Multitask Learning for Radar-Based Characterization of Drones

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

DOI
10.1109/RADAR54928.2023.10371104

Publication date
2023

Document Version
Final published version

Published in
2023 IEEE International Radar Conference, RADAR 2023

Multitask Learning for Radar-Based Characterization of Drones

Apostolos Pappas
Microwave Sensing, Signals & Systems
TU Delft
Delft, The Netherlands
a.pappas-1@student.tudelft.nl

Jacco J.M. de Wit
Department of Radar Technology
TNO
The Hague, The Netherlands
jacco.dewit@tno.nl

Francesco Fioranelli
Microwave Sensing, Signals & Systems
TU Delft
Delft, The Netherlands
f.fioranelli@tudelft.nl

Bas Jacobs
Department of Radar Technology
TNO
The Hague, The Netherlands
bas.jacobs@tno.nl

2023 IEEE International Radar Conference (RADAR) | 978-1-6654-8278-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/RADAR54928.2023.10371104

***Abstract*—For the effective deployment of countermeasures against drones, information on their intent is crucial. There are several indicators for a drone's intent, e.g., its size, payload, and behavior. Within the current study, the focus was on estimating subsets 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. Three *Multitask Learning* (MTL) approaches were analyzed for the simultaneous estimation of subsets 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 generalization capabilities as compared to separately trained single-task neural networks. The results of this initial study show that MTL provides overall better performance than the single-task learning approach, given the available data set of measured drone spectrograms.**

Keywords—multitask learning, radar micro-Doppler, drone classification

I. INTRODUCTION

Over the last few years, the use of drones has grown significantly and is expected to continue to grow in the coming years. Apart from recreational use, drones are now used for a wide variety of commercial applications as well, such as monitoring of critical infrastructure and inspection of building sites. In particular, the anticipated large-scale use of drones by commercial couriers for delivery of packages [1], will deeply embed drones in our daily life.

In the slipstream of the steady development of commercial applications, the unlawful usage of drones increases as well. One example of illegal use is drug trafficking across borders and, on a smaller scale, dropping contraband in prisons, such as drugs, cell phones or small firearms. Another example is the use of drones for politically-driven acts and disruption of public events. In the extreme case, drones could be used for serious acts of terrorism.

Also in the military domain, increasing numbers of drones appear on the battlefield [2], for a variety of applications. Small Unmanned Aerial Vehicles (UAVs), for example, are deployed as weapon carrier, reconnaissance platform or forward observer. Apart from military-grade UAVs, also commercially available drones are deployed on the battlefield as observer or to drop grenades or improvised explosives [3].

In summary; even small drones, i.e., Class I mini-UAVs [4], constitute a threat in the security and defense domains. How to counter them depends on the situation, the intent of the drone and its payload. For instance, when securing a public event, care should be taken that intercepting a drone does not lead to disproportionate collateral damage. Therefore, before intercepting a drone, its intention should be known, e.g., is it carrying a lethal payload? Consequently, as much information as possible should be gathered on the intent of a drone, to ensure effective, and safe, deployment of countermeasures. Considering radar sensors, the track characteristics and micro-Doppler signature of a drone provide relevant information on its number of rotors, wing type, payload, and behavior, which are all indicators for their intent [5]. Within this context, the current study focusses on the approach of *Multitask Learning* (MTL), to simultaneously extract a subset of the following four indicators, i.e., wing type, number of rotors, payload presence and mean rotor rotation rate from a drone's radar micro-Doppler signature. MTL refers to training a neural network simultaneously for different, but related tasks based on a shared data representation [6]. The underlying assumption is that if tasks share features between them, the overall MTL network is easier to train than separate networks optimized for the tasks individually.

The MTL concept is discussed in Section II. The specific multitask problem for radar-based drone characterization is explained in Section III. In this section also the radar data set and the implementation of the multitask networks are discussed. The results are presented in Section IV. Finally, Section V concludes the paper.

II. MULTITASK LEARNING

A. Defining Multitask Learning

As stated, multitask learning is a learning framework intended for training neural networks simultaneously for multiple related tasks, using a shared representation [6]. This comes 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 optimized for each of these tasks separately.

This single-task approach that combines the outputs of these individual models is referred to as Single-Task Learning (STL). STL has been proposed quite early, in works such as Waibel et al [7], to be counterproductive in some cases since it discards useful information that could be exploited by other tasks of the same general problem. In simpler words, feature representations that are produced and possibly discarded by one task might be useful in predicting another. This fact serves as the ground on which MTL was conceived: if tasks share knowledge among them, then the resulting model may learn more easily and generalize better compared to the separate STL counterparts.

From an analysis of the current literature, e.g. [8], 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 can be seen in Fig. 1 (left). 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 Fig. 1 (right).

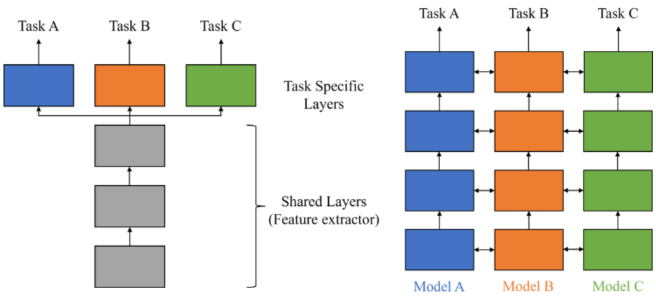

Fig. 1. Schematic representation of the HPS MTL approach: the shared layers serve as the feature extractor of the network while there are three (for this case) independent task specific layers (left). Schematic representation of the SPS MTL approach: in this approach instead of shared layers, N different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

B. *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. [9] proposed a tasks-constrained deep convolutional network to jointly optimize facial landmark detection with a set of related tasks. This approach retains, in its essence, the same configuration as seen in Fig. 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 their architectures can be arbitrarily chosen to fit the purpose of the corresponding task.

Expanding on the notion of HPS, Long et al. [10] 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, which allow the model to learn the relationship between the tasks. Yet, as mentioned in [11], this method is still reliant on the existence of a known sharing structure and might be inadequate for novel applications such as the application addressed in this study: radar-based drone characterization.

Misra et al. [12] developed the idea of cross-stitch networks, see Fig. 2, mainly as a way to counter the issue of deciding where to split in HPS architectures. Dictated by the paradigm of SPS, the authors propose the stitching of pre-defined models for each task using the so called cross-stitch units defined by:

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

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 parametrized by α, varying from 0 to 1, a parameter that is learned during training. Hence, in this case if the diagonal of α is set to 1, meaning that $\alpha_{AB} = \alpha_{BA} = 0$, then the two networks discard the information of one another at that particular layer, corresponding to no joint learning. It is shown that the cross-stitch networks perform and generalize better than both STL and MTL baseline methods.

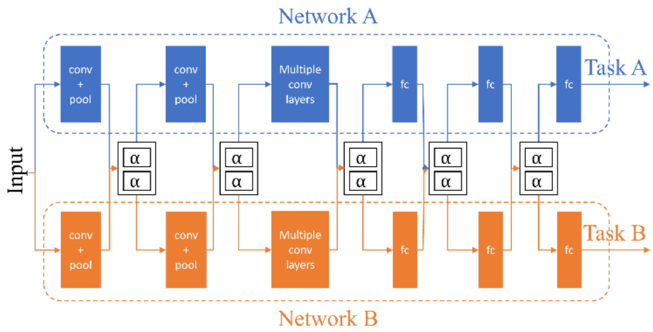

Fig. 2. Schematic representation of the cross-stitch network paradigm introduced by Misra et al. [12]. (Figure adapted from that same paper.)

Ruder et al. [11] built upon the notion cross-stitch networks by introducing a novel meta-architecture named sluice networks. This new 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 model's outer and final layers.

Yet, SPS is not restricted only in sharing activation or feature maps among the individually defined networks. The authors in [13] tackled the question of task relatedness by proposing a dynamic multitask convolutional neural network. This MTL model, comprising multiple per-task convolutional neural networks stitched with task transfer connections, leads to the tasks forming weak groups spontaneously through training progress. In the same manner as in the cross-stitch networks, when communicated information among the networks is ignored, each subnet works independently as an STL model.

C. *MTL in Radar Applications*

The multitask learning paradigm has been previously applied in radar scenarios mainly in target recognition, e.g., using synthetic aperture radar, and human activity recognition. Even though MTL has been used towards drone detection and passive RF characterization in the recent past [14], [15], it was to the best of the authors' knowledge not used towards radar-based drone characterization.

In [16] the authors propose the use of multitask learning towards improving target recognition in targets acquired using synthetic aperture radar. To that end, three distinct tasks are defined, namely: separation of the target from the shadow, estimation of the target's aspect angle, and target recognition. Following an HPS mechanism for multitask learning, the authors consider the target recognition as the main task and the rest as auxiliary tasks, meant to improve the performance of the main task. The authors prove the benefits of multitask learning by performing comparisons on STL versus MTL in varying number of samples. In all cases, training both auxiliary tasks alongside the main task resulted in considerable improvement in target recognition accuracy.

In the domain of human activity recognition, Li et al. [17] proposed a deep multitask network capable of simultaneously predicting human activity and performing person identification in an HPS fashion. A custom convolutional multiscale residual attention network was introduced for this purpose. The feature extractor utilized two different kinds of kernels within the same blocks, one for fine scale-learning and the other for coarse-scale learning. Experimentation showed that the MTL approach provided an improved accuracy score for both tasks, more prominently in the one of person identification where it surpassed STL by 2.54%.

III. MTL for Radar-Based Drone Characterization

A. Radar-Based Drone Characterization

For the effective and safe deployment of proportionate countermeasures against drones, as much information as possible on the intent of a drone should be collected. The main radar observables of drones are its Radar Cross Section (RCS), track behavior and micro-Doppler characteristics. A drone's RCS depends on its size, shape and payload and on the material properties. In general, there is a weak relationship between a drone's RCS and its intent [5]. Furthermore, the observed RCS depends on the aspect angle and typically exhibits large fluctuations as a function of aspect angle. In literature, therefore, the emphasis is on exploiting track behavior and micro-Doppler features for radar-based characterization of drones.

By combining track and micro-Doppler features, it can be determined if a drone is of the fixed-wing or rotary-wing type [18], [19]. The type of drone provides an indicator for its role. For example, rotary-wing drones have some characteristics that makes them particularly suitable for the observer role. They can hover in a fixed position and they can climb very quickly and vertically, suddenly popping up from behind buildings or vegetation. Fixed-wing drones on the other hand can typically carry larger payloads for a longer period of time. Therefore, fixed-wing drones are more suitable for long-term surveillance of large areas. Furthermore, on the basis of micro-Doppler features, the number of rotors and the rotor diameter of a drone can be estimated [20]. These parameters are an indicator for the size of the drone and the maximum weight it is able to carry. Another possible indicator for the existence of payload mounted on a drone may be the rotor rotation rate, as drones carrying some sort of heavy payload would be expected to demonstrate higher rotation rates than non-payload carrying ones.

In view of the above, the focus of this study is applying MTL for simultaneous execution of a subset of the following tasks: wing type classification, number of rotors classification, payload detection, and mean rotor rotation rate estimation. These four tasks provide relevant information on the possible role and intent of a drone. Regarding the wing type, three drone classes with different flight characteristics, are defined: fixed-wing drones, helicopters, and multicopters. The number of rotors is either one, two, four, six, or eight. The payload detection task is treated as a binary classification problem and the mean rotor rotation rate estimation as a single continuous value-regression problem.

B. Dataset

The data considered for this work are produced by real life measurements using an X-band experimental continuous wave 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 durations are available. A spectrogram is generated from each measurement, an example spectrogram is shown in Fig. 3. Since the measurements are of varying durations, 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. Labeling 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, e.g., if the signal-to-noise ratio was too low. This reduced the size of the dataset suitable for the rotation rate estimation problem.

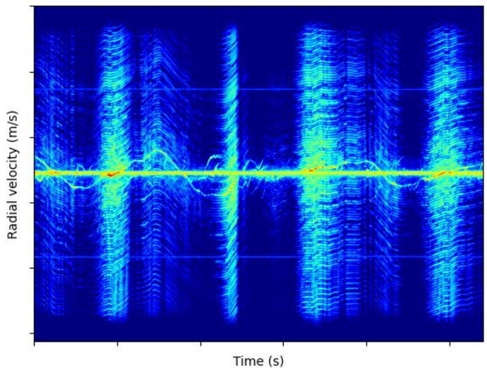

Fig. 3. Example of a measured spectrogram (before the split into 0.5 s segments) of a radio-controlled helicopter.

Considerable class imbalance is present throughout the experiments. This is notable when considering the number of helicopters in the dataset for the wing-type classification task and the number of hexacopters in the dataset for the number of rotors classification task. This imbalance is illustrated with the histograms presented in Fig. 4. 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 fifteen samples of a drone without payload.

C. Multitask-Network Implementation

Three MTL models based on the HPS paradigm (Fig. 1) were defined, all having the ResNet-18 network as backbone. The first two, named Simple HPS and Adjusted HPS models respectively, strictly follow the HPS approach offering different degrees of autonomy for the task streams. The split into two streams takes place just before the fully-connected layer for the Simple HPS model and before the fourth layer for the Adjusted HPS model.

The third MTL model assessed in this work, is the so-called Info-Sharing HPS model, a novel HPS model architecture developed within the framework of this study. The Info-Sharing HPS architecture is shown in Fig. 5. It combines the output of each task stream's fourth layer by concatenating and filtering the resulting feature map with a 1×1 convolutional kernel. Pytorch was used for a deterministic implementation of ResNet-18, ensuring full reproducibility of the results.

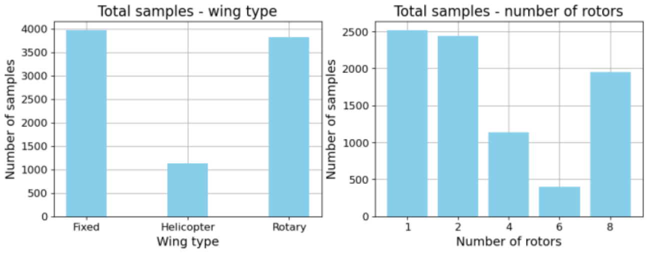

Fig. 4. Class imbalance in the wing type and number of rotors features throughout the available dataset.

Three out of the four tasks assessed in this work can be seen as purely classification problems. Hence, for the wing type, number of rotors classification and payload detection tasks the cross entropy loss alongside with its binary form 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 the STL and MTL models. A very important element in MTL is the combination of all tasks' calculated losses in a single loss. Two approaches were examined in this work: the naïve summation of the two losses and the Loss-Balanced Task Weighting (LBTW) approach developed to reduce the effect of negative transfer among tasks (i.e., cases where multitask models perform worse than their STL counterparts).

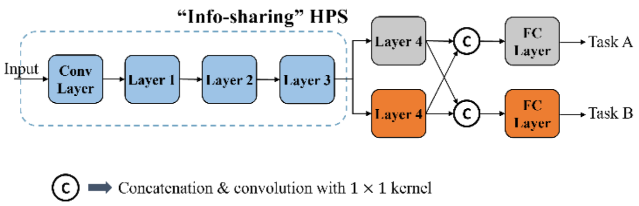

Fig. 5. Schematic representation of the Info-Sharing HPS MTL model. What differentiates this model from the Simple HPS and Adjusted HPS models is the communication of information among the task streams.

The LBTW is defined as the weighted sum of the two separate task losses [21]. 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 original, 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.

In this stage it is also necessary to introduce the ways in which the MTL models may be compared to their STL counterparts. When possible, the comparison of the models is done in two ways: task-wise and jointly. When comparing the corresponding models of the two paradigms task-wisely the F_1 scores (macro-averaged for the multiclass classification problems and positive class F_1 score for the binary classification problems) and MAEs are reported. For the subsets of tasks comprising only classification problems, a joint accuracy is also reported, where for each test sample all task outputs must be correct in order for it to be considered as correctly classified.

A key point in this work is the comparison of the defined models. To make comparisons feasible and to limit the number of free variables regarding the selection of appropriate networks and overall architectures, it was decided that comparisons would be considered between MTL and STL models of the same architecture per task. Furthermore, to provide a proper and fair comparison, all models were trained and optimized independently; thus different hyperparameters were allowed for training the MTL and STL networks.

IV. Results

To examine the possible benefits of MTL in the drone characterization scenario, ten different but nonoverlapping training and test splits were created in a ten-fold cross-validation fashion to test the resilience of the methods to varying distributions of data and targets. These ten train and test set splits were identical for the first two subsets of tasks discussed in Sections IV.A and IV.B. For the third subset of tasks, including the mean rotor rate estimation, the train and test sets are different, since not all samples could be reliably labeled, as mentioned in Section III.B.

For explorative experimentation purposes, all STL and MTL models used the pretrained version of the ResNet-18 neural network as a backbone, trained for 100 epochs with a weight decay of $1 \cdot 10^{-3}$, yet using different optimizers and learning rates. For the MTL models especially, the LBTW loss weighting scheme was employed with $\alpha = 0.5$ as this value was found to suppress the effects of negative transfer most effectively. To reduce overfitting and to artificially increase the amount of different samples given as input to the networks, random horizontal and vertical flips of the spectrograms were performed, i.e., transformations that do not affect the general physical properties of the corresponding spectrograms. The input spectrograms were all in dB scale.

A. Wing Type and Number of Rotors Classification

In this section, the results of the subset of tasks consisting of the wing-type classification problem and the number of rotors classification problem, are discussed.

In all MTL models the Stochastic Gradient Descent (SGD) optimizer 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 optimizer with a learning rate value fixed at $5 \cdot 10^{-5}$. Training the corresponding STL model for the number of rotors classification task was carried out using the same hyperparameters as for the MTL models.

Table I and Fig. 6 show the performance of both the STL and MTL models for the two classification tasks. All MTL models managed to achieve higher performance scores in both of the tasks when compared to the individually optimized single task models. The most significant improvement in F_1 score performance for both tasks is achieved by the Info-sharing HPS model, showing improvements of 1.2% and 2.98% respectively. This is also apparent in the joint accuracy comparison boxplot of Fig. 6, where an increase of 4.53% can be obtained by using the Info-Sharing HPS MTL model.

B. Wing Type and Number of Rotors Classification and Payload Detection

In this section, the results of the subset of tasks consisting of the wing-type classification problem, the number of rotors classification problem and the payload detection, are discussed.

The two STL models for predicting the wing type and number of rotors are reused as defined in the previous subsection. Once again, the MTL models were trained using the SGD optimizer with a learning rate of $1 \cdot 10^{-3}$ and a momentum of 0.9. For the payload detection, a ResNet-18 was trained using the SGD optimizer using a learning rate value fixed at $1 \cdot 10^{-3}$.

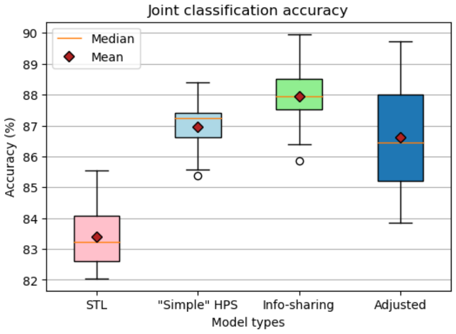

Fig. 6. Joint classification accuracy regarding the wing type and number of rotors classification tasks.

TABLE I. WING TYPE AND NUMBER OF ROTORS CLASSIFICATION F_1 PERFORMANCE COMPARISON

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

It is evident from Table II that compared to the previous subset of tasks, a major drop in performance can be seen in the wing type and number of rotors classification problems. Even though on average the Info-Sharing HPS model manages to provide 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 due to the lack of a strong relation between the two classification problems and the additional payload detection. This is more evident when using the Simple HPS and Adjusted HPS models where for some cases the performance of these models was inferior to the baseline set by the STL models.

On the contrary, the payload detection highly benefitted from the joint training with the other two tasks of this subset. As Table II suggests, an increase in positive (true) class F_1 score of 12.82% was achieved by the Simple HPS model, followed closely in performance by the Info-Sharing HPS model. The generally low F_1 scores can be attributed to the fact that spectrograms of the same drone types as the ones carrying a payload, were present in the test set leading to oftentimes large amounts of false alarms. The improvements in payload detection performance were translated into improvements of joint classification performance as seen in Fig. 7. There, a joint classification accuracy increase of 7.3% with the Info-Sharing HPS model can be observed.

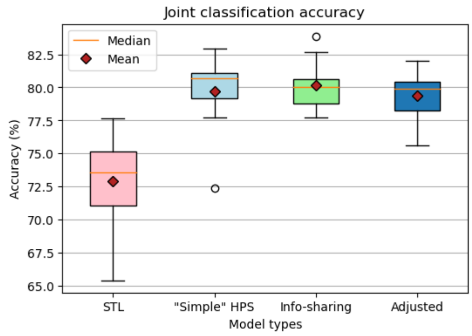

Fig. 7. Joint classification accuracy regarding the wing type and number of rotors classification tasks and the payload detection task.

TABLE II. WING TYPE AND NUMBER OF ROTORS CLASSIFICATION AND PAYLOAD DETECTION F_1 SCORE PERFORMANCE COMPARISON

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

C. Rotor Rotation Rate Estimation and Payload Detection

In this section, the results of the subset of tasks consisting of the mean rotor rotation rate estimation and the payload detection are discussed.

For this last subset, the three aforementioned MTL models are once again utilized, simultaneously predicting the mean rotor rotation rate and the existence of payload mounted on drones. The Simple HPS and Info-Sharing HPS models were trained using the Adam optimizer with a learning rate of $5 \cdot 10^{-5}$. The Adjusted HPS model was trained in the same way as the other two MTL approaches, yet by using the SGD optimizer 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 HPS network.

Regarding the mean rotor rotation rate prediction, all MTL approaches failed to improve upon the MAE performance of their STL counterpart, degrading the MAE performance (i.e., increasing it) by as much as 32.16% when using the Info-Sharing HPS model. Yet, as can be seen in both the positive class F_1 score boxplot in Fig. 8 and in Table III, major improvements were made in payload detection performance when utilizing any of the three proposed MTL models. In this context, the Info-Sharing HPS model provided with 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 seen in both Fig 8. and Table III, both STL and MTL approaches suffered from generally low F_1 score performances. Yet, MTL models were found to be more successful in differentiating between payload and non-payload carrying drones of the same type (i.e., octocopters).

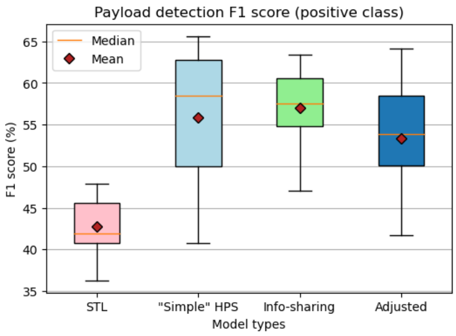

Fig. 8. Positive class F_1 score comparison for the payload detection task between the MTL models and their STL counterpart.

TABLE III. MEAN ROTOR ROTATION RATE ESTIMATION AND PAYLOAD DETECTION PERFORMANCE COMPARISON

Model	Tasks comparison	
	Mean rotor rate (MAE)	Payload detection (F_1)
STL	**36.59 ± 2.6**	42.74 ± 2.49
Simple HPS	42.52 ± 2.95	55.89 ± 6.03
Info-Sharing HPS (% increase)	50.61 ± 3.33	**56.97 ± 3.61 (+ 14.23%)**
Adjusted HPS	43.56 ± 3.53	53.35 ± 5.32

V. Conclusion

In this study, the application of *multitask learning* was investigated for the simultaneous prediction of subsets of the following tasks: a drone's wing type and number of rotors classification, payload detection and mean rotor rotation rate estimation. These features together can provide relevant information on the role of a drone and its possible intent.

The results of this study show that MTL can provide improved performance when compared to separately trained STL models. Significant improvements were introduced when applying MTL for both the wing type and number of rotors classification tasks. Adding the payload detection to this set of tasks led to notable negative transfer to the two classification tasks, where MTL models lost their advantages over their STL counterparts. On the contrary, the payload detection performance was improved by as much as 12.82% when employing the Info-Sharing HPS MTL model. When assessing the joint performance of the mean rotor rotation rate estimation and payload detection, MTL models failed to improve the performance of the rate estimation, yet introduced an improvement of 14.23% in the payload detection. From an operational point of view, payload detection is considered more important, since it is a more direct indicator for the drone's intent.

If some tasks are considered more important than others, given the final objective, tasks may be prioritized within the MTL framework (i.e., attributing a higher weight to their performance). The prioritization of the tasks both within the network's premises and on an operational level still remains an open problem. Furthermore, the relations between tasks should be further investigated, especially in the context of shared layer MTL networks where training unrelated tasks might lead to sever negative transfer.

Acknowledgment

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.

References

[1] "Drone delivery operations underway in 27 countries," Unmanned Airspace, April 2019. Available online: www.unmannedairspace.info (accessed April 18, 2023).

[2] J. Postma, "drones over Nagorno-Karabakh: a glimpse at the future of war?," Atlantisch Perspectief, vol. 45, no. 2, 2021. Available: https://www.jstor.org/stable/48638213?seq=1 (accessed April 20, 2023).

[3] A.E. Kramer, "From the workshop to the war: creative use of drones lifts Ukraine," The New York Times, August 2022. Available: https://www.nytimes.com/2022/08/10/world/europe/ukraine-drones.html (accessed April 20, 2023).

[4] "Strategic concept of employment for unmanned aircraft systems in NATO," Joint Air Power Competence Centre, 2010. Available: https://www.japcc.org/wp-content/uploads/UAS_CONEMP.pdf (accessed April 19, 2023).

[5] J.J.M. de Wit, D. Gusland, and R.P. Trommel, "Radar measurements for the assessment of features for drone characterisation," in Proc. EuRAD, Utrecht, The Netherlands, 2021.

[6] Caruana, R. Multitask Learning. Machine Learning 28, 41–75 (1997).

[7] A. Waibel, H. Sawai and K. Shikano, "Modularity and scaling in large phonemic neural networks," IEEE TASSP, vol. 37, no. 12, pp. 1888-1898, Dec. 1989, doi: 10.1109/29.45535.

[8] Thung, KH., Wee, CY. A brief review on multi-task learning. Multimed Tools Appl 77, 29705–29725 (2018). https://doi.org/10.1007/s11042-018-6463-x

[9] Z. Zhang, P. Luo, C.C. Loy and X. Tang. Facial Landmark Detection by Deep Multi-task Learning. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8694. Springer, Cham. https://doi.org/10.1007/978-3-319-10599-4_7

[10] M. Long, Z. Cao, J. Wang and P. S. Yu, "Learning multiple tasks with multilinear relationship networks," Advances in Neural Information Processing Systems, vol. 30, 2017.

[11] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098,* 2017.

[12] I. Misra, A. Shrivastava, A. Gupta and M. Hebert, "Cross-Stitch Networks for Multi-task Learning," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 3994-4003, doi: 10.1109/CVPR.2016.433.

[13] Y. Fang, Z. Ma, Z. Zhang, X.-Y. Zhang and X. Bai, "Dynamic Multi-Task Learning with Convolutional Neural Network," in IJCAI, 2017, pp. 1668-1674.

[14] R. Akter, V. -S. Doan, A. Zainudin and D. -S. Kim, "An Explainable Multi-Task Learning Approach for RF-based UAV Surveillance Systems," in Proc. ICUFN, Barcelona, Spain, 2022, pp. 145-149, doi: 10.1109/ICUFN55119.2022.9829629.

[15] C. Wang, J. Tian, J. Cao and X. Wang, "Deep Learning-Based UAV Detection in Pulse-Doppler Radar," IEEE TGRS, vol. 60, pp. 1-12, 2022, doi: 10.1109/TGRS.2021.3104907.

[16] W. Du, F. Zhang, F. Ma, Q. Yin and Y. Zhou, "Improving SAR Target Recognition with Multi-Task Learning," in Proc. IGARSS, Waikoloa, U.S.A., 2020, doi: 10.1109/IGARSS39084.2020. 9324210.

[17] X. Li, Y. He and X. Jing, "A deep multi-task network for activity classification and person identification with micro-Doppler signatures," in Proc. Int. Radar Conference, Toulon, France, 2019, doi: 10.1109/RADAR41533.2019.171263.

[18] N. Mohajerin, J. Histon, R. Dizaji, and S. L. Waslander, "Feature extraction and radar track classification for detecting UAVs in civilian airspace," in Proc. IEEE Radar Conf., 2014.

[19] M. Messina and G. Pinelli, "Classification of drones with a surveillance radar signal," in Proc. ICVS, 2019.

[20] N. Regev, I. Yoffe, and D. Wulich, "Classification of single and multi propelled miniature drones using multilayer perceptron artificial neural network," in Proc. Int. Radar Conf., Belfast, U.K., 2017.

[21] S. Liu, Y. Liang, and A. Gitter, "Loss-Balanced Task Weighting to Reduce Negative Transfer in Multi-Task Learning", AAAI, vol. 33, no. 1, pp. 9977-9978, Jul. 2019