Pattern Recognition Letters 176 (2023) 62–68

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A multi-task network for speaker and command recognition in industrial environments

Stefano Bini [a], Gennaro Percannella [a], Alessia Saggese [a,*], Mario Vento [a,b]

[a] Department of Information and Electrical Engineering and Applied Mathematics, University of Salerno, Italy
[b] National Research Council of Italy - Institute for High Performance Computing and Networking (ICAR), Italy

ARTICLE INFO

Editor: Maria De Marsico

Keywords:
Speaker recognition
Speech-Command Recognition
Multi-task network

ABSTRACT

In industrial environments, it is crucial to establish a strong collaboration between humans and robots to enhance productivity. However, the nature of the work demands that workers have the authority to provide specific instructions to the robots. The scientific community has extensively investigated these dual requirements, aiming to develop advanced systems capable of recognizing voice commands and implementing speaker authentication. Nevertheless, in the industrial context, these tasks should be executed simultaneously on low-cost and low-power embedded devices that can be mounted on board the robotic platform. To overcome this challenge, we propose a multi-task network for Speech-Command Recognition and Speaker Identification. Additionally, we employ the *GradNorm* adaptive algorithm to address the issue of task imbalance. To evaluate the proposed system, we introduce a new dataset, MIVIA-ISC, consisting of 20,857 samples uttered by 562 speakers for 31 distinct commands. Our approach significantly reduces the network size by 47% and its execution time by 48% compared to the commonly used methodology, which employs one network for each task. Furthermore, our approach demonstrates a significant improvement in the accuracy of the Speaker Identification task, achieving an 11% increase compared to the corresponding single-task network. Importantly, this enhancement is achieved without compromising the accuracy of the Speech-Command Recognition task, which experiences only a minimal 3% decrease in performance.

1. Introduction

In the last years, the use of robots in industrial environments has increased significantly and the interaction between humans and robots has become a crucial aspect of the manufacturing process. To achieve efficient and safe collaboration, the human operator needs to be able to provide commands to the robot, and the robot needs to be able to recognize these commands accurately and in a short time, to act accordingly. But at the same time, it is essential to ensure that only authorized personnel can issue commands to the robot. This requirement calls for a speaker recognition system to authenticate the identity of the human operator ([1]), but also to allow the system to adapt to specific operators and to profile them [2].

In the literature, the speech recognition problem has been addressed through the following architecture: (i) an Automatic Speech Recognition (ASR) module analyzes an audio sample and generates the corresponding textual transcript; (ii) a Natural Language Understanding (NLU) module starts from the transcript and identifies the underlying intent of the interlocutor. This approach is used, just as an example, by well-known systems such as Siri, Cortana, and Alexa. However, as discussed in [3], to guarantee a high level of generality, the combination of ASR and NLU requires an extensive dataset and a substantial amount of training time. Additionally, it imposes considerable demands on hardware resources. This is of course not feasible in industrial environments, where the elaboration needs to be performed in real-time and over low-size and low-power embedded devices mounted on board of the robot. Indeed, computation on remote servers would not be possible in case of an unstable connection in the working environment, due for instance to poor signal coverage. Furthermore, another important consideration is that the command-based interaction between the human and the robot in industrial environments is typically based on very short and concise commands (e.g. *Start*, *Stop*, *Take the screwdriver*), instead of complex and long sentences with many entities (like it happens for instance in social robotics applications- [4]). Moreover, to interact with the robot, the human is only expected to handle a relatively small set of commands associated with production line operations.

Starting from the above considerations, a quite common approach, as shown in [5], is to formulate the problem in terms of Speech-Command Recognition (SCR). The aim is to directly classify an audio

* Corresponding author.
E-mail addresses: sbini@unisa.it (S. Bini), pergen@unisa.it (G. Percannella), asaggese@unisa.it (A. Saggese), mvento@unisa.it (M. Vento).

https://doi.org/10.1016/j.patrec.2023.10.022
Received 28 April 2023; Received in revised form 13 October 2023; Accepted 21 October 2023
Available online 28 October 2023

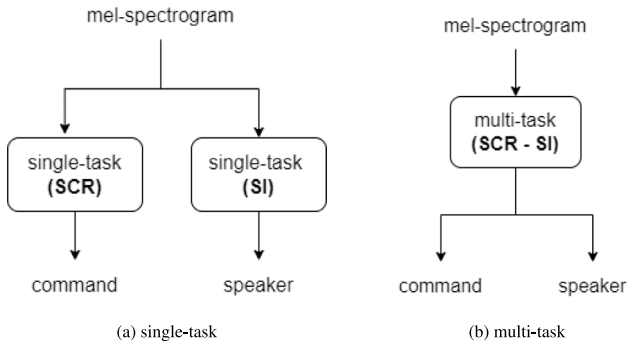

Fig. 1. Difference between (a) two single-task networks (namely one for SCR and one for SI) and (b) a single multi-task network able to deal simultaneously both the two tasks SCR and SI.

chunk as belonging to a specific command with an end-to-end approach. In [6] the authors represent the audio stream containing the commands as an image representing the Mel-scale spectrogram; then, they apply a CNN for image classification to identify the specific command of interest. The commands are single worlds, such as *Yes*, *No*, *On*, having all of them the same duration. Furthermore, standard data augmentation techniques applied on the image (instead of on the audio samples) are employed. In [7] the same representation is used, but the temporal information is explicitly exploited. Indeed, a set of convolutions is also applied only in the time dimension, to extract local relations in the audio stream; then, an LSTM is used for capturing long-term dependencies. The local relationships and the long-term dependencies are used to feed an attention layer, focusing only on the appropriate regions of interest for that audio sample. Even if very promising, the above-mentioned example of SCR is quite expensive from a computational point of view, it is not portable over an embedded platform and can deal with a single task.

As for speaker recognition, the problem can be formulated in terms of verification, identification, re-identification, and diarization, as discussed in [8]. In our context, as suggested in [9], the problem can be formulated in terms of speaker identification, since we are interested in understanding if a specific worker of the company is authorized to carry out a specific command. This is the reason why, in this paper, we focus on Speaker Identification (SI). The common approach for SI, as for SCR, is still the use of Mel-spectrogram in combination with Convolutional Neural Networks (CNNs), as proposed in [10]. The use of the Mel-spectrogram is rationalized by its alignment with the human auditory system, where it arranges pitches in a manner that closely mimics human perception. As for CNNs, a common choice is to employ shallow and small CNNs [11], such as the one by Shahin et al. [12], with less than 1000 parameters, or SpeakerNet proposed in [9], with just a three-layer architecture. Anyway, the number of considered speakers has been only 50 and 6, respectively. When the number of speakers increases, bigger architecture is required. This is why, more recently, ResNet-based architectures have increased their popularity; in [13] the authors employ a ResNet-34, and they evaluate a drop in performance when ranging from 30 to 150 speakers (which is the maximum number of considered speakers) of about 9%, with an accuracy of about 74% with short sentences and 150 speakers.

In our specific context, the challenge is the simultaneous recognition of both commands and speakers; this task is made more complex by the substantial number of speakers, around five hundred. A typical approach would require employing two separate neural networks (for speaker identification and command recognition, respectively), with the effect of doubling the computational load (See Fig. 1(a)). However, we can effectively address this challenge by adopting a multi-task learning approach (See Fig. 1(b)), which has the undeniable advantage of reducing the computational burden. This choice also brings an additional and not negligible property; indeed, it has been theoretically proved in [14] that the multi-task learning paradigm guarantees a more generalized representation, thanks to the regularization action that additional tasks perform on the main one.

This is why in the last years this paradigm has been widely adopted to increase the performance of the main task (e.g. ASR) by exploiting additional related problems (e.g. gender recognition), as proposed in [15–17]. In [18] the authors employed a multi-task network for speech recognition (ASR) and speaker recognition. They used the output of one task (at the previous timestamp) as part of the input of the other (at the current timestamp). This leads to an inter-task recurrent structure that is similar to conventional RNNs, though the recurrent connections link different tasks. In [19], the keyword spotting and speaker verification tasks utilize a common enhancement network to remove noise. Although the two tasks have their feature extractors, the acoustic feature extractor provides a phonetic conditional vector to enhance the speaker feature extractor's capabilities. A pooling network is also employed to combine the outputs from both feature extractors, which generate the keyword and speaker embeddings. The joint classification of spoken keywords and the determination of the speaker has been also proposed in [20], in a multi-tasking approach with soft-sharing parameters: a Mel scale spectrogram employed, and a representation optimized to work with both tasks is computed using a ResNet. Then, two separate branches, one per task, are used. Even if very promising, the system is thought to still manage only a few speakers (the maximum number of considered speakers in their experimentation is eight, with ten keywords), which is not enough to confirm the generalization capability of the network when dealing with hundreds of speakers (as it happens in industrial environments). Furthermore, they are only able to recognize words and not more complex commands, which is instead our aim.

To the best of our knowledge, there are no proposed methods for solving simultaneously the problems of SCR and SI, which is indeed a fundamental problem in industrial environments. One of the main challenges pertains to the availability of data. Furthermore, as discussed in [21], another significant difficulty is the imbalance of losses, as different tasks require the minimization of different risk functions. This problem has been traditionally addressed by weighting the loss contributions (see [22,23]), which can be a time-consuming process as it requires optimizing additional hyperparameters equal to the number of tasks.

Contribution. Starting from the above considerations, we propose a novel multi-task network for end-to-end command-based recognition and speaker identification.

The main benefit deriving from this choice is that a multi-tasking network can reduce the computational complexity and the memory footprint requirements of the overall system. This reduction in computational complexity can lead to faster processing times (about 2x), which is crucial in industrial environments, where real-time decision-making is required. Secondly, these advantages are not paid in terms of accuracy, since a shared representation helps to improve the accuracy of both tasks. Indeed, it captures relevant information from the input data, which might not be possible to achieve using separate networks. This is beneficial, especially for the SI task, for which we need to achieve generalization capabilities even in the presence of a high number of classes and a small size of the network. Furthermore, to deal with the issue of loss balancing, we propose to use a *GradNorm* balancing strategy, enabling the automatic and dynamic adjustment of the loss function weights during training.

Finally, in order to deal with the lack of datasets and the low number of speakers in the dataset, we acquired a new dataset composed of 20,857 audio samples, with 562 speakers and 31 commands. To summarize, the main contributions of this paper are the following:

- We propose multi-task networks for SCR and SI, resulting in comparable or even improved accuracy and reduced computational demands for both tasks compared to their single-task counterparts;
- We deal with the loss balancing issue by introducing a GradNorm strategy;
- We provide a novel dataset for SCR and SI, characterized by a wide range of speakers and acquisition devices.

The performance we achieve confirms that the proposed multi-task network based on GradNorm is able to achieve comparable performance with respect to the single-task network over the SCR task, gaining about 11 percentage points in terms of accuracy on the SI task and reducing the size of the network of almost one half, halving consequently the inference time.

The paper is organized as follows. Section 2 details the proposed method; Section 3 introduces the proposed dataset and the experimentation framework, discussing the obtained results. Finally, we draw some conclusions in Section 4.

2. Proposed method

In this paper, we propose a system for Speech-Command Recognition and Speaker Identification that has to work in industrial environments. Both SCR and SI are formulated as classification problems. As for SCR, given N commands, namely N classes, the classifier is trained with $N+1$ classes, where the last one represents the *non-command class*. This is important since the workers may talk to each other, thus the voice may refer to something that is not a command. Vice-versa, as for the SI task, we suppose that only the workers of the company are allowed to be inside the working area, thus given M workers (that is M speakers), we consider M classes.

2.1. Pre-processing and backbone network

For representing the audio samples, we chose a Mel spectrogram-based representation, widely adopted in the literature [24], since it approximates the response of the human auditory system. Indeed, it allows for better frequency resolution at low frequencies, which is important for capturing the fundamental frequency of speech. This is an important and not negligible feature in our specific applications since both tasks pertain to the human voice.

Another important advantage of this kind of representation is that it results in a 2D matrix (time–frequency), which is the perfect input for feeding one of the well-known 2D-CNNs.

We decided to adopt as backbone ResNet-8, proposed in [25] since it has demonstrated impressive performance in audio analytic tasks while keeping low the computational burden [26]. The latter is a relevant feature since the system has to work over embedded devices.

A visual representation of the *ResNet-8* is shown in Fig. 2(a). The network follows the standard ResNet architectures. It starts with a convolutional layer (*conv_0*) and a Global Average Pooling layer (*GAP*) to set the size of the input and the output of the following six residual blocks to a fixed value. Each residual block (*ResNet block*) uses a bias-free dilated convolution layer with weights $W \in \mathbb{R}^{(l\times h)\times c}$, where l and f are the width and the height, and c the channel of the feature maps. At the convolution output, ReLU activation units feed a Batch Normalization level. The network ends with a Fully Connected layer (*FC*) using the Softmax activation function that releases the probability vector.

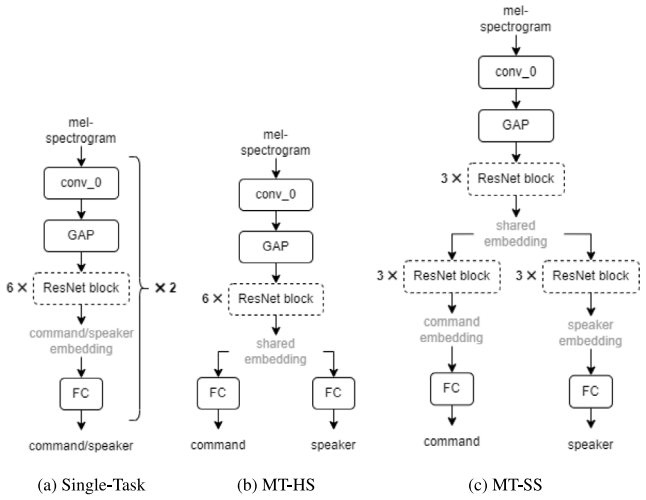

Fig. 2. Architectures. (a) Single-task network doubled to carry out both SCR and SI; (b) MT-HS: multi-task network with a shared embedding extractor and one classification branch for each task; (c) MT-SS: Multi-Task network with a shared feature extractor and two separated branches to produce task-specific embedding for the final classification.

2.2. Multi-task architectures

Two possible versions of the multi-tasking network, namely hard and soft parameters sharing [27], have been evaluated. The two architectures are shown in Figs. 2(b) and 2(c), respectively. A summary with the abbreviation used is reported in Table 1.

The networks, composed of a set of shared modules, produce a common representation and a branch for each task. Each branch takes the common representation (shared embedding) and after processing returns the associated probability vector. The two versions of the multi-task networks share the same backbone network (*ResNet-8*). The difference between the two approaches lies in the bifurcation point, which regulates the trade-off between the degree of sharing and the specialization of features.

Multi-Task Hard Sharing (hereinafter MT-HS) shares the whole feature extractor. It is opposed to the usage of two single-task networks. Using two distinct networks, made up of a feature extractor and a classifier, each network is optimized for its task, obtaining maximum specialization and minimum sharing. Vice-versa, in the MT-HS network, the entire feature extractor is shared (namely *conv_0*, *GAP*, and 6×*ResNet block*). The specialization branches are composed only of the classifier (namely *FC*) that returns the predictions based on the shared embedding. The use of a shared feature extractor not only allows for halving the network size (only the last *FC* layer is task-specific and, thus, doubled) but also normalizes the training due to the interaction of the tasks during the optimization process. This would be from a computational and more generally from a resource point of view the best choice. Anyway, for the two tasks, it could be also beneficial to have some features specialized for that task.

This is why we also employed an intermediate solution between the MT-HS and the two single-task networks, namely a Multi-Task Soft Sharing (MT-SS) approach. The feature extractor is split into two parts: the first part (3×*ResNet blocks*) is shared and extracts high-level features (*shared embeddings*) containing generic audio cues. The second part (3×*ResNet blocks*) is duplicated for each task to obtain more specialized audio features. The split point modulates the trade-off between sharing and specialization, and therefore also regulates the size of the network.

2.3. Multi-task loss function

In multi-task learning approaches the definition of the loss function $L(t)$ to be used during the training plays a crucial role. The most

Table 1
Summary of the two proposed multi-task networks, with the related abbreviations used in the paper (namely MT-HS and MT-SS).

Architecture	Description
MT-HS	Multi-Task network with Hard Sharing of the features (see Fig. 2(b))
MT-SS	Multi-Task network with Soft Sharing of the features (see Fig. 2(c)) features.

common technique is the weighted sum of the single losses:

$$L(t) = \sum_{i=1}^{n} k_i \cdot L_i(t), \tag{1}$$

where n is the number of tasks ($n = 2$ in our case), and k_i is the loss weight for the ith task. In our case the loss for both SCR and SI is a Cross-Entropy:

$$L_i(t) = -\frac{1}{m_i} \sum_{j=1}^{m_i} y_{i,j}(t) \cdot log(\hat{y}_{i,j}(t)), \tag{2}$$

where m_i is the number of labels for task i, $y_{i,j}$ and $\hat{y}_{i,j}$ are the groundtruth and the prediction for the mth label of the ith task, respectively.

The weighted sum method requires an expensive initial grid search to obtain the appropriate weights k_i and can only be used if the losses share the same definition intervals and the tasks have similar complexities. In our case, using the same loss does not pose any issues with definition intervals. However, due to the significant difference in the number of classes to manage (32 commands versus 562 speakers), the two tasks differ significantly in their complexity, resulting in varying training speeds.

In addressing the aforementioned challenges, we recommend adopting the GradNorm strategy, as introduced by Chen et al. [28], for weight selection. This approach facilitates the incorporation of dynamic coefficients into each task loss, thereby enabling the adjustment of the training speed and emphasizing the updating process for each task branch. GradNorm achieves this by dynamically adapting the gradient norms, ensuring that different tasks train at comparable rates, and placing the gradient norms on a common scale.

To compute the loss function with GradNorm, we need first to define:

- $G^i_w(t) = \left\| \nabla_W w_i(t) L_i(t) \right\|_2$, which is the L_2 norm of the gradient of the weighted single-task loss $w_i(t)L_i(t)$ with respect to the weights of the last shared layer W.
- $\bar{G}_W(t) = E[G^i_W(t)]$, which is the average gradient norm across all tasks at training step t, to determine the gradient size.
- $r_i(t) = \frac{\tilde{L}_i(t)}{E[\tilde{L}_i(t)]}$, computed as the relative inverse training rate of task i, useful to balance the gradients. The ratio describes the complexity of the task: the higher the loss of an individual task compared to the average of all tasks, the greater its inherent complexity.

GradNorm loss $L(t)$ is defined as a L_1 loss function between the current gradient norm for the ith task and the desired one, given by the product of the average gradient norm (re-scaling action) and the relative inverse training rate (balancing action).

$$L(t) = \sum_{i=1}^{n} \left| G^i_w(t) - \bar{G}_W(t) \times [r_i(t)]^{\alpha} \right|_1, \tag{3}$$

In simple words, the algorithm calculates the gradient norm for each task $G^i_w(t)$ and a common scale, identified by the average gradient norm $\bar{G}_W(t)$. Then, by applying the L_1 norm on the difference between the two calculated quantities and minimizing the result, the gradient scale is normalized. The product between $\bar{G}_W(t)$ and relative inverse training rate $r_i(t)$ balances the gradient with the complexity. α hyperparameter modulates the restoring force that pulls tasks back to a common training rate.

3. Experimentation

3.1. Dataset

In this paper, we propose a novel dataset called MIVIA-Industrial Speech Commands (MIVIA-ISC in brief). It consists of 31 English commands acquired from both synthetic and real speakers. By using a Telegram bot, human speakers were able to comfortably record samples using their smartphones. This approach not only increased the number of samples acquired but also allowed for greater variability in the acquisition devices used, namely the different types of smartphones used by individuals. The dataset contains 4801 real samples and 4972 synthetic ones. Synthetic samples are obtained from 6 TTS services, 3 paid services (Amazon Polly, Google Cloud, Azure), and 3 free services (Nvidia Nemo, Vocalware, and Naturalreaders). For rejection samples, we use a fraction of the Google Speech Commands [29] and Mozilla Common Voice [30] datasets extracting 6532 and 4552 samples, respectively. Each speaker acquires at least 15 samples (not all the speakers pronounced all the commands). The dataset includes commands for various purposes, such as controlling an adaptive workstation, requesting the handover of tools, managing movement, and conducting standard interactions.[1]

Overall, the dataset is composed of 20,857 samples, spoken by 562 speakers. The gender distribution among the speakers is relatively balanced, with approximately 44% being males and 56% females. In terms of age distribution, most of the samples have been acquired by speakers in the 20–50 age range; this is reasonable in an industrial environment, where we expect that most of the workers lie in this age range. Regarding the distribution of commands within the dataset, approximately 44% of the samples belong to the "reject" class, while the remaining samples are equally distributed among other categories, thus resulting in 32 classes. The class imbalance was introduced to address a key challenge in Human–Robot Interaction (HRI), namely enhancing the system's rejection capability to minimize false positives and prioritize caution. In the experimentation, the dataset is divided into 80% training, 10% validation, and 10% testing sets.

3.2. Implementation details

Starting from the raw audio waveform, we resample to 16,000 Hz and window the signal with a Hann function of size 25 ms, with hops of 12.5 ms to obtain a 50% overlap. Then, we compute the power spectrogram in dB using the Mel scale with 40 filter banks. As a backbone network, we use ResNet-8 with a pool size of $(4, 3)$ and an embed size of 90. We train the network using a batch size of 128 and Adam optimizer with a weight decay of 10^{-4}. We use a "ReduceLROnPlateau" algorithm to schedule the learning rate, which starts from 10^{-2} and arrives at 10^{-5} with the patience of 5. To prevent overfitting, we use the "early stopping" strategy with the patience of 8. Finally, as for GradNorm loss, we limit the restoring force that balances the training rate between tasks setting $\alpha = 0.5$, because the tasks have different complexity.

[1] The commands list includes: *increase the illumination, decrease the illumination, increase the height, decrease the height, increase the inclination, decrease the inclination, bring me the gun screwdriver, take the gun screwdriver, bring me the elbow screwdriver, take the elbow screwdriver, bring me the hammer, take the hammer, bring me the screwdriver, take the screwdriver, bring me the lever, take the lever, bring me the windows control panel, take the windows control panel, bring me the rearview mirror, take the rearview mirror, release, come here, go, start, stop, move to the right, move to the left, move up, move down, move forward, move backward, no command.*

Table 2
Accuracy of the proposed system computed on the test set. The results of both Speech-Command Recognition (SCR) and Speaker Identification (SI) tasks are reported. Furthermore, we report the weights used for the loss function, being the ordering [SCR weight, SI weight]. [1,1] is the traditionally adopted combination, while [0.8, 1.2] has been fixed with the weights automatically found by GradNorm at the end of the training process on MT-HS and MT-SS GradNorm approaches. As we can see, multi-task approaches are better than the counterpart single-task network on the Speaker Identification task (+11%). Furthermore, GradNorm-based approaches are better than the other models, including the ones trained by fixing the weights to the values found by GradNorm.

Approach	Accuracy SCR	Accuracy SI	Weights [SCR,SI]
Single-Task SCR	**97.23**	–	–
Single-Task SI	–	70.07	–
MT-HS	93.20	75.46	[1, 1]
MT-SS	93.59	77.20	[1, 1]
MT-HS GradNorm	93.83	**80.97**	GradNorm
MT-SS GradNorm	94.05	80.55	GradNorm
MT-HS fixed weights	91.73	77.23	[0.8, 1.2]
MT-SS fixed weights	92.47	79.53	[0.8, 1.2]

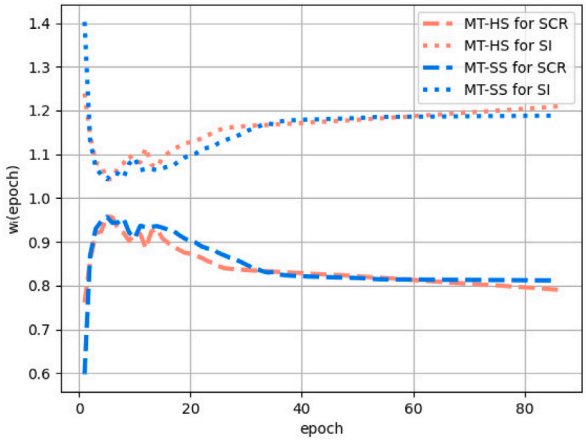

Fig. 3. Weights estimated by *GradNorm* during the training for the Multi-Task Hard-Sharing (MT-HS) and Multi-Task Soft-Sharing (MT-SS) networks in the SCR and tasks. $w_i(epoch)$ is the value of the weight for the ith task (dashed line for SCR and dotted line for SI) as the epochs increase. We can see that the weights are dynamically adapted epoch by epoch, reaching a value around 1 and converging to 0.8 and 1.2 for SI and SCR, respectively.

3.3. Results

We analyze the action of multi-task learning and the GradNorm algorithm in terms of both the accuracy and size of the models, to verify the possibility of running it over embedded devices.

The test set is divided into 20 unpaired folds balanced on both speakers and commands. The results are computed on each of them to obtain a sampling distribution of average accuracies for each approach. Using these mean distributions we have statistically validated the results through the Student's T-test. It is a hypothesis test in which two hypotheses may hold, namely H_0 (there is no significant mean difference between the population and sample mean); H_1 (there is a significant mean difference between the population and sample mean). To test the independence hypothesis H_1 we use the *p-value*. It computes $Pr(H_0) < \alpha$, where α is the reliability threshold that we set to 0.05. If the condition is true, it means that the hypothesis H_1 is true with $Pr(H_1) > 0.95$, implying that the results are statistically independent.

The baseline approach we consider in this paper consists of the use of two *ResNet-8* networks, one for each task. The input for both networks is the Mel spectrogram-based representation. We will refer to this baseline solution as "Single-Task".

Table 2 shows the test set results of each approach on the SCR and SI task. In the first two blocks of Table 2 (rows 1 to 4), we compare the single-task approach against the two multi-task networks. In the last column of the table, we also add the chosen weights for the multi-tasking architectures.

As expected, the different complexity between SCR and SI causes very different performances of single-task networks (97.23% for the SCR task and 70.07% for the SI task). Since multi-task networks use a loss function that combines single ones, the performance disparity strongly modulates their training. Indeed, the combination of losses adds constraints during training, normalizing it and limiting the navigation in feature space to make the training of complex tasks easier. This is reflected in the results on SI, where the MT approaches sharply increase the accuracy with a gain ranging from 5% (by MT-HS) to 7% (by MT-SS) over the single-task network. The beneficial effects of MT are modulated by the level of correlation of the two tasks, i.e., by the goodness of the limits imposed by each task on the others. While for SI the limitations lead the training in the right direction, this is not true for the "simple" task (SCR), where the addition of constraints on the cost function limits the achievement of highly specific features. This results in an accuracy reduction of 4%.

To mitigate the effects generated by the different complexity of the two tasks and favor the need for different training speeds in each of them, we have added GradNorm. Its normalizing action on gradients has led to a further increase in performance on both tasks, with greater emphasis on the more complex one. As we can see in Table 2 (rows 3 to 6), MT networks trained with GradNorm earn about 3−5% on SI and 0.5% on SCR, resulting in an overall gain of 10−11% on SI and a slight loss of 3% on SCR with respect to the single-task. It means that the performance with respect to the single-task is strongly better for the SI task (+11%), and almost comparable for the SCR task (−3%).

Finally, we retrained the networks with the weights found using GradNorm, which are 0.8 for SCR and 1.2 for SI. This analysis is useful to understand if the dynamic action of the GradNorm, which changes the weights epoch by epoch, is important during the training. The results are shown in the last two blocks of Table 2 (rows 5 to 8). We can note the superiority of GradNorm with an increment of 2% in SCR and 1−4% in SI.

The explanation for this can be found by looking at Fig. 3, where we can see how the weights evolve with GradNorm epoch by epoch. The weights do not converge immediately to the final values of 0.8 (SCR) and 1.2 (SI). In the first epoch, the algorithm differentiates between the two tasks and prioritizes the SI task. As the training progresses, the algorithm adjusts the coefficients to approximately equal values (around 1) to learn the generic audio features that are crucial for both tasks. In the end, the algorithm prioritizes the more complex task (SI) to further specialize the system.

It is worth noting that all the improvements and worsening commented on above passed the Student's T-test, except for the comparison between networks trained with *GradNorm* versus those with weights fixed to $[1, 1]$ for the SCR task. Although the accuracy improvement with *GradNorm* was 0.5%, the difference was not statistically significant.

Finally, we assess the enhancements in terms of memory footprint and inference time. For this evaluation, we employ an NVIDIA Jetson Xavier NX, a widely utilized embedded platform in robotics. The obtained results are presented in Table 3. The single-task approach, by duplicating the single-task networks, doubles the number of parameters (930,000), the size of the architecture (3,72 MB), and, therefore, the execution time (22.1 ms). The introduction of the multi-task learning paradigm significantly reduces the computational burden, by sharing the feature extractor or part of it. In fact, in Table 3, we note that the network that shares the entire feature extractor (MT-HS) significantly reduces (−47%) both the number of parameters (492,000) and the size of the network (1.97 MB). This reduction results in a similar reduction (−48%) in the processing time (11.5 ms vs 22.1 ms). Similarly, the

Table 3
Difference in terms of number of parameters, size of the network (in MB), and inference time (in milliseconds) between two single-task networks and a single multi-task network for both tasks. The results are reported for the two versions of multi-task networks, namely Soft Sharing (MT-SS) and Hard Sharing (MT-HS). The percentage reduction of the multi-task networks compared to the two single-task networks is shown in brackets. As we can see, sharing the feature extractor module of the SCR and SI tasks, reduce significantly the number of the model's parameters and thus the size of the model (up to 47%), and also the inference time (up to 48%). The tests have been conducted on an embedded device, namely an NVIDIA Jetson Xavier NX.

Approach	#Params	Size (MB)	Time (ms)
Single-Task SCR	441,000	1.76	10.7
Single-Task SI	489,000	1.96	11.4
Two Single-Task SCR-SI	930,000	3.72	22.1
MT-HS	492,000	1.97 (−47%)	11.5 (−48%)
MT-SS	710,000	2.84 (−24%)	16.5 (−25%)

network that partially shares the feature extractor (MT-SS) reduces (−24%) both the number of parameters (710,000) and the size of the network (2.84 MB), obtaining an almost similar reduction on the inference time (16.5 ms vs 22.1 ms).

To confirm that the proposed system can run on NVIDIA Jetson Xavier NX, we also conducted a comprehensive analysis of the entire execution process, including signal preprocessing and command inference. The preprocessing step takes approximately 8.2 ms. Command and speaker detection requires $8.2+11.5 = 19.7$ ms for the Hard Sharing model, and $8.2 + 16.5 = 24.7$ ms for the Soft Sharing model. As we can see, both approaches enable real-time computation. Furthermore, it is crucial to recognize that in most cases, a robotic platform runs multiple modules concurrently, making computational optimization a strong requirement.

4. Conclusion

In this paper, we propose a system able to simultaneously recognize commands and speakers by the analysis of the voice for industrial environments, where the possibility to work over embedded platforms is an important and not negligible feature. In order to meet the above constraint, we propose a multi-task network able to deal with both tasks. Since the two tasks have different complexities, we propose to use a dynamic adaptation of the loss weights by a GradNorm.

In addition, we have created a pioneering dataset known as MIVIA-ISC to facilitate the training and testing of our proposed networks. The MIVIA-ISC dataset includes 20,857 English voice commands, which have been pronounced by 562 speakers from around the world and are categorized into 31 distinct command classes.

By employing the proposed multi-task learning approach, we have not only reduced the network size by up to 47% and the execution time up to 48%, but also achieved a significant improvement in speaker identification accuracy by +7%, at the expense of a lower command recognition accuracy by −4%. Furthermore, by leveraging *GradNorm*, we have further improved speaker identification gain to +11%, while reducing speaker classification loss to −3%. In summary, our proposed system demonstrates remarkable performance in recognizing commands and authenticating speakers simultaneously, achieving accuracy rates of 94% for command recognition and 81% for speaker identification. Despite its high accuracy, the system has a small size of only 1.97 MB (with 492,000 parameters) and can be executed in real-time (19.7 ms) on embedded devices with low computational capacity.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work has received funding from the EU Horizon 2020 programme under GA No. 101017151 FELICE and from PNRR MUR project PE0000013-FAIR.

References

[1] D. Freire-Obregón, K. Rosales-Santana, P.A. Marín-Reyes, A. Penate-Sanchez, J. Lorenzo-Navarro, M. Castrillón-Santana, Improving user verification in human-robot interaction from audio or image inputs through sample quality assessment, Pattern Recognit. Lett. 149 (2021) 179–184.
[2] S. Rossi, F. Ferland, A. Tapus, User profiling and behavioral adaptation for HRI: A survey, Pattern Recognit. Lett. 99 (2017) 3–12.
[3] Y. Qian, X. Bianv, Y. Shi, N. Kanda, L. Shen, Z. Xiao, M. Zeng, Speech-language pre-training for end-to-end spoken language understanding, in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2021, pp. 7458–7462.
[4] P. Foggia, A. Greco, A. Roberto, A. Saggese, M. Vento, A social robot architecture for personalized real-time human-robot interaction, IEEE Internet Things J. (2023) 1, http://dx.doi.org/10.1109/JIOT.2023.3303196.
[5] P. Warden, Speech commands: A dataset for limited-vocabulary speech recognition, 2018, arXiv e-prints, arXiv–1804.
[6] M. Ayache, H. Kanaan, K. Kassir, Y. Kassir, Speech command recognition using deep learning, in: 2021 Sixth International Conference on Advances in Biomedical Engineering, ICABME, 2021, pp. 24–29, http://dx.doi.org/10.1109/ICABME53305.2021.9604862.
[7] D.C. de Andrade, S. Leo, M.L.D.S. Viana, C. Bernkopf, A neural attention model for speech command recognition, 2018, CoRR URL http://arxiv.org/abs/1808.08929.
[8] Z. Bai, X.-L. Zhang, Speaker recognition based on deep learning: An overview, Neural Netw. 140 (2021) 65–99.
[9] G. Humblot-Renaux, C. Li, D. Chrysostomou, Why talk to people when you can talk to robots? Far-field speaker identification in the wild, in: 2021 30th IEEE International Conference on Robot & Human Interactive Communication, RO-MAN, IEEE, 2021, pp. 272–278.
[10] G. Fenu, M. Marras, G. Medda, G. Meloni, Causal reasoning for algorithmic fairness in voice controlled cyber-physical systems, Pattern Recognit. Lett. 168 (2023) 131–137.
[11] N.N. An, N.Q. Thanh, Y. Liu, Deep CNNs with self-attention for speaker identification, IEEE Access 7 (2019) 85327–85337.
[12] I. Shahin, A.B. Nassif, N. Hindawi, Speaker identification in stressful talking environments based on convolutional neural network, Int. J. Speech Technol. 24 (2021) 1055–1066.
[13] P. Foggia, A. Greco, A. Roberto, A. Saggese, M. Vento, Few-shot re-identification of the speaker by social robots, Auton. Robots 47 (2) (2023) 181–192.
[14] L. Deng, X. Li, Machine learning paradigms for speech recognition: An overview, IEEE Trans. Audio Speech Lang. Process. 21 (5) (2013) 1060–1089.
[15] R. Lotfian, C. Busso, Predicting categorical emotions by jointly learning primary and secondary emotions through multitask learning, 2018, Interspeech 2018.
[16] A. Montalvo, J.R. Calvo, J.-F. Bonastre, Multi-task learning for voice related recognition tasks, in: Proc. Interspeech 2020, 2020, pp. 2997–3001, http://dx.doi.org/10.21437/Interspeech.2020-1857.
[17] W. Ding, L. He, MTGAN: Speaker Verification through Multitasking Triplet Generative Adversarial Networks, in: Proc. Interspeech 2018, 2018, pp. 3633–3637, http://dx.doi.org/10.21437/Interspeech.2018-1023.
[18] Z. Tang, L. Li, D. Wang, Multi-task recurrent model for speech and speaker recognition, in: 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA, 2016, pp. 1–4.
[19] M. Jung, Y. Jung, J. Goo, H. Kim, Multi-task network for noise-robust keyword spotting and speaker verification using CTC-based soft VAD and global query attention, in: Proc. Interspeech 2020, 2020, pp. 931–935, http://dx.doi.org/10.21437/Interspeech.2020-1420.
[20] Y. Li, A. Parsan, B. Wang, P. Dong, S. Yao, R. Qin, A multi-tasking model of speaker-keyword classification for keeping human in the loop of drone-assisted inspection, Eng. Appl. Artif. Intell. (ISSN: 0952-1976) 117 (2023) 105597, http://dx.doi.org/10.1016/j.engappai.2022.105597, URL https://www.sciencedirect.com/science/article/pii/S0952197622005875.
[21] Z. Chen, V. Badrinarayanan, C.-Y. Lee, A. Rabinovich, Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks, in: International Conference on Machine Learning, PMLR, 2018, pp. 794–803.
[22] J. Zhang, Y. Peng, V.T. Pham, H. Xu, H. Huang, E.S. Chng, E2E-based multi-task learning approach to joint speech and accent recognition, in: Proc. Interspeech 2021, 2021, pp. 1519–1523, http://dx.doi.org/10.21437/Interspeech.2021-1495.

[23] S. Sigtia, E. Marchi, S. Kajarekar, D. Naik, J. Bridle, Multi-task learning for speaker verification and voice trigger detection, in: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2020, pp. 6844–6848.

[24] S. Verbitskiy, V. Berikov, V. Vyshegorodtsev, Eranns: Efficient residual audio neural networks for audio pattern recognition, Pattern Recognit. Lett. 161 (2022) 38–44.

[25] R. Tang, J. Lin, Deep residual learning for small-footprint keyword spotting, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2018, pp. 5484–5488, http://dx.doi.org/10.1109/ICASSP.2018.8462688.

[26] R. Vygon, N. Mikhaylovskiy, Learning efficient representations for keyword spotting with triplet loss, in: Speech and Computer: 23rd International Conference, SPECOM 2021, St. Petersburg, Russia, September 27–30, 2021, Proceedings 23, Springer, 2021, pp. 773–785.

[27] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, L. Van Gool, Multi-task learning for dense prediction tasks: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 44 (7) (2022) 3614–3633, http://dx.doi.org/10.1109/TPAMI.2021.3054719.

[28] Z. Chen, V. Badrinarayanan, C.-Y. Lee, A. Rabinovich, Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks, in: International Conference on Machine Learning, PMLR, 2018, pp. 794–803.

[29] P. Warden, Speech commands: A dataset for limited-vocabulary speech recognition, 2018, arXiv e-prints, arXiv–1804.

[30] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F.M. Tyers, G. Weber, Common voice: A massively-multilingual speech corpus, 2019, arXiv preprint arXiv:1912.06670.