464, Jul. 2022.
- [53] Analog Devices. (Oct. 15, 2022). AD8323 Radar Receive Path AFE Datasheet (DocID: SLUSBH2E). [Online]. Available: https://www. analog.com/en/products/ad8283.html
- [54] SZ DJI Technology Co. (Oct. 15, 2022). DJI Mavic Mini 2 Drone Specifications. Shenzen, China. [Online]. Available: https://www.dji. com/mini-2/specs
- [55] M. K. Newey, J. M. Kantor, and G. R. Benitz, "Autofocus methods for moving target imaging in synthetic aperture radar," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–11, 2022.
- [56] T. Wang, R. Liu, and G. Qi, "Multi-classification assessment of bank personal credit risk based on multi-source information fusion," *Expert Syst. Appl.*, vol. 191, pp. 1–15, Apr. 2022.
 [57] Arundo Analytics. (Oct. 15, 2022). *Tsaug—A Python Package for Time*
- [57] Arundo Analytics. (Oct. 15, 2022). Tsaug—A Python Package for Time Series Augmentation. Houston, TX, USA. [Online]. Available: https:// tsaug.readthedocs.io/
- [58] Y. Yao, W. Liu, G. Zhang, and W. Hu, "Radar-based human activity recognition using hyperdimensional computing," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 3, pp. 1605–1619, Mar. 2022.
- [59] S. Di Meo et al., "On the feasibility of breast cancer imaging systems at millimeter-waves frequencies," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 5, pp. 1795–1806, May 2017.



Neda Rojhani (Member, IEEE) received the Ph.D. degree in electronics and electromagnetism engineering from the University of Florence, Florence, Italy, in 2019.

She currently works with the Department of Information Engineering, University of Calgary, Calgary, AB, Canada. She is particularly interested in artificial intelligence, machine learning, deep learning, and optimization algorithms, with the objective of enabling everyone able to take advantage of potentialities given by new emerging technologies. Her

research interests include ground-based synthetic aperture radar (GB-SAR), multi-input-multioutput radar (MIMO), synthetic aperture radar interferometry (InSAR), ground-penetrating radar (GPR), compressive sensing, and Cramer–Rao bound, as well as antenna design and antenna theory.



Marco Passafiume (Member, IEEE) was born in Florence, Italy, in February 1987. He received the master's degree (cum laude) in electronic engineering from the University of Florence, Florence, in 2012, and the Ph.D. degree from the Department of Information Engineering, University of Florence, in 2016.

He is currently working at the Microelectronics and HF Research Group, University of Florence. His research activities cover developing embedded systems and digital systems for communication (also

working with private institutes, such as Autostrade Tech., Florence, for development of new infomobility systems), highly integrated System on Chip and System in Package (SoC/SiP), and human interface devices. Actually, he is studying deeply new development technologies based on field-programmable gate arrays (FPGAs) and he covers in depth various aspects related to computer science evolution (artificial intelligence, and data and signal processing) and digital communications/software-defined radios. His scientific interests also cover the area of data elaboration methodologies with particular attention to algorithmic development. He is particularly interested in human–interface device (HID) development and artificial intelligence applied to biomedics, with the goal of making everyone able to take advantage of potentialities given by new emerging embedded and computational technologies.

Dr. Passafiume was a recipient of the IEEE Microwave Magazine 2016 Best Paper Award.



Mehdi Sadeghibakhi received the master's degree in computer engineering from the Ferdowsi University of Mashhad, Mashhad, Iran, in 2021.

His research interests include computer vision, image processing, deep learning, machine learning, and medical image processing. He is also particularly interested in explainable artificial intelligence (XAI) and data analysis in the field of healthcare.



Alessandro Cidronali (Senior Member, IEEE) received the Laurea and Ph.D. degrees in electronics engineering from the University of Florence, Florence, Italy, in 1992 and 1997, respectively.

From 1999 to 2003, he was a Visiting Researcher with the Physics Science Research Laboratory, Motorola, Tempe, AZ, USA. He is currently a Professor with the Department of Information Engineering, University of Florence, where he teaches courses on electron devices and integrated microwave circuits. His research activities concern

the study of analysis and synthesis methods for nonlinear microwave circuits, the design of broadband microwave integrated circuits and the development of computer-aided design (CAD) modeling for microwave devices and circuits.

Dr. Cidronali was a recipient of the Best Paper Award presented at the 61st ARFTG Conference and the IEEE Microwave Magazine 2016 Best Paper Award. From 2004 to 2006, he was an Associate Editor of the IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES. He is also the Vice-Chair of TMTT-27 Connected and Autonomous Systems Committee a Technical Committee of the Microwave Theory and Techniques Society.



Giovanni Collodi (Member, IEEE) was born in Florence, Italy, in 1966. He received the M.S. degree (Hons.) in electronic engineering and the Ph.D. degree in computer science and telecommunication engineering from the University of Florence, Florence, in 1996 and 2002, respectively. His Ph.D. work focused on monolithic microwave integrated circuit (MMIC) circuit design and device modeling.

From 2001 to 2004, he was a Post-Doctoral Researcher with the Arcetri Astrophysical Observatory, Florence, where he was involved in the

ATACAMA large millimeter array radio telescope project (integrated circuit design), and the Department of Information Engineering, University of Florence. In 2002, he was a Visiting Researcher with the Physical Sciences Research Laboratory, Motorola, Tempe, AZ, USA, and a Contract Professor of electronics with the University of Florence. In 2005, he was an Assistant Professor with the Department of Information Engineering, University of Florence, where he was involved in developing systems for wireless applications. In 2006, he began his involvement also with the Interdepartmental Centre Technology and Microsystems for Quality and Environmental Safety. Since 2007, he has been teaching a course on technology and systems for wireless applications. In 2015, he started teaching a second course on high-frequency electronic microsystems and circuit with a particular emphasis on MMIC.

Open Access funding provided by 'Università degli Studi di Firenze' within the CRUI CARE Agreement