Guided deep embedded clustering regularization for multifeature medical signal classification

Yamil Vindas[a,*], Emmanuel Roux[a], Blaise Kévin Guépié[b], Marilys Almar[c], Philippe Delachartre[a]

[a] *Univ Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM-Saint Etienne, CNRS, Inserm, CREATIS UMR 5220, U1294, F-69100, LYON, France*
[b] *Université de Technologie de Troyes / Laboratoire Informatique et Société Numérique, 10004 Troyes, France*
[c] *Atys Medical, 17 Parc Arbora, 69510 Soucieu-en-Jarrest, France*

Abstract

Medical signal classification often focuses on one representation (raw signal or time frequency). In that context, recent works have shown the value of exploiting different representations simultaneously. We propose a regularized end-to-end trained model for classification in a medical context exploiting both the raw signal and a time-frequency representation (TFR). First, a 2D convolutional neural network (CNN) encoder and a 1D CNN-transformer encoder start by extracting embedded representations from the TFR and the raw signal, respectively. Then, the obtained embeddings are fused to form a common latent space that is used for classification. We propose to guide the training of each encoder by applying two iterated losses. Moreover, we propose to regularize the fused common latent space using deep embedded clustering. Extensive experiments on three medical datasets and ablation studies show the adaptability and good performance of our method for medical signal classification. Our method makes it possible to improve the classification performance from 4% to 12% MCC on a transcranial Doppler dataset, when compared with single-feature counterparts, while giving more stable models. The code is available at:

[⋆] Fully documented templates are available in the elsarticle package on CTAN.
[*] Corresponding author
Email address: `yamil.vindas@creatis.insa-lyon.fr` (Yamil Vindas)

https://github.com/gdec-submission/gdec/[1][2].

Keywords: Multifeature learning, Deep regularization, Guided training, Signal classification, Transcranial Doppler

2010 MSC: 00-01, 99-00

1. Introduction

In the past few years, signal processing has taken advantage of different machine learning techniques to solve tasks such as classification, segmentation, and denoising among others [1]. It is now well known that convolutional neural networks (CNNs) are very effective for almost all image-related tasks [2]. The common component of CNNs is the use of multidimensional convolution kernels with learnable parameters, which are able to extract features from the input images by exploiting their spatial context. However, for temporal-dependent signals there is no such type of model that has proven to be as effective in a large range of tasks by directly exploiting the raw signal. A common practice for handling these signals is to manually extract a time-frequency representation (TFR) and then pass it through a 2D CNN [3, 4]. Even though this strategy can achieve state-of-the-art performance in several tasks, it does not always lead to good results [5].

In this paper, we focus on temporal-dependent signals and, more specifically, on medical signals from the monitoring of cerebral blood flow by means of transcranial Doppler (TCD) for the detection of emboli. We also focus on signals to monitor heart activity (electrocardiograms, ECGs) for heartbeat categorization, and brain activity (electroencephalograms, EEGs) for epilepsy seizure recognition (ESR). These three tasks are important for public health since stroke, cardiovascular diseases (which are the leading cause of death and disability worldwide [6]), and epilepsy (one of the most common neurological diseases) can be detected using TCD, ECG, and EEG, respectively. Some works have

[1] Username gdec-submission and password $Gjaq * \& * K7vq44azu$

[2] Mail: gdec.submission@gmail.com, Password: $1\#tU6mKAXqGT8\#CY$

tried to solve these tasks by using classic signal processing techniques and machine learning techniques [7, 8, 9]. However, few works have directly exploited the raw signal [10, 5] as handcrafted features (or TFR) are often extracted to be fed to the different models [8, 11, 12, 13, 14].

Furthermore, inspired from natural language processing (NLP), several methods have been proposed to exploit the temporal context of time-dependent signals. These models range from recurrent neural networks [15] and 1D CNNs [16, 10] to convolutional deep belief networks [17] and transformers [18, 4]. Even though different representations (TFR or raw signal) are used to solve the task, the question of the optimal representation remains open. The current trend is to use multiple representations of a single signal to solve the task [19, 11], but few works consider the raw signal as a representation itself, and usually only different handcrafted features or TFRs are used. What is more, these methods are often designed for a very particular task, do not exploit the temporal/spectral complementarity, or are not end-to-end trainable.

Inspired by the aforementioned motivations, we propose a regularized hybrid CNN-transformer capable of exploiting both the temporal and spectral information through the use of the raw signal and a TFR. More specifically, the model is composed of four main components. First, two encoders, a 1D CNN-transformer and a 2D CNN, are used to extract features from the raw signal and the log-magnitude spectrogram, respectively. Second, a fusion layer takes the extracted encoding of both of the initial representations to create a common encoding, which is then passed through a classifier. Third, to guide the learning of the encodings of each representation, we propose to use an iterated loss [20]. Finally, in order to enforce clustering in the different latent spaces, we propose to regularize the model using deep embedded clustering (DEC) [21].

Our main contributions can be summarized as follows:

- End-to-end joint trained multifeature model, capable of simultaneously exploiting complementary information of different representations.
- Regularization strategy to guide the learning of the encoding of each in-

dividual input representation, thanks to the use of an iterated loss.

- Regularization strategy of the common (fused) feature space, encouraging more separable and dense clusters based on deep embedded clustering.

- Extensive evaluation using one private dataset and two public datasets, validating the effectiveness and adaptability of our proposed method.

The rest of the paper is structured as follows. In section 2 we introduce the works related to our method. In section 3 we present in detail the different components of our method. In section 4 we describe the experimental set-up that we use to validate our approach as well as the results and their discussion. Finally, in section 5 we conclude and present the guidelines of our future work.

2. Related works

2.1. Machine learning signal classification

Drawing on 2D convolutions that are able to exploit the spatial context in an image, several methods propose the use of 1D convolutions on raw signals or 2D convolutions on TFR [16, 22, 23]. Lee et al. [16] proposed creating a multilevel representation of the raw signal to perform audio classification using 1D convolutions with residual connections, average pooling, and fully connected (FC) layers. Pu et al. [22] proposed using 1D Morlet filters with learnable parameters to extract a TFR from the raw signal. Then, they used 2D separable Morlet filters to get an embedding that is passed through a CNN or deep neural network (DNN) to perform speaker identification and acoustic event recognition. Sharan et al. [23] showed that combining different TFRs helps increase the classification performance of a 2D CNN model.

Moreover, other types of models such as recurrent neural networks (RNNs) or deep belief networks (DBN) have been used in signal classification [15, 24, 18, 17]. Indeed, RNNs can take into account the temporal dependencies in a signal better than CNNs can. Okawa et al. [24] proposed two representations based on the binary encoding of the raw signal amplitudes. They input this to different models (2D CNN, LSTM and bidirectional gated recurrent unit) to solve various tasks: acoustic event detection, music classification, and speech classification. Scarpiniti et al. [17] proposed the use of a DBN for audio classification and construction site monitoring using as input different handcrafted statistic features computed from the mel-frequency cepstral coefficients (MFCC) of the raw signal.

Furthermore, with the rise of transformers [25] in the NLP community, several works have applied these types of models to signal classification [18, 20, 10, 4, 26]. Karita et al. [18] carried out a comparative study using 15 different audio speech recognition datasets, showing that transformers can be superior to RNN models (they outperformed the RNNs on 13 of the datasets). Che et al. [10] used a 1D CNN to extract features from the raw signal before feeding them to a transformer encoder in a channel-wise manner with respect to the output of the 1D CNN. Finally, Tjandra et al. [20] suggested a transformer model guided by an iterated loss and feature re-presentation to carry out audio and video classification. To do this, they extract features from the raw signal using a mel filterbank, which are then fed to a sequence of transformer modules. At different levels, the authors re-introduce input features of the model (feature re-presentation) and make intermediate predictions using the learned features at that particular level (iterated loss).

2.2. Multimodality and multiple-feature learning

Multimodality is an important and vast topic of research in the machine learning community [27]. The global aim is to search for strategies to combine data of different nature in a way that complementary information is well exploited. For instance, for video action recognition and audio event classification, images and audio can be combined [28, 26]. Ortega et al. [28] proposed performing emotion recognition using three different modalities: video, audio, and text. They extracted features from each modality using a DNN and then fused the obtained representations into a joint representation by concatenation.

Akbari et al. [26] extracted features from each modality (video, audio, and text) using an FC layer, and then passed each feature into a transformer encoder. Then, the extracted embeddings were projected in different common spaces with different granularities (data scale) to complete the classification.

Inspired by the benefits that multiple modalities can bring to classification problems, several works have applied these methods to different representations coming from the same modality. On the one hand, in the computer vision community, several works have tried to combine different features extracted from a single image [29, 30]. Zhu and Jian [30] extracted global features from images using 2D PCA and local features using local binary patterns. They then fused these features and passed it through a CNN to perform face recognition. Mao et al. [29] proposed performing object detection using iterative RELIEF to select the top three color components of an image in order to obtain three new images. Then, they fed these three images to a multipath CNN to extract features and fused them by concatenation. Finally, they projected the obtained feature using PCA and passed the resulting feature to a support vector machine (SVM) classifier.

On the other hand, in the signal processing community, several works focus on extracting TFRs and other handcrafted features to combine them afterward [31, 32, 33, 11]. Kim and Lee [31] extracted three TFR (spectrogram, mel-spectrogram, and MFCC) from the raw signal, concatenated them and passed the obtained feature through an LSTM model for power signal analysis. Feng et al. [32] computed different features from the raw signal (wavelet packet decomposition, gradient extreme value, fast Fourier transform, etc.) which were fused by concatenation and passed through a DBN. The DBN outputs were then selected by removing redundancy via the maximum information coefficient and were used for the classification step. Finally, Ahmad et al. Ahmad et al. [11] used ECG signals to classify heartbeats. They created three different images from the raw signal (Gramian angular field (GAF), recurrent plot (RP), and Markov transition field (MTF)) and fused them using two different strategies: an early fusion approach (images are fused before an AlexNet model), and an intermediate fusion approach (feature extraction by AlexNet from each of the three images and then fusion of the extracted features before an SVM classifier).

2.3. Regularization in deep learning

Building on Kukačka et al. [34], we can identify different regularization types.

First, regularization can be achieved through data. This can be done by adding some noise or perturbations at different levels [35, 36, 37], by using dropout [38], or by using normalization [39].

Moreover, regularization can be obtained via the model architecture. One can use (dilated) convolutions [40] to obtain lighter models without loss of prediction capacity, or add other components such as skip connections [41] and residual connections [42].

Otherwise, regularization can be performed through a regularization term in the loss. Different terms can be added, such as weight decay [43], Jacobian penalty [44], Hessian penalty [45], or loss-invariant backpropagation [46].

Other regularization approaches are early stopping [47] and mutual exclusivity [48]. Mutual exclusivity uses the unlabeled samples of a partially labeled dataset to move the classifier decision boundaries in zones with few samples. For a more detailed explanation of regularization techniques, we refer readers to [34].

3. Proposed method

In this work, we propose a regularized end-to-end multifeature joint trained model exploiting complementary information of different representations. Regularization is done through: (1) two iterated losses allowing to guide the training of each single-feature encoder and (2) deep embedded clustering (DEC) on the joint embedding space and adapted to a supervised-learning context, allowing to improve generalization and partially handle imbalanced datasets. Even though we design our method to be suitable to other architectures and representations,

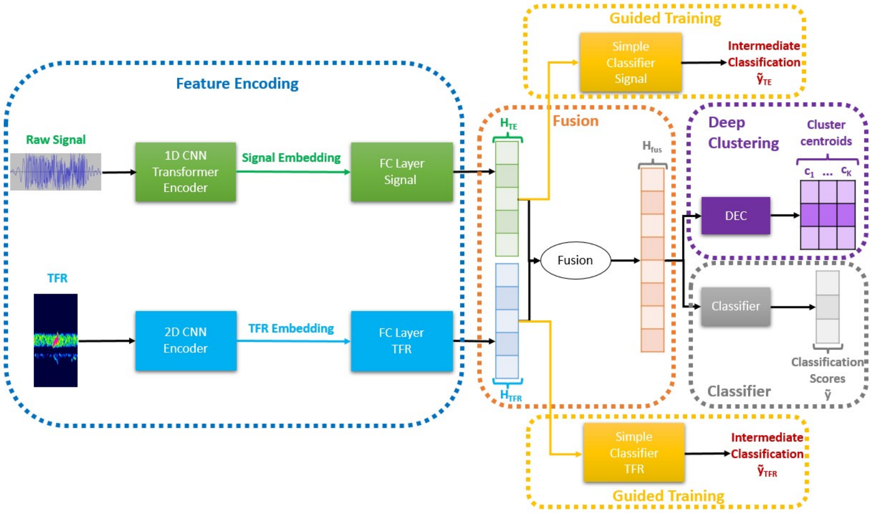


Figure 1: Global pipeline of the proposed method. The method has five modules: a feature encoding module (dotted blue box) composed of one encoder for each input representation, one guided training module to individually guide their training (dotted yellow boxes), a fusion module (dotted orange box) to create a joint embedding space, a classifier (dotted gray box) using the obtained joint representation, and a deep embedded clustering (DEC) module (dotted purple box) to regularize the joint embedding space.

we are going to focus on a hybrid CNN-transformer exploiting both the temporal information through the raw signal and the spectral information through a TFR. In this section, we present in detail each stage of our proposed method.

Let us define some notations. Suppose that we have a dataset composed of N labeled samples $\{\mathbf{X}_i\}_{i\in[1,N]}$ and K classes. We suppose that each sample has two different representations, a raw signal $\mathbf{X}^i_{TE} \in \mathbb{R}^{L\times C}$ composed of L samples and C channels, and the TFR $\mathbf{X}^i_{TFR} \in \mathbb{R}^{F\times M}$ composed of F frequency bins and M time bins. The aim is to investigate the classification task improvement that can be achieved by using both representations instead of a single one.

3.1. Global pipeline

The global pipeline of the proposed method is introduced in figure 1. It is composed of five modules: one encoding module, one fusion module, one guided training module, one classification module, and one deep clustering module.

The encoder module is composed of two encoders, one for each input feature,

$\mathbf{X}_{TE}$ and $\mathbf{X}_{TFR}$. The raw signal input is encoded by the 1D CNN-transformer encoder part of [5] denoted $\mathcal{E}_{TE}$, whereas the TFR input is encoded by the 2D CNN encoder of the same work, denoted $\mathcal{E}_{TFR}$. The detailed architectures of both encoders can be found in figure 2. To enable the sum of the two encodings (fusion strategy), two FC layers (FC_{TE} and FC_{TFR}) project them into spaces of the same dimension d_{com}. We denote as $\mathbf{H}_{TE} = FC_{TE}(\mathcal{E}_{TE}(\mathbf{X}^{TE})) \in \mathbb{R}^{d_{com}}$ and $\mathbf{H}_{TFR} = FC_{TFR}(\mathcal{E}_{TFR}(\mathbf{X}^{TFR})) \in \mathbb{R}^{d_{com}}$ the projected embeddings of the raw signal and TFR, respectively. These embeddings are then used in the guided training and fusion modules.

Furthermore, we guide the training of each projected embedding using an iterated loss, as in [20]. The main idea is to have a vanilla classifier using the projected embedding to perform an intermediate classification. This module will be detailed in section 3.2.

Moreover, the fusion module combines the projected embeddings, $\mathbf{H}_{TE}$ and $\mathbf{H}_{TFR}$, of each input feature in a joint embedding space. We propose to explore two fusion methods, concatenation and weighted sum, to form the joint representation $\mathbf{H_{fus}}$:

- Concatenation:

  $$\mathbf{H_{fus}} = \mathbf{H_{TE}} \oplus \mathbf{H_{TFR}}$$

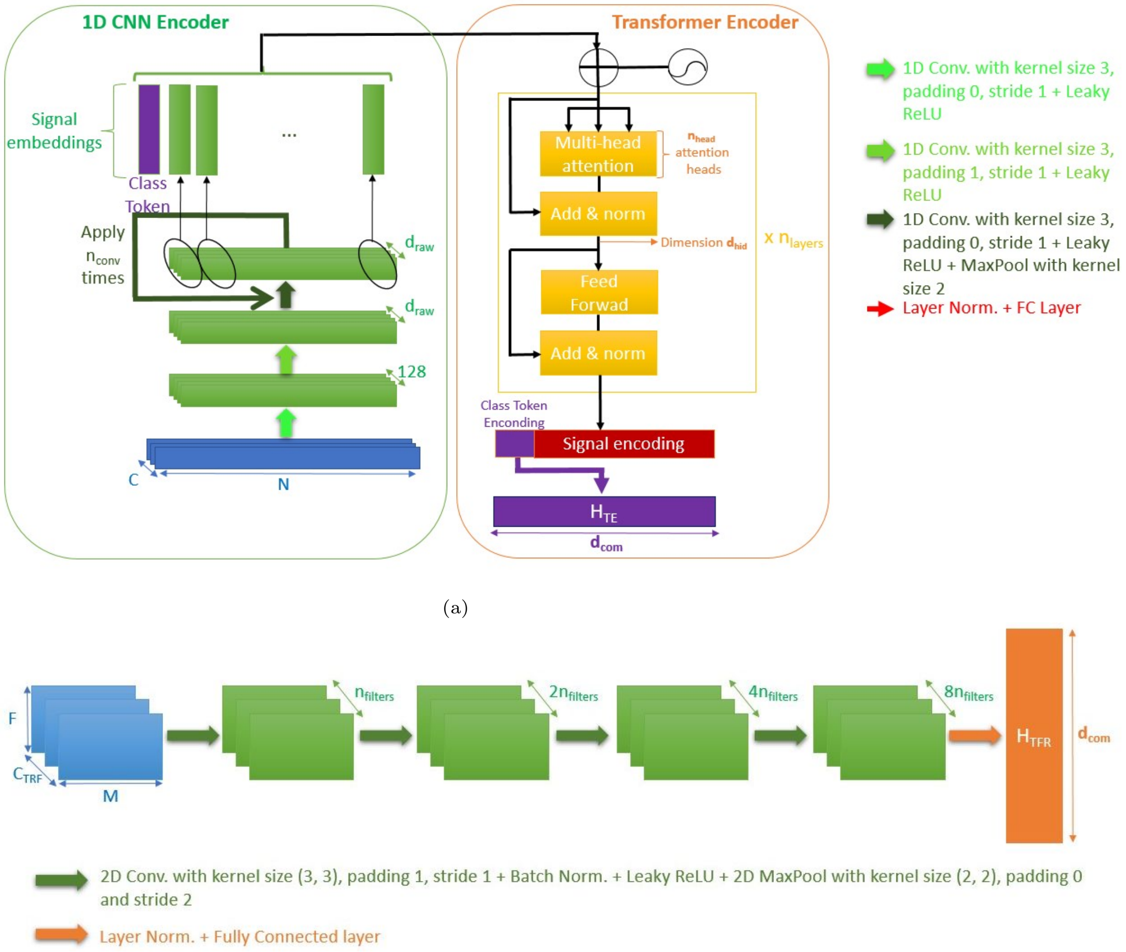
  where $\oplus$ is the concatenation operator. In this case, $\mathbf{H_{fus}} \in \mathbb{R}^{2\times d_{com}}$.

- Weighted sum:

  $$\mathbf{H_{fus}} = w_{TE} \times \mathbf{H_{TE}} + w_{TFR} \times \mathbf{H_{TFR}}$$

  where $w_{TE} \in \mathbb{R}$ and $w_{TFR} \in \mathbb{R}$ are two learnable parameters between 0 and 1 such that $w_{TE} + w_{TFR} = 1$. In this case, $\mathbf{H_{fus}} \in \mathbb{R}^{d_{com}}$.

This joint representation is then fed to the classification module to perform the final classification, and to a DEC module to cluster the input samples in the joint embedded space. This last module will be detailed in section 3.3.


1D CNN Encoder
Transformer Encoder
Signal embeddings
Class Token
...
Apply n_{conv} times
d_{raw}
d_{raw}
128
C
N
Multi-head attention
n_{head} attention heads
Add & norm
Dimension d_{hid}
x n_{layers}
Feed Forwad
Add & norm
Class Token Enconding
Signal encoding
H_{TE}
d_{com}
1D Conv. with kernel size 3, padding 0, stride 1 + Leaky ReLU
1D Conv. with kernel size 3, padding 1, stride 1 + Leaky ReLU
1D Conv. with kernel size 3, padding 0, stride 1 + Leaky ReLU + MaxPool with kernel size 2
Layer Norm. + FC Layer
F
C_{TRF}
M
$n_{filters}$
$2n_{filters}$
$4n_{filters}$
$8n_{filters}$
H_{TFR}
d_{com}
2D Conv. with kernel size (3, 3), padding 1, stride 1 + Batch Norm. + Leaky ReLU + 2D MaxPool with kernel size (2, 2), padding 0 and stride 2
Layer Norm. + Fully Connected layer

(a)

(b)

Figure 2: Encoder models used to extract embeddings from the raw signal and the TFR. (a) 1D CNN-transformer raw signal encoder. (b) 2D CNN TFR encoder.

3.2. Training guidance: iterated loss

We propose to guide the training of each encoder by using an iterated loss as in [20]. Hence, during training, two vanilla classification models are trained : one with $\mathbf{H}_{TE}$ as input, while the other one is fed by $\mathbf{H}_{TFR}$. It forces the encoders to produce structured intermediate embedding spaces ($\mathbf{H}_{TE}$ and $\mathbf{H}_{TFR}$) that are discriminative enough at this stage. Another advantage is that classification remains possible even if one of the input representations is not available. Indeed, even if one input features is not able to learn enough with the global classification and with DEC, the iterated loss will guide its training to make its projected embedding discriminative enough by itself[3].

The classifiers used for each input representation are two consecutive blocks of one normalization layer and one FC layer for $\mathbf{H}_{TE}$, and one FC layer and dropout for $\mathbf{H}_{TFR}$. Let us denote by $\tilde{\mathbf{y}}_{\mathbf{TE}}$ and $\tilde{\mathbf{y}}_{\mathbf{TFR}}$ the intermediate classification outputs of the guided trainings of the raw signal projected embedding ($\mathbf{H}_{TE}$) and the TFR projected embedding ($\mathbf{H}_{TFR}$), respectively. The iterated losses are defined as the cross entropy (CE) loss between the intermediate classification outputs ($\tilde{\mathbf{y}}_{\mathbf{TE}}$ and $\tilde{\mathbf{y}}_{\mathbf{TFR}}$) and the true labels of the samples. We denote them as $\mathcal{L}_{\mathbf{TE}}$ and $\mathcal{L}_{\mathbf{TFR}}$ (raw signal and TFR, respectively).

3.3. Regularization through feature clustering using deep embedded clustering (DEC)

The last module of our proposed method is the clustering module. This is done by applying DEC [21] to the joint representation embedding space. The rationale behind this clustering is twofold. First, we want to improve the generalization of the trained models by creating a more clustered latent space that will be used for classification. Second, we want to handle imbalanced datasets, which can be done by applying robust clustering methods such as DEC [49].

The main idea of DEC is to form K clusters using the embeddings obtained by an encoder model instead of using the original samples. In our particular case, we propose to apply DEC to the joint representations $\mathbf{H}_{fus}^{1}, \ldots, \mathbf{H}_{fus}^{N}$ obtained by the encoder and fusion modules, and the number of clusters is the number of classes K. We denote as $\mathbf{c}_1, \ldots, \mathbf{c}_K$ the centroids of the different clusters, and they are initialized using k-means. The objective of DEC is to jointly optimize these centroids and the weights of the encoder models.

[3] This can be seen as a decoupling of the global classification task and the feature encoding task.

Moreover, contrary to the original DEC paper, we avoid the pre-training stage by introducing a hyperparameter e_{init}, corresponding to the epoch from which DEC will be activated. The DEC regularization term is defined as follows:

$$\tilde{\mathcal{L}}_{DEC}(e_c) = \mathbb{1}_E(e_c) \times \mathcal{L}_{DEC} \tag{1}$$

where e_c is the current epoch number, $E = \{e \in \mathbb{N}/e \geq e_{init}\}$, and $\mathcal{L}_{DEC}$ is the DEC loss defined in [21].

The influence of e_{init} is discussed in [49], but the main idea is to choose an epoch where the joint embedded space starts producing some clusters, in order to avoid a bad initialization of the centroids using k-means.

3.4. Final loss function

We aim at achieving supervised classification using two representations of a signal: the raw signal and a TFR. Therefore, we optimize a CE loss function between the predicted labels, $\tilde{\mathbf{y}}$, obtained by passing the joint representation through a classifier (as shown in figure 1) and the true labels, $\mathbf{y}$. We denote this loss as $\mathcal{L}_{CE}$. The final loss function is optimized using gradient descent and is defined by combining the different terms mentioned in 3.2 and 3.3:

$$\mathcal{L} = \mathcal{L}_{CE} + \alpha \times \mathcal{L}_{TFR} + \beta \times \mathcal{L}_{TE} + \gamma \times \tilde{\mathcal{L}}_{DEC} \tag{2}$$

where α, β, and γ are hyperparameters regulating the importance of each regularization term to the final loss.

4. Experiments

4.1. Datasets

In order to validate our proposed method, we conducted different experiments using three datasets: one private TCD high intensity transient signals (HITS) dataset for cerebral emboli classification, one ECG dataset for heartbeat categorization, and one EEG dataset for epileptic seizure recognition. For all the experiments and all the datasets, we computed class weights using the Scikit Learn implementation. Without loss of generality, the models used on the different datasets have the same structure, but the hyperparameters of the architectures were adapted based on the dataset. Moreover, the core of our method (multifeature fusion, guided training and DEC regularization) was applied in the same way for all the datasets.

4.1.1. TCD HITS dataset

Our main interest is the classification of TCD HITS to help clinicians in preventing stroke. Therefore, we evaluated our proposed method on a HITS dataset composed of 1680 HITS extracted from 50 subjects, and distributed in three classes: 608 solid emboli, 616 gaseous emboli, and 456 artifacts. Each HITS sample is composed of an audio file representing the raw signal and an image representing the logarithmic scale spectrogram. We split the dataset according to the subjects into three subsets (a subject cannot be in the two subsets at the same time): one for training, with 58% of the samples, one for validation, with 34% of the samples, and one for testing, with 8% of the samples. It is important to note that the test HITS do not necessarily follow the exact same distribution as the train or validation ones, as some test signals can have a length greater than 1400 points, the maximal length observed in the train and validation sets.[4]

For more details about this dataset and the pre-processing steps, we refer the reader to [5].

4.1.2. ECG dataset

The PTB dataset, composed of 14 552 ECG lead-II signal, focuses on the identification of myocardial infarction and comprises two imbalanced classes: 1 0506 normal and 4 046 abnormal heartbeats. We used the standardized and

[4] This has also an impact on the TFR computation, as more points are used to obtain it.

pre-processed version from [50][5], and the logarithmic scale spectrograms were computed as in [5].

Moreover, we split the dataset intro three subsets: train (64%), validation (16%), and test (20%). The hyperparameters were selected using the validation set, then the model was retrained by regrouping the train and validation sets, to make a fair comparison with other state-of-the-art methods which only used train/test splits. For more details about this dataset and the pre-processing steps (padding, normalization, and spectrogram computation), we refer the reader to [50, 5].

4.1.3. EEG dataset

We used the Epileptic Seizure Recognition dataset (ESR) [51] from the UCI repository, composed of pre-processed EEG signals[6]. For the pre-processing details, we refer the reader to the description in the official UCI repository. The dataset has 11 500 samples distributed in five classes (equally distributed): (1) seizure activity, (2)–(5) no seizure activity. As in most other works, we focus on binary classification, where the first class is (1) and the second one regroups (2)–(5), obtaining an imbalanced dataset.

Each EEG signal is sampled at 178 Hz, and the logarithmic scale spectrograms were computed using $n_{fft} = 32$, $n_{overlap} = 4$ and a Blackman window.

Finally, for a fair comparison, we proposed to randomly split the dataset using 90% of the samples for training and 10% of the samples for testing as in [14][7].

4.2. General experimental set-up

We studied our proposed approach with three experiments. The objective of experiment 1 is to highlight the advantage of using the proposed regularized

[5] We used the public available versions found at `https://www.kaggle.com/datasets/shayanfazeli/heartbeat`

[6] We used the public available version found at `https://www.kaggle.com/datasets/harunshimanto/epileptic-seizure-recognition`

[7] Because of the structure of this dataset, it is very difficult to obtain a subject-wise train/test split.

end-to-end training for multifeature models. The objective of the second and third experiment is to study the influence of the guided training and DEC regularization (respectively) on the classification performance of the trained models.

4.2.1. Training and model parameters

The training and model parameters used in the different experiments can be found in tables 1 and 2, respectively. Models (a) and (b) are single-feature models from [5]: (a) a 1D CNN-transformer model taking as input the raw signal, and (b) a 2D CNN model taking as input the TFR. Model (c) is a late fusion model as in [5] using the two previous models (a) and (b) as base classifiers. Models (d) and (e) are the models that we propose in this paper, with the difference that model (d) is not regularized whereas model (e) is regularized with both DEC and guided training. Models (d) and (e) were trained with two different intermediate fusion strategies: concatenation and weighted sum (c.f. column *Fusion* in table 1). Additionally, all the models were trained with Adam optimizer and a batch size of 32 except for the late fusion model, which was trained with a batch size of 16. In experiment 1, all the models are evaluated, whereas in experiments 2 and 3, only (e) is studied.

For statistical purposes, all the experiments are repeated 10 times.

4.2.2. Evaluation metrics

In all the datasets, we evaluated the classification performance of the trained models using three main metrics: the Matthews correlation coefficient (MCC), the F1 score, and the classification accuracy. The MCC and F1 score are well suited for evaluating classification models on imbalanced datasets. For the multiclass datasets, we used the macro-averaged F1-Score.

4.2.3. Implementation details

All the codes were implemented using Pytorch and Scikit-Learn. The different experiments were executed on two high-performance computing clusters: one with 25 heterogeneous machines (each machine with between 16 Gb and 128

Table 1: Training parameters of the different models. α, β, and γ correspond to the importance of $\mathcal{L}_{TFR}$, $\mathcal{L}_{TE}$, and $\tilde{\mathcal{L}}_{DEC}$, respectively. *Cat* stands for concatenation and *Weight. Sum* for weighted sum. Two versions of our end-to-end trained model are evaluated: one without regularization, named *Ours (No Reg.)*, and another with the proposed regularization, named *Ours (Reg.)*. The 1D CNN-transformer and 2D CNN models are single/feature models, taking as input the raw signal and the TFR, respectively. The other models are multifeature models, taking as input the raw signal and the TFR. The *Late Fusion* model is the same as that of [5] but with different hyperparameters.

Dataset	**Model**	**Epochs**	**Learning rate**	**Weight decay**	α	β	γ	e_{init}	**Fusion**
HITS	(a) 1D CNN-trans.	150	0.07	$1e^{-7}$	-	-	-	-	-
	(b) 2D CNN	50	0.001	$1e^{-7}$	-	-	-	-	-
	(c) Late Fusion	15	0.01	$1e^{-8}$	-	-	-	-	
	(d) Ours (No Reg.)	150	0.3	$1e^{-7}$	-	-	-	-	Cat.
					-	-	-	-	Weight. Sum
	(e) Ours (Reg.)				0.01	0.1	0.01	50	Cat
					0.001	1	0.1	50	Weight. Sum
PTB	(a) 1D CNN-trans.	150	0.1	$1e^{-7}$	-	-	-	-	-
	(b) 2D CNN	50	0.0001	$1e^{-7}$	-	-	-	-	-
	(c) Late Fusion	15	0.01	$1e^{-8}$	-	-	-	-	
	(d) Ours (No Reg.)	150	0.3	$1e^{-7}$	-	-	-	-	Cat.
					-	-	-	-	Weight. Sum
	(e) Ours (Reg.)				0.01	1	0.0001	50	Cat.
					0.01	0.1	0.1	50	Weight. Sum
ESR	(a) 1D CNN-trans.	100	0.3	0.0001	-	-	-	-	-
	(b) 2D CNN	100	0.001	0.00001	-	-	-	-	-
	(c) Late Fusion	15	0.001	$1e^{-8}$	-	-	-	-	
	(d) Ours (No Reg.)	200	0.3	0.0001	-	-	-	-	Cat.
					-	-	-	-	Weight. Sum
	(e) Ours (Reg.)				0.01	1	0.0001	50	Cat. Weight. Sum

Table 2: Parameters of the models based on the dataset used. We refer the reader to figure 2 for the definition of the model parameters. Models (a), (b), (d), and (e) are the same as those in table 1. Model (c) from table 1 is not presented here since the base models of (c) are (a) and (b).

Dataset	**Model**	$\mathbf{n_{head}}$	$\mathbf{d_{hid}}$	$\mathbf{n_{layers}}$	$\mathbf{p_{dropout}}$	$\mathbf{n_{proj}}$	$\mathbf{d_{raw}}$	$\mathbf{n_{conv}}$	$\mathbf{n_{filters}}$	$\mathbf{d_{com}}$	Pool
HITS	(a) 1D CNN-trans.	8	64	8	0.1	10	128	2	-	-	-
	(b) 2D CNN	-	-	-	0.2	-	-	-	64	-	Max
	(d) Ours (No Reg.) (e) Ours (Reg.)	4	64	4	0.1	10	128	2	64	64	Max
PTB	(a) 1D CNN-trans.	8	64	8	0.1	10	128	4	-	-	-
	(b) 2D CNN	-	-	-	0.2	-	-	-	64	-	Max
	(d) Ours (No Reg.) (e) Ours (Reg.)	4	64	4	0.1	10	128	2	64	64	Max
ESR	(a) 1D CNN-trans.	4	8	4	0.3	4	64	2	-	-	-
	(b) 2D CNN	-	-	-	0.2	-	-	-	64	-	Max
	(d) Ours (No Reg.) (e) Ours (Reg.)	4	8	4	0.3	4	64	2	64	64	Max

Gb of RAM, CPUs with 8–32 cores, and different types of Nvidia Quadro RTX and Tesla GPUs), and another with NVIDIA Tesla V100 GPUs[8]. The GitHub for the PTB and ESR experiments can be found at: https://github.com/gdec-submission/gdec/[9][10].

4.3. Results

4.3.1. Experiment 1: Advantage of end-to-end training

The objective of this experiment is twofold: (1) to show the increase in classification performance yielded by our proposed method, and (2) to compare our method with state-of-the-art methods on different datasets. To this end, we trained models (a)–(e) on the HITS, PTB, and ESR datasets. The experiment results are presented in table 3. We also give the number of parameters of each model, as well as the number of multiplication and additions (mult-adds) done per model and per sample (in billions, G).

First, if we compare the single-feature 1D CNN-transformer (a) and 2D CNN (b) models with our proposed model we can see that, for all datasets, our regularized multifeature model (e) outperforms the single-feature models. Moreover, the variability of the results is reduced by at least 0.32% in terms of MCC, hence giving more stable models, except for the weighted sum regularized model in the HITS dataset.

Second, if we compare our proposed models with other state-of-the-art models [5, 11, 13, 14], we achieve state-of-the-art results on both the HITS and PTB datasets. On the HITS dataset, the best performing model is the proposed regularized multifeature model, outperforming the other models by a margin greater than 3.61% in terms of MCC. In particular, the regularized model using concatenation (intermediate fusion) also reduces the variability by a up to 8.35%. Moreover, on the PTB dataset, this same model outperforms the other models,

[8] For a detailed description of this cluster, we the reader to http://www.idris.fr/jean-zay/jean-zay-presentation.html

[9] Username gdec-submission and password $Gjaq * \& * K7vq44azu$

[10] Mail: gdec.submission@gmail.com, Password: 1$\#tU6mKAXqGT8\#CY$

Table 3: Experiment 1. Test classification performance of our proposed model and other state-of-the-art models on three medical datasets: HITS, PTB, and ESR. The results confirm the appeal and adaptability of our method as it can outperform the single-feature models, 1D CNN-trans. and 2D CNN, on the three datasets. The proposed method achieves state-of-the-art performance on two datasets, the HITS and PTB datasets, and excellent performance on the ESR dataset. The number of multiplications and additions (mult-adds) is given in billions (G).

Dataset	**Model**	**Modality**	**Fusion method**	**MCC**	**F1-Score**	**Accuracy**	**No. Parameters**	**No. mult-adds (G)**
HITS	1D CNN-trans.	Raw signal	-	79.17 ± 6.64	84.37 ± 6.62	85.61 ± 4.74	766 271	0.173
	2D CNN	TFR		87.09 ± 4.31	90.98 ± 2.95	91.29 ± 2.96	1 681 923	1.23
	Late Fusion [5]	Both	Weight. Sum	84.66 ± 10.99	88.67 ± 9.32	89.14 ± 8.35	27 073 416	19.87
	Late Fusion (ours)			87.94 ± 2.60	91.44 ± 1.91	91.80 ± 1.83	2 448 072	1.40
	Ours (No Reg.)		Cat	84.53 ± 1.58	89.61 ± 1.06	89.71 ± 1.02	4 833 727	1.40
	Ours (No Reg.)		Weight. Sum	85.93 ± 1.21	90.56 ± 0.78	90.58 ± 0.82	4 876 233	1.40
	Ours (Reg.)		Cat	91.89 ± 2.64	94.31 ± 1.66	94.53 ± 1.74	4 833 727	1.40
	Ours (Reg.)		Weight. Sum	88.28 ± 6.91	91.69 ± 4.81	92.01 ± 4.73	4 876 233	1.40
PTB	1D CNN-trans.	Raw signal	-	98.31 ± 0.43	99.16 ± 0.22	99.32 ± 0.17	765 876	0.026
	2D CNN	TFR		97.03 ± 1.22	98.51 ± 0.61	98.80 ± 0.50	1 555 842	0.063
	[11]	GAF MTF RP	Weight. Sum	-	98	99.2	9 259 427	-
	Late Fusion [5]	Both	Weight. Sum	99.29 ± 0.21	99.65 ± 0.10	99.71 ± 0.08	1 156 732	0.119
	Late Fusion (ours)			98.45 ± 0.49	99.22 ± 0.25	99.38 ± 0.20	2 321 594	0.089
	Ours (No Reg.)		Cat	97.11 ± 0.43	98.6 ± 0.22	98.84 ± 0.18	2 128 820	0.236
	Ours (No Reg.)		Weight. Sum	97.29 ± 0.50	98.64 ± 0.25	98.91 ± 0.20	2 130 366	0.236
	Ours (Reg.)		Cat	**99.28 ± 0.11**	**99.64 ± 0.05**	**99.71 ± 0.04**	2 128 820	0.236
	Ours (Reg.)		Weight. Sum	99.18 ± 0.25	99.59 ± 0.13	99.67 ± 0.10	2 130 366	0.236
ESR	1D CNN-trans.	Raw signal	-	95.14 ± 1.67	97.55 ± 0.87	98.40 ± 0.59	109 942	0.008
	2D CNN	TFR		92.81 ± 3.53	96.33 ± 1.88	97.59 ± 1.35	1 555 842	0.062
	[13]	Raw signal		**99.09**	**98.89**	98.67	-	-
	[14]			–	98.59	**99.39**	-	-
	Late Fusion (ours)	Both	Weight. Sum	97.45 ± 1.49	98.71 ± 0.77	99.16 ± 0.51	1 665 724	0.070
	Ours (No Reg.)		Cat	93.40 ± 1.32	96.67 ± 0.68	97.89 ± 0.45	1 801 590	0.123
	Ours (No Reg.)		Weight. Sum	93.01 ± 2.22	96.45 ± 1.22	97.77 ± 0.69	1 803 456	0.123
	Ours (Reg.)		Cat	96.51 ± 0.46	98.25 ± 0.23	98.88 ± 0.15	1 801 590	0.123
	Ours (Reg.)		Weight. Sum	96.85 ± 0.70	98.42 ± 0.35	98.98 ± 0.23	1 803 456	0.123

and in particular the model of [11], which uses three features as input whereas only two are required in our model. Additionally, the model achieves the same performance as our previously published model [5] but it reduces the variability by half, giving a more stable model. However, on the ESR data our proposed regularized model is not able to outperform the state-of-the-art models. Indeed, the best performing model is that of [13], outperforming our regularized models by a margin of 2.24% and 0.47% in terms of MCC and F1 score, respectively. However, despite not being specifically designed for this particular ESR dataset, our proposed model still achieves excellent (96.85 ± 0.70 of MCC) performance that is close to that of the best performing model (99.09 of MCC).

4.3.2. Experiment 2: Influence of guided training

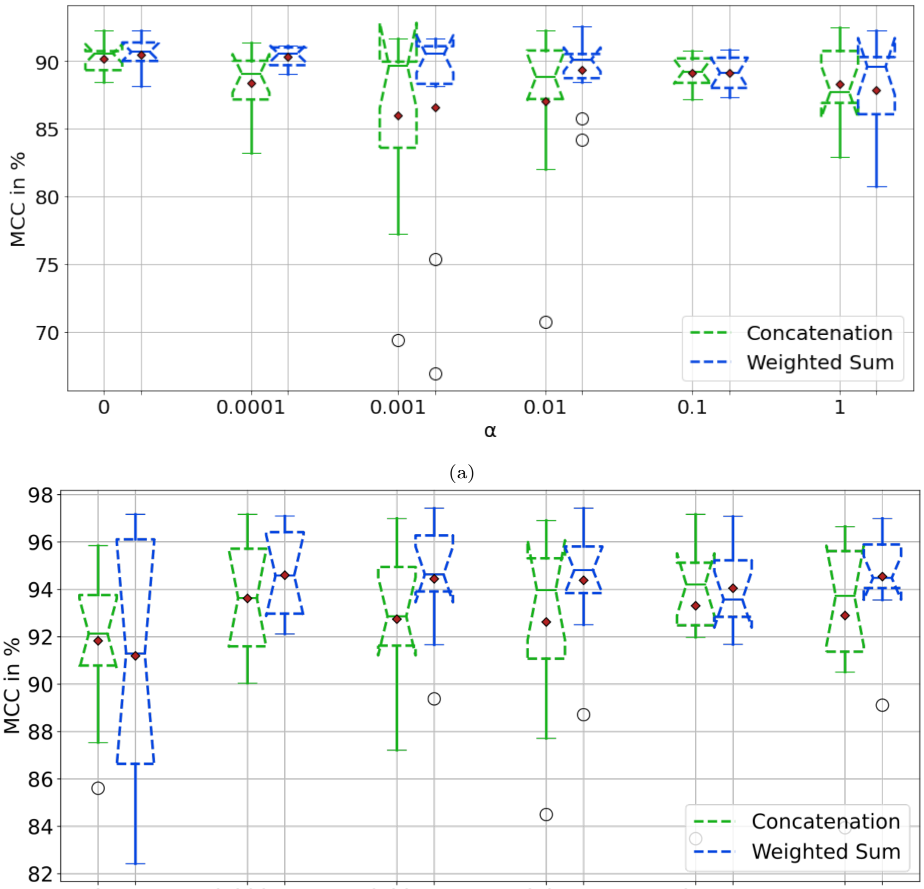
The objective of this experiment is to study the importance of guided training (see α and β in equation 5). To this end, we trained different models without

DEC ($\gamma = 0$) and with the guided training applied at different places: (1) in the latent space of the TFR only ($\alpha > 0$ and $\beta = 0$), (2) in the latent space of the raw signal only ($\alpha = 0$ and $\beta > 0$), (3) and in both latent spaces ($\alpha > 0$ and $\beta > 0$). Moreover, we varied the importance of each regularization term in the range: $\{1, 0.1, 0.01, 0.001, 0.0001\}$. The first two regularization sub-experiments, (1) and (2), were trained on the HITS and PTB datasets, whereas the third one (3) was trained only on the HITS dataset due to resources and energy limitations. The results of (1) and (2) are presented in figures 3 and 4, and the results of (3) are shown in figure 5.

First, from figure 3 we can observe that, for both datasets, applying guided training on the latent space of the TFR encoder yields similar results to those of the unguided model. Indeed, for the HITS dataset and the weighted sum fusion strategy, $\alpha = 1e^{-4}$ achieves an MCC of 90.28 ± 0.77 compared to 90.49 ± 1.21 for the unguided model. For the PTB dataset and the concatenation fusion strategy, $\alpha = 0.01$ achieves 93.31 ± 3.65 MCC versus 91.83 ± 3.13 for the unguided model, which represents a bigger gap than with the HITS dataset.

Second, from figure 4 we can see that, for both datasets, guiding the training of the 1D CNN-transformer encoder has a beneficial effect on the classification performance of the model. Indeed, this guiding makes it possible to outperform the unguided model by a margin greater than 0.84% for the HITS dataset and 6.68% for the PTB dataset. In particular, for both datasets, globally, the importance of $\mathcal{L}_{TE}$ (value of β) is not crucial, as different values achieve similar results.

Finally, using figure 5, we note that globally, when $\alpha \leq \beta$, the performance of the models increases, achieving better performance than the unguided model. Furthermore, we can see that the previous results are still valid when both losses, $\mathcal{L}_{TFR}$ and $\mathcal{L}_{TE}$, are applied. Indeed, for a fixed value of α (importance of $\mathcal{L}_{TFR}$), we observe that increasing the value of β (importance of $\mathcal{L}_{TE}$) also increases the classification performance of the trained models. Moreover, for a fixed value of β, the classification performance of the different models is relatively stable with respect to α, especially for large values of β.



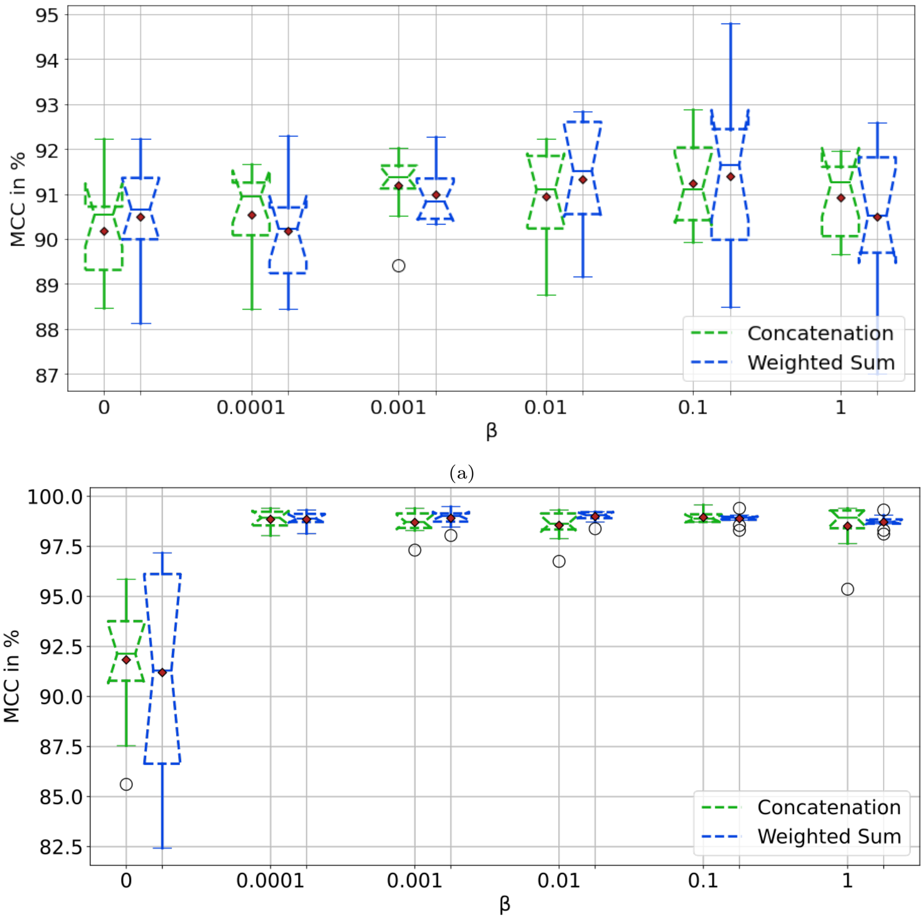
(a)

(b)

Figure 3: Experiment 2. Validation classification performance (MCC) of two end-to-end trained multifeature models using guided training only on the 2D CNN encoder space, without DEC (i.e, $\alpha > 0$, $\beta = 0$, $\gamma = 0$). (a) HITS dataset, (b) PTB dataset. α corresponds to the importance of $\mathcal{L}_{TFR}$. Globally, guiding the training of the 2D CNN encoder does not improve the classification performance of the model considerably with respect to the unregularized model.

405 *4.3.3. Experiment 3: Influence of DEC regularization*

The objective of this experiment is to study the importance of DEC in the clinical dataset in the application of interest in our project (HITS). To this end, we trained different versions of model (e) without the guided training ($\alpha = \beta = 0$) and with DEC ($\gamma > 0$) on the HITS and PTB datasets. Moreover, we varied



(a)

(b)

Figure 4: Experiment 2. Validation classification performance (MCC) of two end-to-end trained multifeature models using guided training only on the 1D CNN-transformer encoder space, without DEC (i.e., $\alpha = 0$, $\beta > 0$, $\gamma = 0$). (a) HITS dataset, (b) PTB dataset. β corresponds to the importance of $\mathcal{L}_{TE}$. We observe that guiding the training of the 1D CNN-transformer encoder can considerably increase the classification performance of the model with respect to the unregularized model, especially in the PTB dataset (imbalanced dataset).

the importance of the DEC loss, γ, in the range $\{1.0, 0.1, 0.01, 0.001, 0.0001\}$. The results are illustrated in figure 6.

First, we can see that the results for the HITS dataset are consistent with the results in [49]. Indeed, the HITS dataset is a balanced dataset, and thus the DEC alone achieves similar results to the unregularized method. However, we can see that both fusion strategies, concatenation and weighted sum, allow us to

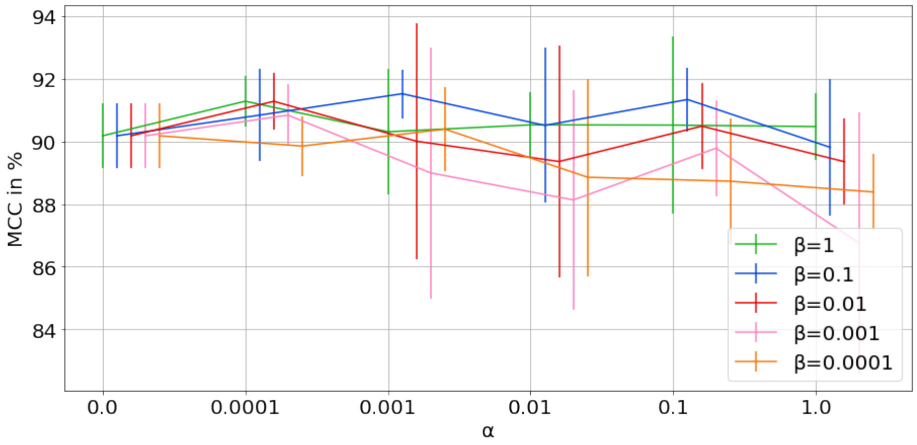
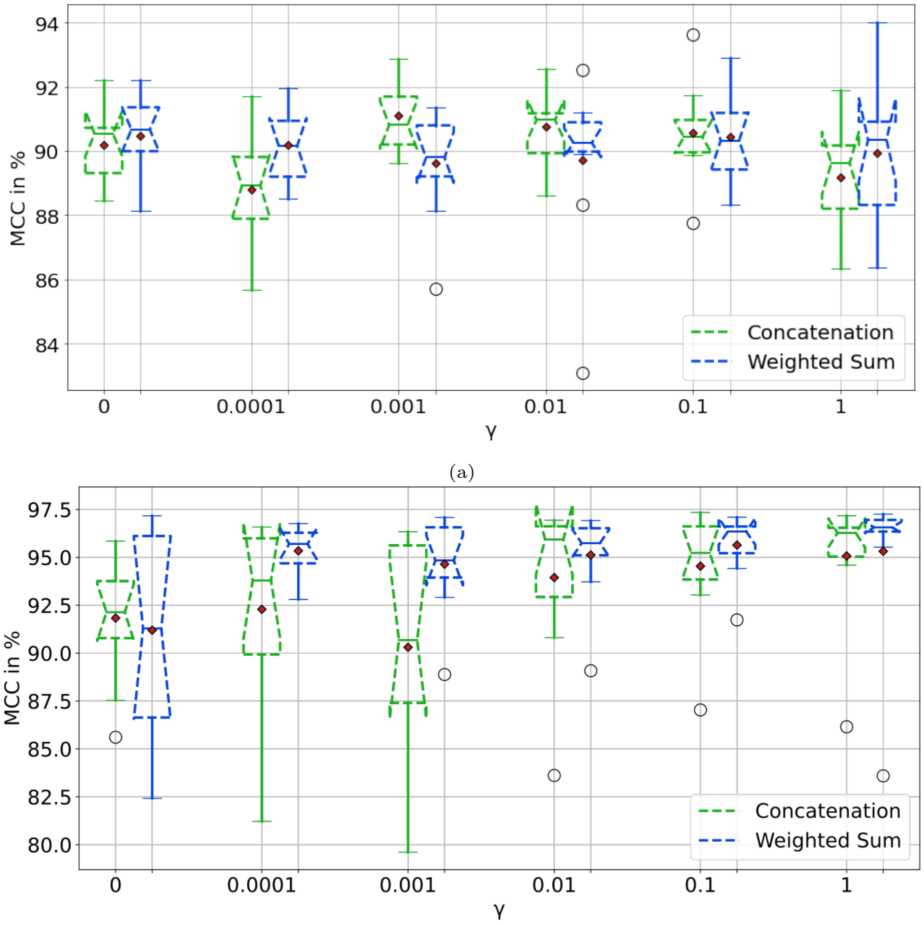


Figure 5: Experiment 2. Guided training on the two latent spaces (2D CNN TFR encoder and 1D CNN-transformer raw signal encoder) for the model using concatenation as intermediate fusion strategy on the HITS dataset. α and β correspond to the importance of $\mathcal{L}_{TFR}$ and $\mathcal{L}_{TE}$, respectively. We see that the guiding of the 1D CNN-transformer encoder is more important than that of the 2D CNN encoder. Indeed, for a fixed α, when β decreases, the MCC tends to decrease, whereas for a fixed β, when α decreases, the MCC remains relatively stable.

achieve a similar or better classification performance when DEC regularization is applied, with 91.09 ± 1.09 ($\gamma = 0.001$) and 90.45 ± 1.43 ($\gamma = 0.1$) MCC versus 90.18 ± 1.03 and 90.49 ± 1.21 MCC for the concatenation and weighted sum fusion strategies, respectively.

Second, we observe that the results for the PTB dataset are also consistent with the results in [49]. Indeed, as the PTB dataset is an imbalanced dataset, DEC allows us to achieve better results than the unregularized models. When we increase γ (importance of DEC), the classification performance also increases and the variability decreases. By the same token, the best performing models are the ones using DEC regularization with 95.07 ± 3.07 and 95.64 ± 1.52 MCC versus 91.83 ± 3.12 and 91.19 ± 5.09 for the unregularized concatenation and weighted sum strategies models, respectively.



(a)

(b)

Figure 6: Experiment 3. Validation classification performance (MCC) of two end-to-end trained multifeature models using DEC on the common fused space, without guided training (i.e., $\alpha = \beta = 0$ and $\gamma > 0$). (a) HITS dataset. (b) PTB dataset. γ corresponds to the importance of $\tilde{\mathcal{L}}_{DEC}$. The results are consistent with [49] as DEC can improve the classification performance on both datasets, especially on the PTB dataset (imbalanced).

4.4. Discussion

4.4.1. Experiment 1: Advantage of end-to-end training

The results of this experiment prove the effectiveness of our proposed regularized multifeature end-to-end trained model for medical signal classification. Our proposed regularization makes it possible to have more clustered latent spaces, especially for the latent space of the raw signal encoder (guided training) and the fused latent space (DEC). To show this, we project the embedded representations of the different latent spaces on a 2D plane using Uniform Manifold Approximation and Projection (UMAP)[11] [52]. Figure 7 shows the 2D projections of the different latent spaces on the HITS dataset. As the results showed, the difference in TFR latent spaces between the regularized and unregularized models is not evident (silhouette scores of 0.21 ± 0.05 and 0.19 ± 0.04, respectively, on the HITS dataset). However, if we focus on the raw signal encoder latent space and on the fused latent space, the difference is more striking. Indeed, for the raw signal encoder latent space, the regularized model has a more separable and clustered structure, which is also shown by the silhouette score (0.38 ± 0.04 for the regularized model versus -0.03 ± 0.01 for the unregularized one, on the HITS dataset). Moreover, on the fused latent space, we do not observe an important difference between the regularized and unregularized models, with silhouette scores on the HITS dataset of 0.36 ± 0.11 and 0.39 ± 0.05, respectively[12]. A similar behavior is observed on the PTB dataset (see supplementary materials).

What is more, our approach is able to perform similarly or outperform other state-of-the-art methods because it takes advantage of the complementarity of different representations thanks to joint training. Indeed, the two used representations (raw signal and TFR) are complementary commonly used in signal processing (and are not specific to a type of medical signal). Using joint training with these two complementary features allows each encoder to compensate for the weaknesses of the other (similar behavior was studied in [53]). However, as the results in table 3 show it, joint training is not enough, if the individual features or latent spaces are not discriminative enough.

Furthermore, we observe that, on the three datasets, one of the single-feature models is able to outperform the unregularized end-to-end trained multifeature

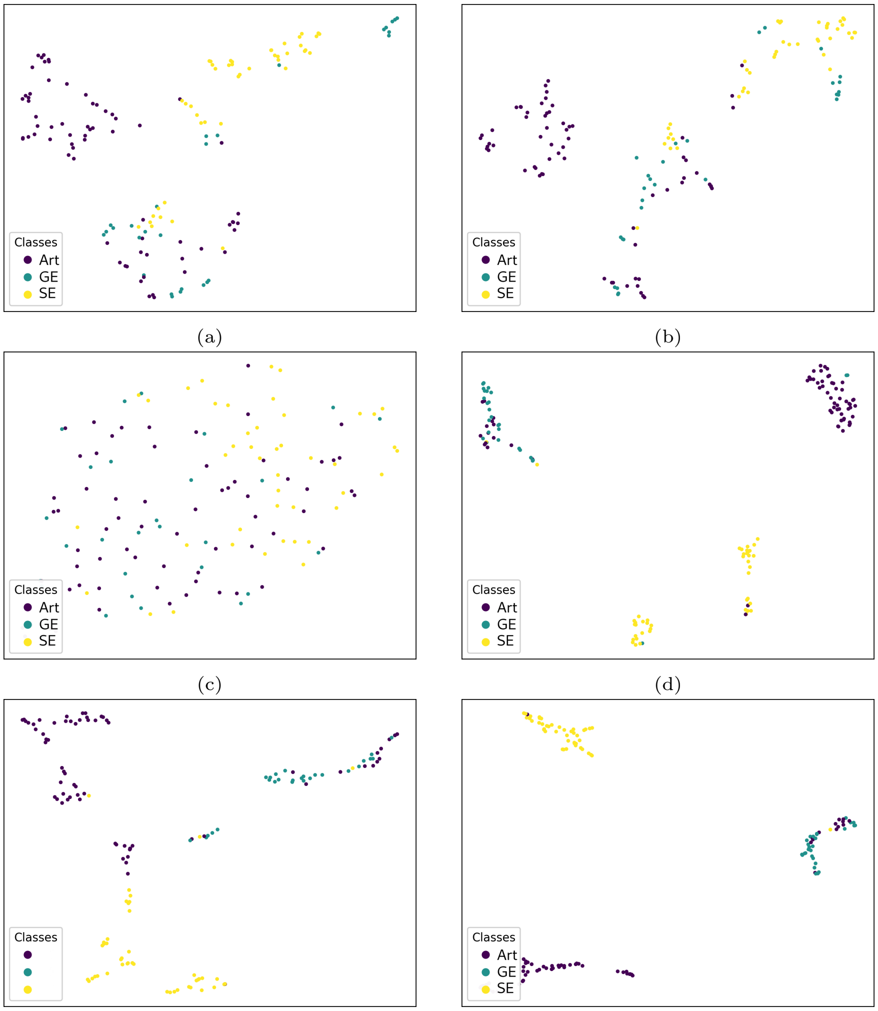
[11] We used the default parameters of the *umap-learn* library, except $n_{neighbors}$, which was fixed to 5

[12] Although, on the validation samples, the regularized model seems to give a more clustered fused latent space, with a silhouette score of 0.75 ± 0.06 against 0.71 ± 0.05 for the unregularized model (see supplementary materials).

models. Indeed, training the multifeature models with the two input features makes the latent spaces of each single-feature encoder harder to learn, and thus penalizing the final classification performance, hence the importance of guiding the training and using semi-supervised DEC.

By the same token, on the ESR data, our proposed regularized end-to-end trained model was not able to outperform the state-of-the-art models. Several factors can explain this. First, the comparison is not easy to make as [13, 14] do not give the standard deviation of their evaluation metrics, and thus we are not able to measure how far our results are from their results. Additionally, the evaluation strategy is not the same between works: some authors use a certain train/test split, and others use cross-validation. Another key point is that, in this ESR dataset, it is very difficult to perform a subject-wise train/test split or subject-wise cross-validation as not all subjects have samples for all the classes, and even those having samples for the same classes do not have the same quantity. This can lead to overconfident evaluation metrics, as samples from the same subject can be both in the train and in the test sets. In addition, the models of [13, 14] were specifically designed for ESR and optimized for the ESR UCI dataset, whereas our model was designed for emboli classification on TCD data, and tested on other medical signal classification tasks. Even so, contrary to the late fusion models, our approach is easier to train as joint training is used, and all the encoders and classification models are trained simultaneously, whereas in the late fusion models, optimization has to be done separately.

Finally, if we compare the number of parameters and number of mult-adds (in billions, G) of the best performing models on the HITS dataset, we note some interesting results. First, on the HITS dataset, our regularized end-to-end trained model has a total of 4 833 727 to 4 876 233 parameters versus 27 073 416 parameters for the model in [5]. The former models are able to outperform the latter model by over 4.3% in terms of MCC while reducing its variability by 2.07%, and this with 5.6 fewer parameters. The same behavior is observed for the number of mult-adds, as our proposed model reduces the number of operations by a factor of 14. However, this is not observed in the PTB or



(a)

(b)

(c)

(d)

(e)

(f)

Figure 7: Experiment 1. Test embeddings of the regularized and unregularized models on the HITS dataset. (a) 2D CNN encoder TFR latent space without regularization, (b) 2D CNN encoder TFR latent space with regularization, (c) 1D CNN-transformer raw signal encoder latent space without regularization, (d) 1D CNN-transformer raw signal encoder latent space with regularization, (e) fused common latent space without regularization, (f) fused common latent space with regularization. For the regularized model we used $\alpha = 0.01$, $\beta = 0.1$, $\gamma = 0.01$, and $e_{init} = 50$.

ESR datasets, since our proposed models have between 1 801 590 and 2 130 366 parameters versus 1 156 732 and 2 321 594 parameters for the late fusion models. The same trend is observed for the number of mult-adds.

4.4.2. Experiment 2: Influence of guided training

The results of this experiment revealed the influence of guided training on the classification performance of the trained models on two datasets, HITS and PTB.

First, the results show the importance of guiding the training, especially for the raw signal encoder. Indeed, training both encoders (the 2D CNN TFR encoder and the 1D CNN-transformer raw signal encoder) jointly makes the training more difficult. This can be seen by the performance of the unregularized models on the three datasets and by the 2D projections of the latent spaces of the same models (figure 7). Unregularized models are not able to learn discriminative features from the raw signal, but guiding the training of these features increases their discriminative power.

Second, this experiment also gives guidelines on how to choose the values of α and β. Indeed, we see that for $\alpha > \beta$, the classification performance of the guided trained models tends to decrease. In fact, giving too much importance to the TFR can reduce performance even in cases where the single-feature TFR models have a lower performance than the single-feature raw signal models. Thus, we recommend users choose α and β such that $\alpha \leq \beta$, since the guiding of the raw signal encoder is more important than that of the TFR in a joint training context.

Finally, the results showed that the guiding of the 2D CNN TFR model does not always have a significant (positive or negative) impact on the classification performance of the models. However, the guiding of the 1D CNN-transformer raw signal model can have a significant positive impact on the classification performance of the models. Thus, keeping both guiding strategies during training can be mostly beneficial. Moreover, keeping both guiding strategies makes it possible to have more robust models against missing features, since one can deactivate one encoder and still carry out classification. This is indirectly confirmed by the learned latent spaces, where for the different datasets we obtain latent spaces with test silhouette scores higher than 0.2.

4.4.3. Experiment 3: Influence of DEC regularization

The results of this experiment were coherent with the results obtained in the synthetic dataset in [49].

First, this experiment showed that DEC regularization can help to deal with a real-life imbalanced dataset, improving its generalization capability.

Second, this experiment revealed that DEC regularization alone is able to improve the classification performance of a multifeature model on balanced and imbalanced datasets: by reducing the variability, it produces more stable models. Thus, with a good choice of hyperparameters, one is able to improve the classification performance of the models.

Finally, the results showed that in the worst-case scenario, DEC regularization does not degrade considerably the classification performance of the trained models, especially for imbalanced datasets. This means that, with our proposed method, adding DEC regularization with guided training is not problematic because even when the hyperparameters are not precisely optimized, the final models can benefit from the whole regularization strategy.

4.4.4. Limitations

Although our proposed method enables end-to-end training of a multifeature model, achieving great classification performance on several datasets, some limitations can be highlighted.

First, our method has several hyperparameters that need to be optimized (α, β, γ, and e_{init}), which makes the training more difficult. We studied the influence of these hyperparameters on the classification performance of the models and [49] give some guidelines on how to select them, but more extensive experiments should be carried out to validate the generality of our method.

Second, we only evaluated our proposed method on two type of features and one type of model architecture per feature (2D CNN for the TFR and 1D CNN-Transformer for the raw signal).

Finally, we focused on medical datasets, which is our main interest, but more extensive studies can be performed on more datasets of different nature (environmental sound recognition, voice recognition, music classification, etc.).

5. Conclusion and future work

In this work, we presented a regularized end-to-end guided trained classification model for medical signals, exploiting both the TFR and the raw signal through intermediate fusion. The method guides the training of the encoder of each input representation through two iterated losses, and regularizes the fused joint common space through deep embedded clustering. Extensive experiments and ablation studies show the generalizability of our proposed method to different medical signal classification tasks, achieving state-of-the-art results on two of the three datasets tested, without the need for designing a distinct model with specific inputs for each dataset.

As future work, we plan on improving the selection of the hyperparameters (α, β, γ, and e_{init}) using Bayesian optimization. Moreover, we aim to study the generalizability of our method to a nonmedical context, using nonmedical datasets as well as other input features and models/architectures. Finally, as DEC regularization does not depend on the labels of the input samples, we intend to combine it with robust loss functions to partially handle noisy-labeled datasets.

Acknowledgments

This work was carried out in the context of the CAREMB project funded by the Auvergne-Rhône-Alpes region, within the *Pack Ambition Recherche* program. This work was performed within the framework of the LABEX CELYA (ANR-10-LABX-0060) and PRIMES (ANR-11-LABX-0063) of Universite de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR). This work was performed using HPC resources from GENCI-IDRIS (Grant 2022-AD011013616)

References

[1] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, T. Sainath, Deep learning for audio signal processing, IEEE Journal of Selected Topics in Signal Processing 13 (2019) 206–219.

[2] W. Rawat, Z. Wang, Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review, Neural Computation 29 (2017) 2352–2449.

[3] A. Natarajan, Y. Chang, S. Mariani, A. Rahman, G. Boverman, S. Vij, J. Rubin, A wide and deep transformer neural network for 12-lead ecg classification, in: 2020 Computing in Cardiology, 2020, pp. 1–4.

[4] Y. Gong, Y.-A. Chung, J. Glass, AST: Audio Spectrogram Transformer, in: Proc. Interspeech 2021, 2021, pp. 571–575.

[5] Y. Vindas, B. K. Guépié, M. Almar, E. Roux, P. Delachartre, An hybrid cnn-transformer model based on multi-feature extraction and attention fusion mechanism for cerebral emboli classification, in: Proceedings of the 7th Machine Learning for Healthcare Conference, Proceedings of Machine Learning Research, PMLR, 2022.

[6] W. H. Organization, The top 10 causes of death, 2020.

[7] B. K. Guepie, M. Martin, V. Lacrosaz, M. Almar, B. Guibert, P. Delachartre, Sequential emboli detection from ultrasound outpatient data, IEEE Journal of Biomedical and Health Informatics 23 (2019) 334–341. Number: 1.

[8] P. Sombune, P. Phienphanich, S. Phuechpanpaisal, S. Muengtaweepongsa, A. Ruamthanthong, C. Tantibundhit, Automated embolic signal detection using deep convolutional neural network, in: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 2017, pp. 3365–3368.

[9] M. Wasimuddin, K. Elleithy, A.-S. Abuzneid, M. Faezipour, O. Abuzaghleh, Stages-based ecg signal analysis from traditional signal processing to

machine learning approaches: A survey, IEEE Access 8 (2020) 177782–177803.

[10] C. Che, P. Zhang, M. Zhu, Y. Qu, B. Jin, Constrained transformer network for ecg signal processing and arrhythmia classification, BMC Medical Informatics and Decision Making 21 (2021).

[11] Z. Ahmad, A. Tabassum, L. Guan, N. M. Khan, Ecg heartbeat classification using multimodal fusion, IEEE Access (2021).

[12] Y. Vindas, B. K. Guépié, M. Almar, E. Roux, P. Delachartre, Semi-automatic data annotation based on feature-space projection and local quality metrics: an application to cerebral emboli characterization, Medical Image Analysis (2022) 102437.

[13] A. M. Hilal, A. A. Albraikan, S. Dhahbi, M. K. Nour, A. Mohamed, A. Motwakel, A. S. Zamani, M. Rizwanullah, Intelligent epileptic seizure detection and classification model using optimal deep canonical sparse autoencoder, Biology 11 (2022).

[14] G. Xu, T. Ren, Y. Chen, W. Che, A one-dimensional cnn-lstm model for epileptic seizure recognition using eeg signal analysis, Frontiers in Neuroscience 14 (2020).

[15] H. Nishizaki, K. Makino, Signal classification using deep learning, 2019, pp. 1–4.

[16] J. Lee, T. Kim, J. Park, J. Nam, Raw waveform-based audio classification using sample-level cnn architectures, ArXiv abs/1712.00866 (2017).

[17] M. Scarpiniti, F. Colasante, S. Tanna, M. Ciancia, Y. Lee, A. Uncini, Deep belief network based audio classification for construction sites monitoring, Expert Systems with Applications 177 (2021) 114839.

[18] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. Yalta, R. Yamamoto, X. fei Wang, S. Watanabe, T. Yoshimura, W. Zhang, A comparative study on transformer vs rnn in speech applications, 2019, pp. 449–456.

[19] J. Jin, S. Yang, B. Zhao, L. Luo, W. L. Woo, Attention-block deep learn-

ing based features fusion in wearable social sensor for mental wellbeing evaluations, IEEE Access 8 (2020) 1–1.

[20] A. Tjandra, C. Liu, F. Zhang, X. Zhang, Y. Wang, G. Synnaeve, S. Nakamura, G. Zweig, Deja-vu: Double feature presentation and iterated loss in deep transformer networks, in: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 6899–6903.

[21] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML'16, JMLR.org, 2016, p. 478–487.

[22] J. Pu, Y. Panagakis, M. Pantic, Learning separable time-frequency filterbanks for audio classification, in: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 3000–3004.

[23] R. Sharan, H. Xiong, S. Berkovsky, Benchmarking audio signal representation techniques for classification with convolutional neural networks, Sensors 21 (2021) 3434.

[24] M. Okawa, T. Saito, N. Sawada, H. Nishizaki, Audio classification of bit-representation waveform, in: INTERSPEECH, 2019, pp. 2553–2557.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017.

[26] H. Akbari, L. Yuan, R. Qian, W.-H. Chuang, S.-F. Chang, Y. Cui, B. Gong, Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text, Advances in Neural Information Processing Systems 34 (2021).

[27] T. Baltrusaitis, C. Ahuja, L.-P. Morency, Multimodal machine learning: A survey and taxonomy, IEEE Transactions on Pattern Analysis and Machine

Intelligence PP (2017).

[28] J. D. S. Ortega, M. Senoussaoui, E. Granger, M. Pedersoli, P. Cardinal, A. L. Koerich, Multimodal fusion with deep neural networks for audio-video emotion recognition, ArXiv abs/1907.03196 (2019).

[29] S. Mao, Y. Li, Y. Ma, B. Zhang, J. Zhou, K. Wang, Automatic cucumber recognition algorithm for harvesting robots in the natural environment using deep learning and multi-feature fusion, Computers and Electronics in Agriculture 170 (2020).

[30] Y. Zhu, Y. Jiang, Optimization of face recognition algorithm based on deep learning multi feature fusion driven by big data, Image and Vision Computing 104 (2020) 104023.

[31] J.-G. Kim, B. Lee, Appliance classification by power signal analysis based on multi-feature combination multi-layer lstm, Energies 12 (2019).

[32] X. Feng, Q. Feng, S. Li, X. Hou, S. Liu, A deep-learning-based oil-well-testing stage interpretation model integrating multi-feature extraction methods, Energies 13 (2020).

[33] X. Chen, Z. Cheng, S. Wang, G. Lu, G. Xv, Q. Liu, X. Zhu, Atrial fibrillation detection based on multi-feature extraction and convolutional neural network for processing ecg signals, Computer Methods and Programs in Biomedicine 202 (2021) 106009.

[34] J. Kukacka, V. Golkov, D. Cremers, Regularization for deep learning: A taxonomy, ArXiv abs/1710.10686 (2017).

[35] C. M. Bishop, Training with noise is equivalent to tikhonov regularization, Neural Computation 7 (1995) 108–116.

[36] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10, Omnipress, Madison, WI, USA, 2010, p. 807–814.

[37] T. Miyato, S.-i. Maeda, M. Koyama, S. Ishii, Virtual adversarial training: A regularization method for supervised and semi-supervised learning, IEEE Transactions on Pattern Analysis and Machine Intelligence PP (2017).

[38] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, CoRR abs/1207.0580 (2012).

[39] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: F. Bach, D. Blei (Eds.), Proceedings of the 32nd International Conference on Machine Learning, volume 37 of *Proceedings of Machine Learning Research*, PMLR, Lille, France, 2015, pp. 448–456.

[40] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, CoRR abs/1511.07122 (2016).

[41] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Los Alamitos, CA, USA, 2015, pp. 3431–3440.

[42] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[43] D. C. Plaut, S. J. Nowlan, G. E. Hinton, Experiments on learning back propagation, Technical Report CMU–CS–86–126, Carnegie–Mellon University, Pittsburgh, PA, 1986.

[44] S. Rifai, P. Vincent, X. Muller, X. Glorot, Y. Bengio, Contractive auto-encoders: Explicit invariance during feature extraction, in: Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML'11, Omnipress, Madison, WI, USA, 2011, p. 833–840.

[45] S. Rifai, X. Glorot, Y. Bengio, P. Vincent, Adding noise to the input of a model trained with a regularized objective, ArXiv abs/1104.3250 (2011).

[46] C. Lyu, K. Huang, H.-N. Liang, A unified gradient regularization family for adversarial examples, 2015 IEEE International Conference on Data Mining (2015) 301–309.

[47] R. Collobert, S. Bengio, Links between perceptrons, mlps and svms, in: Proceedings of the Twenty-First International Conference on Machine

Learning, ICML '04, Association for Computing Machinery, New York, NY, USA, 2004, p. 23.

[48] M. Sajjadi, M. Javanmardi, T. Tasdizen, Regularization with stochastic transformations and perturbations for deep semi-supervised learning, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, Curran Associates Inc., Red Hook, NY, USA, 2016, p. 1171–1179.

[49] Y. Vindas, E. Roux, B. K. Guépié, M. Almar, P. Delachartre, Deep embedded clustering regularization for supervised imbalanced cerebral emboli classification using transcranial doppler ultrasound, . Submitted to the European Signal Processing Conference (EUSIPCO) 2023.

[50] M. Kachuee, S. Fazeli, M. Sarrafzadeh, Ecg heartbeat classification: A deep transferable representation, in: 2018 IEEE international conference on healthcare informatics (ICHI), IEEE, 2018, pp. 443–444.

[51] R. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, C. Elger, Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state, Physical review. E, Statistical, nonlinear, and soft matter physics 64 (2002) 061907.

[52] T. Sainburg, L. McInnes, T. Q. Gentner, Parametric umap embeddings for representation and semisupervised learning, Neural Computation 33 (2021) 2881–2907.

[53] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, H. Shah, Wide deep learning for recommender systems, in: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS 2016, Association for Computing Machinery, New York, NY, USA, 2016, p. 7–10. URL: `https://doi.org/10.1145/2988450.2988454`. doi:`10.1145/2988450.2988454`.