Multitask Learning Approaches Towards Drone Characterisation With Radar

Pappas, Apostolos; de Wit, Jacco J.M.; Fioranelli, Francesco; Jacobs, Bas

DOI
10.1049/rsn2.70012

Publication date
2025

Document Version
Final published version

Published in
IET Radar, Sonar and Navigation

IET Radar, Sonar & Navigation

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

Multitask Learning Approaches Towards Drone Characterisation With Radar

Apostolos Pappas[1] | Jacco J. M. de Wit[2] | Francesco Fioranelli[1] | Bas Jacobs[2]

[1]Microwave Sensing, Signals and Systems Group, TU Delft, Delft, The Netherlands | [2]Department of Radar Technology, TNO, The Hague, The Netherlands

Correspondence: Apostolos Pappas (a.pappas-2@tudelft.nl)

Received: 3 June 2024 | **Revised:** 9 December 2024 | **Accepted:** 18 February 2025

Handling Editor: Brian W.-H. Ng

Funding: The authors received no specific funding for this work.

Keywords: drone classification | multitask learning | radar micro-Doppler

ABSTRACT

For the effective deployment of countermeasures against drones, information on their intent is crucial. There are several indicators for a drone's intent, for example, its size, payload and behaviour. In this paper, a method is proposed to estimate two or more of the following four indicators: a drone's wing type, its number of rotors, the presence of a payload and its mean rotor rotation rate. Specifically, three multitask learning (MTL) approaches are analysed for the simultaneous estimation of several of these indicators based on radar micro-Doppler spectrograms. MTL refers to training neural networks simultaneously for multiple related tasks. The assumption is that if tasks share features between them, an MTL model is easier to train and has improved generalisation capabilities as compared to separately trained single-task neural networks. The proposed MTL approaches are validated with experimental data and in a variety of combined classification and regression tasks. The results show that MTL approaches can provide improvement in several tasks compared with conventional approaches.

1 | Introduction

The usage of drones has seen significant growth which is projected to persist in the foreseeable future. Beyond recreational purposes, drones are increasingly utilised in many commercial applications including surveillance of vital infrastructure and construction sites, and potential delivery services for package transportation [1].

In parallel with the ongoing advancement of commercial applications, the unlawful utilisation of drones is also on the rise. An example of such use involves the smuggling of drugs across borders and, on a smaller scale, the delivery of contraband into prisons including substances such as drugs, cell phones or small firearms. Another example is the abuse of drones for politically motivated actions and the disruption of public gatherings. In the most extreme scenarios, drones could be employed for vicious acts of terrorism. Moreover, within the military domain, there is a growing presence of drones on the battlefield, serving various purposes. Small unmanned aerial vehicles (UAVs), for instance, are deployed as weapon carriers, reconnaissance platforms or forward observers. Additionally, aside from military-grade UAVs, commercially accessible drones are also utilised in military operations as observers or for the delivery of grenades or improvised explosives [2].

In summary, even small drones (i.e., with a maximum weight of 20 kg), pose a significant threat in security contexts. The approach to countering them varies depending on the specific situation, the drone's intent and its payload. For instance, in safeguarding a public event, caution must be taken to avoid disproportionate collateral damage when intercepting a drone.

Hence, before taking action it is crucial to ascertain intent [3] including whether the drone carries a potentially lethal payload. Consequently, thorough gathering of information on a drone's intent is essential for the effective and safe deployment of countermeasures.

Regarding radar sensors, both the track characteristics and micro-Doppler signature of a drone provide relevant information on their attributes, such as the number of rotors, wing type, payload presence and behaviour, all of which serve as indicators of its intent. In this context, the paper focuses on employing multitask learning (MTL) to simultaneously extract a subset of four indicators, namely, wing type, number of rotors, payload presence and/or mean rotor rotation rate, from a drone's radar micro-Doppler signature. MTL involves training a neural network concurrently for different yet interconnected tasks based on a shared data representation [4]. The underlying assumption is that if tasks share features, the overall MTL network is easier to train compared to separate networks optimised for individual tasks.

Compared to the preliminary results [5], this extended study aims to provide a more comprehensive view on the use of MTL for radar-based characterisation of drones. To this end, we provide a more extensive literature review on the state-of-the-art of MTL in radar applications. Moreover, we expand on our description of the proposed models as well as the types of tests created considering the size and diversity of the dataset at hand. Finally, we present the results of this more extensive and complete assessment alongside its derived conclusions. It is shown that MTL can improve classification performances of a task considered priority thanks to its joint combination with auxiliary tasks, as for example, demonstrated for the payload detection task that has considerable operational importance.

The rest of the paper is organised as follows. The MTL concept is discussed in Section 2, where the state-of-the-art of MTL is evaluated as well as applications of it in the field of radar. The specific multitask problem for radar-based drone characterisation is explained in Section 3, along with the radar dataset and the implementation of the multitask networks. Results are presented in Section 4 with conclusions given in Section 5.

2 | Multitask Learning

Multitask learning is a learning framework intended for training neural networks simultaneously for multiple related tasks using a shared data representation [4]. This may be counter-intuitive in what is often seen as the norm in machine learning which is to address complex problems or tasks by segmenting them into smaller and preferably independent problems. These smaller tasks are then addressed individually by defining independent machine learning models (e.g., neural networks), trained and optimised for each of these tasks separately.

This single-task approach that combines the outputs of individual models is referred to as single-task learning (STL). STL was shown in works such as Waibel et al. [6] to be counterproductive in some cases as it discards useful information that could be exploited by other tasks related to the same general problem. In simpler words, feature representations that are produced and possibly discarded by one task might be useful in performing another. This fact serves as the ground on which MTL was conceived: if tasks share knowledge, then the resulting MTL model may learn more easily and generalise better compared to the separate STL counterparts.

From an analysis of the current literature, such as the work of Thung et al. [7], it can be concluded that the main families of MTL models are the so-called hard parameter sharing (HPS) and soft parameter sharing (SPS) approaches. In the first case a common feature extractor is used, followed by individual branches depending on the number of tasks as is illustrated in Figure 1 (left) for three tasks. On the other hand, SPS approaches involve the co-existence of multiple per-task models that are trained in a joint manner by exchanging information (i.e., activation maps or weight values) among them. This approach is shown in Figure 1 (right) for three tasks.

2.1 | State-of-the-Art in MTL

A plethora of research works have directly adopted HPS as means towards performing MTL. For instance, Zhang et al. [8] proposed a task-constrained deep convolutional network to

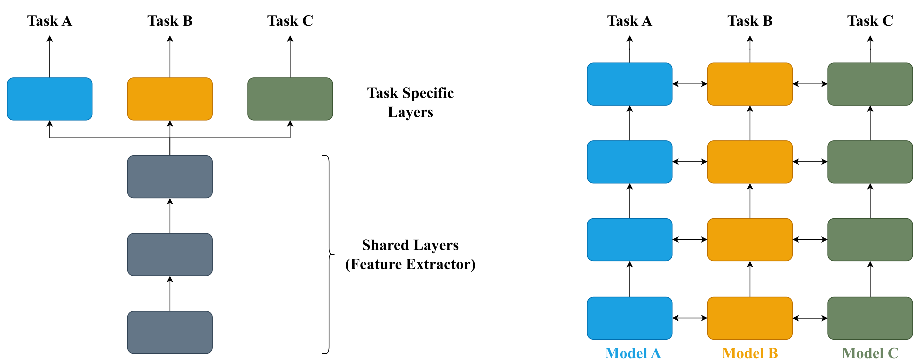


FIGURE 1 | Schematic representation of the HPS MTL approach (left): the shared layers serve as the feature extractor of the network while there are three independent task specific layers. Schematic representation of the SPS MTL approach (right): in this approach instead of shared layers, three different models are created.

jointly optimise facial landmark detection with a set of related tasks. This approach retains, in its essence, the same configuration as seen in Figure 1 (left). It consists of a feature extraction unit, common for every task, followed by the branching dictated by the defined tasks. The task specific layers and architectures can be arbitrarily chosen to fit the purpose of the corresponding task.

Expanding on the notion of HPS, Long et al. [9] proposed the so-called deep relationship network. There, following the shared convolutional layers are the task-specific fully-connected layers. The novelty of this work is that matrix priors are placed on the fully-connected layers that allow the model to learn the relationship between tasks. Yet, as denoted by Ruder [10], this method is still reliant on the existence of a known sharing structure and might be inadequate for novel applications such as in this study on radar-based drone characterisation.

Misra et al. [11] developed the idea of cross-stitch networks, see Figure 2, mainly as a way to counter the issue of deciding where to split the HPS architecture into task-specific layers. Dictated by the paradigm of SPS, the authors propose the stitching of predefined models for each task using the so-called cross-stitch units defined by the following equation:

$$\begin{bmatrix} \tilde{x}_A^{ij} \\ \tilde{x}_B^{ij} \end{bmatrix} = \begin{bmatrix} w_{AA} & w_{AB} \\ w_{BA} & w_{BB} \end{bmatrix} \begin{bmatrix} x_A^{ij} \\ x_B^{ij} \end{bmatrix} \tag{1}$$

where (i, j) indicate a specific part of the activation map and A and B define the corresponding task layers. The linear combination of the activation maps is parameterised by w, varying from 0 to 1, a parameter that is learnt during training. Hence, in this case if the diagonal of w is set to 1, meaning that $w_{AB} = w_{BA} = 0$, then the two networks discard the information of one another at that particular layer, corresponding single-task learning. It is shown that the cross-stitch networks perform and generalise better than both STL and MTL baseline methods.

Ruder [10] built upon the notion of cross-stitch networks by introducing a novel meta-architecture named sluice networks. This approach encompasses a variety of methods such as HPS, SPS and cross-stitch units. Their approach enables learning of what layers should be shared and with what weighting while also offering a number of skip connections at the outer and final layers.

Yet, the concept of SPS is not limited to in sharing activation or feature maps among the individually defined networks. The authors in Ref. [12] tackled the question of task relatedness by proposing a dynamic multitask convolutional neural network. The use of this MTL model, comprising multiple per-task convolutional neural networks stitched with task-transfer connections, results in the tasks forming weak groups spontaneously during training. As in the cross-stitch networks, when communicated information among the networks is ignored, each subnet works independently as an STL model.

2.2 | MTL in Radar Applications

Few works may be found in literature that employ MTL as a means to perform joint classification and/or parameter estimation. The existing works mainly focus on classification of synthetic aperture radar (SAR) data and human activity recognition. It is also worth noting that in all of these works, the number of tasks to be predicted is limited to two or maximum three. It is furthermore notable that common ground in all these works is the preference towards hard parameter sharing approaches.

Du et al. [13] proposed the use of multitask learning towards improving target recognition in targets acquired using SAR. To that end, three distinct tasks were defined, namely: separation of the target from the shadow, target's aspect angle estimation and target recognition. Following an HPS mechanism for multitask learning, the authors consider the target recognition as the main task and the rest as auxiliaries, performed only to improve the performance on the main task. The authors manage to prove the benefits of multitask learning by comparing the performance of STL and MTL models, using a varying number of samples. In all cases, training both auxiliary tasks alongside the main one resulted in considerable improvement in target recognition accuracy.

Furthermore, in the field of SAR target recognition, Chen et al. [14] combined the concepts of meta and multitask learning and designed a multitask metalearning recognition algorithm framework (MMRAF) alongside a variable combinatorial loss to

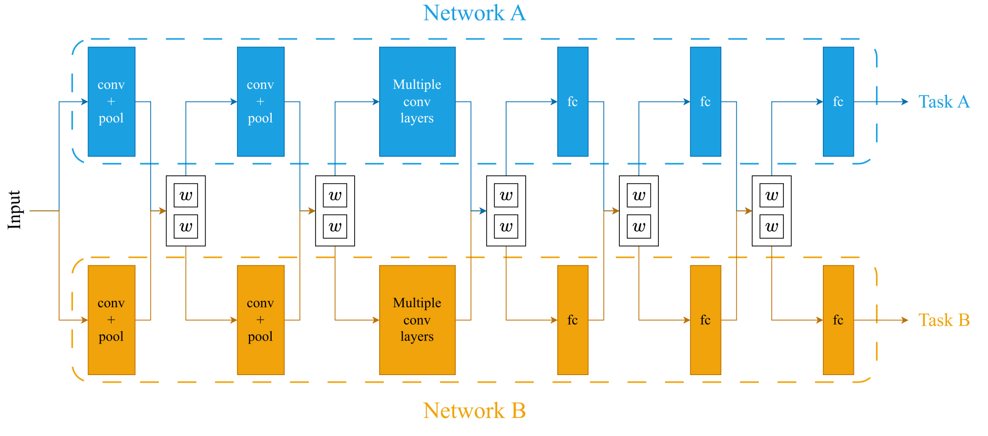


FIGURE 2 | Schematic representation of the cross-stitch network paradigm introduced by Misra et al. [11]. Figure adapted from that same paper.

mitigate the network overfitting problem while extracting more discriminative features for the main task. Their work was carried out in the context of very small training sets. Once again, N auxiliary metatasks are constructed, each one consisting of a main recognition and an auxiliary task. In the proposed network, the main and the auxiliary tasks all share the same ResNet-18 backbone network, leaving only a fully connected layer as a per-tasks unique layer. The application of the proposed MMRAF framework on the publicly available MSTAR dataset showcased the ability of the framework to generalise better and achieve superior performance compared to other methods.

In the domain of human activity recognition, Li et al. [15] proposed a deep multitask network capable of predicting human activity and also perform person identification simultaneously using the motion capture (MOCAP) dataset provided by Carnegie Mellon University. This joint classification task was achieved by introducing a custom convolutional multi-scale residual attention network (MRA-Net). This network was composed of two major parts; the feature extractor and the multitask classifier. The feature extractor takes spectrograms as its input and extracts the features, whereas the multitask classifier performs the two classification tasks. To make the training of the model possible, the authors defined the overall loss as a weighted sum of the corresponding losses of the two tasks. These weights were then defined by a greedy search approach based on the accuracy achieved by the models for each task separately. It is shown in their results that the MTL approach provided an improved accuracy score for both tasks, more prominently in the person identification where it surpassed the STL approach by 2.54%.

Regarding radar detection of drones, MTL was proven successful in the work by Wang et al. [16]. Inspired by the advancements of MTL in image recognition, the authors propose the use of a multitask learning model as an alternative of the well-known constant false alarm rate (CFAR) detector algorithm. The authors propose an MTL model predicting two tasks simultaneously with a shared backbone, leading to a HPS model with two branches: one for classifying the input range-Doppler map patch into target present or target absent, and the other for the regression of offset between the target and the patch centre. The results of this work were very promising for the use of multitask learning for drone detection. Performing experiments on both measurements and simulations showed that the proposed method was able to locate the target more accurately and achieve a much lower false alarm rate at a comparable detection rate than a CFAR detector.

Drone characterisation using MTL with radio frequency (RF) signals as input is presented in the work of Akter et al. [17]. There, the authors employed a three-branched HPS model for three different tasks: drone detection, type identification and activity recognition. The drone detection task was defined as a two-class classification task (drone or not a drone). The type identification task was defined as a four-class classification task (Parrot Bebop drone, AR drone, DJI Phantom drone or noise). Finally the activity recognition task was defined as a twelve-class classification task (12 different modes of operation were defined). The drone detection task was not improved compared to the STL approach, whereas an increase in performance of 1.9% and 6.59% was shown for the type identification and activity recognition tasks respectively, compared to the STL approach.

3 | MTL Framework for Radar-Based Drone Characterisation

To support the efficient and safe deployment of proportionate countermeasures against drones, it is necessary to gather as much information as possible regarding their intent. Key radar observable variables include radar cross section (RCS), tracking patterns and micro-Doppler characteristics. The RCS of a drone is influenced by factors such as its size, shape, payload and material properties. Generally, there exists a limited correlation between a drone's RCS and its intentions [3]. Moreover, the observed RCS can fluctuate significantly as function of aspect angle. Therefore, the literature emphasises the utilisation of tracking behaviour and micro-Doppler features for radar-based drone characterisation.

By combining both track and micro-Doppler characteristics, it becomes possible to discern whether a drone belongs to the fixed-wing or rotary-wing category [18, 19]. This classification offers insights into the drone's intended function. For instance, rotary-wing drones are well-suited for observation tasks. They can maintain a stationary position and ascend rapidly and vertically, emerging suddenly from obstructed areas such as buildings or foliage. Conversely, fixed-wing drones typically have greater payload capacities and endurance, making them preferable for prolonged surveillance over large regions. Additionally, micro-Doppler features enable estimation of the drone's rotor count and rotor diameter, serving as proxies for its size and maximum payload capacity [20]. Furthermore, the rotation rate of the rotors may indicate the presence of a payload with multicopter drones carrying heavier loads generally exhibiting higher rotation rates compared to those without payloads.

Based on the aforementioned considerations, this study proposes MTL approaches to concurrently perform two or more of the four following tasks:

1. Classification of the wing type. In this study, three different fixed-wing and rotary-wing drone classes are distinguished: fixed-wing drones, helicopters and multicopters.
2. Classification of the number of rotors. This task is defined as an eight-class classification problem; the number of rotors can be one to eight.
3. Payload detection. This task is defined as a two-class classification problem; payload present or no payload present.
4. Mean rotor rotation rate estimation. This task is defined as a single continuous value-regression task.

The outcomes of these four tasks collectively yield insight into the potential role and intent of a drone.

3.1 | Proposed Multitask Models

Three MTL architectures are proposed in an effort to assess the performance of different approaches for varies combinations of tasks. All three architectures are based on the HPS paradigm and use the pre-trained ResNet-18 model as a backbone network. The ResNet-18 model is provided by the TorchVision package. It includes a 7 × 7 kernel as first layer, four convolutional layers and finally a single fully-connected layer. Pytorch was used for a deterministic implementation of ResNet-18, ensuring full reproducibility of the results. The three proposed MTL architectures differ in the position where the ResNet-18 backbone network is split into task-specific layers and how information is shared between the task-specific layers. The three proposed MTL architectures are referred to as the *Simple Model*, the *Adjusted Model* and the *Info-Sharing Model,* respectively:

- In the *Simple Model*, the backbone network is split into task-specific layers after the fourth convolutional layer. This approach results in similar fully connected task-specific layers. The shared layers keep the pre-trained weights of the original ResNet-18 backbone as seen in Figure 3 (left). The weights of the task-specific fully connected layers are tuned to the specific task.
- In the *Adjusted Model*, the backbone network is split into task-specific layers between the third and fourth convolutional layers. This approach results in two task-specific layers, one convolutional layer and a fully connected layer, giving more emphasis to the individual tasks. With this approach, as seen in Figure 3 (centre), the two task-specific layers may be tuned more efficiently to the specific task.
- In the *Info-Sharing Model*, the backbone network is also split into task-specific layers between the third and fourth convolutional layers but information is shared again between the task-specific branches after the fourth convolutional layer, see Figure 3 (right). This approach is partially inspired by the soft parameter sharing paradigm and the multiscale residual attention network of Li et al. [15]. The influence of soft parameter sharing can be seen in the exchange of information between the task-specific branches in an effort to benefit from each other's extracted features. Considering the two-task example of Figure 3 (right), information sharing begins with a simple concatenation of the feature maps A and B of the fourth convolutional layers. Considering the dimensions of each feature map (H, W, C), where H is the height, W the width and C the number of channels, and the dimensions of the concatenated feature maps are $(H, W, 2C)$. In this way, each branch exploits its own produced feature space and the feature space produced by the parallel branch. In an SPS approach, the two extracted feature maps would be combined together via a weighted sum such as in the case of the previously mentioned cross-stitch networks. In the info-sharing model, which is in the HPS realm, the feature maps are concatenated by applying a 1 × 1 convolution kernel. Hence, the concatenated feature maps can be seen as a weighted linear sum of the individual task-specific feature maps. This operation is illustrated in Figure 4, where the unit convolution not only combines the task-specific feature maps but also reduces the dimensions back to (H, W, C). In the case of more than two tasks, the same approach is followed, leading to concatenated feature maps with dimensions (H, W, nC), where n denotes the number of tasks in the MTL model.

Note that the MTL model architectures are based on the full ResNet-18, comprising all four convolutional layers. During this study, it was noticed that using a shallower version of ResNet-18 as a backbone, in some circumstances, led to better performance. However, it was deemed outside the scope of the current study to investigate different architectures and analyse this phenomenon in detail.

3.2 | Choice of the MTL Loss Function

Three out of the four tasks assessed in this study are classification problems. Hence, for the wing type and number of rotors classification tasks the categorical cross-entropy loss was utilised, whereas for the payload detection the binary cross-entropy loss was employed. For the mean rotor rotation rate regression task the mean average error (MAE) was used as a cost function during training of both STL and MTL models. A very important element in MTL is the combination of all tasks losses in a single overall loss. Two approaches were examined in this work: the naive summation of the individual losses and the loss-balanced

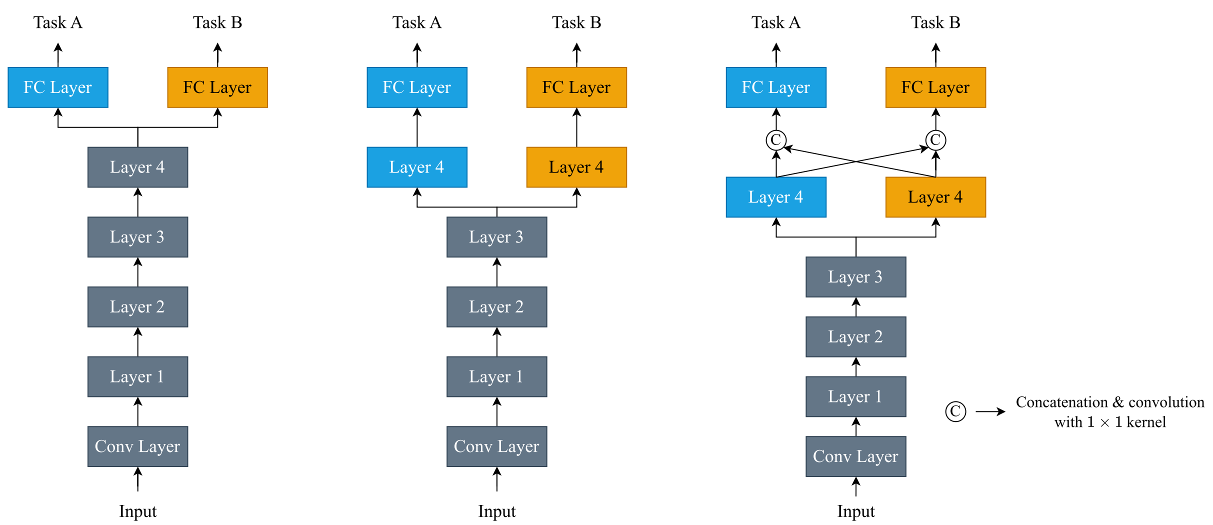


FIGURE 3 | Schematic representation of the MTL models proposed in this study: the simple model (left), the adjusted model (middle) and the info-sharing model (right).

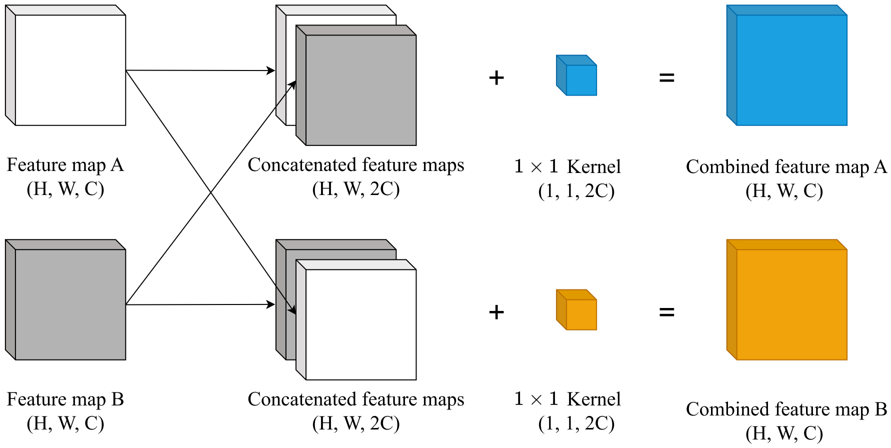


FIGURE 4 | Schematic representation of the information sharing module in the info-sharing model.

task weighting (LBTW) approach developed by Liu et al. [21] to reduce the effect of negative transfer among tasks (i.e., cases where multitask models perform worse than their STL counterparts). LBTW is defined as the weighted sum of the individual task losses. For each task and epoch, LBTW stores the first batch's loss of each task. It then sets each task weight as the ratio of the current batch loss to the benchmark loss, raised to the power of a free parameter α, the value of which can be selected following a greedy approach. In this manner, poorly trained tasks (i.e., those where the loss is progressively increasing within the epoch) produce ratios closer to 1, having a greater effect on the joint loss.

3.3 | Comparing STL and MTL Models

In assessing the potential gains in performance that MTL provides when used in radar applications, two major challenges are presented. Firstly, comparing the performance of the MTL models proposed in this work to others found in the literature is complex due to the very different data used for training and testing. Secondly, as to the best of our knowledge no works applying MTL to radar data can be found in the open literature, therefore no direct comparisons with other approaches can be made. Hence, the MTL models are directly compared to their STL counterparts that are seen as the most correct baseline for comparison. The remainder of this section is dedicated to describing the comparison conditions, presenting restrictions regarding the models and defining the performance metrics.

3.3.1 | Model Restrictions

One additional reason for selecting a specific backbone network was to pose boundaries to the architecture of the proposed models and to define a fair platform for performance comparisons. Posing such boundaries is required to constrain the optimisation space, since performing a full optimisation at model level would be very time consuming (and outside the scope of the current study). The use of a backbone network reduces the optimisation procedure to only optimising the defined models by hyperparameter tuning. However, this approach may disadvantage tasks for which the selected backbone network is not optimal.

To make a fair comparison between STL and MTL models, the following restriction is applied: the path from input to output for each task must be identical in the related STL and MTL models. The info-sharing model is an exception to this restriction, since it includes a processing step that cannot be implemented in an STL fashion without being considered as a joint classification approach. Yet, the architectures of the Info-Sharing Model and its STL counterparts remain comparable. What is not constrained is the optimisation of each STL or MTL model. The optimisation and training process is unique for each model, based on the notion that to provide a fair comparison, the training of the STL models should be not influenced by the training of the MTL models (i.e., different hyperparameters were allowed for training the MTL and STL networks).

3.3.2 | Performance Metrics

The performance comparison of the STL and MTL models is done in two ways: task-wise and jointly. In the task-wise comparison, the F_1 score is reported for the individual classification problems and the MAE is reported for the regression task. For the two-class classification problem, the positive class F_1 score is provided, whereas for the three-class classification problem, the macro-averaged F_1 score is given. In case of task combinations including only classification problems, the joint accuracy is also given, where for each sample all individual classification outcomes must be correct in order for the sample to be considered as correctly classified.

3.3.3 | Conducted Tests

To evaluate the performance of the proposed MTL models, two types of tests were conducted. A test based on the cross-validation approach and tailored robustness tests to evaluate the performance in more difficult circumstances:

- *Cross-Validation Approach* The size and variability of the available data make a simple train/test split susceptible to external biases when manually, or even randomly, dividing the data into training and test set. It might occur for instance that the test set comprises the hardest targets to classify and vice versa. Thus, the results can be either too optimistic or too pessimistic, depending on the types of targets and the signal-to-noise ratio (SNR) regimes found in the test set. To mitigate this issue, a recursive train/test split method, inspired by the well-known cross-validation approach, is proposed. The dataset containing 0.5 s spectrograms is first shuffled, then, after shuffling, 10 unique and nonoverlapping subsets are defined. Then, 10 models are trained from scratch with nine subsets of data and tested with the remaining 10th subset of data.

- *Robustness Test* The test configuration described above, assesses the performance of the classifiers using different train and test splits, minimising the effect of bias and obtaining statistics on the models' performance by training and testing using performing multiple unique splits. However, this approach comes with a flaw: since the train/test split of each fold is performed on the segmented spectrograms level, spectrograms belonging to the same target track can be found in both the training and test set. This results in the models learning and adapting to different SNR conditions for the same targets, leading to possibly overly optimistic outcomes. In an effort to assess the performance of all models in every task combination scenario possible, the dataset is manually split in specific training and test sets in order to assess generalisation over unseen measurement scenarios or even unseen targets. The manual splits are discussed in the related sections.

4 | Results of the Proposed MTL Models

In this section, the proposed MTL models are evaluated with different combinations of the tasks listed in Section 3. Three different task combinations are considered within this study:

1. The combination of the wing type classification and number of rotors classification tasks. The idea behind this combination is that fixed-wing drones generally have only one or two propellers, whereas multicopters may have four, six or even eight rotors. So there is a relation between the wing type and the expected number of rotors.
2. The combination of the wing type classification, number of rotors classification, and payload detection tasks. The ground for this combination is that information on the wing type and the number of rotors of a drone, may provide a rough indication of its size. A larger drone is more likely to carry a heavy payload. In this case, the assumed relation between the tasks may be weak.
3. The combination of the mean rotor rotation rate estimation and payload detection tasks. The idea underlying this combination is that in particular multicopters control lift by adapting the rotor rotation rates. When carrying a heavy payload, multicopters therefore have to increase the rotor rotation rates to produce more lift.

Please note that from an operational point of view, detecting whether a drone carries a payload is most valuable. Consequently, the payload detection task can be seen as the main task with highest priority. The other tasks are therefore considered auxiliary tasks, performed simultaneously in an MTL fashion to improve the performance on the payload detection task.

4.1 | Dataset

The data considered for this work are produced by experimental measurements using an X-band continuous wave (CW) radar. The targets present in the dataset are various types of drones ranging from commercially available to homemade drones. In total, 110 different measurements of different duration are available. A spectrogram is generated from each measurement, with an example shown in Figure 5. Since the measurements are of varying duration, so are the spectrograms. In order to create an appropriate dataset, these initial spectrograms were split into segments of constant duration of 0.5 s resulting in a total of 8298 samples. Labelling of the mean rotors rotation rate was carried out manually on the basis of cepstrograms of the dataset. In some cases, the mean rotation rate could not be reliably estimated, for example, if the signal-to-noise ratio was too low. This reduced the size of the dataset suitable for the rotation rate estimation problem.

Considerable class imbalance is present in the dataset. This is notable when considering the number of helicopters for the wing type classification task and the number of hexacopters for the number of rotors classification task. This imbalance is illustrated with the histograms presented in Figure 6. A serious imbalance is also present in the dataset used for the payload detection task; for each sample of a drone carrying a payload there are about 15 samples of a drone without payload.

4.2 | Wing Type Classification and Number of Rotors Classification

The task combination first presented comprises the wing type and number of rotors classification tasks. In all MTL models the

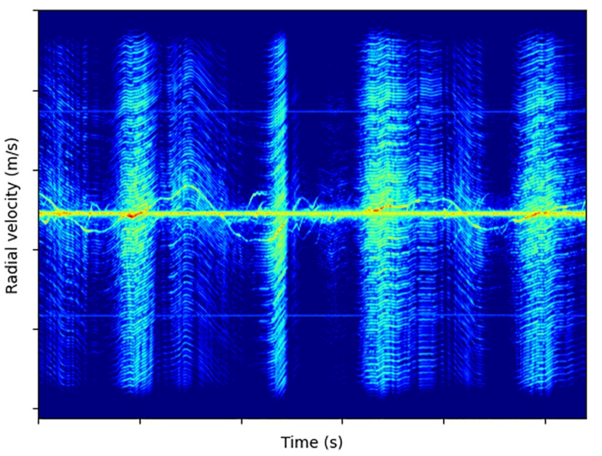


FIGURE 5 | Example of a measured spectrogram (before the split into 0.5 s segments) of a radio-controlled helicopter. It should be noted that both axes tick labels have been omitted because of data confidentiality reasons. Also, the effect of clutter present in the measurements at the line formed around the zero Doppler mark is possibly due to moving branches and leaves of trees present in the measurement scene.

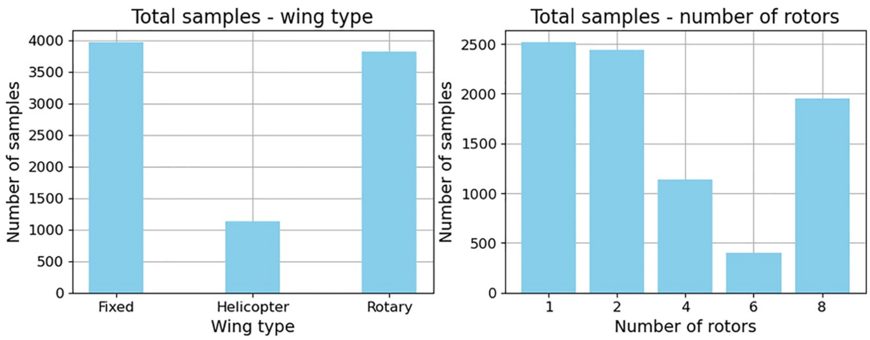


FIGURE 6 | Histograms showing class imbalance in the wing type and number of rotors features throughout the available dataset.

stochastic gradient descent (SGD) optimiser was used with a learning rate of $1 \cdot 10^{-3}$ and a momentum of 0.9. For the wing type classification task, an STL model was trained using the Adam optimiser with a learning rate value fixed at $5 \cdot 10^{-5}$. Training the STL model for the number of rotors classification task was carried out using the same hyperparameters as for the MTL models.

4.2.1 | Cross-Validation Approach

The results of the cross-validation are presented in Table 1 and Figure 7 for both the STL and MTL models. All MTL models achieved higher performance for both tasks when compared to the individually optimised STL models. The most significant improvement in F_1 score performance for both tasks is achieved by the info-sharing model, showing improvements of 1.2% and 2.98%, respectively. This is also apparent in the joint accuracy comparison boxplot of Figure 7; an increase of 4.53% is obtained by using the info-sharing model.

4.2.2 | Robustness Test—Varying Measurement Conditions

To further assess the performance of the proposed classifiers in a more realistic manner, a robustness test was defined in which the data from a complete measurement campaign were put in the test set. In this way, it was guaranteed there were no samples of the same measurement campaign in both the training set and test set. Since the measurement conditions differ from campaign to campaign, this test assesses the generalisation capabilities of the classifiers when facing different measurement conditions, for example, different SNR regimes and different flight manoeuvres. For this test only measurements of the common, commercially available drone types were manually selected. Details on the resulting test set are summarised in the corresponding Table A1.

Training on the rest of the dataset and testing on the aforementioned selected samples yielded the joint accuracy scores presented in Table 2. There, an increase of 2.93% can be found when the test set is comprised of commercially available drones and the info-sharing model is utilised. It is important to highlight that all MTL models managed to outperform their STL counterparts with varying performances.

TABLE 1 | Wing type and number of rotors classification tasks F_1 score performance comparison.

	F_1 scores	
Model	Wing type	Number of rotors
STL	94.36 ± 0.5	86.47 ± 0.66
Simple model	95.27 ± 0.59	88.58 ± 1.05
Adjusted model	95.35 ± 0.45	88.32 ± 1.16
Info-sharing model (% Increase)	**95.56 ± 0.41** (**+1.2%**)	**89.45 ± 1.22** (**+2.98%**)

Note: The best performance in each case and the corresponding increase are highlighted in bold.

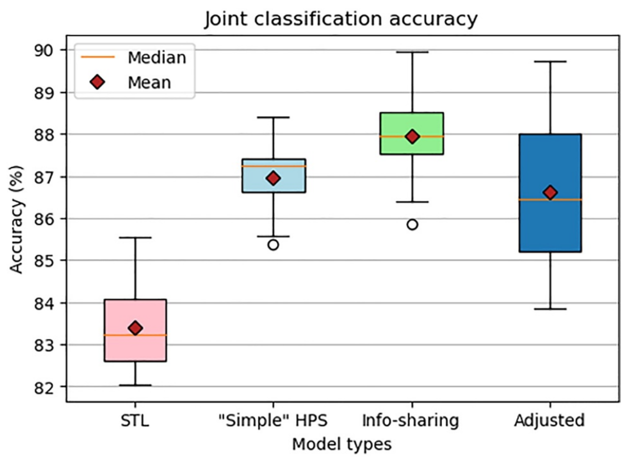


FIGURE 7 | Joint classification accuracy regarding the wing type classification and number of rotors classification tasks.

4.2.3 | Robustness Test—Irregular Target Types

A second robustness test was performed to test the generalisation capabilities of the classifiers regarding unseen targets and in this case even irregular target types. For this test, all measurements of irregular target types, that is, targets exhibiting some unique characteristics, were manually selected and put in the test set. With this approach it was guaranteed that the test set only included irregular drone types that were not represented in the training set. The resulting test set is summarised in the corresponding Table A2.

TABLE 2 | Joint accuracy (%) performance comparison between the MTL models and their STL counterparts for the case where the test set contains known targets in varying measurement conditions.

Model	STL	MTL simple model	MTL adjusted model	MTL info-sharing model
Joint accuracy	76.36	77	78.87	**79.29**

Note: The best performance is highlighted in bold.

Training on the rest of the dataset and testing on the aforementioned selected samples yielded the joint accuracy scores presented in Table 3 below. Even though the performance of all classifiers in this case does not yield satisfactory values, an increase in joint accuracy when employing the MTL can be observed. Each MTL model manages to improve the STL performance. More specifically, the simple model manages to achieve the best performance in this case, providing an improvement of 7.96%.

4.2.4 | Discussion

At first glance, it would appear that the use of either paradigm does not yield accurate results yyet, some considerations are needed, since the presented joint accuracy does not convey the task-wise advantages and the per-class classification performance of each approach. For instance, it was noticed that significant improvements were introduced in the number of rotors classification task when utilising the MTL models. Other factors not immediately obvious in this analysis regard the nature and the target types included in the provided dataset. The diversity of target types within the dataset was so significant that confusion between the targets and their classes in certain cases was possible. For instance, a few of the custom-made drones present in the dataset would introduce some ambiguity when having more than two rotors but were not characterised as rotary wing drones. Hence, by only having micro-Doppler information because of the utilised CW radar, useful information regarding the observed drones that in other cases would be available are instead discarded.

Another point of discussion is the consideration of correlation between the task results. With the architectures proposed in this work, potential or known correlations between task results (i.e., a fixed wing drone cannot have eight rotors) are not explicitly enforced upon the models. This may lead to the proposed models predicting classes that may be contradictory or without any physical meaning. Hence, to prevent this phenomenon, each task branch could be informed about the predicted class of the other in an effort to artificially impose rules to the networks. Furthermore, additional modifications in the loss function could be imposed, so as to further punish the models for producing predictions that when aggregated lead to an unrealistic outcome.

4.3 | Wing Type Classification, Number of Rotors Classification, and Payload Detection

The task combination discussed in this section includes the wing type classification, number of rotors classification and payload detection tasks. Recalling that the payload detection task is the main task, the other two tasks are considered auxiliary tasks. The two STL models for predicting the wing type and number of rotors are reused as defined in the previous section. The MTL models were again trained using the SGD optimiser with a learning rate of $1 \cdot 10^{-3}$ and a momentum of 0.9. For the payload detection, the ResNet-18 backbone was trained using the SGD optimiser using a learning rate value fixed at $1 \cdot 10^{-3}$.

4.3.1 | Cross-Validation Approach

It is shown in Table 4 that compared to the combination of two classification tasks discussed in the previous section, a major drop in performance can be observed for the wing type and number of rotors classification problems. Even though on average the info-sharing model provides a 0.3% increase in F_1 score in both of them, it is not substantial. This drop in performance may be attributed to possible negative transfer between the three tasks because of the lack of a strong relation between the two multiclass classification problems and the additional payload detection task. This is more evident when using the simple model and adjusted model which in some cases performed worse than the baseline set by the corresponding STL models.

On the contrary, the payload detection task highly benefited from the joint training with the other two tasks. As Table 4 suggests, an increase in positive (true) class F_1 score of 12.82% was achieved by the simple model, followed closely in performance by the info-sharing model. The generally low F_1 scores can be attributed to the fact that measurements of the same drone type, with and without payload, were present in the training and test sets, oftentimes leading to a high number of false alarms. The improvement in payload detection performance also led to an improvement of the joint classification performance as seen in Figure 8. There, a joint classification accuracy increase of 7.3% with the info-sharing model can be observed.

4.3.2 | Robustness Test—Balanced Test Set

In this robustness test, further emphasis was put on the payload detection task. The test set comprised 144 samples of an octocopter carrying a payload and 138 samples of drones not carrying a payload. More specifically, a commercially available fixed-wing drone and a custom-built quadcopter constitute the drone types without a payload in the test set. This manual selection resulted in a small but balanced test set, balanced in terms of payload presence. The test set is summarised in the corresponding Table A3.

TABLE 3 | Joint accuracy (%) performance comparison between the MTL models and their STL counterparts for the case where the test set only contains irregular target types.

Model	STL	MTL simple model	MTL adjusted model	MTL info-sharing model
Joint accuracy	40.85	**48.81**	47.2	45.45

Note: The best performance is highlighted in bold.

TABLE 4 | Wing type classification, number of rotors classification and payload detection tasks F_1 score performance comparison.

	F_1 scores		
Model	Wing type	Number of rotors	Payload detection
STL	94.36 ± 0.5	86.47 ± 0.66	47.04 ± 3.71
Simple model	94.09 ± 0.84	86.37 ± 1.44	**59.86 ± 4.09 (+12.82%)**
Adjusted model	94.12 ± 0.69	86.66 ± 1.01	58.57 ± 3.9
Info-sharing model (% Increase)	**94.66 ± 0.83 (+0.3%)**	**86.77 ± 1.19 (+0.3%)**	59.79 ± 3.59

Note: The best performance in each case and the corresponding increase are highlighted in bold.

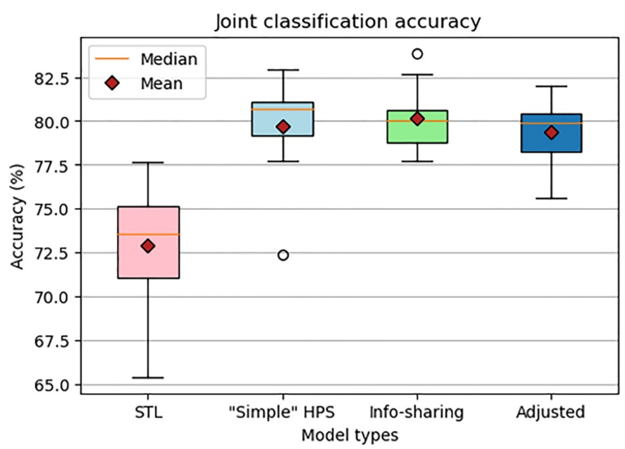


FIGURE 8 | Joint classification accuracy regarding the wing type classification, number of rotors classification and payload detection tasks.

The joint accuracy obtained with the discussed test set, is presented in Figure 9a. As can be seen, the MTL models outperform the STL models. The highest accuracy of 73.4% is obtained with the info-sharing model. An increase of 8.86% as compared to the accuracy obtained with the STL counterparts. For reference, the joint accuracy of the wing type and number of rotors classification tasks is also presented in the figure. Regarding the performance of this combination of these two classification tasks, the performance of all models is similar.

For a more in-depth analysis of the per-task performance, let us consider the F_1 score plot of Figure 9b. There, the negative transfer introduced by the additional payload detection task is evident. In the wing type classification task, all MTL models perform worse than their STL counterparts, whereas the MTL models perform better on the number of rotors classification task. For the latter task, an improvement of 2.41% was achieved when using the adjusted model. The task for which the use of MTL proved to be largely advantageous was the payload detection task.

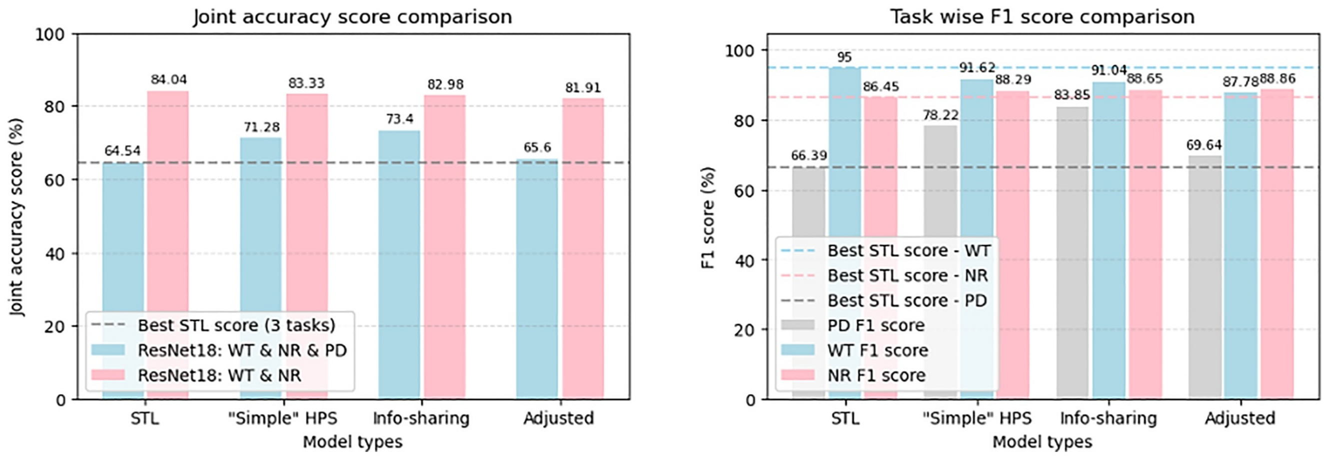
The ability of the simple and info-sharing models to correctly distinguish among drones carrying a payload and others that did not, as well as the much smaller number of false alarms, is made clear in the payload detection's F_1 score. The info-sharing model, in particular, obtained a score of 83.85%. This high score can be explained when considering the very small number of false alarms produced in the test set, having misclassified less than 10 samples as payload-carrying drones.

4.3.3 | Robustness Test—Imbalanced Test Set

The previous test gave some insight about the possible robustness of the algorithms when they are trained with a sufficient amount of data yet, the test set in that particular case was balanced, something that is very rarely the case in real-life scenarios. To account for this, a second robustness test was carried out, using the test set described in Table A1 supplemented with additional samples of an octocopter carrying a heavy payload. This particular split produced 478 drone samples without payload and 144 octocopter samples carrying a heavy payload. Table A4 shows more details of the drone types present in this test set.

In similar fashion as in the previous test, three separate task-specific STL models are trained, tested and then compared to the three defined MTL models created to predict the three tasks of this group in a joint manner. The joint accuracy is shown in the bar plot of Figure 10a. There, it is shown that exploiting MTL to jointly train these three tasks leads to superior performance over the STL counterparts. An increase in joint accuracy up to 8.04% can be seen when using the info-sharing model, following the conclusions of the previous test. The advantages of using MTL in this test are also evident when considering only the wing type and number of rotors classification tasks. There, all three MTL models achieve higher accuracy when compared to the baseline STL models with the most significant increase obtained by info-sharing model.

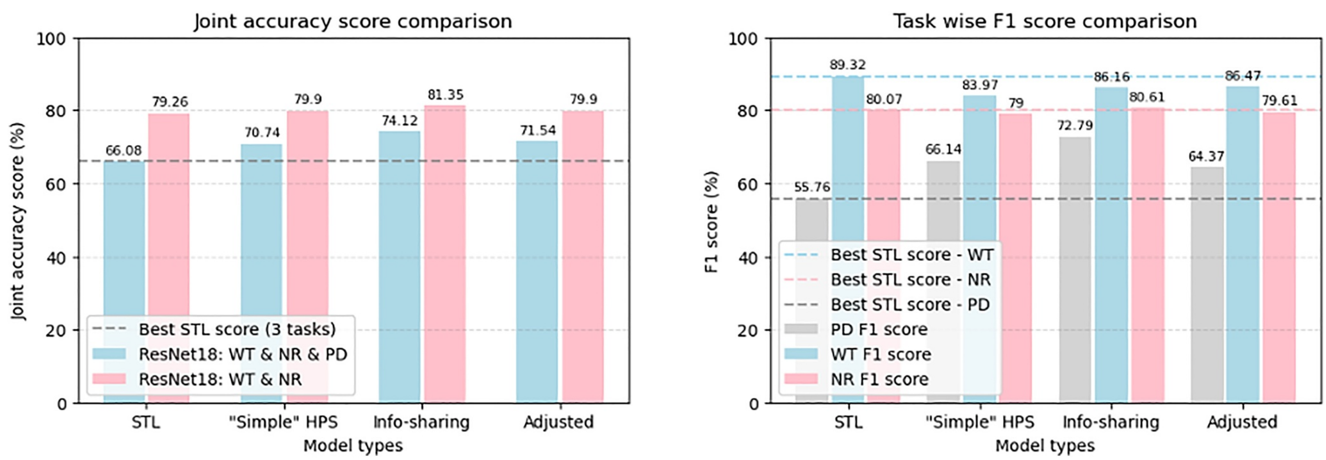
What was shown in the test of this section was also the possible negative transfer induced from the newly added payload



(a) Joint accuracy score comparison between MTL and STL models.

(b) Task-wise F_1 score comparison between MTL and STL models.

FIGURE 9 | Joint classification accuracy and task-wise F_1 score comparison plots for the wing type classification, number of rotors classification and payload detection tasks using the test set described in Table A3.



(a) Joint accuracy score comparison between MTL and STL models.

(b) Task-wise F_1 score comparison between MTL and STL models.

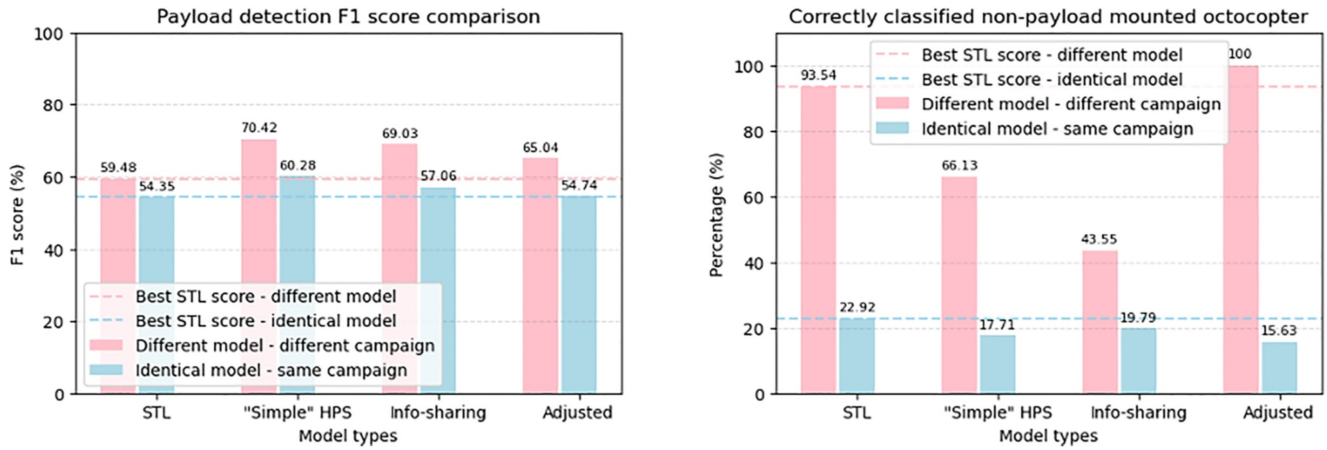
FIGURE 10 | Joint classification accuracy and task-wise F_1 score comparison plots for the wing type classification, number of rotors classification and payload detection tasks using the test set described in Table A4.

detection to the other two tasks of the group. This can be also observed in the F_1 score comparison plot of Figure 10b. There, the MTL models perform worse than the STL models, with the exception of the info-sharing model's performance regarding the number of rotors classification task, where it performed marginally better than its STL counterpart. The payload detection task, on the other hand, was much improved when exploiting MTL, with a +17.03% increase in F_1 score (from 55.76% to 72.79%) when utilising the info-sharing model.

4.3.4 | Discussion

As can be concluded from the two robustness tests, the only drone equipped with payload in each test set is a specific octocopter model. One could therefore wonder whether the classifiers really distinguish the existence of a payload mounted on a drone, or whether they learn that the specific octocopter model comes equipped with a heavy payload.

To prove the ability of the classifiers in this context, extra tests were carried out. For this, the imbalanced test set was modified by adding further measurements of an octocopter without payload. Two new test sets were defined: one set with added measurements of a different type of octocopter acquired during another measurement campaign and one set with added measurements of the same type of octocopter acquired during the same measurement campaign, as the measurements with the payload. Figure 11a presents the F_1 score performance of all proposed models on both test sets. As can be seen, all MTL models outperform their STL counterparts. This comparison based on the F_1 scores is fair but it does not provide any information on the actual percentage of the newly added measurements that were misclassified as drones carrying a payload. In the operational domain this is an important measure of



(a) F_1 score comparison between the MTL and STL models for the payload detection task.

(b) Percentage of correctly classified octocopter samples without payload.

FIGURE 11 | The payload detection performance of the STL and MTL models, when extra measurements of octocopters without a payload are added to the test set.

performance. Figure 11b demonstrates the potential disadvantages of MTL in this context. With the exception of the adjusted model, the MTL models performed poorly in differentiating between octocopters with and without payload. A possible explanation is the fact that, as the STL model is purposefully trained and optimised for this specific task, it might be able to learn more features associated with the RCS of the targets with payload. Nevertheless the STL model produced a high number of false alarms caused by samples of different types of drones.

4.4 | Rotor Rotation Rate Estimation and Payload Detection

In this section, the combination of the mean rotor rotation rate estimation and payload detection tasks is discussed. Again the payload detection task is considered to be the primary task. The simple and info-sharing models were trained using the Adam optimiser with a learning rate of $5 \cdot 10^{-5}$. The adjusted model was trained in the same way as the other two MTL approaches by using the SGD optimiser with $1 \cdot 10^{-3}$ as the learning rate. The two STL models were trained for each split both using the same hyperparameters used for training the simple model.

4.4.1 | Cross-Validation Approach

Regarding the mean rotor rotation rate estimation, all MTL models failed to improve upon the MAE performance of their STL counterpart, degrading the MAE performance by as much as 32.16% when using the info-sharing model. However, as can be seen in both the positive class F_1 score box plot in Figure 12 and in Table 5, major improvements were made in payload detection performance were made when using any of the three proposed MTL models. In this context, theinfo-sharing model achieved an F_1 score improvement of 14.23%. As was the case with the previous subset of tasks, it occurred to some extent that spectrograms belonging to the same drone types without payload were present in the test sets in each split, leading to a considerable amount of false alarms. Hence, as can be seen in both Figure 12 and Table 5,

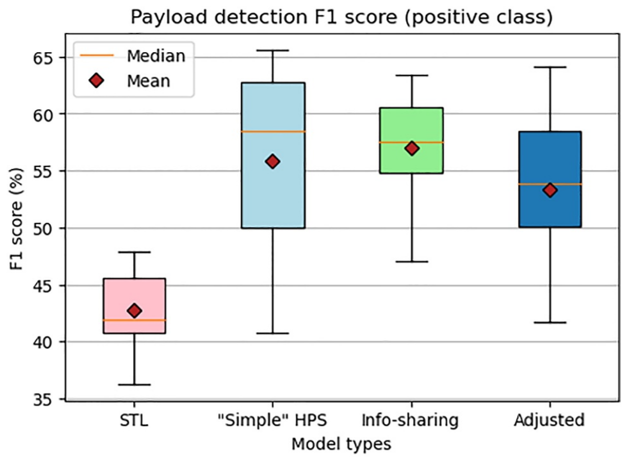


FIGURE 12 | Positive class F_1 score comparison between the MTL and STL models for the payload detection task.

both the STL and MTL models achieved generally only low F_1 scores yet, MTL models were found to be more successful in differentiating between payload and nonpayload carrying drones of the same type (i.e., octocopters).

4.4.2 | Robustness Test—Imbalanced Test Set

One of the key differences between the robustness tests involving the payload detection task of this section and the ones presented previously in Section 4.3.2, is the difference in number of samples in the formed training sets (for both robustness tests). Because of dataset limitations, not all available spectrograms produced cepstrograms enabling reliable rotor rotation rate extraction. This severely impacted the training set part of both robustness tests. The consequence was that the training set of the first robustness test scenario of Section 4.3.2 produced classifiers that could not predict the payload task sufficiently. It is possible that the missing training samples provided some sort of class separation that benefited the learning of the models. Hence, although both STL and MTL models performed quite

TABLE 5 | Mean rotor rate estimation and payload detection tasks F_1 score performance comparison.

Model	Performance	
	MAE	F_1 score (%)
STL	**36.59 ± 2.6**	42.74 ± 2.49
Simple model	42.52 ± 2.95	55.89 ± 6.03
Adjusted model	43.56 ± 3.53	53.35 ± 5.32
Info-sharing model (% Increase)	50.61 ± 3.33	**56.97 ± 3.61** (+**14.23**%)

Note: The best performance in each case and the corresponding increase are highlighted in bold.

well in terms of rotor rotation rate estimation, all struggled with the payload detection task.

On the contrary, classifiers trained on the corresponding set of the second robustness test proved able to distinguish among cases with and without payload. Only the second robustness test of Section 4.3.2 is utilised and presented in this subsection. It is important to mention that the test set was not affected in any way and remained intact, as all corresponding cepstrograms enabled rotation rate extraction. Details on the content of the test set for this robustness test can be found in Table A4. The benefits of exploiting MTL in this setting is made clear when observing the F_1 comparison plot of Figure 13. There, a significant increase of 22.95% can be found when utilising the info-sharing model, which achieved an F_1 score of 80.5% versus a much poorer 57.55% achieved by the STL counterparts. The relatively high F_1 score of the info-sharing model is almost solely attributed to the extremely high precision of the classifier, having no false alarms.

It would appear that in this particular train/test split, the mean rotor rate estimation capabilities of the adjusted and info-sharing models are roughly equivalent to their STL counterpart, whereas the simple model appears inferior as can be seen in the MSE and MAE plots of Figure 14. Even though the adjusted and info-sharing models provide a decrease in MSE of 3.59% and 7.15%, respectively, this improvement should be treated with some caution, especially when considering the sensitivity of the metric to outliers in the estimation. This is made more apparent when observing the plot of the mean rotation rates of all targets in Figure 15a, alongside with the absolute error for each instance in Figure 15b. There it can be seen that a relatively small number of large errors can dominate over the metric even though it is an average over all given samples.

The MAE comparison plot of Figure 14b proves this point, as it can be seen that the models are perform all similar in terms of the mean absolute error. The constant incorrect estimations seen in certain cases in Figure 15a can be attributed to the manner in which the representative rotor rotation rate for each case is calculated. The rotor rotation rate extraction used to obtain the ground truth was carried out by imposing specific restrictions that were not imposed to the regression models. For example, helicopters give rise to two distinct harmonics related to the main and tail rotor respectively yet only the harmonic related to the main rotor was selected to represent the ground truth. The regression model may converge to either one.

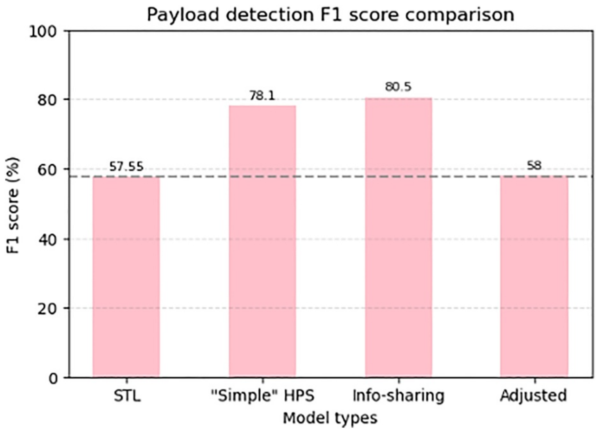
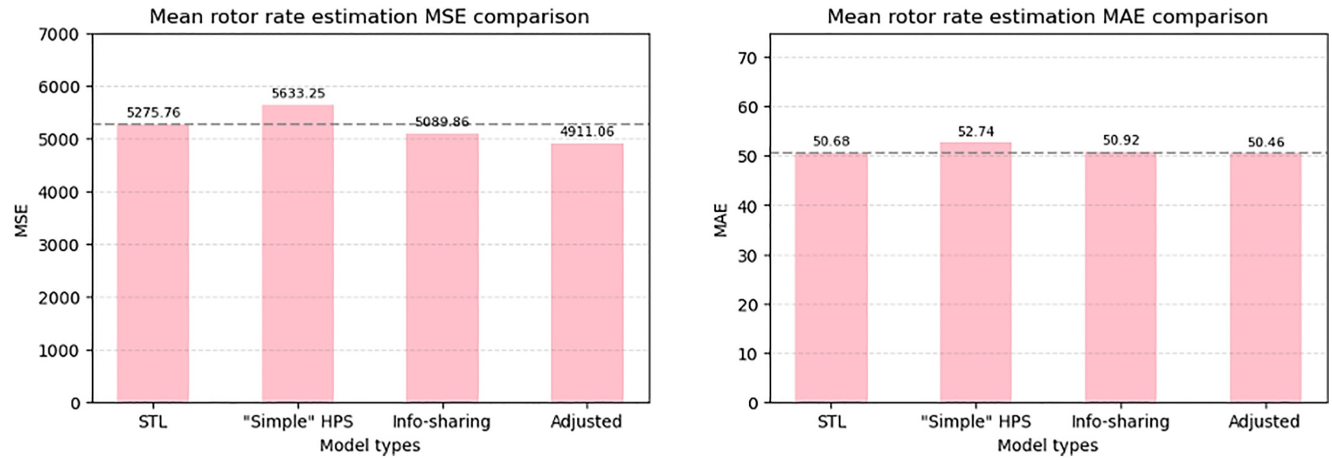


FIGURE 13 | Payload detection F_1 score comparison between the STL and MTL models using the test set described in Table A4.

Even though the conclusions of the cross-validation approach should not be directly compared to those of the robustness test as they depend on vastly different training procedures, it is true that in both cases the MTL models were more capable of detecting the presence of a payload than their STL counterparts. Another common conclusion is that the STL models were, in the best case scenarios, comparable to their STL counterparts in terms of mean rotor rotation rate regression capabilities. When compared to the corresponding robustness test of Section 4.3.2, the performance of the MTL models for the current two-task combination is worse. It appears that the mean rotor rotation rate estimation alongside the payload detection reduces the performance of the latter task.

4.4.3 | Discussion

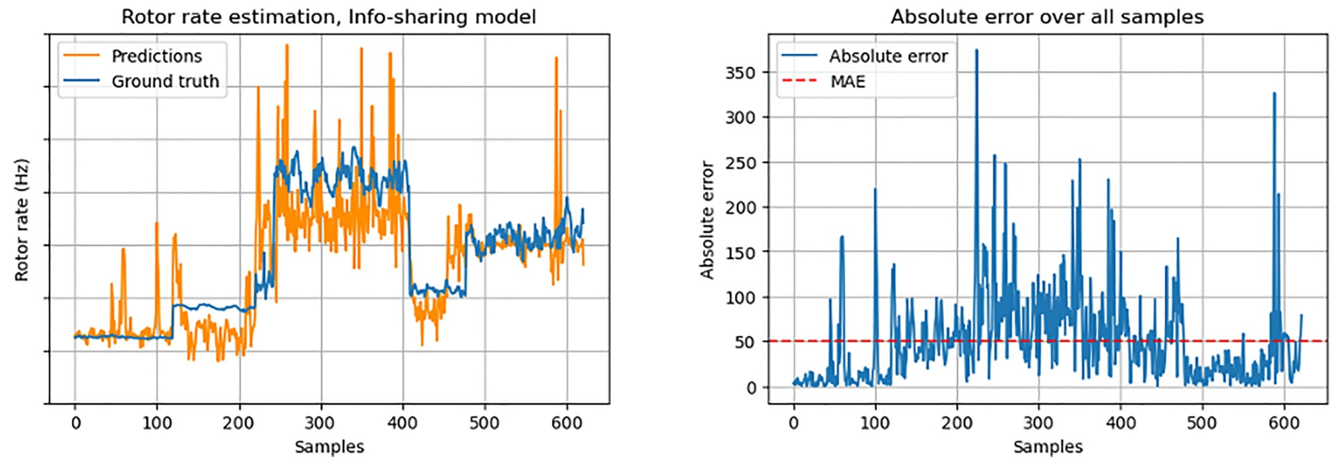
The same question asked in Section 4.3.2 can be asked here. It would also be of interest to compare what effect predicting a different task, both in type and nature, would have on the differentiation between two structurally equivalent, or even identical, drone types. Interesting conclusions may be drawn from the bar plots of Figure 16 on the differentiation capabilities of the proposed MTL models. Firstly, it is shown in Figure 16a that exploiting MTL provides better performances in F_1 score for either of the two types of octocopters introduced in the test set. More specifically, the simple model achieved an F_1 score improvement of 6.46% over the best score achieved by an STL counterpart. This improvement is attributed to the higher number of false alarms. The STL model in this case produced only 21 false alarms yet it missed 51.39% of the drones carrying a payload. A higher improvement in F_1 score is seen when a measurement of the same drone type is transferred from the training to the test set. As expected, the presence of this particular drone type in the test set induces a high number of false alarms for both the STL and MTL models, leading to a drop in performance for all models. Nevertheless, all proposed MTL models perform better than their STL counterparts. The highest improvement is observed when employing the simple model, achieving an F_1 score equal to 65.28%, an increase of 12.49% over the STL counterparts.



(a) Mean rotor rate MSE comparison between the MTL and STL models.

(b) Mean rotor rate MAE comparison between MTL and STL models.

FIGURE 14 | Rotor rate regression performance comparison between the MTL and STL models.



(a) Rotor rotation rate estimates (orange) and ground truth values (blue).

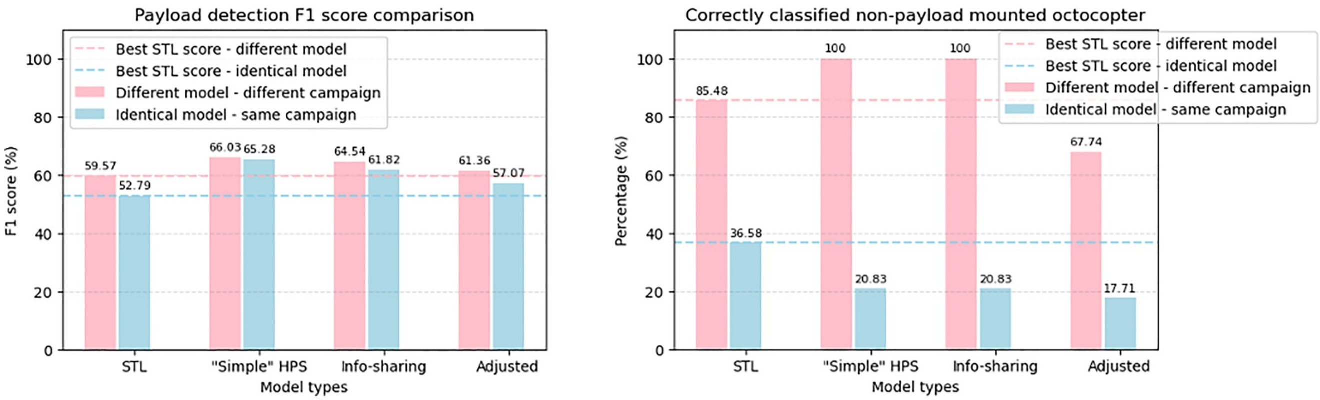
(b) Absolute error in the rotor rate estimates over all samples across all measurements present in the test set. The MAE is also plotted (dashed red line).

FIGURE 15 | The mean rotor rotation rate estimates and the absolute error in the estimates, as produced by the model performing best on the payload detection task: the info-sharing model. Note that for simplicity, all mean rotation rates are presented in these plots; each sample present in the plots represents one rotation rate calculated from the relevant cepstrograms.

Although in general all MTL models performed considerably better than the STL baseline, the question of how efficient they are in distinguishing among drones of the same type with or without payload still holds. To answer this, consider the comparison plot regarding the percentage of correctly classified octocopters without a payload of Figure 16b. There, it can be observed that both the simple and info-sharing models achieve 100% separation between octocopters with and without payload when the latter is of a different type. This comes as a 14.52% improvement over the one achieved by the STL counterpart. Hence, for these two MTL models, the false alarms are induced solely by misclassifying other drone types. On the contrary, all models achieve a very low separation when it comes to a specific octocopter type with payload versus the same type without a payload yet, the STL model was found to be considerably better when compared to the MTL models. Other thresholds than the traditional 0.5 one for class assignment were proven to be beneficial for the payload detection task, yet this benefit came at the expense of a major drop in the F_1 score. For instance, setting the threshold for the info-sharing model to 0.54 gives a percentage of correctly classified octocopters without payload of 40.63% (a 19.8% increase of the percentage compared to using the 0.5 threshold), at the same time causing the F_1 score to drop to 50%.

5 | Conclusions

In this extended version of our initial paper [5], the application of MTL was investigated to simultaneously address combinations of the following tasks: classification of a drone's wing type, classification of the number of rotors, payload detection and estimation of the mean rotor rotation rate. All the aforementioned classification and regression tasks, and their combinations, can be



(a) F_1 score comparison between MTL and STL models regarding the payload detection task.

(b) Percentage of correctly classified octocopters without payload.

FIGURE 16 | The payload detection performance of the STL and MTL models when extra samples of octocopters without payload are added to the test set.

considered as possible indicators for the intent of a drone. Specifically, significant improvements were introduced when applying MTL for both the wing type classification and number of rotors classification tasks where MTL models managed to outperform their STL counterparts on a consistent basis.

The introduction of payload detection as an additional task to be jointly predicted led to notable negative transfer to the other two classification tasks throughout all experiments. Although all MTL models assessed were found to provide better overall performance regarding the payload detection task, the existence of drone targets of the same type as the one carrying payload in the test set led to a serious increase in false alarms. What was far more concerning in this case was the fact that the MTL models would correctly classify a smaller percent of these drone without payload than their STL counterpart. This might be showing that the MTL models would associate the given drone type as always carrying a payload. When assessing the joint performance of the mean rotor rotation rate estimation and payload detection, negative transfer among the tasks was also noticeable, with MTL models failing to improve the performance of the rotation rate estimation. In payload detection, on the other hand, MTL proved highly advantageous, introducing an important increase in detection performance.

To sum up, MTL can certainly be applied to drone characterisation and offers overall substantial and in many cases robust improvement in performance across various tasks of different nature. However, a few considerations shall be made, a number of which are inherent from the general concept of deep learning. Throughout this study it was made clear that the availability, diversity and length of the dataset at hand played a crucial role in the results of all proposed tests. Additionally, a rather concerning issue of deep learning in general, amplified by the nature of MTL at least when implemented through the HPS approach, is the lack of explainability. By adding multiple outputs and defining more sophisticated loss functions in order to train the proposed models impose more complexity, making explainability of the models in hand even more challenging.

Another fact that one has to keep in mind when employing MTL in general is whether the goal is to improve all examined tasks, a subgroup or even only one of them. From an operational point of view, payload detection is assumed to be more important, as a potentially more direct indicator for the drone's intent. If some tasks are considered operationally more important than others, they may be prioritised within the proposed MTL framework (i.e., by attributing a higher weight to their performance). The most suitable strategy for this implementation remains an open research question for future work. Furthermore, the relations between tasks should be further investigated, especially in the context of shared-layer MTL models where training unrelated tasks might lead to negative transfer.

Naturally, as it is the case with most deep learning applications, MTL models could benefit by the introduction of more data acquired by additional measurement campaigns which would expand the available dataset. Although one of the premises of MTL is that it introduces some resilience to overfitting, increasing the network's complexity by adding more tasks could lead to inadequate results either due to small datasets such as the one used here, or negative transfer as was discussed earlier in this work. A larger and more task-wise diverse dataset would allow for further experimentation with networks examining all proposed tasks of this paper jointly as well as additional ones. We consider this step as further future work on MTL for drone characterisation.

Author Contributions

Apostolos Pappas: conceptualization, data curation, formal analysis, investigation, methodology, validation, visualization, writing. **Jacco J. M. de Wit:** conceptualization, methodology, supervision, writing. **Francesco Fioranelli:** methodology, supervision, writing. **Bas Jacobs:** conceptualization, methodology, supervision.

Acknowledgements

The radar data used for this study have been kindly made available by Thales Nederland B.V. within the framework of D-RACE: the Dutch Radar Centre of Expertise, a strategic alliance of Thales Nederland B.V. and TNO.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

Research data cannot be shared with anyone in general because of their commercial confidentiality; enquiries can be directed the authors in case.

References

1. "Drone Delivery Operations Underway in 27 Countries," (2019), www.unmannedairspace.info.

2. A. E. Kramer and D. Guttenfelder, "From the Workshop to the War: Creative Use of Drones Lifts Ukraine," *International New York Times* (2022): NA.

3. J. J. M. De Wit, D. Gusland, and R. P. Trommel, "Radar Measurements for the Assessment of Features for Drone Characterization," in *2020 17th European Radar Conference (EuRAD)* (IEEE, 2021), 38–41.

4. R. Caruana, "Multitask Learning," *Machine Learning* 28, no. 1 (1997): 41–75, https://doi.org/10.1023/a:1007379606734.

5. A. Pappas, J. J. M. De Wit, F. Fioranelli, and S. Jacobs, "Multitask Learning for Radar-Based Characterization of Drones," in *2023 IEEE International Radar Conference (RADAR)* (IEEE, 2023), 1–7.

6. A. Waibel, H. Sawai, and K. Shikano, "Modularity and Scaling in Large Phonemic Neural Networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37, no. 12 (1989): 1888–1898, https://doi.org/10.1109/29.45535.

7. K. H. Thung and C. Y. Wee, "A Brief Review on Multi-Task Learning," *Multimedia Tools and Applications* 77, no. 22 (2018): 29705–29725, https://doi.org/10.1007/s11042-018-6463-x.

8. Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial Landmark Detection by Deep Multi-Task Learning," in *Computer Vision – ECCV 2014* (Springer, 2014), 94–108.

9. M. Long, Z. Cao, J. Wang, and P. S. Yu, "Learning Multiple Tasks With Multilinear Relationship Networks," *Advances in Neural Information Processing Systems* 30 (2017).

10. S. Ruder, "An Overview of Multi-Task Learning in Deep Neural Networks," *arXiv Preprint*. arXiv:1706.05098 (2017).

11. I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-Stitch Networks for Multi-Task Learning," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 3994–4003.

12. Y. Fang, Z. Ma, Z. Zhang, X. Y. Zhang, and X. Bai, "Dynamic Multi-Task Learning With Convolutional Neural Network," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)* (2017), 1668–1674.

13. W. Du, F. Zhang, F. Ma, Q. Yin, and Y. Zhou, "Improving SAR Target Recognition With Multi-Task Learning," in *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium* (IEEE, 2020), 284–287.

14. Z. Chen, Y. Lin, L. Zhuang, and H. Yu, "Meta Multi-Task Learning for Small Sample SAR Target Recognition," (2022).

15. X. Li, Y. He, and X. Jing, "A Deep Multi-Task Network for Activity Classification and Person Identification With Micro-Doppler Signatures," in *2019 International Radar Conference (RADAR)* (IEEE, 2019), 1–5.

16. C. Wang, J. Tian, J. Cao, and X. Wang, "Deep Learning-Based UAV Detection in Pulse-Doppler Radar," *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021): 1–12, https://doi.org/10.1109/tgrs.2021.3104907.

17. R. Akter, V. S. Doan, A. Zainudin, and D. S. Kim, "An Explainable Multi-Task Learning Approach for RF-Based UAV Surveillance Systems," in *2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN)* (IEEE, 2022), 145–149.

18. N. Mohajerin, J. Histon, R. Dizaji, and S. L. Waslander, "Feature Extraction and Radar Track Classification for Detecting UAVs in Civillian Airspace," in *2014 IEEE Radar Conference* (IEEE, 2014), 0674–0679.

19. M. Messina and G. Pinelli, "Classification of Drones With a Surveillance Radar Signal," in *Computer Vision Systems (ICVS 2019)* (Springer, 2019), 723–733.

20. N. Regev, I. Yoffe, and D. Wulich, "Classification of Single and Multi Propelled Miniature Drones Using Multilayer Perceptron Artificial Neural Network," in *International Conference on Radar Systems (Radar 2017)* (IET, 2017), 1–5.

21. S. Liu, Y. Liang, and A. Gitter, "Loss-Balanced Task Weighting to Reduce Negative Transfer in Multi-Task Learning," *Proceedings of the AAAI Conference on Artificial Intelligence* 33, no. 01 (2019): 9977–9978, https://doi.org/10.1609/aaai.v33i01.33019977.

Appendix A

This appendix includes four tables with details of the different test data used for the different robustness tests performed in the paper.

TABLE A1 | Composition of the test set used for the robustness test discussed in Section 4.2.2 considering samples acquired in different measurement conditions.

Description of target	Wing type	Number of rotors	Number of samples
Single pusher propeller drone	Fixed wing	1	188
Helicopter drone	Helicopter	2	221
Quadcopter drone	Multicopter	4	69

TABLE A2 | Composition of the test set used for the robustness test discussed in Section 4.2.3 considering samples of irregular drone types only.

Description of target	Wing type	Number of rotors	Number of samples
Single pusher propeller drone	Fixed wing	1	323
Fixed wing drone with two rotors one behind the other	Fixed wing	2	49
Custom built VTOL flying wing drone	Fixed wing	4	121
Hexacopter with paired rotors	Multicopter	6	168
Octocopter with person walking underneath	Multicopter	8	558

Abbreviation: VTOL: vertical take-off and landing.

TABLE A3 | Composition of the test set for the robustness test discussed in Section 4.3.2 with a small but balanced test set, that is, balanced in terms of payload presence.

Description of target	Wing type	Number of rotors	Payload	Number of samples
Single pusher propeller drone	Fixed wing	1	False	91
Quadcopter drone	Multicopter	4	False	47
Octocopter drone	Multicopter	8	True	144

TABLE A4 | Composition of the test set for the robustness test discussed in Sections 4.3.3 and 4.4.2 with an imbalanced test set, that is, imbalanced in terms of drone types and payload presence.

Description of target	Wing type	Number of rotors	Payload	Number of samples
Single pusher propeller drone	Fixed wing	1	False	188
Helicopter drone	Helicopter	2	False	221
Quadcopter drone	Multicopter	4	False	69
Octocopter drone	Multicopter	8	True	144