Weakly-supervised learning for emboli characterization with Transcranial Doppler (TCD) monitoring

Yamil Vindas Yassine

N°d'ordre NNT : 2023ISAL0064

THESE de DOCTORAT DE L'INSA LYON,
Membre de l'Université de Lyon

Ecole Doctorale N° 160
Électronique, Électrotechnique, Automatique

Spécialité/discipline de doctorat :
Traitement du Signal et de l'Image

Soutenue publiquement le 12/10/2023, par :
Yamil Vindas-Yassine

Weakly-supervised learning for emboli characterization with Transcranial Doppler (TCD) monitoring

---

Devant le jury composé de :

Ruan, Su	Professeure, Université de Rouen-Normandie	Rapporteure
Rousseau, François	Professeur, IMT Atlantique	Rapporteur
Mandic, Danilo	Professor, Imperial College London	Examinateur
Delachartre, Philippe	Professeur, INSA Lyon	Directeur de thèse
Roux, Emmanuel	Maître de conférences, Université Claude Bernard Lyon 1	Co-encadrant de thèse
Guépié, Blaise Kévin	Professeur Assistant, Université de Technologie de Troyes	Co-encadrant de thèse
Almar, Marilys	Ingénieure R&D, Atys Medical	Invitée

INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Département de la Formation par la Recherche et des Études Doctorales (**FEDORA**)

Bâtiment INSA direction, 1er étage
37, av. J. Capelle
69621 Villeurbanne Cédex
fedora@insa-lyon.fr

Référence : TH0993_Yamil VINDAS-YASSINE

L'INSA Lyon a mis en place une procédure de contrôle systématique via un outil de détection de similitudes (logiciel Compilatio). Après le dépôt du manuscrit de thèse, celui-ci est analysé par l'outil. Pour tout taux de similarité supérieur à 10%, le manuscrit est vérifié par l'équipe de FEDORA. Il s'agit notamment d'exclure les auto-citations, à condition qu'elles soient correctement référencées avec citation expresse dans le manuscrit.

Par ce document, il est attesté que ce manuscrit, dans la forme communiquée par la personne doctorante à l'INSA Lyon, satisfait aux exigences de l'Etablissement concernant le taux maximal de similitude admissible.

INSA LYON
Campus LyonTech La Doua
20, avenue Albert Einstein - 69621 Villeurbanne cedex - France
Tél. +33 [0]4 72 43 83 83 - Fax +33 [0]4 72 43 85 00
www.insa-lyon.fr

Département FEDORA – INSA Lyon – Ecoles Doctorales

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	**CHIMIE DE LYON** **https://edchimie-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage secretariat@edchimie-lyon.fr	**M. Stéphane DANIELE** C2P2-CPE LYON-UMR 5265 Bâtiment F308, BP 2077 43 boulevard du 11 Novembre 1918 69616 Villeurbanne directeur@edchimie-lyon.fr
E.E.A.	**ELECTRONIQUE, ELECTROTECHNIQUE, AUTOMATIQUE** **https://edeea.universite-lyon.fr** Sec. : Stéphanie CAUVIN Bâtiment Direction INSA Lyon Tél : 04.72.43.71.70 secretariat.edeea@insa-lyon.fr	**M. Philippe DELACHARTRE** INSA Lyon Laboratoire CREATIS Bâtiment Blaise Pascal, 7 avenue Jean Capelle 69621 Villeurbanne CEDEX Tél : 04.72.43.88.63 philippe.delachartre@insa-lyon.fr
E2M2	**EVOLUTION, ECOSYSTEME, MICROBIOLOGIE, MODELISATION** **http://e2m2.universite-lyon.fr** Sec. : Bénédicte LANZA Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.88.62 secretariat.e2m2@univ-lyon1.fr	**Mme Sandrine CHARLES** Université Claude Bernard Lyon 1 UFR Biosciences Bâtiment Mendel 43 boulevard du 11 Novembre 1918 69622 Villeurbanne CEDEX sandrine.charles@univ-lyon1.fr
EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTE** **http://ediss.universite-lyon.fr** Sec. : Bénédicte LANZA Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.88.62 secretariat.ediss@univ-lyon1.fr	**Mme Sylvie RICARD-BLUM** ICBMS – UMR 5246 CNRS – Université de Lyon Bâtiment Raulin – 2ème étage Nord 43 boulevard du 11 Novembre 1918 69622 Villeurbanne Cedex Tél : 04.72.44.82.32 sylvie.ricard-blum@univ-lyon1.fr
INFOMATHS	**INFORMATIQUE ET MATHEMATIQUES** **http://edinfomaths.universite-lyon.fr** Sec. : Renée EL MALHEM Bât. Blaise Pascal, 3e étage Tél : 04.72.43.80.46 infomaths@univ-lyon1.fr	**M. Hamamache KHEDDOUCI** Université Claude Bernard Lyon 1 Bâtiment Nautibus 43 boulevard du 11 Novembre 1918 69622 Villeurbanne Cedex Tél : 04.72.44.83.69 hamamache.kheddouci@univ-lyon1.fr
Matériaux	**MATERIAUX DE LYON** **http://ed34.universite-lyon.fr** Sec. : Yann DE ORDENANA Tél : 04.72.18.62.44 yann.de-ordenana@ec-lyon.fr	**M. Stéphane BENAYOUN** Ecole Centrale de Lyon, Laboratoire LTDS 36 avenue Guy de Collongue 69134 Ecully Cedex Tél : 04.72.18.64.37 stephane.benayoun@ec-lyon.fr
MEGA	**MECANIQUE, ENERGETIQUE, GENIE CIVIL, ACOUSTIQUE** **http://edmega.universite-lyon.fr** Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bâtiment Direction INSA Lyon mega@insa-lyon.fr	**M. Jocelyn BONJOUR** INSA Lyon Laboratoire CETHIL Bâtiment Sadi-Carnot 9 rue de la Physique 69621 Villeurbanne Cedex jocelyn.bonjour@insa-lyon.fr
ScSo	**ScSo*** **https://edsciencessociales.universite-lyon.fr** Sec. : Mélina FAVETON INSA : J.Y. TOUSSAINT Tél : 04.78.69.77.79 melina.faveton@univ-lyon2.fr	**M. Christian MONTES** Université Lumière Lyon 2 86 rue Pasteur 69365 Lyon Cedex 07 christian.montes@univ-lyon2.fr

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Eventually, doctors will adopt AI and algorithms as their work partners. This leveling of the medical knowledge landscape will ultimately lead to a new premium: to find and train doctors who have the highest level of emotional intelligence.

Eric Topol, Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again

Acknowledgement

I would like to express my deepest appreciation to the members of the jury for accepting to examine my work and take part in the defense, despite their busy schedules and the distance.

This endeavor would not have been possible without my PhD thesis supervisor, Philippe Delachartre, and my two co-supervisors, Emmanuel Roux and Blaise Kévin Guépié, who guided and advised me throughout my doctoral thesis. I would particularly like to thank them for all the revisions of my different works, as well as that of the present manuscript, which have enabled me to greatly improve their quality. I would also like to extend my sincere thanks to Marilys Almar of Atys Medical, who taught me various aspects of manual embolus classification, as well as for her help in the annotation of the transcranial Doppler data used in this work.

Thanks should also go to the different interns who helped me to develop various tools to make it possible to use the models described in this work.

I am also grateful to the various permanent and non-permanent (PhD students and master's students) members of the laboratory, who have contributed to a pleasant working atmosphere. In particular, I would like to thank the team directors who created a stimulating research atmosphere, making it easier for me to learn new techniques useful in my research and future career.

I would like to acknowledge Auvergne-Rhône-Alpes region who funded my research in the context of the CAREMB project, 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-0 0 07) operated by the French National Research Agency (ANR).

I could not have undertaken this journey without my family and friends. Particularly, I want to thank my parents, my siblings, and my partner, who supported me these past three years. Without them, this work would not have been possible.

Contents

List of Figures

List of Tables

Introduction

Context and objective

According to the World Health Organization (WHO), stroke is one of the most common neurological disorders (Organization, 2006; Feigin, Nichols, Alam, Bannick, Beghi, Blake, Culpepper, Dorsey, Elbaz, Ellenbogen, Fisher, Fitzmaurice, Giussani, Glennie, James, Johnson, Kassebaum, Logroscino, Marin and Vos, 2019) and one of the leading causes of death and disability in the world (Johnson, Onuma, Owolabi and Sachdev, 2016). The most common type of stroke is ischemic (Meschia and Brott, 2018), which is caused when a foreign body, blocks a cerebral artery. Some of these extraneous objects can be cerebral emboli (Meyer, Mutamatsu and Shirai, 1996; Rosenkranz, Fiehler, Niesen, Waiblinger, Eckert, Wittkugel, Kucinski, Röther, Zeumer, Weiller and Sliwka, 2006), which can be of two natures, solid or gaseous. Therefore, to ensure better patient management and help clinicians prevent stroke, accurate and early detection of foreign bodies is necessary.

Moreover, cerebral emboli can be generated by different sources, such as atheromatous plaques (Rosenkranz *et al.*., 2006), or the fragments generated by some surgical procedures such as transcatheter aortic valve implantation (TAVI) (Aggarwal, Delahunty RN, Menezes, Perry, Wong, Reinthaler, Ozkor and Mullen, 2018). They can also be spawn by other medical procedures such as patent foramen ovale (PFO) tests (Serena, Jimenez-Nieto, Silva and Castellanos, 2010), or cerebral angiography (Markus, Israel, Brown, Loh, Buckenham and Clifton, 1993a). In addition, cerebral emboli can occur randomly (Georgiadis, Grosset, Kelman, Faichney and Lees, 1994; Spencer, 1992), therefore, to be able to detect them in such heterogeneous situations, a flexible long-duration monitoring of the cerebral arteries is crucial. This can be achieved with a transcranial Doppler (TCD) ultrasound device, which has the advantage of being relatively cheap and non-invasive compared to other techniques, such as magnetic resonance imaging (MRI) or computed tomography (CT). Additionally, to increase the duration of the patient monitoring, a portable TCD device such as the TCD-X Atys Medical Holter can be used without being overly obtrusive to the patient. The objective of this thesis is to establish the extent to which these devices can be used to detect emboli. The main idea is to use an ultrasound probe to measure the cerebral blood flow, mainly using the middle cerebral artery (MCA). Then, potential emboli are detected through high intensity transient signals (HITS), which correspond to over intensities in the acquired signal with respect to the intensity of the cerebral blood flow (Ringelstein, Droste, Babikian, Evans, Grosset, Kaps, Markus, Russell and Siebler, 1998).

The work was conducted in collaboration with Atys Medical thanks to the regional project CAREMB (funded by the Auvergne-Rhône-Alpes region) in order to enhance cerebral emboli classification, adapted to industrial constraints (which will be detailed in chapter 1). Therefore, in this thesis, I shall to focus on semi-supervised learning methods with the aim to enhance the classification of cerebral emboli right after HITS detection, in order to help clinicians in both their research and patient management.

Main contributions

In this thesis, we present three primary contributions. First, we introduce an innovative semi-automatic data annotation method designed for the efficient and accurate annotation of data with controlled annotation error. While our primary focus is on cerebral emboli detection using TCD devices, our method is versatile and applicable to various other types of data.

Secondly, we propose a novel multi-feature learning approach to enhance the precision of medical signal classification. Our method leverages diverse signal representations to capitalize on their complementary nature. Furthermore, our approach demonstrates relative robustness against label noise in samples, making it compatible with our semi-automatic data annotation method.

Lastly, we introduce an innovative model compression technique employing pruning and quantization. Our approach involves utilizing the statistics of the weights of the model to determine which weights should be removed (pruning). Subsequently, we ternarize the retained weights based on their sign. This method was designed to be versatile, applicable to models derived from the preceding contributions.

Thesis structure

The first chapter is dedicated to the introduction of the medical and scientific context through the presentation of cerebral emboli, transcranial Doppler (TCD) ultrasonography, and the current challenges of HITS detection and classification.

The second chapter presents an overview of the state-of-the-art of the four main topics related to my research: (1) cerebral emboli detection and classification, (2) automatic and semi-automatic data annotation, (3) multi-feature classification, and (4) model compression.

The third chapter will introduce and detail the method that we developed for semi-automatic data annotation. This method was used to generate and annotate different HITS datasets used in the following chapters. The results and validation of this method will also be presented in this third chapter.

The fourth chapter tackles directly the different classification models that we proposed to improve cerebral emboli classification. These models focus on multi-feature classification, and are validated on three medical datasets.

The fifth chapter is devoted to the model compression approach that we developed in order to reduce the resource and energy requirements of the models developed in the previous chapter. This is an important step in our research as we want these models to be compatible with the clinical context, where the devices may have resources limitations.

Finally, in chapter six, we conclude and present some perspectives that could push further the boundaries of cerebral emboli classification.

1 Context

This chapter will introduce the general context in which my work takes place. First, I will introduce the medical context regarding cerebral emboli and its associated risks. Secondly, I will introduce the scientific context, concerning the post-processing techniques applied to obtain the features used for emboli detection and classification. This chapter does not intend to do an exhaustive review of cerebral emboli and its detection, but presents the main concepts for our study.

Contents

General objective

Stroke is one of the leading causes of dead and disability in the world (George, 2017). The most common type of stroke, ischemic stroke, can often be caused by cerebral emboli (Meyer *et al.*., 1996; Rosenkranz *et al.*., 2006; Iguchi, Kimura, Kobayashi, Ueno, Shibazaki and Inoue, 2008). Thus, cerebral emboli detection and classification is an important tool to help clinicians prevent stroke and manage their patients.

Moreover, transcranial Doppler (TCD) ultrasonography have improved cerebral emboli detection, as it is more flexible (non-invasive, relatively cheap, easy to transport and to adjust) and allows long-duration recordings compared to other modalities. However, no methods in the literature are able to accurately do cerebral emboli classification with portable TCD data, according to the main three classes: solid embolus, gaseous embolus, and artifact.

Therefore, in this work we will focus on the classification and characterization of cerebral emboli using the portable TCD device from Atys Medical, TCD-X.

I Medical context

I.1 Cerebral emboli

I.1.A Origin

Definition and associated risks. Cerebral emboli are extraneous materials that can circulate in the blood flow and block an artery supplying blood to the brain (Chung and Caplan, 2007; Haines, 2018). When this happens, ischemic stroke can occur as a result of the occlusion, which prevents the brain of having a correct supply of oxygen (Roth, 2011). Based on the type of ischemic stroke, the affected zone of the brain, and the degree of the occlusion, the effects of the stroke can vary. First, let us define the two types of ischemic strokes according to (*Types of Stroke*, 2022): thrombotic and embolic. The former occurs when the source of the blockage is a blood clot originated in a blood vessel directly connected to the brain or inside it. This happens, for instance, when the middle cerebral artery (MCA) is blocked by a fragment of an atheromatous plaque due to carotid stenosis (Stork, Kimura, Levi, Chambers, Abbott and Donnan, 2002). Moreover, in some cases, this type of stroke can be preceded by transient ischemic attacks, which can be an indicator of a possible stroke. On the other side, the embolic stroke takes place when the material blocking the blood vessel is generated elsewhere in the body, for instance due to a cardiac problem such as atrial fibrillation. Furthermore, the stroke can occur in different zones in the brain, and based on the zone, the symptoms may differ (*Effects of Stroke*, 2023, 2022). Globally, as one side of the brain controls the opposite side of the body, the effects of stroke in one side of the brain are materialized in the opposite side of the body (*Effects of Stroke*, 2023). What is more, not only the side of the brain touched by the stroke is important, but also the part of the brain that is concerned (*Effects of Stroke*, 2022). The brain is mainly divided in three parts: the brainstem, the cerebellum, and the cerebrum (see Figure 1.1). The brainstem is located at the bottom of the brain, and it connects the spinal cord and the brain. It controls some involuntary functions such as the regulation of the body temperature, breathing, heart rate, blood pressure, balance, hearing, and swallowing. Therefore, a stroke touching this area may disrupt some of these functions, and in some serious cases, it could lead to coma. By the same token, the cerebellum can be found at the back of the brain, between the brainstem and the cerebrum. It controls motor commands such as balance, posture, coordination of muscle movements, as well as some cognitive functions such as language and attention. Thus, a stroke affecting this zone can create motor and coordination problems, as well as headaches, nausea,

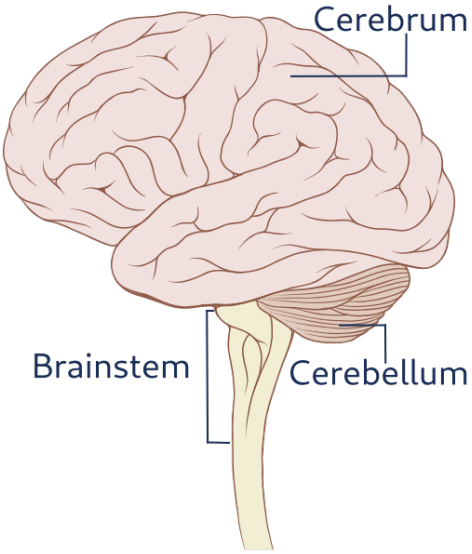


Figure 1.1: Three main parts of the brain: brainstem, cerebellum, and cerebrum. Medical Illustrations by Patrick Lynch, generated for multimedia teaching projects by the Yale University School of Medicine, Center for Advanced Instructional Media, 1987-2000.

and dizziness. Last but not least, the cerebrum corresponds to the largest part of the brain and it is composed of two parts, the right and left hemispheres. It controls many important functions such as learning, speech, vision, hearing, reasoning, emotions, muscle functions, etc. As a consequence, a stroke touching the cerebrum may cause (based on the hemisphere) memory issues, weakness and/or paralysis of a part of the body, visual and speech problems, behavioral changes, etc.

Characterization with transcranial Doppler. In this work, we focus on cerebral emboli detection and classification using portable transcranial Doppler ultrasound devices. Therefore, for the detection task, we used the identification criteria established on the consensus committee of the 9^{th} international cerebral hemodynamic symposium (*Basic Identification Criteria of Doppler Microembolic Signals*, 1995). The criteria are the following:

- The duration of microembolic signals is smaller than 300 ms, as they are transient signals and we search to remove artifacts (voice of the patient, probe movement, etc.) which tend to be of longer duration.
- The increase in intensity of microembolic signals with respect to the signal of the blood flow is of at least 3 dB.
- Microembolic signals are unidirectional in the time-frequency representation of the Doppler signal. This is an important criterion allowing to distinguish solid and gaseous emboli from artifacts (see Figure 1.2).
- Cerebral emboli signals have a music sound, similar to a "snap" or "chirp".

These criteria are not absolute, and they depend on different factors such as the recording duration, the ultrasound probe frequency, the fast Fourier transform (FFT) parameters, etc. Therefore, a new committee was established in 1998 (Ringelstein *et al.*, 1998) to discuss more guidelines about the use of TCD devices for emboli detection. However, the 1995 consensus guidelines remains used in clinical practice.

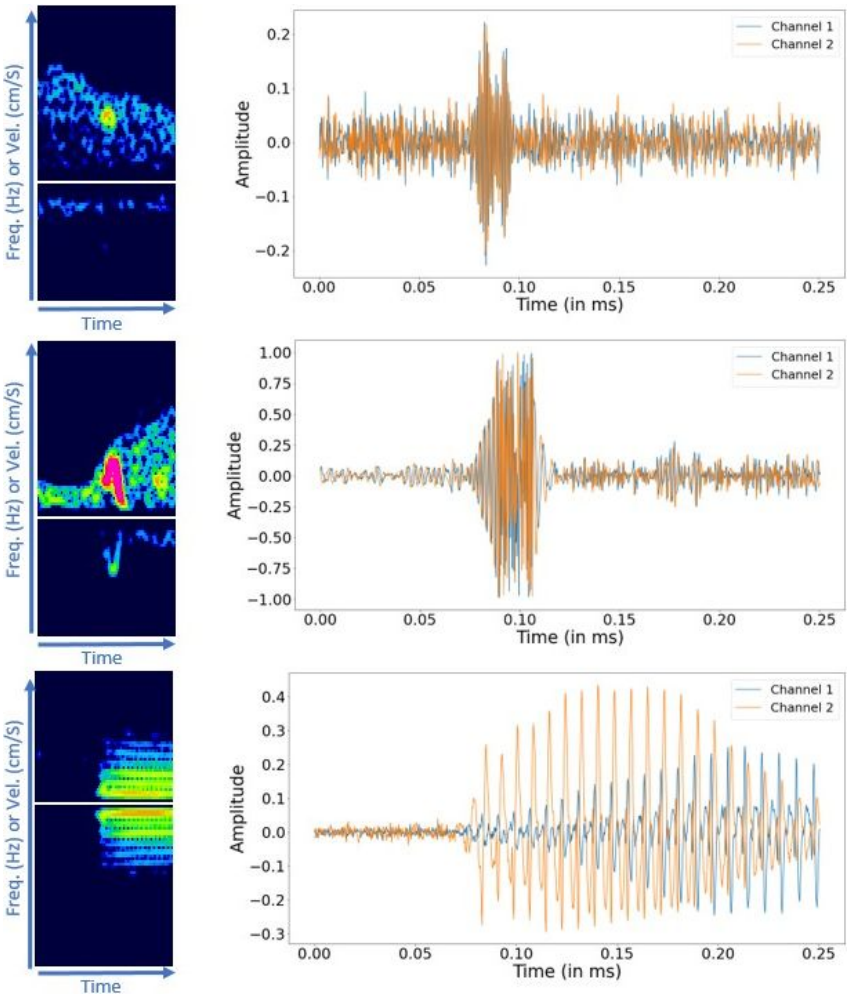


Figure 1.2: Example of time-frequency representations and raw signals of the three types of HITS. The first row corresponds to a solid embolus, the second row corresponds to a gaseous embolus, and the third row corresponds to a (voice) artifact. The first column correspond to the time-frequency representations (where the baseline is the horizontal white line, indicating the 0 Hz (or cm/s) value), and the second column correspond to the amplitude of the in quadrature (IQ) signal (composed of two channels) as a function of the time. We can see that the two emboli (solid and gaseous) are unidirectional on the time-frequency representation, contrary to the artifact. Indeed, the artifact presents a symmetry with respect to the baseline (zero frequency line), which is not present for the emboli. If an embolus exhibits symmetry concerning the baseline, with both positive and negative velocities simultaneously, it implies that it is traveling in two opposing directions simultaneously—along with the blood flow direction and against it. This scenario is not physically plausible.

I.1.B Types

As mentioned before, cerebral emboli can be detected using TCD ultrasonography. To do this, HITS are detected in the patient TCD signal recording. These high intensity transient signals correspond to zones in the Doppler signal having a higher intensity than the one of the background (mainly the blood flow)[1]. Therefore, to be able to perform cerebral emboli classification, it is important to understand the characteristics of the different types of HITS. Here, we are going to describe the three main types of HITS (solid embolus, gaseous embolus, and artifacts) based on the seminar of Dr. Ruud W.M. Keunen (Keunen, 2022).

Zero-crossing dynamics. According to the consensus committee of the 9^{th} International Cerebral Hemodynamic Symposium (*Basic Identification Criteria of Doppler Microembolic Signals*, 1995), emboli have a characteristic musical sound, similar to a "chirp", "snap" or "moan". However, in that comittee, this was not quantified. Dr. Keunen quantified this through the zero-crossing dynamics of the signal (Keunen, Hoogenboezem, Wijnands, Van den Hengel and Ackerstaff, 2008). Indeed, the musical sound of emboli come from two factors. First, there is the intensity increase with respect to the background, with higher amplitude variations. However, artifacts can also have these amplitude variations. The second factor, which is the most important one for emboli according to (Keunen *et al.*., 2008), is the influence of the velocity and the regularity of the zero-crossings. Dr. Keunen and his collaborators noted that, in the background signal (i.e. the blood flow), the zero-crossings are quite irregular compared to microembolic signals. To quantify this, they introduced the zero-cross index (ZCI), which measures the different frequencies at which the zero-crossings occurs. Thus, its value lies between 0 and infinity, smaller values indicating more regular zero-crossings, and higher values indicating more irregular zero-crossings.

Thanks to this, Dr. Keunen quantified the musical characteristics of solid and gaseous emboli by its ZCI range values, that will be detailed for the different type of HITS hereafter.

Solid embolus. Solid emboli are solid particles (with a mass) that can circulate in the cerebral blood flow. As the blood flow in the vessels is laminar, higher velocities are observed at their center, thus solid emboli will tend to circulate at the center of the vessels at higher velocities. Therefore, in a time-frequency representation, solid emboli will tend to have higher frequencies (see first row of Figure 1.2).

Moreover, Dr. Keunen identified the following criteria to characterize solid emboli with TCD devices:

- An intensity increase with respect to the background signal in the range of 3 to 5 dB;
- A duration smaller than 30 ms;
- A ZCI value between 25-50;
- Unidirectional in a time-frequency representation.

Furthermore, the source of a solid embolus can be related to its position in the cardiac cycle[2] (Keunen, 2022). On the one hand, if the source of the solid embolus is in the bifurcation between the internal and external carotid arteries (see Figure 1.4.a), e.g., coming from an unstable plaque

[1] More details about the detection mechanism will be specified in Section II.3.

[2] The rationale behind Dr Keunen explanation is that, as a solid particle, a solid embolus will circulate in the cerebral blood stream at a certain velocity, so based on the origin, it is more probable to observe it at some cardiac cycle positions than others.

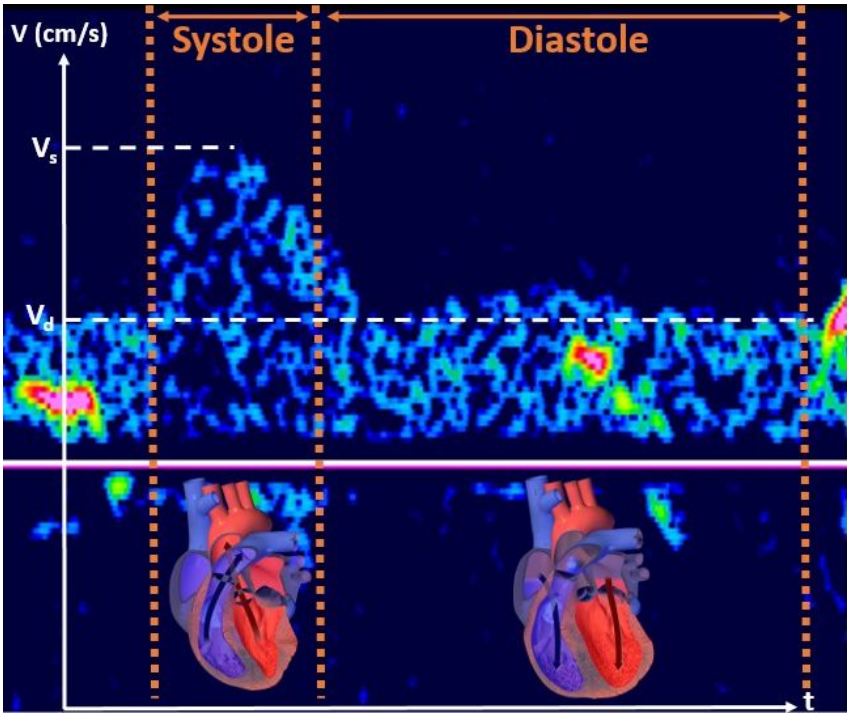


Figure 1.3: Systole and diastole, observed in a time-frequency representation of a TCD signal. The systole corresponds to the contraction of the heart. The diastole corresponds to the expansion of the heart. The systolic velocity, V_s, is defined as the maximum velocity during the systole, which happens at the systolic peak. The diastolic velocity, V_d, is defined at the velocity at the end of the diastole. Heart mages from BruceBlaus. English: Systole vs Diastole. See a full animation of this medical topic. 5 novembre 2015. Own work, Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Systolevs_Diastole.png.

in a carotid stenosis patient, the embolus will be released during the systole (corresponding to the contraction of the heart, see Figure 1.3), and it will be observed in the middle cerebral artery (MCA during diastole, corresponding to the expansion of the heart). Therefore, if an important number of solid emboli are observed only during the diastole, the origin of these emboli could be at the bifurcation of the two aforementioned arteries. On the other hand, if the source of the solid embolus is an unstable plaque located near the bifurcation between the internal carotid artery and the anterior cerebral artery (see Figure 1.4.b), then if the embolus is released during the systole, it will probably appear in the MCA during the systole too.

What is more, the impact of solid emboli on a patient depends on cerebral hemodynamics (Ogasawara, Suga, Sasaki, Chida, Kobayashi, Yoshida, Otawara and Ogawa, 2008a; Keunen, 2022). Indeed, higher cerebral blood flow velocities allow a better clearance of solid emboli than low flow velocities. Therefore, small solid emboli are more dangerous in poorly perfused brains as it has more chances of being stuck. Additionally, as these small solid emboli are asymptomatic, particular care have to be taken for patients with that profile[1].

Gaseous embolus. Contrary to solid emboli, gaseous emboli do not have a mass, so their movement in the blood vessels is more chaotic, going from one wall to another. That is why, in a time-frequency representation, one gaseous embolus can alternate between high and low frequencies (or velocities), having a more elongated shape than solid emboli, or a *v-shape* (see

[1]A solid asymptomatic embolus do not produce any evident symptoms, but as indicated, they can (partially) block a cerebral artery and cause ischemic stroke, thus they can be dangerous.

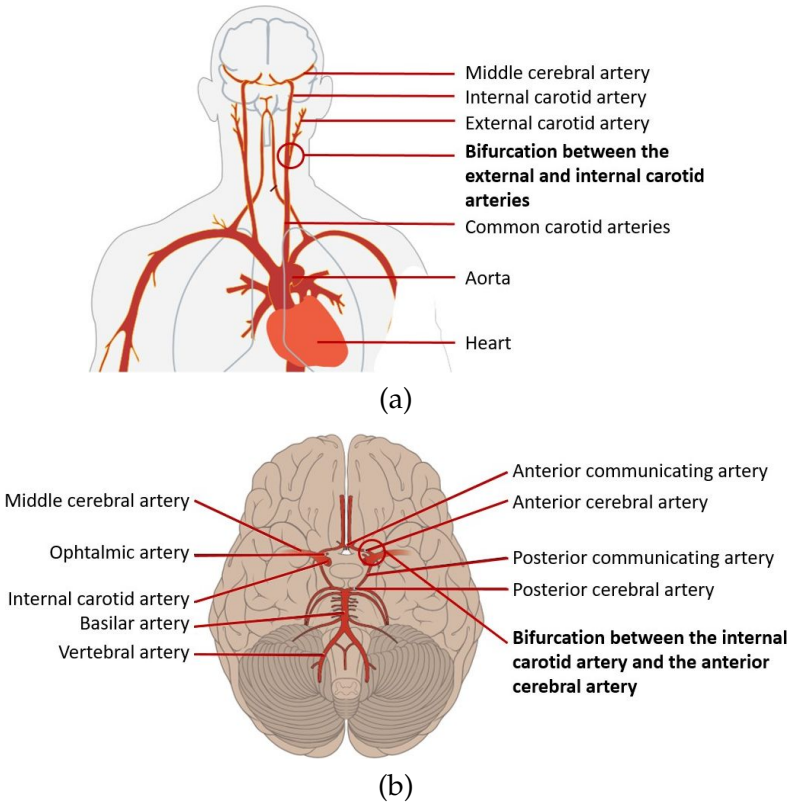


Figure 1.4: Main components of the blood supply to the brain. Base images from: (a) Żbiczek, LadyofHats, translated by. Polski: Uproszczony schemat układu tętniczego u człowieka. May 3, 2009. translation of File:Arterial System en.svg, Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Arterial_system_pl.svg. (b) College, OpenStax. Illustration from Anatomy Physiology, Connexions Web site. http://cnx.org/content/col11496/1.6/, Jun 19, 2013. Anatomy Physiology, Connexions Web site. http://cnx.org/content/col11496/1.6/, Jun 19, 2013., Wikimedia Commons, https://commons.wikimedia.org/wiki/File:2123_Arteries_of_the_Brain.jpgfilelinks. Referred to the 12/06/2023.

second row of Figure 1.2). Additionally, gaseous emboli are strong reflectors of the ultrasound signal, therefore, when they enter the sample volume, they can overload the signal and create increase of the amplitudes with respect to the background signal.

Moreover, the main characteristics identified by Dr. Keunen for gaseous emboli with TCD devices are the following:

- An intensity increase with respect to the background signal in a range of 5 to 15 dB (higher intensities than solid emboli);
- A duration in the range of 30-100 ms (longer than solid emboli);
- A ZCI value between 0-25 (smaller than solid emboli);
- Unidirectional in a time-frequency representation;
- There is no relation between the occurrence of a gaseous embolus and the cardiac cycle position.

We can see that, even if the basic identification criteria of gaseous and solid emboli are different, in some cases they can be confused. For instance, a small gaseous embolus can be mixed up with a large solid embolus, when the intensity increase with respect to the background signal is of 5 dB, the duration around 30 ms and the ZCI of 25.

Furthermore, gaseous emboli have been less studied than solid emboli with TCD devices, as their identification is not an easy task. However, they can have an impact on the brain. Indeed, too much air in the brain can generate cognitive dysfunctions after surgery, and gaseous emboli can occlude capillaries which can lead to small lesions in the brain. More details about the impact of gaseous emboli on the brain are going to be discussed in Section I.1.C.

Artifact. Artifacts are parasite and undesired HITS that may be artificially generated and do not give any useful information to the clinicians. They can be generated by different sources, such as the voice of the patient, the movement of the patient, the movement of the robotic probe (for devices equipped with robotic probes such as Atys Medical TCD-X device or WAKIe R3), electrical noise, etc.

The basic identification criteria established by Dr. Keunen for artifacts are the following (see third row of Figure 1.2):

- Lower frequencies than solid or gaseous emboli.
- An intensity increase with respect to the background signal in a range of 5 to 15 dB, similar to gaseous emboli.
- Duration between 15 and 300 ms (300 ms is the maximal duration of a HITS according to the 1995 consensus).
- They can be bidirectional. This translates in the time-frequency representation as symmetric with respect to the time (horizontal) axis.
- ZCI values larger than 200, much higher than the ZCI values of solid and gaseous emboli.

Artifacts are HITS that are not clinically useful. However, it is important to identify them to avoid confusion with emboli, specially for portable TCD devices which are more prone to artifacts. Indeed, confusing an artifact as a solid or gaseous embolus can be serious, as it can mislead the treatment given to the patients.

I.1.C Possible sources

Cerebral emboli can be generated by different sources. They often come from the heart (cardiogenic) or proximal arteries such as the internal carotid artery. To better understand the origin of the emboli, we divided the sources between pathologies, medical procedures, and external sources.

Pathologies.

Carotid stenosis. A common source of cerebral emboli is carotid stenosis. Its link with ischemic stroke has been largely studied by the medical community for symptomatic and asymptomatic patients (Stork *et al.*., 2002; Sitzer, Müller, Siebler, Hort, Kniemeyer, Jäncke and Steinmetz, 1995; Markus and MacKinnon, 2005; Siebler, Kleinschmidt, Sitzer, Steinmetz and Freund, 1994). Carotid stenosis is the narrowing of a carotid artery due to the formation of plaques in the walls of the artery (see Figure 1.5.a). This plaque can be composed of fat, cholesterol, calcium or other substances (*Carotid Artery Stenosis*, 2022). Moreover, patients suffering from carotid stenosis can have strokes in two contexts: (1) when the narrowing is large enough to directly block the supply of blood to the brain in the formation place, and (2) when a fragment of the plaque separates from it (becoming a solid embolus), follows the blood flow, and blocks it at another location (in the same artery or another one). Similarly to carotid stenosis, recent studies have established a link between ischemic stroke and aortic arch atheroma (Viguier, Pavy le Traon, Massabuau, Valton and Larrue, 2001; Zavala, Amarrenco, Davis, Jones, Young, Macleod, Horky and Donnan, 2006; Viedma-Guiard, Guidoux, Amarenco and Meseguer, 2021). Indeed, atheromatous plaques can also appear in the aortic arc, and cerebral emboli can be generated from it if a fragment detaches from the plaque.

Atrial fibrillation. Atrial fibrillation (AF) can also be a source of cerebral emboli (*Atrial Fibrillation and Stroke | National Institute of Neurological Disorders and Stroke*, 2023). AF is a type of arrhythmia where the two upper chambers of the heart, the atria, beat at an irregular rhythm (CDC, 2022), generating fast contractions with small amplitudes. Because of this, the blood flow between the upper and lower chambers of the heart is slow and irregular. Therefore, blood can be quasi-static, generating blood clots that can become emboli. These emboli can then go to the brain and cause cardiogenic embolic stroke, as it has been shown by several clinical studies (Tsao, Aday, Almarzooq, Alonso, Beaton, Bittencourt, Boehme, Buxton, Carson, Commodore-Mensah, Elkind, Evenson, Eze-Nliam, Ferguson, Generoso, Ho, Kalani, Khan, Kissela, Knutson, Levine, Lewis, Liu, Loop, Ma, Mussolino, Navaneethan, Perak, Poudel, Rezk-Hanna, Roth, Schroeder, Shah, Thacker, VanWagner, Virani, Voecks, Wang, Yaffe and Martin, 2022; Kamel, Okin, Elkind and Iadecola, 2016; Freedman, Potpara and Lip, 2016; Stirling, Muramatsu and Shirai, 1996).

Patent Foramen Ovale. Patent foramen ovale (PFO) can be associated to ischemic stroke (*Patent Foramen Ovale (PFO)*, 2011; Schminke, Ries, Daffertshofer, Staedt and Hennerici, 1995). Contrary to carotid stenosis or AF, PFO do not generate emboli. However, they allow particles and materials generated elsewhere in the body to go to other parts (such as the brain) becoming cerebral emboli that can cause stroke (Anzola, Magoni, Guindani, Rozzini and Volta, 1999; Serena, nez Nieto, Silva and Castellanos, 2010). Indeed, PFO is a type of hole between the left and right upper chambers of the heart that occurs when the foramen ovale do not close after birth. Normally, materials circulating in the bloodstream are filtered by the lungs after passing by the right atria and ventricle. However, in patients with PFO, these materials can bypass the

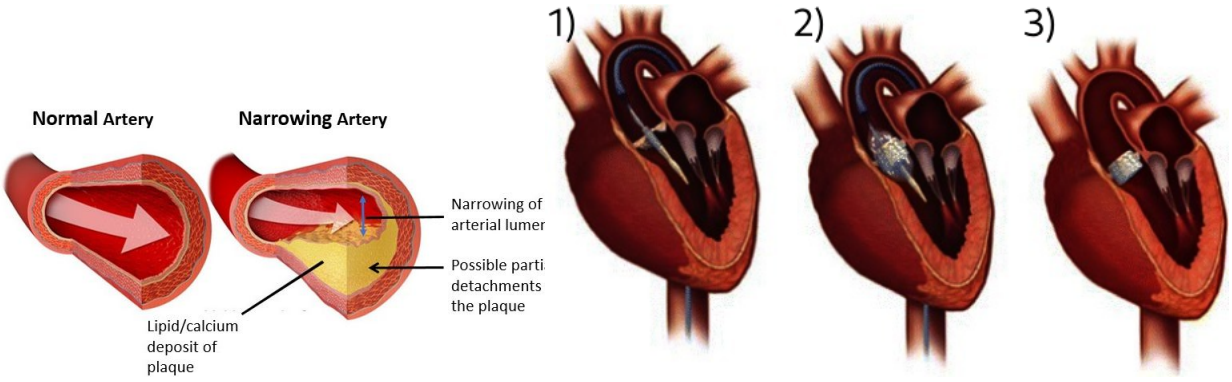
lungs and go directly to the left atria and ventricle, which can then go to other organs such as the brain (*Patent Foramen Ovale (PFO)*, 2011) causing paradoxical stroke. Additionally, the size of the PFO also has an influence on the risk of stroke (Schuchlenz, Weihs, Horner and Quehenberger, 2000). Because of this, PFO existence tests have been developed (Kamel *et al.*., 2016; Telman, Kouperberg, Sprecher, Goldsher and Yarnitsky, 2005), where contrast agents are injected into the patient, and TCD monitoring is used to detect potential microbubbles (gaseous emboli) in the cerebral arteries (MCA) associated to the presence of a PFO.

Other pathologies. In addition, other pathologies such as mitral annular calcification, pulmonary arterionvenous malformation, or valvular vegetations can be a source of cerebral emboli. For more details about this, we refer the reader to (Marcoff and Homma, 2014).

Medical procedures. Different types of emboli can be generated during different medical procedures, surgical or nonsurgical.

Carotid endarterectomy. A common surgical procedure producing emboli is carotid endarterectomy (CEA) (Ackerstaff, Jansen, Moll, Vermeulen, Hamerlijnck and Mauser, 1995; Ackerstaff, Moons, van de Vlasakker, Moll, Vermeulen, Algra and Spencer, 2000; Jansen, Ramos, van Heesewijk, Moll, van Gijn and Ackerstaff, 1994; Ogasawara, Suga, Sasaki, Chida, Kobayashi, Yoshida, Otawara and Ogawa, 2008b), as it is strongly related to carotid stenosis. Indeed, CEA consists in removing the atheromatous plaque formed in the carotid artery in patients with carotid stenosis (*Carotid endarterectomy*, 2017). This is an important procedure when the narrowing of the concerned carotid artery has become important or the plaque is unstable, becoming a potential source of cerebral emboli (Bonati, Kakkos, Berkefeld, de Borst, Bulbulia, Halliday, van Herzeele, Koncar, McCabe, Lal, Ricco, Ringleb, Taylor-Rowan and Eckstein, 2021). Because of the nature of the procedure, some debris of the plaque can go to the brain and cause embolic stroke (Ackerstaff *et al.*., 1995, 2000; Spencer, Thomas, Nicholls and Sauvage, 1990), which is paradoxical. As a consequence of this paradox, several studies have shown that CEA is not always the best alternative to treat carotid stenosis. Therefore, a recent guideline has been proposed by the European Stroke Organization to handle carotid stenosis with CEA (Bonati *et al.*., 2021), while trying to minimize the emboli generation risks.

Transcatheter aortic valve implantation. Another surgical procedure generating different types of emboli is transcatheter aortic valve implantation (TAVI) (Szeto, Augoustides, Desai, Moeller, McGarvey, Walsh, Bannan, Herrmann and Bavaria, 2011), which is used to replace damaged aortic valves unable to properly open because of (aortic) stenosis (see Figure 1.5.b). The procedure is minimally invasive and consists in accessing the aortic valve using a catheter guided through the femoral artery or through the apex of the heart. Then, the catheter is used to deploy a new artificial valve within the damaged aortic valve to restore the function of the latter. The procedure is composed of several stages where emboli can be generated. Indeed, solid emboli can be generated from atheroma in the aorta, calcification of the aortic valve, or by the catheter when moving through the artery (Masson, Kovac, Schuler, Ye, Cheung, Kapadia, Tuzcu, Kodali, Leon and Webb, 2009; Van Mieghem, El Faquir, Rahhab, Rodríguez-Olivares, Wilschut, Ouhlous, Galema, Geleijnse, Kappetein, Schipper and de Jaegere, 2015). Gaseous emboli can also be generated from the left ventricular cannulation (Masson *et al.*., 2009), contrast agent injection, or catheter flushing (Aggarwal *et al.*., 2018). For more details about the different stages and embolization risk, we refer the reader to (Masson *et al.*., 2009; Aggarwal *et al.*., 2018).



(a) (b)

Figure 1.5: Different sources of emboli. (a) Narrowing of a carotid artery due to atherosclerosis. Oregon State University. Artery disease. photo, April 18, 2016. Flickr, https://www.flickr.com/photos/oregonstateuniversity/26481606576/. (b) Transcatheter aortic valve implantation, TAVI (Source: Edwards Lifesciences). The aortic valve is replaced by an artificial valve without narrowing, allowing it to open properly. First (1), a balloon catheter is put into the heart valve passing through the aorta. Then (2), the transcatheter valve is positioned over the defective aortic valve. Finally (3), when the new synthetic valve is in place, the catheter is retired and the procedure finished.

Other procedures. Other procedures such as cerebral angiography (Ikizceli, Donmez, Kahveci and Kahriman, 2021; Markus, Israel, Brown, Loh, Buckenham and Clifton, 1993b) or cardiopulmonary bypass (CPB) (Abu-Omar, Balacumaraswami, Pigott, Matthews and Taggart, 2004) can generate solid and gaseous emboli. Cerebral angiography can generate solid emboli from the manipulation of the catheter (Ikizceli *et al.*., 2021) and gaseous emboli from the injection of the contrast agent (Markus *et al.*., 1993b). CPB can also generate emboli, mainly gaseous, at different stages: perfusionist interventions, aortic cannulas, or cross-clamp (Borger and Feindel, 2002; Abu-Omar *et al.*., 2004).

External sources. As we have seen in the previous paragraph (*Medical procedures*), emboli can be of artificial nature. For instance, gaseous emboli can be generated during the injection of contrast agents as it is done during PFO tests (Schminke *et al.*., 1995) and cerebral angiography (Markus *et al.*., 1993b), or created during surgical procedures such as cardiac surgeries (Mitchell and Gorman, 2002; Abu-Omar *et al.*., 2004). However, other external sources can be at the origin of solid and gaseous emboli. Hereafter, we present two types of external sources of cerebral emboli based on (Judge, Mello, Bradley and Harbison, 2017): pressure-related gaseous emboli and (solid) foreign bodies.

Pressure-related gaseous emboli. Gaseous emboli can be artificially generated because of pressure-related causes. This can occur in several circumstances such as diving, hyperbaric treatments or cabin decompression in flights. In the literature, diving accidents are one of the most common sources of gaseous emboli. Indeed, according to Boyle's law, at a fixed temperature, the pressure is inversely proportional to the volume of gas. When diving, the external (water) pressure changes, modifying the volume of the spaces filled with gas in the body. During descent, the water pressure will increase, compressing (i.e. reducing) the volume of the spaces filled with gas in the body. Then, according to Boyle's law, as the volume of the gas has decreased, the pressure inside those spaces will increase, which can lead to damages in

those spaces[1] and the formation of arterial gas emboli. During ascent, the inverse phenomenon happens: as the water pressure decreases, the gas inside the body is decompressed (i.e. the volume increases). Therefore, as the gas expands, bubbles can be liberated in the tissues, creating gaseous emboli. Because of these reasons, it is important to regulate the volume of gas inside the body. However, this is not always done, which can lead to decompression sickness or stroke due to gaseous cerebral emboli.

Foreign bodies. Another external (but less common) source of cerebral emboli are foreign bodies such as gun pellets (mainly happening in military populations) or illicit drugs. Indeed, in the military domain, fragments from gun pellets or other military munitions can become emboli if they go inside the body. In some rare cases, they reach the brain, and cause ischemic stroke. Moreover, illicit drugs are also a source of emboli as they are often mixed with unknown materials which do not dilute well. Because of these, for intravenous illicit drugs, solid foreign bodies can be introduced in the blood flow, becoming potential cerebral emboli.

I.2 Transcranial Doppler ultrasound

I.2.A General principle

In this section, we are going to review the general principle of transcranial Doppler (TCD) ultrasonography. This neuroimaging tool is the core of our work with Atys medical, as it is the technology that we used to acquire all the TCD data utilized in this work. To do this, we are going to build on (Purkayastha and Sorond, 2013; Naqvi, Yap, Ahmad and Ghosh, 2013).

Physical principle. Transcranial Doppler ultrasonography is a noninvasive, relatively cheap and portable ultrasound technique allowing a real-time monitoring of the cerebral arteries over long periods of time.

The principle at the core of TCD ultrasonography is the Doppler effect. Indeed, TCD devices are composed of an ultrasound probe that is placed on one of the acoustic windows of the skull (see Figure 1.6). This probe is composed of piezoelectric crystals allowing to transmit and receive ultrasound waves. When a wave is emitted at a frequency f_e, it will go through the skull[2] and encounter moving blood composed of red cells. Because of the movement of the red cells, the frequency f_r of the received wave will not be the same as the one of the emitted wave (Doppler effect). As one can measure the frequency f_r of the received wave, and the frequency f_e of the emitted wave is known, one can use the Doppler effect to get the velocity v_r of the moving red cells:

$$v_r = \frac{|f_r - f_e| \times c}{2 \times f_e \times cos(\theta)} \tag{1.1}$$

where c is the speed of the emitted wave in the propagation medium (for soft tissues it is assumed constant with a value of 1541 m/s) and θ is the Doppler angle of the wave with respect to the direction of the blood flow.

For TCD measurements, one important assumption regarding θ is made to be able to compute the velocity v_r. Indeed, the angle θ is not always easy to measure and it has an impact on the precision of the estimated v_r. In the ideal case, to have an accurate measurement of v_r, $\theta = 0$ which means that the emitted wave is parallel to the blood flow. However, this is

[1] This is because, the tissue will fill these spaces to equalize pressure, which can lead to its rupture.

[2] In fact, this depends on the chosen emission frequency: higher frequencies are more attenuated through their path, so they may not penetrate the skull. That is why, in general, for TCD ultrasonography, low frequencies are chosen, below 2 MHz.

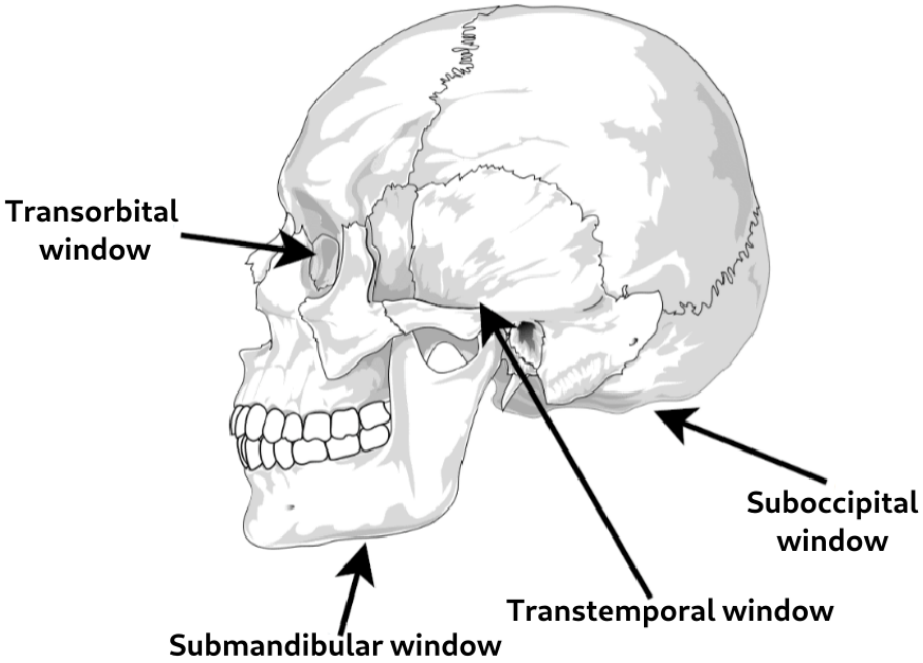


Figure 1.6: Acoustic windows commonly used in transcranial Doppler ultrasound monitoring. The different windows allow to insonate different arteries. The transtemporal window is the one used in this work, as it allows to insonate the middle cerebral artery (MCA), as well as the anterior and posterior cerebral arteries (however, in this work we focus on the MCA). The transorbital window allows to insonate the internal carotid artery and the ophtalmic artery. The submandibular window allows to insonate the MCA. The suboccipital window allows to insonate the vertebral arteries as well as the basilar artery. Image from (Weir, 2016).

not possible in practice, so this angle should be reduced as much as possible to decrease the approximation error of v_r. According to (Purkayastha and Sorond, 2013), for $\theta \geq 30$, the made error is higher than 15%, so an angle of $\theta \leq 30$ should be used to guarantee a good quality of the measurement.

Acoustic windows. As mentioned before, different acoustic windows can be used with TCD ultrasonography: the transtemporal window, the suboccipital window, the transorbital window, and the submandibular window. It is important to place the probe at an acoustic window and not elsewhere as they correspond to thin regions of the skull where the used ultrasound waves can penetrate it. For emboli detection, the most commonly used window is the transtemporal window, as it allows to insonate the middle cerebral artery (MCA), which can be found at depths of 35-55 mm depending on the patient. Moreover, in this artery, the flow is unidirectional towards the probe up to the MCA trifurcation, where it becomes bidirectional. By the same token, the transtemporal window is particularly interesting for Atys Medical portable TCD device, the TCD-X Holter, as it allows the device to be attached using a headband or a glasses-inspired attachment (see Figure 1.7).

Types. There are mainly two types of TCD devices: duplex and non-duplex. The main difference between them is that, in duplex devices the imaging of the organs is done in parallel to the blood flow measurement. Therefore, in duplex devices the monitored vessels are directly known, and thanks to the imaging, an estimation of the Doppler angle can be done, improving the quality of the estimated velocities. On the contrary, in non-duplex devices, such as Atys

Medical TCD-X Holter, the vessels are not directly identified, but the acquired information (audible signal, time-frequency representation, used acoustic window, etc.) has to be used to determine the monitored vessel. In this work, we focus on non-duplex TCD devices, as it is the only one allowing a long-duration real-time monitoring of the cerebral blood flow.

Advantages and drawbacks. TCD devices have different advantages and drawbacks. From the one hand, the main advantages of TCD devices are that they are noninvasive, they have good temporal resolution, they are relatively cheap, they are more portable than other medical devices (such as magnetic resonance imaging, MRI, or computed tomography, CT), and they are the only medical devices allowing a real-time monitoring of the cerebral blood flow. This last point is particularly interesting, as many applications require real-time monitoring of the cerebral blood flow, such as TAVI, cardiopulmonary bypass, or AF ablation (AFA). On the other hand, TCD devices have some drawbacks which can limit the quality of the monitoring. Indeed, TCD devices make some assumptions to be able to measure the velocities of the bodies in movement in the blood flow (mainly assumptions about the Doppler angle and the diameter of the vessels). Moreover, TCD ultrasound measurements depend on the used acoustic window (some patients can have inadequate acoustic windows), as well as the operator (training, experience, etc.). This last point has been partially solved by Atys Medical TCD-X Holter, which uses a robotized probe to automatically detect the best signal in the MCA, using the transtemporal window.

I.2.B Atys medical robotized probe

In this work, the TCD data were acquired using the portable TCD device from Atys Medical, the TCD-X Holter (Figure 1.7). It can be used on ambulatory patients to monitor their cerebral blood flows on the MCA using the transtemporal window, for long term monitoring sessions with a duration going up to 10 hours[1] (signal recorded in an SD card). To achieve this, a robotized probe automatically finds the best MCA signal during the recording, allowing to avoid loosing the Doppler signal. This property of the TCD-X is particularly interesting as cerebral emboli are rare events, so long duration monitoring are required to catch these events. However, the main inconvenient of this is that it tends to generate more artifacts than classical TCD devices, as the patient is in free movement, so they can talk, walk, or do other activities during the recording.

Furthermore, the main two components allowing portable TCD recordings with the TCD-X Holter are the following: (1) a single robotized ultrasound probe[2] and (2) a headset, allowing to fixate the probe to the transtemporal window of the patient during all the recording (see Figure 1.7.b). First, the robotized probe is able to automatically find the signal in the MCA. To do this, the probe tries different orientations in order to search for the Doppler signal, and then it chooses the orientation maximizing the signal. This is interesting as, in long term recordings on ambulatory patients, the Doppler signal can easily be lost (patient can freely move), requiring a recalibration to find the signal. Classically, this is done by clinicians or qualified technicians, but this is not practical for ambulatory patients. The proposed robotized probe allows overcoming this difficulty, making long duration monitoring on ambulatory patients more reliable. Secondly, in order to keep the robotized probe in the transtemporal window of the patient, the TCD-X Holter uses a headset as in Figure 1.7.b. The headset is composed of a spectacle frame, the robotized probe, a counterweight, and a headband. The

[1] The monitoring duration depends on the pulse repetition frequency, see Appendix 1 for more details.

[2] Atys medical has now developed a new bilateral TCD-X Holter composed of two probes, allowing to monitor the cerebral blood flow using the two transtemporal windows.

(a) (b)

Figure 1.7: TCD-X Holter device. (a) Recorder module. (b) headset used for fixating the robotized probe to the temporal window. Images from Atys Medical.

spectacle frame and the headband allow supporting the probe and linking the probe to the counterweight. The counterweight allows mainly balancing the spectacle frame and ensuring the tension of the headband. This fixation mechanism of the probe is important allows as it allows having a reliable and comfortable portable TCD device usable on ambulatory patients, specially for long term recordings. Indeed, as the patient is in movement, so will be the probe if it is not properly fixed (it can also be moved by external sources, for instance when the recording is done during a surgical procedure). Moreover, the robotized probe must remain in the transtemporal window, close to a region where the signal can be found, to be able to stay on the best Doppler signal: the presented headset allows this.

At last, the acquired Doppler signal is recorded in an SD card, which can then be analyzed with the proprietary software, Atys data management software (ADMS). This software will detect all the possible HITS present in the signal, classify them between potential artifacts and emboli (no distinction between solid and gaseous), and compute many features that can be analyzed by the user, such as the HITS-to-blood ratio (HBR), the HITS velocity, its position in the cardiac cycle, the systolic, diastolic and mean velocities, the embolic relative duration, etc.

II Scientific context

II.1 Used representations

II.1.A Raw signal

The raw signal of a HITS corresponds to the Doppler signal acquired by the TCD device. Indeed, the received ultrasound wave (acoustic perturbation reflected by the moving bodies such as red cells or emboli in the cerebral blood flow) is processed by the TCD device, and demodulated using a quadrature demodulation. Because of this demodulation, we obtain a signal $S \in \mathbb{R}^{2\times N}$ of length N, composed of two channels, $I \in \mathbb{R}^N$ and $Q \in \mathbb{R}^N$, corresponding to the in-phase and quadrature components, respectively. This stereo signal can be used to identify emboli according to the criteria mentioned in Section I.1.A.

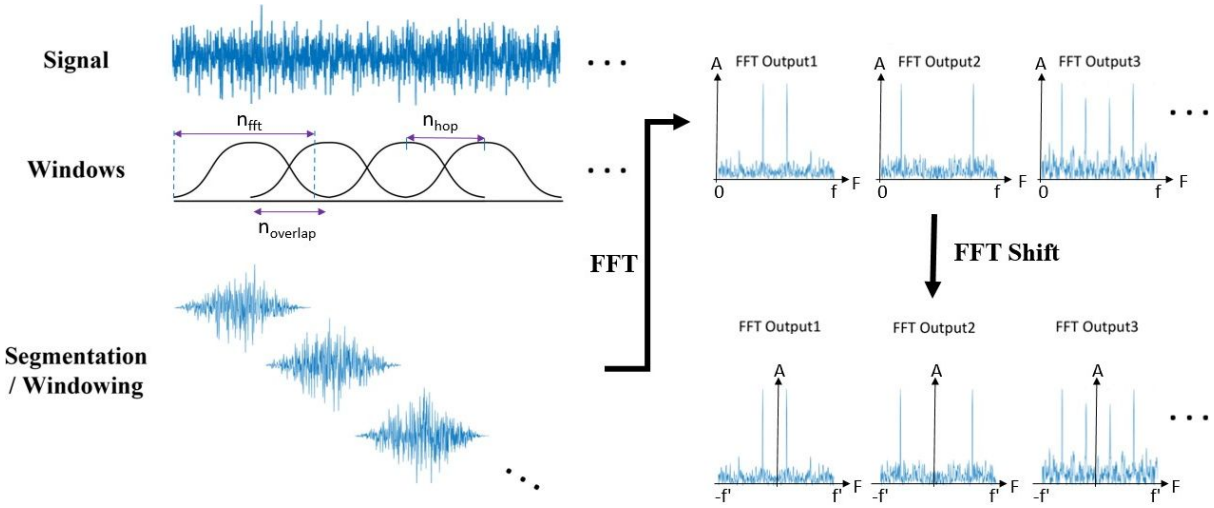


Figure 1.8: Overview of the short-time Fourier transform (STFT). Image from (Jeon *et al.*., 2020)

II.1.B Time-frequency representation

Time-frequency representations (TFRs) are commonly used to identify HITS and classify them between solid embolus, gaseous embolus, and artifact. These representations have several advantages. First, it allows giving a visual representation of the Doppler signal, which is easier to analyze by clinicians, specially for long duration recordings. Second, it allows measuring different useful quantities such as the systolic, diastolic and mean velocities. Indeed, this can be achieved as we work with Doppler signals, so there is a direct correspondence between velocity and frequency according to equation 1.1.

Logarithmic scale spectrogram. The most commonly used time-frequency representation (and the one used by ADMS) is the logarithmic scale spectrogram. To obtain this representation, we detail hereafter the various steps that are applied (Guepie *et al.*., 2018): (1) extraction of a complex IQ signal from the stereo signal, (2) signal filtering, (3) short-time Fourier transform (STFT), (4) spectrogram computation, (5) logarithmic scaling, (6) fast Fourier transform (FFT) shifting, and (7) amplitude filtering based on the obtained spectrogram's statistics.

First, from the stereo signal S, we extract a complex mono signal S_C defined as follows:

$$S_C = I + i \times Q \tag{1.2}$$

Then, the so obtained signal is filtered using a 4^{th} order Butterworth digital high-pass filter, with a cut-off frequency of 150 Hz, to obtain a filtered signal S_C^F.

Afterwards, the STFT of the filtered signal S_C^F is computed (see Figure 1.8), using a Hamming window of length n_{FFT}, with an overlap between two consecutive windows' centers of $n_{overlap}$ points, and a hop length between two consecutive windows of $n_{hop} = n_{FFT} - n_{overlap}$. For the HITS, we used $n_{FFT} = 128$ and $n_{overlap} = 8$.

Subsequently, the logarithmic scale spectrogram is computed:

$$Spec_{log}(S) = 10 \times log_{10}(|STFT(S_C^F)|^2) \tag{1.3}$$

This logarithmic scale spectrogram is a 2D matrix, composed of $F = n_{FFT}$ rows corresponding to the frequency bins, and $T = \lfloor \frac{N}{n_{hop}} \rfloor$ columns corresponding to the time bins.

Next, FFT shifting is applied to $Spec_{log}(S)$ to center the spectrogram around the zero frequency component, obtaining $Spec_{log}^{shift}(S)$. Finally, a new filtering is done on the amplitudes of

the shifted spectrogram, to remove extreme amplitude values, obtaining the final spectrogram $Spec_{final}$. To do this, two thresholds are defined as follows:

$$min_{dB} = \mu_{Spec} + a \times \sigma_{Spec} \tag{1.4}$$

$$max_{dB} = \mu_{Spec} + b \times \sigma_{Spec} \tag{1.5}$$

where μ_{Spec} and σ_{Spec} correspond to the mean and standard deviation of the logarithmic scale spectrogram, respectively, and a and b ($a \leq b$) correspond to two adjustable parameters, fixed to $a = 0.5$ and $b = 6$. The filtering is then performed according to:

$$\forall i \in [1, F], j \in [1, T], Spec_{final}(S)[i, j] = \begin{cases} min_{dB} \text{ if } Spec_{log}^{shift}(S)[i, j] < min_{dB} \\ Spec_{log}^{shift}(S)[i, j] \text{ if } Spec_{log}^{shift}(S)[i, j] \in [min_{dB}, max_{dB}] \\ max_{dB} \text{ if } Spec_{log}^{shift}(S)[i, j] > max_{dB} \end{cases}$$

This filtered logarithmic scale spectrogram is the used TFR in the rest of this work, as well as the representation computed by ADMS TCD data processing software. It is important to note that this representation was optimized by Atys Medical for emboli detection by modifying the classical STFT, where the proposed normalization improves the visual quality of the TFRs, as well as emboli detection. Therefore, the chosen parameters abovementioned were chosen accordingly.

Other time-frequency representations. Other types of TFR exist and are commonly used in the signal processing and machine learning fields.

One commonly used representation is the Mel spectrogram, which works in a different frequency scale, the Mel scale, defined as follows:

$$m = 2595 \times log_{10}(1 + \frac{f}{100})$$

where f is a frequency in Hertz and m is a frequency in Mels.

This scale tries to get closer to the non-linear way that sounds are perceived by humans, where lower frequencies are more discriminative than higher frequencies. In this way, in the Mel scale, we have more lower frequencies than higher frequencies (see figure 1.9), as more Hz frequencies are mapped into lower mel frequencies than high mel frequencies.

Moreover, from a mel spectrogram, it is possible to obtain another compressed representation, the mel-frequency cepstral coefficients (MFCC). To do this, a discrete cosine transform is applied to the mel spectrogram to decorrelate some coefficients obtained by applying the mel filter bank.

By the same token, another popular TFR based on the human ear is the cochleagram (Lyon, 2017), which is based on a gammatone filter bank. The idea of this TFR is to mimic the functioning of the cochlea, to have a representation that is similar to what is perceived by humans. To do this, a gammatone filter bank is applied to the input signal, and then the energy of each frequency channel (center frequency of each gammatone filter) is computed.

Furthermore, the chromagram, or chroma feature, is another commonly use TFR, specially for music analysis. These feature is more centered on the musical characteristics of the input signal, based on the pitch classes.

However, the main inconvenient with all the abovementioned TFRs is that they are not necessarily the ones the more adapted to HITS classification as, for humans, HITS are not easy to detect nor classify using the Doppler signal. Indeed, there are fine details in the Doppler signal that are not easy to perceive by the human ear. Thus, translating the Doppler signal into an image in the same way that humans perceive it may not be optimal.

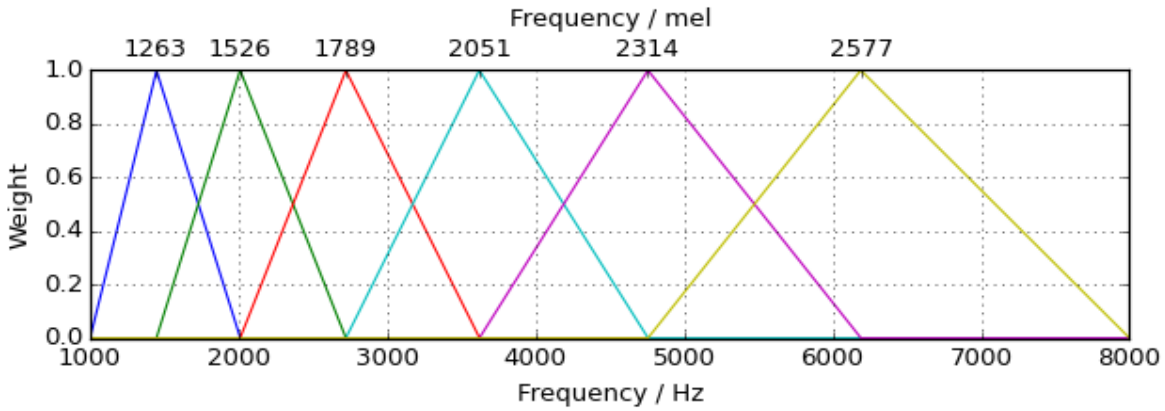


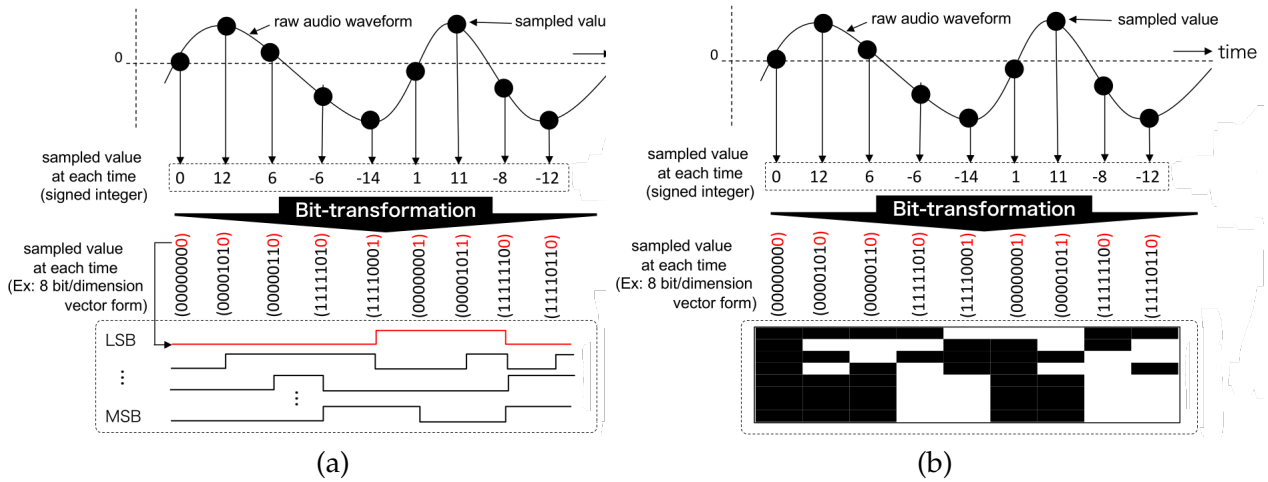
Figure 1.9: Mel filter bank. In the upper x-axis we have the frequencies in Mel and in the lower x-axis we have the frequencies in Hertz. Each filter have a triangular response, centered at a given mel frequency. Image from PyFilterbank: Mel Filter Bank — PyFilterbank devN documentation. https://siggigue.github.io/pyfilterbank/melbank.html. Referred to June 12, 2023.

II.2 Other representations

Time-frequency representations are a good way to represent signals as they make explicit the spectral characteristics of the signal. However, theoretically, the spectral information is implicitly present in the raw signal as this information is computed using the signal. Therefore, one can use other signal representations that are not frequency-based, in order to study the signal from other points of view.

Two interesting representations of a signal without loss of information are their binary encodings: bit-based pulse waveform and bit pattern image (Okawa, Saito, Sawada and Nishizaki, 2019). The first representation (bit-based pulse waveform) correspond to a binary multichannel signal (see Figure 1.10.a). If we suppose that we have a signal where the amplitudes are encoded using B bits, then, for each value $S(i), i \in [1, N]$ of the raw signal, we compute its binary encoding using B bits. Then, we concatenate the obtained binary encodings channel-wise to obtain a new signal of length N, with B channels, and with binary values. The second representation (bit pattern image) is a visual representation extracted similarly to the bit-based pulse waveform (see Figure 1.10.b). Indeed, the only difference with respect to the bit-based pulse waveform is that, the binary encodings of each element of S are concatenated column-wise, in order to form an image where the i^{th} column represents the binary encoding of $S(i)$, using B bits. Therefore, this representation is an image of height B and width N.

These representations can be interesting to exploit for emboli classification. However, few works have explored their use for signal classification (Okawa *et al.*., 2019), and some difficulties come with them. The main difficulty for both representations is the length of the representations. Indeed, for both representations, the length remains the same (N), whereas we increase the number of channels or columns, leading to larger models to process these type of inputs. Moreover, for bit pattern images, we end up with an image of shape $B \times N$, with $B << N$, with makes feature extraction more difficult (large image with wide temporal context (width) and very small amplitude dimension (height dimension).



(a) (b)

Figure 1.10: Binary representations of a temporal-dependent signal. (a) Bit-based pulse waveform. (b) Binary pattern image. Images from (Okawa *et al.*., 2019)

II.3 Emboli detection

The principles used to detect emboli are based on the two consensus of 1995 and 1998 (*Basic Identification Criteria of Doppler Microembolic Signals*, 1995; Ringelstein *et al.*., 1998), as mentioned in Section I.1.A.

First, before doing emboli detection, a 4^{th} order digital Butterworth high pass filter with a cut-off frequency of 150 Hz (3 dB attenuation) is applied to the raw Doppler signal corresponding to the recording. This allows to remove some artifacts coming from the movement of the patient.

Second, emboli are detected through HITS, which correspond to high intensity signals of brief duration, detected using the hits to blood ratio (HBR), measured in dB. The HBR can be computed sequentially, as in (Guepie *et al.*., 2018). Let us denote a $S \in \mathbb{R}^{2\times N}$ a pre-processed Doppler IQ signal of length N, f_s the sampling rate and, for all $n \in [1, N]$, $S(n) \in \mathbb{R}^2$ the sample n of S. Likewise, we denote as $B \in \mathbb{R}^{2\times N}$ the signal corresponding to the blood flow, where the first f_s values are initialized using S ($\forall n \in [1, f_s], B(n) = S(n)$) and the following values are iteratively computed. The average power, $P(n)$, of the signal at discrete time $n \in [f_s + 1, N]$ is defined as follows:

$$P(n) = \frac{1}{f_s} \times \sum_{i=0}^{f_s-1} |B(i + n - f_s)|^2$$

Note that, to compute $P(n)$ for all $n \in [f_s + 1, N]$, we only need the values $B(0), ..., B(n-1)$ and not $B(n)$. The HBR at discrete time $n \in [f_s + 1, N]$ is then defined as follows:

$$HBR(n) = 10 \times log_{10}(\frac{|S(n)|^2}{P(n)})$$

For $n \geq f_s + 1$, the HBR is computed by computing $B(n)$ and $P(n)$ iteratively:

$$B(n) = \begin{cases} B(n - f_s) \text{ if } HBR(n) \geq h \\ S(n) \text{ if } HBR(n) < h \end{cases}$$

$$P(n) = \begin{cases} P(n-1) \text{ if } HBR(n-1) \geq h \\ P(n-1) + \frac{|S(n-1)|^2 - |B(n-1-f_s)|^2}{f_s} \text{ if } HBR(n-1) < h \end{cases}$$

where h is the threshold used to detect HITS ($h > 3$ dB). Then $HBR(n+1)$ can be computed, and the process be restarted.

Moreover, all the detected HITS are not necessarily emboli, as some artifacts generated during the recording can have similar features to emboli. To further remove evident artifacts, other criteria are used based on the abovementioned consensuses:

- HBR larger than 3 dB. This allows to remove artifacts such as speckle noise. Atys Medical recommends using an intensity threshold between 7 dB and 12 dB for their TCD-X Holter[1];
- Duration smaller than 300 ms. This allows to remove an important number of voice artifacts, as they tend to be longer;
- No symmetry with respect to the baseline in the time frequency domain (horizontal 0 Hz (cm/s) value line). This allows to remove some voice artifacts as well as probe movement artifacts.

Even though the emboli detection method used by Atys Medical allows removing an important number of artifacts, we include some of them on the different HITS datasets used in this work, in order to create an independent system from ADMS for Atys Medical, which can enhance artifacts rejection.

III Challenges

III.1 Medical challenges

Different medical challenges exist when it comes to cerebral emboli, its detection, and classification with TCD devices.

First, the characterization of micro-embolic signals using TCD has not been largely studied in the medical domain. All microembolic signals do not have the same TCD signature, therefore, there may exist different subtypes of emboli (within the solid and gaseous type). These subtypes of emboli may be characterized by the pathologies of the patients (carotid stenosis, AF, etc.), their treatments (coagulants, shots, etc.), the recording conditions (ambulatory patient, surgical intervention, diving, etc.), or the position in the cardiac cycle when the micro-embolic signal occurs. This characterization is important and interesting from a medical point of view, as it can help clinicians to better understand emboli and their origin, allowing a better prevention and management of patients.

By the same token, one important challenge related to emboli characterization, is the origin of the embolus. Indeed, based on the origin of the emboli, the treatment, and the care unit handling the patient, may not be the same. For instance, patients with cerebral emboli originated in the heart (e.g., AF or mitral annular calcification patients) are mainly treated by cardiovascular units, whereas patients with emboli originated near the brain (e.g., atheromatous plaques in cerebral arteries) are mainly treated by neurovascular units. Even though it is not possible to accurately determine the origin of an embolus with the current TCD devices and emboli detection methods, exploiting the whole long-duration TCD recording can guide clinicians to determine their origin as mentioned in Section I.1.B.

Finally, another important medical challenge concerns the detection of gaseous emboli and its distinction with respect to solid emboli. Indeed, current methods allowing to do this are

[1]In the official documentation of the TCD-X, the recommended thresholds are 9 dB to 15 dB. However, the research and development engineers of Atys medical recommend 7 dB to 12 dB, which is more adapted for pathologies such as carotid stenosis.

based on non-portable TCD devices, often equipped with two probes allowing an emission at two different frequencies, 2 MHz and 2.5 MHz (Russell and Brucher, 2002; Markus and Punter, 2005), which facilitates classification between solid and gaseous emboli as the relative intensity of both types of HITS will differ based on the insonation frequency (solid emboli reflect more at higher frequencies, whereas gaseous emboli are stronger reflectors for lower frequencies). Additionally, (Markus and Punter, 2005) showed that the proposed TCD device and emboli detection method do not allow an accurate enough distinction between solid and gaseous emboli to be usable in clinics. Moreover, current portable TCD devices do not allow any distinction between solid and gaseous emboli. From a medical point of view this is important, as the treatments are not the same for solid or gaseous emboli as seen in I.1.C. Additionally, an accurate distinction between the two types of emboli will allow more clinical studies on gaseous emboli to determine their consequences on public healthcare.

III.2 Scientific challenges

The main objective of this work is to develop novel deep learning (DL) techniques to improve the analysis (detection/classification) of cerebral emboli monitored by portable TCD devices. Therefore, a portable TCD signals database is necessary to train the different models, allowing their study. However, there is no publicly available database of portable TCD signals. Therefore, in order to develop DL models, these databases need to be created.

Moreover, several challenges arise when creating a portable TCD database. First, annotation of HITS is not an easy task, and even clinicians or experts can face difficulties classifying some HITS between solid embolus, gaseous embolus and artifacts. This creates noisy labels, as HITS can be wrongly labeled. Second, with portable TCD long duration recordings, an important amount of HITS can be generated (it can go to more than 10 000 HITS per patient for some specific procedures such as heart surgery, but on average there are 14 HITS per minute), so it becomes nearly impossible to manually label all the data. Third, classes are highly imbalanced, as solid emboli are rare events (while being the riskiest ones for the patient), and gaseous emboli and artifacts are more common. All these difficulties associated to the creation and annotation of a portable TCD signals dataset can be observed in Figure 1.11, where we projected 68 491 HITS on a 2D plane, using a dimensionality reduction technique (t-distributed stochastic neighbor embedding, t-SNE (Maaten and Hinton, 2008)). In this figure, we can note that we have a very important number of HITS, with the vast majority of the HITS unlabeled (purple points), whereas the solid emboli (teal points) are a minority compared to artifacts (red points) and gaseous emboli (yellow points).

Furthermore, as we work with medical signals, we have a temporal aspect that we need to take into account. Indeed, inside a signal, there is a temporal dependency between the different elements composing it. This means that, for all $i, j \in [1, N]$, such that $i < j$, $S(i)$ may have an influence on $S(j)$. Therefore, the different models that are used to study these signals have to take this temporal dependency into account. Moreover, as we are working with signals representing a physical phenomenon evolving over time, the question of which type of representation of this signal is optimal for a specific task raises. As we saw in Section II.1, a same signal can be represented under different forms, which can be complementary. Based on the used representation, the used model is not necessarily the same. For instance, for raw signals, 1D convolutional neural networks (CNNs), long short term memory (LSTM) networks, or transformers models can be used, whereas for time-frequency representations can be interpreted as images, thus 2D CNNs can be used. What is more, contrary to image classification, in DL there is not proven global type of model able to achieve good performances for different types of signals with different characteristics, contrary to images, where 2D CNNs achieve globally great performances in different tasks. This is related to the fact that temporal

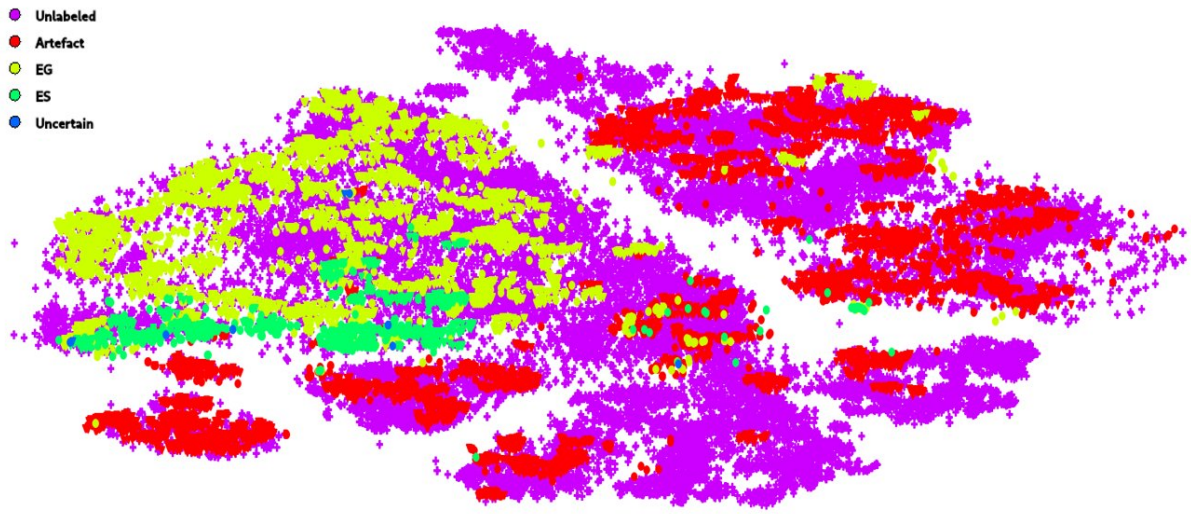


Figure 1.11: 2D projection of 68 491 HITS, using t-distributed stochastic neighbor embedding. Over all the available HITS, only 8 685 (13%) are labeled. Moreover, the dataset is highly imbalanced, as the solid emboli represent less than 10% of all the labeledHITS.

dependent signals have additional characteristics to images: long sequences, multiple channels, different representations, etc. This point will be further discussed in Chapter 2.

Finally, one important remaining scientific and industrial challenge (as we work in collaboration with a company) is the memory, energy, time, and computation resources needed by the developed models. Indeed, it is well known that deep learning models tend to be over-parametrized, often having from hundreds of thousands to hundreds of millions of parameters, requiring important memory resources, as well as important computation resources and high levels of energy consumption (Lane, Bhattacharya, Georgiev, Forlivesi and Kawsar, 2015). This makes their use in embedded systems and clinical practice difficult, as they have often limited resources. In the particular case of TCD portable devices, it is important to develop lite models requiring the least amount of memory and computation resources possible. If this is not taken into account, the developed models can be difficult to use in practical applications. By the same token, related to the used computation resources, DL models can have large inferences times per samples, which can also limit their use in practice, not only for embedded devices. Indeed, one patient undergoing a TCD examination can have thousands of HITS, which need to be analyzed in a few minutes to be usable by clinicians.

IV Conclusion

In this chapter, we have provided an overview of the medical and scientific context of cerebral emboli classification using transcranial Doppler ultrasonography.

From a medical point of view, emboli are solid or gaseous particles that can circulate in the cerebral blood flow and create stroke or transient ischemic attacks by blocking an artery supplying blood to the brain. Different sources can generate cerebral emboli, such as pathologies, medical procedures, or external sources. Transcranial Doppler (TCD) ultrasonography is the only relatively cheap and noninvasive technique allowing a real-time monitoring of the cerebral blood flow. Thus, it is one of the most used and promising techniques for cerebral emboli detection and classification.

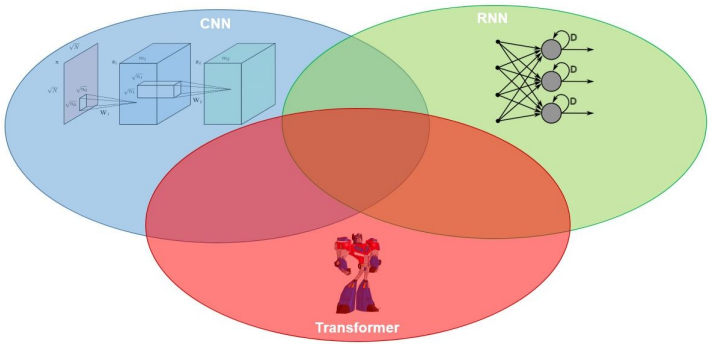
Furthermore, from a scientific point of view, the data acquired by TCD devices are represented as signals with a temporal dependency. Therefore, this temporal dependency has to be taken into account when studying cerebral emboli using TCD. What is more, it is important to choose a good representation to study these signals, as the different representations do not show the same information.

Finally, to make the developed models usable in practice, it is important to take into account their memory and energy consumption footprint, as well as the needed computation resources and inference times.

In the following chapter, we shall review the state-of-the-art related to the abovementioned challenges.

2 State-of-the-art

This chapter presents a brief overview about the different topics related to my work. First, in Section I we focus on cerebral emboli detection and classification using TCD signals and three families of techniques (signal processing, machine learning and deep learning). Afterwards, in Section II we introduce semi-automatic data annotation methods allowing to label large datasets, as well as noisy-labels resistant methods to perform classification with the obtained datasets. Then, in Section III we explore some works on multi-feature signal classification, allowing to take advantage of the complementarity of different representations of a signal. Finally, in Section IV we overview some model compression methods allowing to reduce the memory, computation, and energy resources of different deep learning (DL) models.



Contents

I Cerebral emboli detection and classification

I.1 Signal processing and machine learning

I.1.A General overview

Cerebral emboli detection and classification using TCD ultrasonography have been largely studied since the 1990s. Different approaches have been taken to do this, based on different fields: signal processing, machine learning and deep learning.

The first approaches, which are still being used nowadays, are mainly based on signal processing handcrafted features such as Fourier or wavelet transforms (Markus and Punter, 2005; Keunen *et al.*, 2008; Uğuz, 2012; Gencer, Bilgin and Aydin, 2013; Serbes and Aydin, 2014; Karahoca and Tunga, 2015; Sombune, Phienphanich, Muengtaweepongsa, Ruamthanthong and Tantibundhit, 2016; Sombune, Phienphanich, Phuechpanpaisal, Muengtaweepongsa, Ruamthanthong, Chazal and Tantibundhit, 2018), and mainly focus on artifact vs. emboli classification, without distinction between solid and gaseous emboli. (Keunen *et al.*, 2008) proposed to extract different feature from the Doppler signal to then pass them through a neural network. These features are mainly based on statistical characteristics of the signal and its zero-crossings (mean duration, intensity, and ZCI, maximum intensity, minimum ZCI, number of zero-crossings). (Gencer *et al.*, 2013) showed that features extracted from well-designed time-frequency representations (TFRs) can help cerebral emboli detection. They suggest using the fractional Fourier transform (FrTF), which can be seen as a generalization of the classical Fourier transform, to extract handcrafted features such as the HBR, the quotient between the maximum value and the mean value of the FrTF, etc. (Serbes and Aydin, 2014) proposed to improve the HBR computation by using modified dual-tree complex wavelet transform, which can denoise Doppler signal and has the advantage of being relatively invariant against shifting. This is an important step as the HBR is used in most of the cerebral emboli detection and classification methods based on TCD signals. (Karahoca and Tunga, 2015) proposed to classify HITS between embolus (no distinction between solid and gaseous), artifacts, and Doppler speckle, using a polynomial based algorithm. (Sombune *et al.*, 2016, 2018) introduced a cerebral emboli detection and classification system (no distinction between solid and gaseous) based on adaptive gain control (AGC), adaptive wavelet packet transform, and adaptive neuro-fuzzy classifier. At last, (Markus and Punter, 2005) focused on solid vs. gaseous emboli classification, thanks to a dual frequency TCD capable of insonating at two ultrasound frequencies (2 MHz and 2.5 MHz). As a result of this, they obtain two signals, one for each frequency, so they can compute the difference between intensity increases (with respect to the background signal) of these two signals. This value is then used to detect and classify possible and definite solid and gaseous emboli.

I.1.B Related work

More recently, different machine learning approaches have been combined with signal processing approaches, to enhance the detection and classification of cerebral emboli using portable TCD data (Imaduddin, LaRovere, Kussman and Heldt, 2019; Guépié, Sciolla, Millioz, Almar and Delachartre, 2017; Guepie *et al.*, 2018). Moreover, to our knowledge, only the last two works (Guépié *et al.*, 2017; Guepie *et al.*, 2018) used data acquired with portable TCD devices, which are more prone to artifacts. First, (Imaduddin *et al.*, 2019), proposed a time-frequency based approach for cerebral emboli detection and characterization using TCD data. To start, a weighted-frequency Fourier linear combiner estimates the power of the Doppler signal and classifies it as a HITS if a certain adaptive threshold is exceeded (this threshold depends on

the power variance with respect to the baseline, and it is not the same as the HBR). Then, the obtained HITS is separated to obtain individual potential embolic signals, as often they can superimpose. This is done by extracting a TFR from the signal (wavelet transform), processing the TFR as an image to do morphological segmentation, and then going back to the time domain. Finally, from each individual embolic signal, different time and spectral features are extracted, and then passed through a logistic regression classifier.

The method proposed in (Guépié *et al.*, 2017) does HITS detection and classification between artifact and embolus, based on signal processing (detection, pre-processing and feature extraction) and machine learning (classification) techniques. Compared to previous methods for HITS detection and classification (Gencer *et al.*, 2013; Serbes and Aydin, 2014), the proposed method is non-parametric as no hypothesis are made about the probability distribution of the background noise. First, HITS are detected using a TFR of the initial quadrature Doppler signal and a two hypothesis statistical test where a pixel value of the TFR is considered as the realization of a random variable. Then, the average blood-flow magnitude is estimated using spectral kurtosis, in order to get rid of parts of the Doppler signal/TFR which do not contain a HITS. Afterwards, evident artifacts are removed (symmetric, small magnitude, small surface, small number of pixels), based on the average magnitude of the HITS and bounding boxes around them. Finally, a classification of the remaining HITS is done using a support vector machine (SVM), taking as input the ratio between the height and the width of the HITS bounding boxes, and the magnitude of the HITS. This method showed impressive results, allowing to reduce the number of artifacts predicted as emboli by 92% between the first and last steps. Moreover, the same team introduced another approach in (Guepie *et al.*, 2018), also based in signal processing and machine learning, for automatic sequential HITS detection and classification. First, sequential calculation of the HITS-to-blood ratio (HBR) on the quadrature Doppler signal is done as specified in Chapter 1: this HBR is used to select potential HITS based on a prdefined threshold, fixed to 3 dB using the Elbow method. Then a log-magnitude spectrogram is computed, and regions of interest (ROIs) are automatically extracted. These ROIs are computed only on the positive frequencies and are nonsymmetrical. Once this is done, handcrafted features are extracted using the HITS and the selected ROI (see Figure 2.1). Finally, these features are fed to a machine learning classifier (SVM, naive Bayes and decision tree). This approach was able to divide the number of artifacts detected as emboli by a factor of more than 10 compared to previous methods (Karahoca and Tunga, 2015; Guépié *et al.*, 2017), passing from 6 detected artifacts per minute in (Guépié *et al.*, 2017) to 0.6, for the same amount of detected emboli.

I.2 Deep learning

With the raising of deep learning techniques in the past years, these techniques have also been explored for TCD signals classification (Seera, Lim, Tan and Liew, 2017; Sombune *et al.*, 2017; Tafsast, Ferroudji, Hadjili, Bouakaz and Benoudjit, 2018).

I.2.A General overview

The work done in (Seera *et al.*, 2017) proposed to use recurrent neural networks (RNNs) and ensemble methods to classify TCD signals between two classes: (1) signals from patients with intracranial arterial stenosis, and (2) signals from patients without intracranial arterial stenosis. To do this, they acquired seven TCD signals coming from different sources: anterior cerebral artery, basilar artery, internal carotid artery, middle cerebral artery, ophthalmic artery, intracranial vertebral artery, and carotid siphon. Then, they extracted five features per signal (depth, upper and lower mean flow velocities, and the upper and lower pulsatility index), and

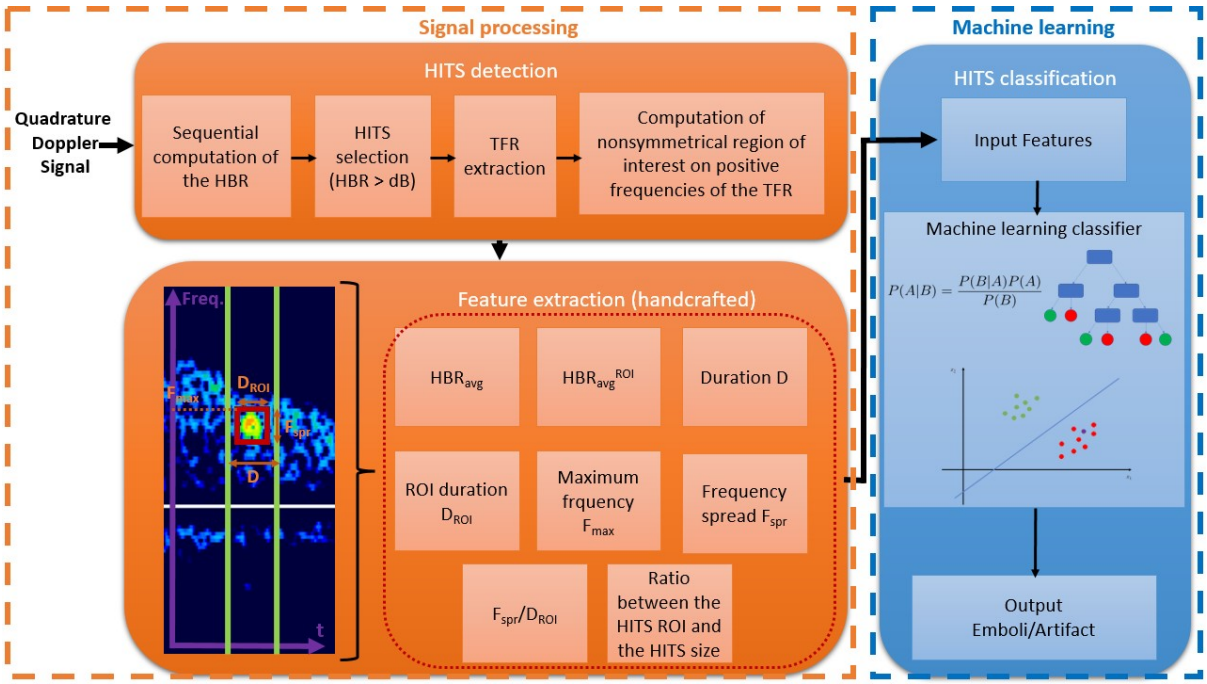


Figure 2.1: HITS detection and classification pipeline proposed in (Guepie *et al.*, 2018). The used features are not learned, but manually designed.

passed them thought a RNN. This work showed the interest of deep learning for TCD signal classification, but it is out of the scope of our work as different TCD signals (not only the one of the MCA) were used and no emboli detection nor classification was done.

I.2.B Related work

What is more, (Sombune *et al.*, 2017) suggest to use convolutional neural networks (CNNs) to classify TCD signals between three classes: (1) embolic signals ES (no distinction between solid and gaseous), (2) artifacts, and (3) normal interval (NI) signal. To do this, the authors proposed to used AGC to detect the HITS, and then compute their spectrogram. Afterwards, the obtained spectrograms were transformed into grayscale images to be fed to the proposed CNN. The CNN architecture was relatively simple, composed of two 5×5 convolutional layers with 48 filters, followed by a 2×2 max-pooling layer for feature extraction, and two fully-connected (FC) layers for classification (see Figure 2.2). However, their work did not outperform their previous ANFIS-based method, which was justified by the small number of samples for training and the simple used architecture due to resource limitations. Additionally, this paper did not take into account gaseous emboli for the classification task. To our knowledge, the only work doing this is the work of (Tafsast *et al.*, 2018), but this was done with in vitro TCD acquired data[1].

We can see that, even though much effort has been dedicated to improve cerebral emboli detection and classification with TCD data, most of the techniques are based on classical signal processing and machine learning approaches which require time-consuming handcrafted features. Moreover, most of these works do not use portable TCD data, and none does *in vivo* distinction between solid and gaseous emboli. Exploring more complex deep learning methods could help do a better emboli discrimination with automatic feature extraction, as well as allow

[1]The acquired gaseous emboli were obtained with a phantom under controlled conditions, which is not the case with *in vivo* data.

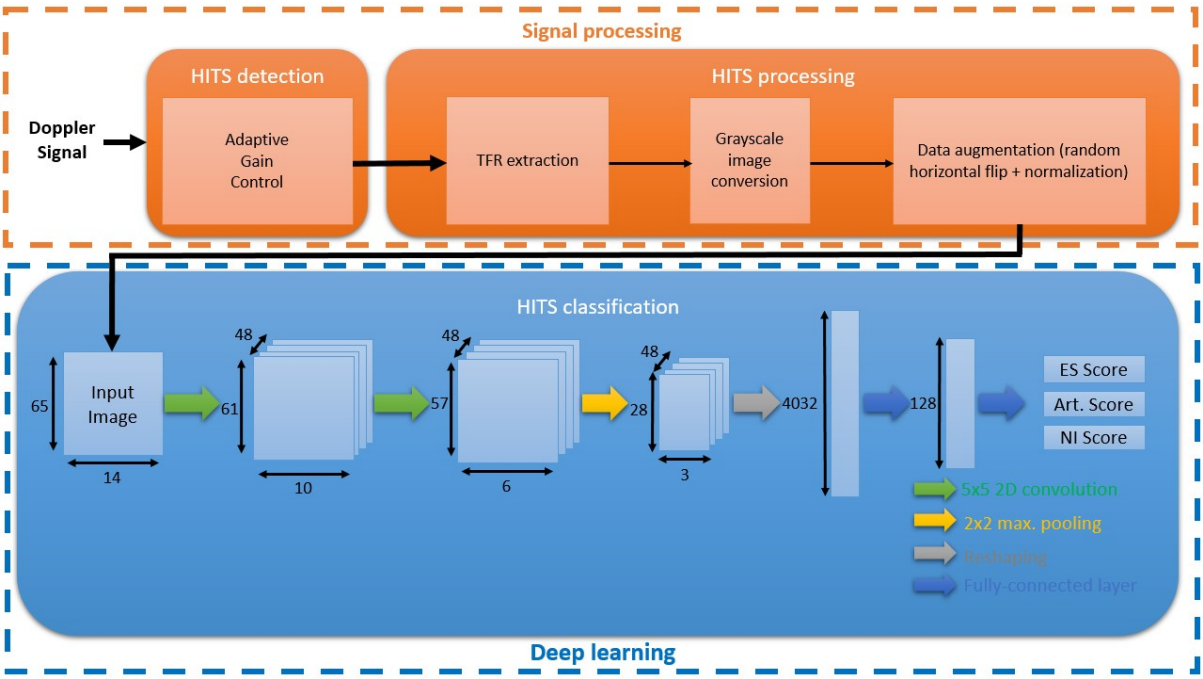


Figure 2.2: A CNN-based method used for cerebral emboli classification in (Sombune *et al.*., 2017). ES stands for embolic signal, Art. for artifact, and NI for normal interval.

and improve solid vs. gaseous emboli classification.

II Data annotation

Data acquisition and annotation is one of the main difficulties in several deep learning tasks. This is particularly true in the medical domain, where annotation is costly and difficult to perform. However, these steps are essential in the development of deep learning methods and models as they are often data-hungry, needing important amounts of samples to achieve good generalization.

II.1 Representation learning and semi-automatic data annotation

II.1.A General overview

Different semi-supervised learning methods have tried to tackle the data annotation problem based on the *structure assumption* (Chapelle, Scholkopf and Zien, 2009), which states that samples that are in the same structure (cluster or manifold), are likely to have the same label. Therefore, techniques such as self-training (Rosenberg, Hebert and Schneiderman, 2005), generative models (Kingma, Rezende, Mohamed and Welling, 2014), or label propagation (Zhu and Ghahramani, 2002; Weston, Ratle, Mobahi and Collobert, 2012; Benato, Telea and Falcao, 2018; Benato *et al.*., 2021), have been used for semi-automatic data annotation.

(Rosenberg *et al.*., 2005) iteratively annotate a dataset using a classification model that is trained multiple times. In addition, at each iteration, the classifier is trained with the new labeled dataset, composed of the originally labeled samples, plus the new ones annotated with the classifier from the previous iteration. (Kingma *et al.*., 2014) used a different approach, where the labels of the unlabeled samples are considered as latent variables which can be generated

using a learned distribution, obtained through generative models. (Zhu and Ghahramani, 2002) approach the problem differently by propagating the labels from the labeled samples to the unlabeled ones, based on a K-nearest neighbor (KNN) strategy. This last method can be related to Laplacian support vector machine (LapSVM) (Sindhwani, Niyogi and Belkin, 2005; Belkin, Niyogi and Sindhwani, 2006) using the formalism of (Weston *et al.*., 2012). Let us suppose that we have a dataset $\mathcal{D} = \{(X_1, y_1), ..., (X_A, y_A), (X_{A+1}, y_{A+1}), ..., (X_{A+U}, y_{A+U})\}$ composed of a large number of unlabeled samples, $\mathcal{U} = \{(X_{A+1}, y_{A+1}), ..., (X_{A+U}, y_{A+U})\}$, and a small number of annotated samples, $\mathcal{A} = \{(X_1, y_1), ..., (X_A, y_A)\}$ (usually $|\mathcal{A}| << |\mathcal{U}|$, and $y_{A+1}, ..., y_{A+U}$ are unknown). Let us denote as $W \in \mathbb{R}^{(A+U)\times(A+U)}$ an adjacency matrix where, for all $i, j \in [1, A+U]$, W_{ij} measures the similarity (or dissimilarity) of sample X_i with sample X_j. We aim to find a function f, mapping each point $X \in \mathcal{D}$ to an embedding $f(X)$, which usually is a predicted label. We can then compare label propagation and LapSVM by the loss function to optimize:

$$\min_f = \sum_{i=1}^{A} ||f(X_i) - y_i||^2 + \lambda_I ||f||_I^2 \qquad \text{(Label propagation)} \tag{2.1}$$

$$\min_f = \sum_{i=1}^{A} F(y_i, f(X_i)) + \lambda_{Am} ||f||_{Am}^2 + \lambda_I ||f||_I^2 \qquad \text{(LapSVM)} \tag{2.2}$$

where:

- F is a given loss function, often the Hinge loss $H(y_i, f(X_i)) = max(0, 1 - y_i \times f(X_i))$ or the squared error $SE(y_i, f(X_i)) = (y_i - f(X_i))^2$.
- $||f||_I^2 = \sum_{i,j=1}^{A+U} w_{ij} \times ||f(X_i) - f(X_j)||^2$ is the intrinsic regularizer and λ_I its associated hyperparameter.
- $||f||_{Am}^2$ is the ambient regularizer and λ_{Am} its associated hyperparameter.

We can note in the previous formalism that both methods are similar as they both exploit the similarity between samples, whether they are labeled or not, thanks to the intrinsic regularizer.

II.1.B Related work

More recent work has mainly focused on deep learning methods for semi-automatic data annotation through label propagation (Benato *et al.*., 2018, 2021). These works, based on representation learning techniques, use auto-encoders (AE) to extract features from high-dimensional data, as they have been proven effective for this task (Doersch, Gupta and Efros, 2015; Chen, Shi, Zhang, Wu and Guizani, 2017; Zhang, Isola, Efros, Shechtman and Wang, 2018). The core idea of these works is the following (see Figure 2.3). The first step consists in extracting features of the data, using all the available samples from $\mathcal{D}$. Even though this step allows reducing the dimensionality of the original samples, the dimensionality remains too important to be easily usable by human annotators. Therefore, an additional step of dimensionality reduction is used, based on t-distributed stochastic neighbor embedding (t-SNE) (Maaten and Hinton, 2008), allowing to project the extracted features (previous step), into a 2D space. This space is then used for semi-automatic annotation through label propagation. Indeed, the user can select the regions to propagate the labels from the labeled to the unlabeled samples, or directly annotate some samples. Moreover, in this last step, automatic label propagation is done using two different methods: LapSVM or semi-supervised optimum-path forest (OPF-semi) (Amorim, Falcão and Carvalho, 2014). OPF-semi is selected to compute the confidence values defining if a sample should be manually labeled or not. This confidence score is computed as

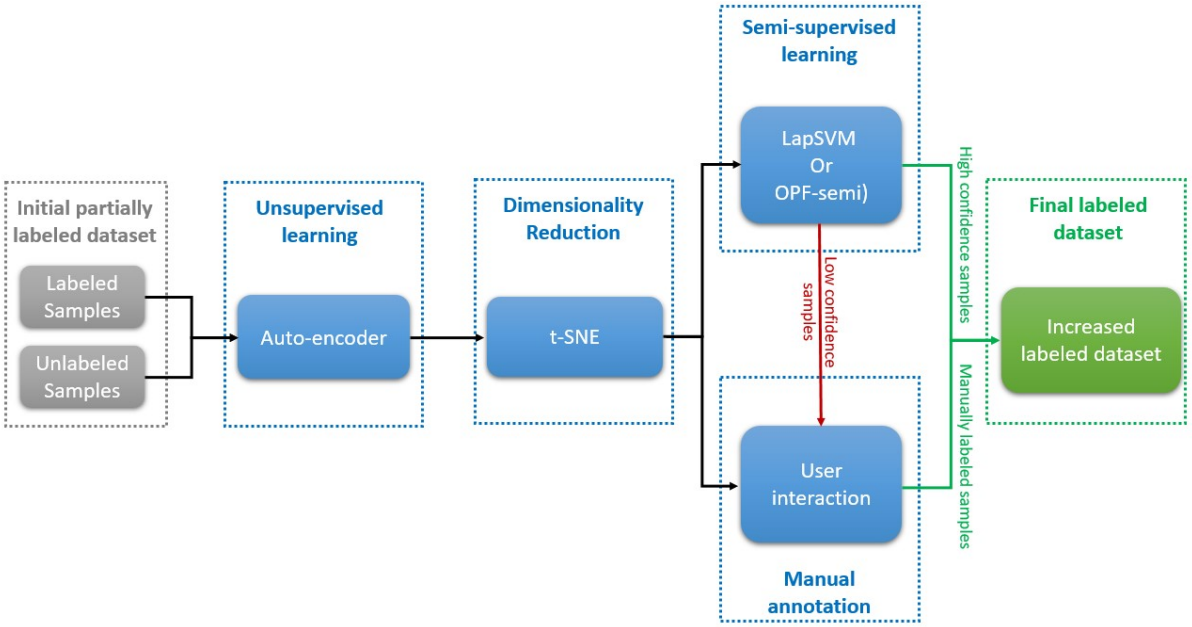


Figure 2.3: Semi-automatic data annotation method proposed by (Benato *et al.*., 2021)

follows. Let us suppose that we have a cost function $\mathcal{L}$ and that all the 2D samples are mapped as nodes in a graph $G = (V, E)$, where V is the set of nodes and E the set of edges. A node (sample) $s \in V$ is considered as conquered (labeled) by the root $R \in V$ (a labeled sample) if it leads to a minimum cost $\mathcal{L}(s, R)$ (i.e. $\forall P \in V, \mathcal{L}(s, R) \leq \mathcal{L}(s, P)$). Therefore, if we denote as $R_1, R_2 \in V$ two roots such that $\forall P \in V, \mathcal{L}(s, R_1) \leq \mathcal{L}(s, R_2) \leq \mathcal{L}(s, P)$, then the confidence score, $c(s) \in [0, 1]$, is defined as follows

$$c(s) = 1 - \frac{\mathcal{L}(s, R_1)}{\mathcal{L}(s, R_1) + \mathcal{L}(s, R_2)} \tag{2.3}$$

where $c(s) = 0$ indicates a low confidence in the annotation, whereas $c(s) = 1$ indicates high confidence. In that way, if the cost $\mathcal{L}(s, R_2)$ is larger than $\mathcal{L}(s, R_1)$ (i.e. the second optimal path to s is much longer than the optimal path), then the sample is confidently labeled ($c(s) \approx 1$)[1]. Then, using a pre-defined threshold, we can define high or low confidence samples. Based on this, if the sample is of high confidence, it is automatically annotated, if not, it can be annotated by an expert.

However, the approach of (Benato *et al.*., 2021) has two main limitations: (1) it does not take into account the quality of the embedding space used for semi-automatic data annotation, and (2) for the classification task, it does not take into account the noise introduced in the labels due to automatic annotation errors.

II.2 Dimensionality reduction quality evaluation

II.2.A Related work

The first limitation of the work in (Benato *et al.*., 2021) can be tackled using metrics allowing to measure the global and local quality of the projection of a high dimensional space into a lower

[1]On the contrary, if the cost $\mathcal{L}(s, R_2)$ is relatively close to $\mathcal{L}(s, R_1)$ (i.e. the second optimal path to s has a similar length than the optimal path), then the sample is labeled with low confidence ($c(s) \approx 0$).

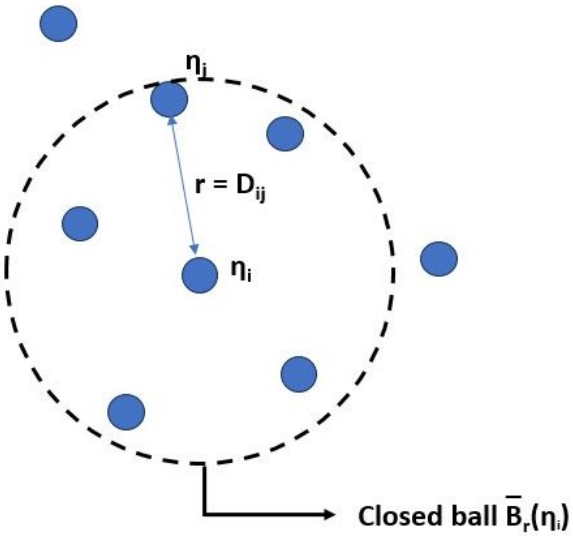


Figure 2.4: Illustration of how ranks are defined in the co-ranking framework of (Lueks *et al.*., 2011). In this example, the rank R_{ij} of sample i with respect to sample j is equal to 5 as there are five neighbors in the closed ball $\overline{B}_r(\eta_i)$ of center on η_i and of radius $r = D_{ij}$.

dimensional one (Lueks *et al.*., 2011). These metrics are often based on the co-ranking matrix, which translates the preservation of the samples' neighborhoods when the dimensionality reduction (DR) is done. We are going to build on (Lueks *et al.*., 2011) to introduce the co-ranking framework.

Suppose that we have a set of high-dimensional samples $H = \{\eta_1, \eta_2, ..., \eta_N\}$ with for all $i \in [1, N], \eta_i \in \mathbb{R}^M$, and a lower dimensional representation set $\Lambda = \{\lambda_i, i \in [1, N], \lambda_i = f(\eta_i) \in \mathbb{R}^m\}$ obtained by applying a DR mapping $f : \mathbb{R}^M \to \mathbb{R}^m$ ($m << M$). Let us denote for all $i, j \in [1, N]$, D_{ij} the distance between η_i and η_j, and d_{ij} the distance between λ_i and λ_j. We can now define the rank R_{ij} of sample η_j with respect to sample η_i as follows:

$$\forall i, j \in [1, N], R_{ij} = |\{\eta_k, D_{ik} \leq D_{ij} \text{ and } k \in [1, N] \backslash j\}| \tag{2.4}$$

In other terms, R_{ij} denotes the number of samples that are closer to η_i than η_j (i.e. number of samples within the closed ball $\overline{B}_r(\eta_i)$ of center η_i and of radius $r = D_{ij}$, see Figure 2.4). We denote for all $i, j \in [1, N]$, r_{ij} the rank of sample λ_j with respect to sample λ_i (in the lower dimensional space)[1]. We can now define the co-ranking matrix $Q \in \mathbb{R}^{(N-1)\times(N-1)}$, where each element Q_{ij} corresponds to the number of points that have a rank of i in H and a rank of j in Λ. More formally:

$$\forall i, j \in [1, N-1], Q_{ij} = |\{(n, p), R_{np} = i \text{ and } r_{np} = j\}| \tag{2.5}$$

This co-ranking matrix gives a general overview about the neighborhood errors made by f when reducing the dimension of H:

- Ideally, if all the neighborhoods of H were preserved by f when obtaining Λ, then Q is a diagonal matrix;
- If $i < j$, then we have an *extrusion* as for some samples the rank increased when applying f[2];

[1]It is important to note that the ranks are not symmetric as, for instance, the rank of η_j with respect to η_i can be 1 but the rank of η_i with respect to η_j can be greater than 1. This is true in the high and low dimensional spaces.

[2]This means that, for some samples in H, some of its neighbors got further away from them in Λ, because some other samples got closer in Λ.

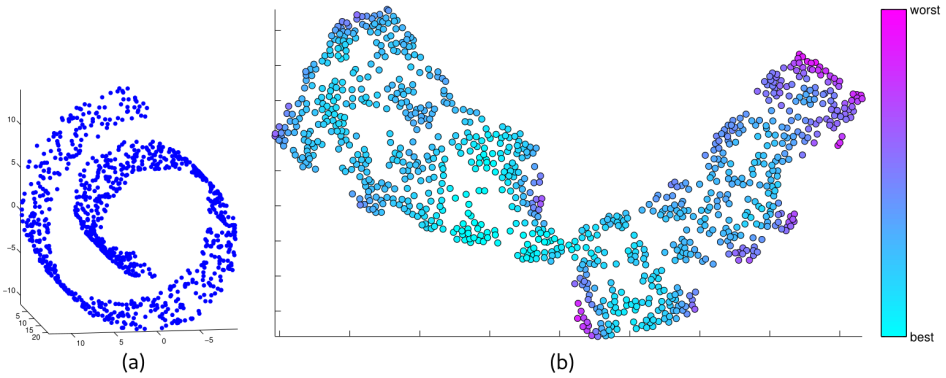


Figure 2.5: Illustration of the local quality metrics on the Swiss roll benchmark dataset from (Lueks *et al.*., 2011). (a) Original space in $\mathbb{R}^3$. (b) Unrolled embedded 2D space obtained by t-SNE. The points in (b) are colored by local quality. We can notice that the points that were in the center of the Swiss roll in $\mathbb{R}^3$ are the lowest local quality points in $\mathbb{R}^2$, as those are the points where the neighborhood changed the most.

- If $i > j$, then we have an *intrusion*, as for some samples the rank decreased when applying f[1].

Moreover, using this co-ranking matrix (through the ranks), we can define two metrics, one global Q_G and one local Q_L^i for $i \in [1, N]$, measuring how well the neighborhood of the samples was preserved during the projection by f:

$$Q_G(k_s, k_t) = \frac{1}{k_s \times N} \times \sum_{i,j=1}^{N} \mu_t(R_{ij}, r_{ij}, k_t) \times \mu_s(R_{ij}, r_{ij}, k_s) \tag{2.6}$$

$$Q_L^i(k_s, k_t) = \frac{1}{2 \times k_s \times N} \times \sum_{j=1}^{N} (\mu_t(R_{ij}, r_{ij}, k_t) \times \mu_s(R_{ij}, r_{ij}, k_s) + \mu_t(R_{ji}, r_{ji}, k_t) \times \mu_s(R_{ji}, r_{ji}, k_s)) \tag{2.7}$$

where μ_t and μ_s are two functions indicating the rank error tolerance (size of the tolerated rank errors) and rank significance (rank errors to be considered), respectively, and k_t and k_s are two hyperparameters of these functions[2]. These functions are defined for all $i, j \in [1, N]$ as follows:

$$\mu_t(R_{ij}, r_{ij}, k_t) = \begin{cases} 1 \text{ if } |R_{ij} - r_{ij}| \leq k_t \\ 0 \text{ else} \end{cases} \tag{2.8}$$

$$\mu_s(R_{ij}, r_{ij}, k_s) = \begin{cases} 1 \text{ if } R_{ij} \leq k_s \text{ or } r_{ij} \leq k_s \\ 0 \text{ else} \end{cases} \tag{2.9}$$

Using these metrics, it is possible to evaluate whether the obtained 2D space and/or samples can be used for label propagation (low local quality samples are not reliable, see Figure 2.5).

[1]This means that, for some samples in H, some of its neighbors got closer in Λ, by replacing neighbors that were closer in H.

[2]The influence of k_t and k_s will be studied in Chapter 3.

II.3 Working with noisy-labels

II.3.A General overview

The second limitation of (Benato *et al.*., 2021) can be handled using robust methods against noisy labels. Different approaches exist allowing to train deep learning classification models with noisy-labels. We are going to build on (Song, Kim, Park, Shin and Lee, 2022) to do a quick overview of the different concepts and approaches.

First, a set of labels is considered as noisy if we are not certain that they are correct (i.e. some *noise* could be present in the labels). The typical approach for modeling noise in labels makes the assumption that the noise is independent of the data features when the true label is given. Formally, we can define a *label transition matrix* T where $T_{ij} := p(\tilde{y} = j|y = i)$ is the probability that a true label $y = i$ is corrupted to become a noisy label $\tilde{y} = j$. From this, we can define two types of noise:

- *Symmetric* or *uniform*: A true label can be corrupted to another label with the same probability. If we have a noise rate $\zeta \in [0, 1]$ and a set of labels $C = [1, ..., C]$, then

$$\forall i, j \in [1, C] T_{ij} = p(\tilde{y} = j|y = i) = \begin{cases} 1 - \zeta \text{ if } i = j \\ \frac{\zeta}{C-1} \text{ otherwise} \end{cases}$$

- *Asymmetric* or *label-dependent*: A certain label has a higher probability of being corrupted to some labels than to others. In other words, there exists three different labels i, j and k such that $T_{ij} > T_{ik}$. In our application, we will have to model the label noise using this type of noise, since solid emboli are more likely to be mislabeled into gaseous emboli than artifacts.

In the following paragraphs we present different approaches that can be used to handle noisy-labels: robust architectures, robust regularization, loss adjustment, sample selection, meta-learning, semi-supervised learning, and robust loss functions.

Robust architectures: The main idea is to model the label transition matrix of a dataset using the architecture of a neural network (NN). The two main methods are: noise adaptation layer (Goldberger and Ben-Reuven, 2017) and dedicated architectures (Xiao, Xia, Yang, Huang and Wang, 2015). The former adds a layer in order to mimic the noise process of the labels, whereas the latter uses dedicated architectures to learn the label transition matrix, for instance, using two networks, one to predict the noise type (e.g., noise free, random noise, etc.) and another one to model the label transition matrix.

Robust regularization: Regularization techniques are known to be simple and efficient approaches to avoid overfitting. Some classical regularization methods such as data augmentation, L2 penalty, dropout, and batch normalization can be used to improve the classification performances when working with noisy-labeled data. Other techniques such as adversarial training (Goodfellow, Shlens and Szegedy, 2015), label smoothing (Pereyra, Tucker, Chorowski, Kaiser and Hinton, 2017) or mixup (Zhang, Cisse, Dauphin and Lopez-Paz, 2018) have been introduced to enhance the training process with noisy-labels. Adversarial training (Goodfellow *et al.*., 2015) gives to the model original inputs samples and perturbed input samples (without perturbing the labels) during the training process in order to encourage the model to be resistant to perturbations and therefore to eventual noise. Label smoothing (Pereyra *et al.*., 2017) tries to estimate the effect of noisy-labels during training and then improves the generalization capability of the model by avoiding the model to be confident on noisy training samples. Mixup

(Zhang, Cisse, Dauphin and Lopez-Paz, 2018) encourages a linear behavior between training samples by constructing a new training dataset composed of linear combinations of the original noisy-training samples (both the samples themselves and the labels are used to construct the new samples).

Loss adjustment: The idea is to reduce the negative influence of noisy labeled samples by adjusting them before updating the weights of the model. There are three main categories: loss correction (Patrini, Rozza, Menon, Nock and Qu, 2017), loss reweighting (Wang, Liu and Tao, 2018), and label refurbishment (Reed, Lee, Anguelov, Szegedy, Erhan and Rabinovich, 2015). Loss correction tries to estimate the label transition matrix to correct the wrongly labeled samples before the forward or backward passes. Loss reweighting assigns different weights to different samples: greater weights are assigned to samples with correct labels and smaller weights to samples that are potentially wrongly labeled. Thanks to this, the wrongly labeled samples will have a little influence on the learning process of the model. Label refurbishment adjusts the labels of the samples by creating new labels as a convex combination of the original label and the predicted label by the model.

Sample selection: The main idea is to select clean data samples (i.e. with good labels) to avoid introducing noise to the learning process. Because we want to exploit all the data that we have, we did not investigate deeply this family of methods.

Meta-learning: Meta learning aims at creating models that can quickly adapt to different tasks and that can learn some meta-parameters by themselves (Gaier and Ha, 2019). This can be used for noisy-labels learning in different ways. First, fast adaptation (Garcia, Carvalho and Lorena, 2016) proposes to create a model able to perform several tasks and adapt quickly to make it robust against wrongly labeled samples. Second, learning to update (Li, Yang, Song, Cao, Luo and Li, 2017) learns the loss adjustment rule to lower the negative effect of the wrongly labeled samples during the learning process of the model.

II.3.B Related work

Robust loss functions: In this work we mainly focus on robust loss functions, as they do not depend on the model architecture, can easily adapt to new datasets, do not need pre-training, allow exploiting all the available samples, and do not need to know the noise rate nor a clean validation set. The idea is to design loss functions that are robust to noise in the labels. We define a loss function as noise-tolerant if a classifier trained on a noisy labeled dataset with it has the same misclassification probability as a classifier trained on a clean dataset (i.e. with no noise). More formally, if we have a dataset $\mathcal{D} = \{(X_1, y_1), ..., (X_A, y_A)\}$ composed of A labeled samples with C classes, a loss function $\mathcal{L}$ is noise-tolerant under symmetric noise if (Ghosh, Kumar and Sastry, 2017):

- The noise rate, ζ, verifies: $\zeta < \frac{C-1}{C}$.
- $\mathcal{L}$ is symmetric: $\forall X \in \mathcal{D}, \forall f \in \mathcal{F}, \sum_{i=1}^{C} \mathcal{L}(f(X, \Theta), y = i) = P$, with $\mathcal{F}$ the space of functions where we search for the model, Θ the parameters of the model f, and $P \in \mathbb{R}$ a constant.

This can be extended to *asymmetric* noise by imposing that $\mathcal{R}_{\mathcal{L}}(f^*) = 0$ where f^* is a global minimizer of the risk $\mathcal{R}_{\mathcal{L}}$, defined as follows:

$$\forall f \in \mathcal{F}, \mathcal{R}_{\mathcal{L}}(f) = \mathbb{E}_{\mathcal{D}}[\mathcal{L}(f(x), y)]$$

$\mathcal{R}_{\mathcal{L}}$ is the risk, and f^* is a global risk minimizer of $\mathcal{R}_{\mathcal{L}}$.

Furthermore, even though classical categorical cross entropy (CE) loss function has fast convergence and high generalization capability, this function is not noise-tolerant. That is why, theoretically, other loss functions, such as Mean Absolute Error (MAE) (Ghosh *et al.*., 2017), have better generalization capabilities when working with noisy labels. However, the main inconvenient of MAE is that it has slow convergence and their performances drop when using complicated data. Therefore, other close to noise-tolerant loss functions were proposed such as generalized cross entropy (GCE) (Zhang and Sabuncu, 2018) which combines the advantages of both CE and MAE, and symmetrical cross entropy (SymCE) (Wang, Ma, Chen, Luo, Yi and Bailey, 2019) which combines a noise-tolerant term (the reverse cross entropy Loss) with CE. Formally, if we denote as $f \in \mathcal{F}$ the classifier, $\overline{y}$ the true one-hot encoded label of a sample $X \in \mathcal{D}$, and H the cross-entropy, the GCE and SymCE loss functions are defined as follows:

$$\mathcal{L}_{GCE}(f(X), \overline{y}) = \sum_{k=1}^{K} \frac{\overline{y}_k - f_k(X)^q}{q} \tag{2.10}$$

$$\mathcal{L}_{SymCE}(f(X), \overline{y}) = \alpha \times H(f(X), \overline{y}) + \beta \times H(\overline{y}, f(X)) = -\sum_{k=1}^{K} (\alpha \times f(X)_k \times log(\overline{y}_k) + \beta \times \overline{y}_k \times log(f(X)_k)) \tag{2.11}$$

where $\overline{y}_k$ and $f_k(X)$ are the k^{th} components of $\overline{y}$ and $f(X)$, and q, α, β are hyperparameters. For GCE, the hyperparameter q allows controlling the noise tolerance and the convergence speed:

- when $q \to 1$, we obtain (ignoring a multiplication factor) the MAE loss function, which is known to be noise tolerant but with slow convergence speed,
- when $q \to 0$, we arrive at the CE loss function, which is known to have fast convergence speed but which is not noise tolerant.

III Multi-feature signal classification

In the past years, several works have focused on image classification using deep learning methods such as deep neural networks (DNNs) and convolutional neural networks (CNN) (Rawat and Wang, 2017). Nevertheless, fewer works have focused on signals with a temporal dependence, such as audio signals or sensors signals. Yet, temporal dependency is particularly interesting in the medical field as different devices such as TCD ultrasound, electrocardiogram (ECG) or electroencephalogram (EEG), produce signals with a rich temporal dimension.

III.1 Image classification

In this section, we focus on 2D images of the form $I \in \mathbb{R}^{H \times W \times D}$, having a height of H, a width of W, and D channels. Several of the notions seen here can be extended to 3D images, but they are out of the scope of this work.

CNNs are convolution-based models capable of automatically extracting features from the input data. This is an important advantage as it avoids the time-consuming task of manual feature design that is done to fed classical machine learning techniques such as SVMs, decision trees, naive Bayes, etc. We are going to build on (Rawat and Wang, 2017) to explain the general principle of CNNs.

Usually, CNNs are composed of two sub-networks: an encoder and a classifier (see Figure 2.6.a). The former is composed of convolutional blocks, consisting often of convolutional

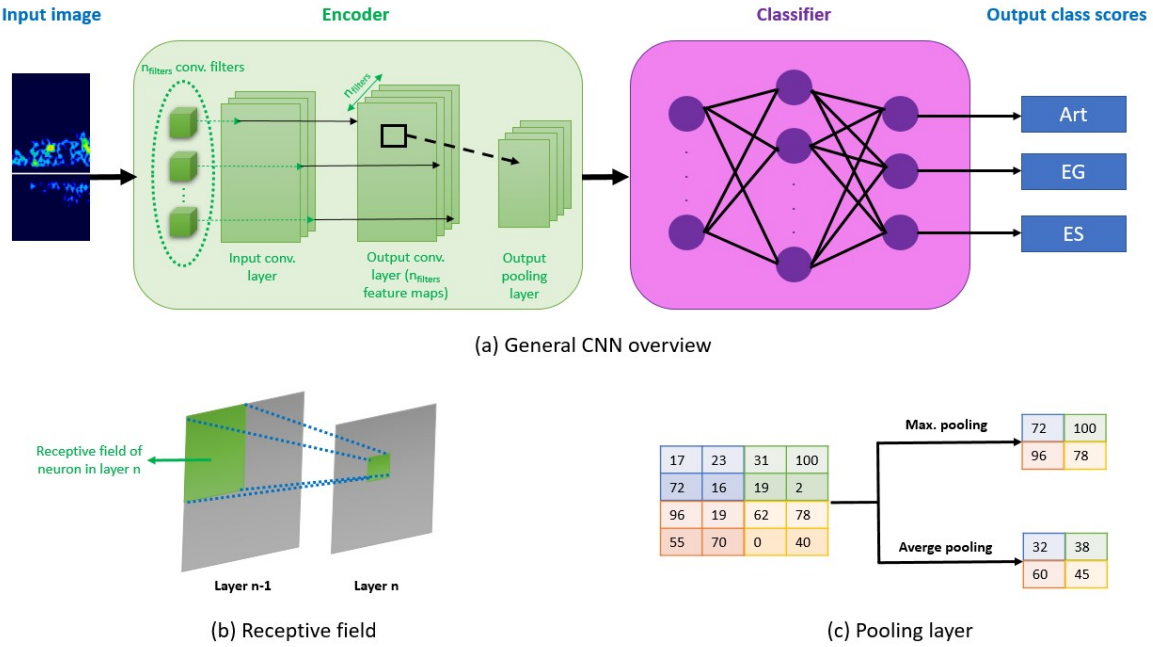


Figure 2.6: Convolutional neural networks overview. (a) General overview. (b) Receptive field of convolutional layers. (c) Pooling layer principle.

layers followed by pooling layers helping to reduce dimensionality (see Figure 2.6.c), ending with a (non-linear) activation function. The classifier is often composed of a multi-layer perceptron (MLP), consisting in a sequence of fully-connected layers followed by non-linear activation functions. Moreover, the main component of CNNs are the convolutional layers that serve as feature extractors. These layers are composed of several convolutional filters that are applied iteratively to the whole input, to give an output composed of several features maps (each feature map is the output of a convolutional filter). Each feature map is composed of neurons, which have a receptive field corresponding to the neighboring neurons in the previous layer (see Figure 2.6.b). This receptive field is an important factor that has an influence on the performances of the model because it defines the temporal or spatial context that we take into account to extract features and solve the task. In audio signal processing terms, the receptive field corresponds to the number of samples or spectra used to compute the prediction; when we fix the CNN architecture, we fix the receptive field. In order to increase the receptive field, we can increase the number of convolutional layers and/or the size of the convolutional kernels, or we can use dilated convolutions (Yu and Koltun, 2016) which allows keeping a reasonable number of parameters for the model.

Furthermore, several variants of CNNs have been proposed, achieving state-of-the-art performances in image classification tasks, such as AlexNet (Gao, Zhang and Wang, 2019), VGG (Simonyan and Zisserman, 2015), GoogLeNet with inception modules (Szegedy, Liu, Jia, Sermanet, Reed, Anguelov, Erhan, Vanhoucke and Rabinovich, 2015), or ResNet (He, Zhang, Ren and Sun, 2016). What is more, recent works on signal processing have exploited CNN techniques, specially for audio related tasks (Purwins, Li, Virtanen, Schlüter, Chang and Sainath, 2019), by applying 2D convolutions on TFRs (Park and Yoo, 2020; Yeh, Mahadeokar, Kalgaonkar, Wang, Le, Jain, Schubert, Fuegen and Seltzer, 2019; Okawa *et al.*, 2019; Pu, Panagakis and Pantic, 2021; Sharan, Xiong and Berkovsky, 2021). (Park and Yoo, 2020) propose to do environmental sound classification by creating a TFR from the raw signal using a learnable

gammatone filterbank, and then passing it through a series of 2D convolutions. (Pu *et al.*, 2021) did speaker identification and acoustic event recognition by feeding a 2D CNN and a DNN with a carefully designed TFR. This TFR was created by applying 1D Morlet filters with learnable parameters to the raw signal, and then using 2D separable Morlet filters to obtain the final embedding. (Yeh *et al.*, 2019) did speech recognition by applying 2D causal convolutions on a log-mel spectrogram extracted from the raw signal and then feeding the result to a transformer model. (Sharan *et al.*, 2021) combined different TFRs to increase the performance of a 2D CNN model for sound scene and speech commands classification.

III.2 Signal classification

As seen in Section I, classical signal processing techniques can be combined with machine learning models to do signal classification (Guepie *et al.*, 2018; Sombune *et al.*, 2017; Wasimuddin, Elleithy, Abuzneid, Faezipour and Abuzaghleh, 2020). However, in this work we focus on deep learning techniques, allowing automatic feature extraction from raw signals and/or TFRs.

Furthermore, temporal-dependent signals have some distinctive characteristics with respect to images. Indeed, these signals are often of large variable length, can have several channels (more than three), but only one dimension (instead of two spatial dimensions for images). Therefore, classical 2D CNN models cannot be easily used with this type of signals, unless using TFRs. Therefore, other types of DL models have to be used, such as 1D CNNs, RNNs, or transformers models (Purwins *et al.*, 2019).

III.2.A General overview

One-dimensional CNNs are a direct adaptation of classical 2D CNNs for image classification, that can be easily used with temporal dependent-signals (Dieleman and Schrauwen, 2014; Lee, Park, Kim and Nam, 2017; Nguyen, Nguyen, Zeng, Nguyen, Tran, Nguyen, Sridharan and Fookes, 2021). (Dieleman and Schrauwen, 2014) compared raw signals and log-mel spectrograms for music automatic tagging, using a 1D CNN. Even though the TFR representation achieves higher performances, the raw signal model was able to extract spectral features from it. (Lee *et al.*, 2017) performed audio classification by creating a multilevel representation of the raw signal that is fed to a 1D CNN with residual connections. (Nguyen *et al.*, 2021) carried out multi-modal emotion recognition using two convolutional AE: a 2D convolutional AE to handle face images, and a 1D convolutional AE to take care of the voice audio. Then, the embedded spaces of both models were fused by concatenation, and the formed feature passed through an LSTM. Into the bargain, these models can also be combined with RNNs or transformers (Okawa *et al.*, 2019; Natarajan *et al.*, 2020; Che, Zhang, Zhu, Qu and Jin, 2021), to enhance their performances.

What is more, the main advantage of RNNs with respect to CNNs is that they can take into account more context (by an indefinitely large receptive field), thus doing a better modeling of the temporal dependencies within the inputs. However, one of the main disadvantages of RNNs is vanishing gradients during training. If we see in more detail the vanishing gradient problem, it means that the weights of the first layers will be updated with a very small gradient, so they won't really be updated: the firsts layers have a forgetting problem. A way to solve this gradient problem is to use Long Short Term Memory (LSTM) models which use different gates and memory cells to keep the important information to remember over time, mitigating the information flow. These type of models can be extended to model audio signals across time and frequency domains by using local filters and recurrent connections to capture translational invariance (F-LSTMs (Li, Mohamed, Zweig and Gong, 2015)) or by using local filters and

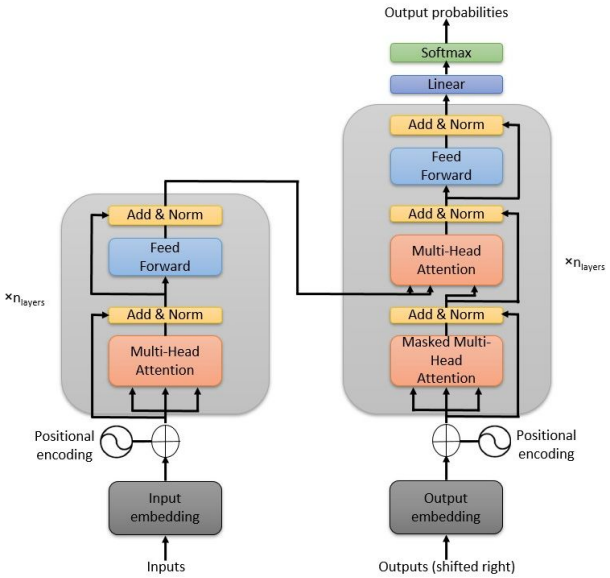


Figure 2.7: Original transformer architecture from (Vaswani *et al.*., 2017).

recurrent connections to model spectral and temporal variations (TF-LSTMs (Sainath and Li, 2016)). Likewise, (Nishizaki and Makino, 2019) applied an LSTM model with FC layers to the power spectrum of the raw signal to do sound classification and tennis swing motion classification. As well, (Okawa *et al.*., 2019) created two binary representations of the raw signals, by doing a binary encoding of its amplitudes. They fed these representations to different models (2D CNN, LSTM and bidirectional gated recurrent unit) to tackle different tasks (acoustic event detection, music classification, and speech classification). On top of that, CNNs and RNNs can be combined to create convolutional recurrent neural networks (McFee and Bello, 2017) where the output of a CNN is the input of an RNN. In convolutional RNN, the CNN is used to extract features, whereas the RNN is used to combine the features over a temporal context.

On the other hand, a newer family of models that has shown some success for signal processing, is the transformer family. This architecture was originally designed for sequence-to-sequence modelling in natural language processing (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser and Polosukhin, 2017), where the main idea is to transform an input sequence into an output sequence (for instance a phrase in English into a phrase in French). These types of models are mainly based on attention mechanisms and do not require recurrence or convolutional layers. These models have an encoder-decoder architecture and are autoregressive i.e. the output is computed using only the previous generated outputs to create the next output of the model. The main parts of the transformers are the following (see Figure 2.7):

- Encoder: The input sequence is given to an embedding layer that will transform this input into an embedded representation. Then, to take into account the temporal information, a positional encoding is added to these embeddings. Afterwards, we find a stack of several identical layers, composed of a multi-head attention mechanism and a point-wise fully-connected layer.

- Decoder: It aims at decoding the encoded representations obtained by the encoder into the output sequence. In this work we focus on classification rather than sequence-to-sequence modelling, thus this component is not going to be detailed.

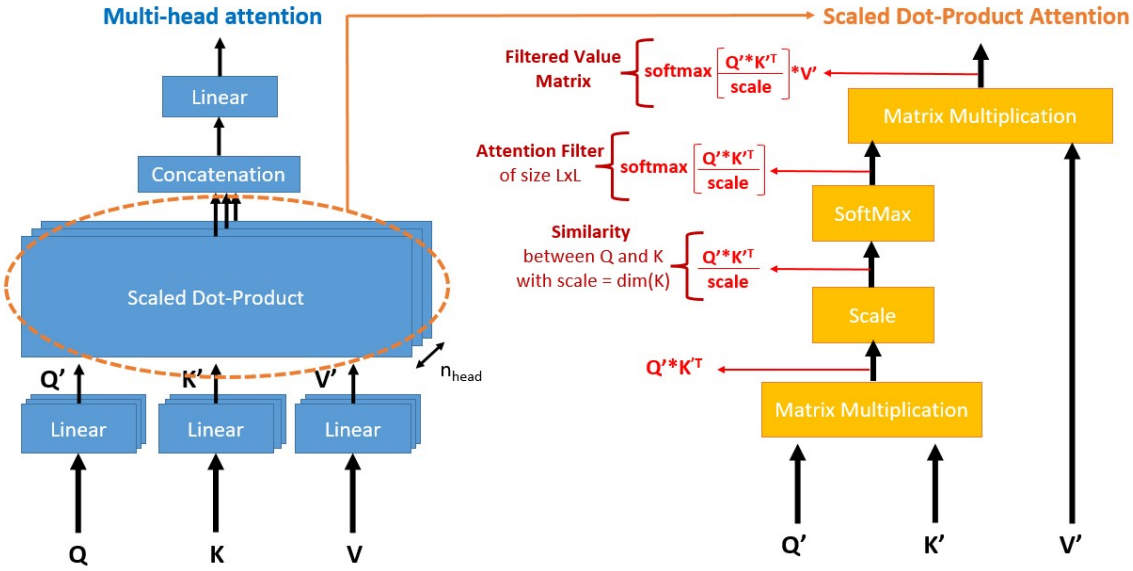
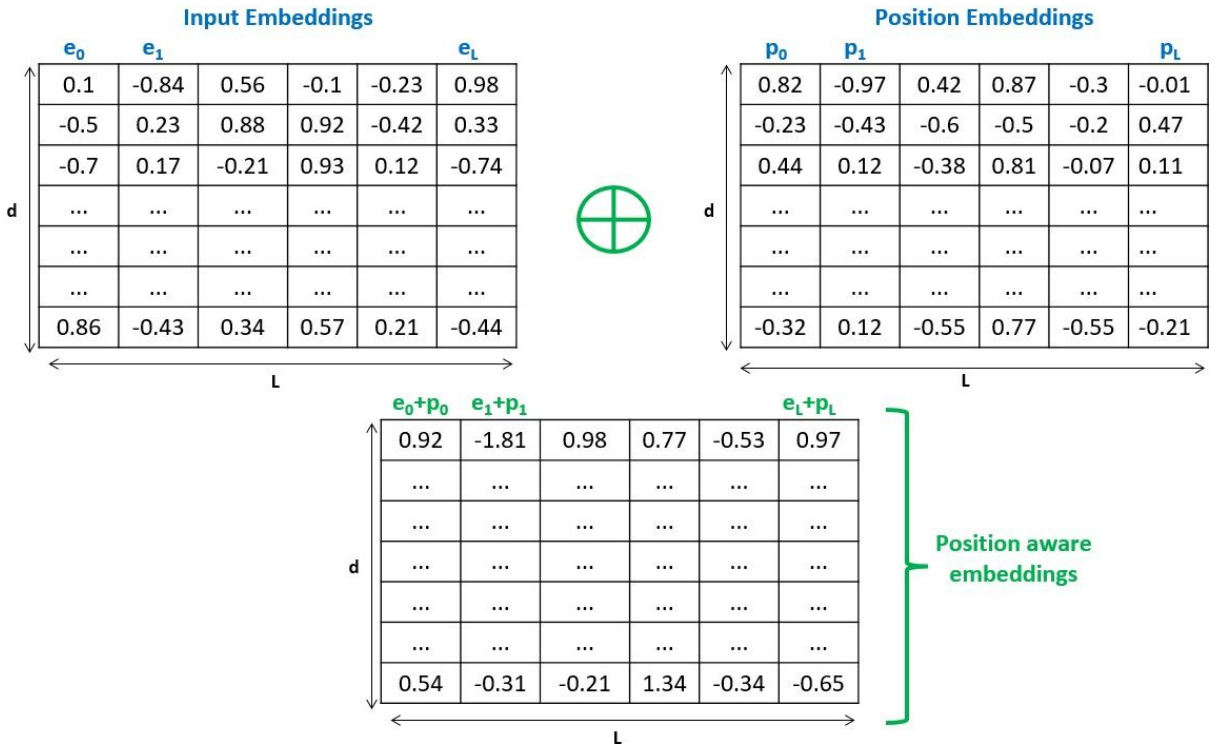


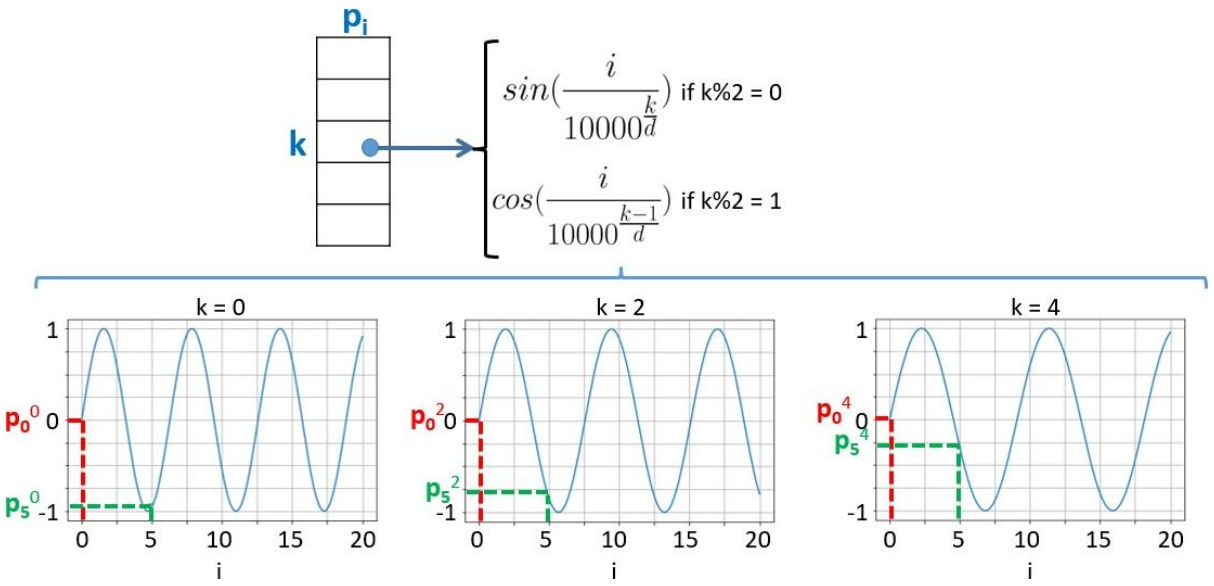
Figure 2.8: Multi-head attention mechanism based on the scaled dot product attention, where L corresponds to the length of the input sequence.

- Multi-head attention (Figure 2.8): The idea is to exploit the information of different representations sub-spaces at different positions. To do this, the input is copied three times, to form a query Q, key K, and value V tensors. They are then projected into a new space using linear layers, to obtain Q', K', and V'. The results are then passed through a scaled dot product attention module, where the softmax of a scaled cosine similarity is computed between Q' and V', to obtain an attention filter. Then, this filter is multiplied by V' to obtain the filtered value matrix by the attention weights.
- Position-wise Feed-Forward Networks: It is composed of two FC layers, each one followed by a ReLU activation. They are applied to each position identically, so the weights are shared within a layer.
- Embeddings and softmax: The inputs of the encoder and decoder (the targets/labels fed to the decoder) are encoded into an embedded representation thanks to embedding layers that are learned. Softmax is used to transform the output of the decoder into probabilities.
- Positional Encoding (Figure 2.9): It is one of the key aspects of the transformers, as it is used to compensate the lack of recurrence or convolutions while exploiting the ordering/temporal information. The main idea is to add to the learned embeddings some positional information. The positional embeddings can be learned or fixed, but in the original paper they are fixed.

Transformers can achieve state-of-the-art performances in several tasks while being less computationally expensive than some CNNs or RNNs models (Vaswani *et al.*., 2017), easier to parallelize, and able to learn long-range dependencies between inputs or outputs. This is particularly interesting for signal processing, where several works have applied this type of architecture (Karita, Wang, Watanabe, Yoshimura, Zhang, Chen, Hayashi, Hori, Inaguma, Jiang,



(a)



(b)

Figure 2.9: Positional encoding used on transformer architectures to compensate the lack of convolutions, recurrence, and take into account the positional/temporal information. Thanks to the proposed sinusoidal encoding, two different positions cannot have the same positional encoding.

Someki, Soplin and Yamamoto, 2019; Mohamed, Okhonko and Zettlemoyer, 2019; Yeh *et al.*, 2019; Boes and Van Hamme, 2019; Tjandra, Liu, Zhang, Zhang, Wang, Synnaeve, Nakamura and Zweig, 2020; Che *et al.*, 2021; Gong, Chung and Glass, 2021; Akbari, Yuan, Qian, Chuang, Chang, Cui and Gong, 2021; Ding, Jia, Miao and Cao, 2022).

(Karita *et al.*, 2019) compared different transformers and RNNs on 15 different audio speech recognition datasets, reaching the conclusion that transformers are often superior to RNN models, as they outperform them in 13 of the 15 datasets.

(Mohamed *et al.*, 2019) carried out speech recognition using a encoder-decoder transformer-based model. From the one hand, the encoder of this model is composed of a 2D CNN feature extractor taking as input a log-mel spectrogram, followed by a transformer block. On the other hand, the decoder is composed of a 1D CNN taking as input the previously predicted word, a multi-head attention layer taking as input the output of the encoder, and a transformer block combing both previously generated outputs.

(Ding *et al.*, 2022) and (Gong *et al.*, 2021) proposed to apply a transformer model to a TFR of the raw signal to do fault diagnosis of rolling bearings and audio classification. The main difference between the two approaches is that the latter slices the TFR into patches before projecting each patch into a 1D vector which is then used as input to the transformer, whereas the former gives as input to the transformer all the frequency components of the TFR at a given time.

III.2.B Related work

(Tjandra *et al.*, 2020) performed speech recognition based on a transformer model guided by an iterated loss doing feature re-presentation (and not representation) at different depths of the model. To accomplish 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 do intermediate predictions using the learned features at that particular level (iterated loss). Let us denote as $\mathcal{M}$ a DL classification model with l layers, $\mathcal{L}$ a loss function to optimize using a dataset $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$, for all $k \in [1, l]$ O_k the output of layer k, and m_k a classification model (for instance and MLP) using O_k. $\mathcal{M}$ is trained with an iterated loss strategy by optimizing the following loss function:

$$\forall X \in \mathcal{X}, \forall y \in \mathcal{Y}, \mathcal{L}_{It}(\mathcal{M}(X), y) = \mathcal{L}(\mathcal{M}(X), y) + \lambda \times \sum_{i \in l_{sub}} \mathcal{L}(m_i(O_i), y) \tag{2.12}$$

where $l_{sub} \subset \{1, ..., l\}$ is a subset of layers, and λ a hyperparameter. With this formulation, we can see the feature re-presentation done at intermediate levels, thanks to the second term. Even though this method is general, Tjanadra et al. applied it in the context of transformers to do speech recognition.

(Natarajan *et al.*, 2020) proposed a wide and deep transformer-based model for ECG signal classification. They mixed handcrafted features extracted from the raw signal with automatically extracted features using a deep learning model. The deep learning model was a 1D CNN model followed by a transformer encoder (see Figure 2.10), and it is composed of three parts. The first part is a 1D CNN feature extractor used to extract embeddings from the raw signal, replacing thus the embedding layer of the original transformer architecture. These embeddings are then passed through a sequence of transformer encoder layers, which allows taking into account the positional (temporal) information of the input samples, as well as to apply attention mechanisms. Then, average pooling and fully connected layers allow obtaining the final encoding that is then fused with the wide handcrafted features for final classification.

Similar to the previous work, (Che *et al.*, 2021) used a 1D CNN-transformer architecture to classify 12 lead ECG signals (no wide feature used). The main difference with (Natarajan *et al.*,

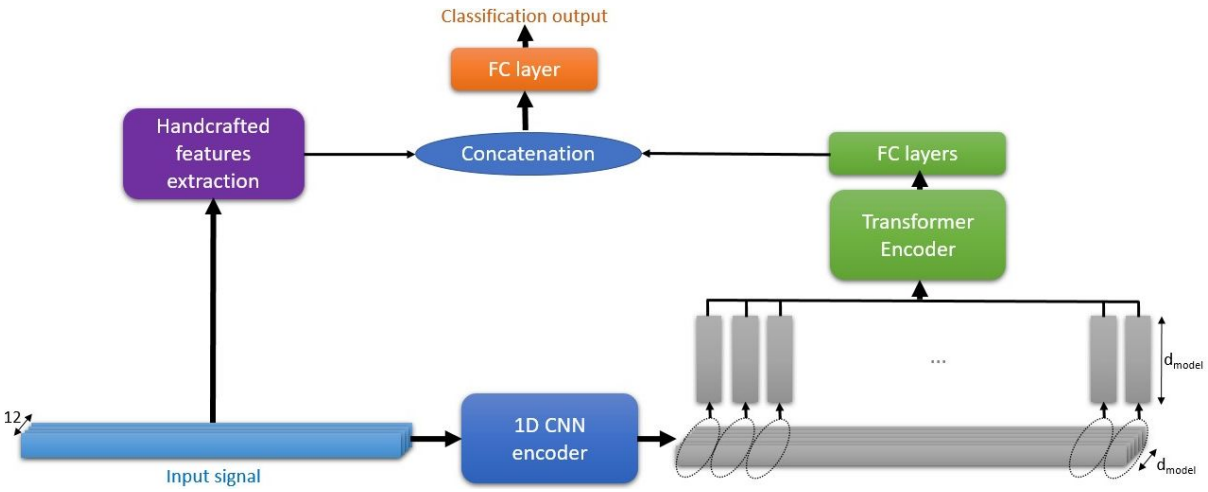


Figure 2.10: 1D CNN transformer wide and deep transformer-based architecture for ECG signal classification (Natarajan *et al.*, 2020), where d_{model} corresponds to the dimension of a token of the input sequence.

2020) is the proposed regularization technique based on link constraints (Qu, Liu, Zhang, Xiao, Jin and Xiong, 2021), allowing to increase classification performance while handling imbalanced datasets. Let us suppose that we have a labeled dataset $\mathcal{D} = \{(X_1, y_1), \ldots, (X_N, y_N)\}$, composed of N labeled samples distributed in C classes, where the samples are $X_1, \ldots, X_N$ and the labels are y_1, ..., y_N. Let us suppose that we have a classification model $\mathcal{M} = C \circ \mathcal{E}$ composed of an encoder $\mathcal{E}$ and a classifier C. We denote as $E_1 = \mathcal{E}(X_1), ..., E_N = \mathcal{E}(X_N)$ the embeddings of the N samples, $X_1, ..., X_N$, and given two embeddings E_i, E_j with $i, j \in [1, N]$, we make the two following assumptions (Che *et al.*, 2021): (1) if the correlation between E_i and E_j is high, it is likely that $\mathcal{M}$ will put X_i and X_j on the same class, (2) if the correlation between E_i and E_j is small, it is likely that $\mathcal{M}$ will put X_i and X_j on the different classes. Based on these two assumptions, (Che *et al.*, 2021) proposed the following regularization term:

$$\mathcal{I}(\mathcal{M}) = \lambda \times \sum_{ij=1}^{N} \frac{1}{2} \times ||E_i - e_{ij} \times E_j||^2 \tag{2.13}$$

$$\forall i, j \in [1, N] e_{ij} = \begin{cases} 1 \text{ if } y_i = y_j \\ -1 \text{ otherwise} \end{cases} \tag{2.14}$$

where λ is a hyperparameter, and e_{ij} is the link between samples X_i and X_j.

Even tough this approach is very interesting, it has the disadvantage of directly depending on the labels, which makes it less robust against noisy labels. One alternative solution, is to adapt similar unsupervised strategies such as deep embedded clustering (DEC) (Xie, Girshick and Farhadi, 2016), as they do not depend on the labels of the samples, but rather on the hidden structure/features.

Keeping the notations of the previous paragraph, DEC (Xie *et al.*, 2016) propose to do clustering on the latent space of the encoder $\mathcal{E}$ of an AE, by using the embedded representations. We denote as $\mathbf{c}_1, \ldots, \mathbf{c}_J$ the centroids of the different J clusters, initialized using k-means. We can now define, for all $i \in [1, N]$ and $j \in [1, J]$, the soft assignments q_{ij} (interpreted as the

probability to assign sample $\mathcal{E}(X_i)$ to the cluster of centroid c_j) using the Student's t-distribution:

$$\forall i \in [1, N], \forall j \in [1, J], q_{ij} = \frac{\sum_{p=1}^{J} (1 + \frac{||\mathcal{E}(X_i) - \mathbf{c}_p||^2}{\alpha})^{\frac{\alpha+1}{2}}}{(1 + \frac{||\mathcal{E}(X_i) - \mathbf{c}_j||^2}{\alpha})^{\frac{\alpha+1}{2}}} \tag{2.15}$$

where α is the degrees of freedom of the Student's t-distribution, fixed to $\alpha = 1$ in (Xie *et al.*., 2016) . We denote as Q the predicted labels' distribution obtained by the q_{ij}. Moreover, in order to train the encoder model and the centroids, (Xie *et al.*., 2016) introduce a target distribution, P, having three main properties: (1) cluster purity, (2) focus on high confidence samples, and (3) normalization of the contribution of each centroid to the final loss. First, we want to have distinct clusters without too much overlapping with neighboring clusters. Second, we want to give more importance to samples assigned to clusters with high confidence (large values of q_{ij}) than to samples assigned to a cluster with small confidence (small values of q_{ij}). Finally, we want to avoid modifying too much the embedding space by the presence of large clusters. To obtain these three properties, (Xie *et al.*., 2016) propose to define P as follows:

$$\forall i \in [1, N], j \in [1, J], p_{ij} = \frac{\frac{q_{ij}^2}{f_j}}{\sum_{p=1}^{J} \frac{q_{ip}^2}{f_p}} \tag{2.16}$$

where $f_j = \sum_{p=1}^{J} q_{pj}$ is the soft frequency of cluster j. The DEC module is then optimized using the Kullback-Leibler (KL) divergence between the target distribution P and the predicted labels' distribution Q:

$$\mathcal{L}_{DEC} = KL(P||Q) = \sum_{i=1}^{N} \sum_{j=1}^{J} p_{ij} \times \log(\frac{p_{ij}}{q_{ij}}) \tag{2.17}$$

Furthermore, transformer were originally designed for sequence-to-sequence modelling tasks in natural language processing (NLP), for instance, translation from one language to another one. Therefore, it had an encoder-decoder architecture. Several works adapt the architecture for classification tasks, using the mean or max of the output scores, or using a class token (Dosovitskiy, Beyer, Kolesnikov, Weissenborn, Zhai, Unterthiner, Dehghani, Minderer, Heigold, Gelly, Uszkoreit and Houlsby, 2021; Devlin, Chang, Lee and Toutanova, 2018). Indeed, (Dosovitskiy *et al.*., 2021; Devlin *et al.*., 2018) proposed to add a learnable class token, that is used for classification purposes (see Figure 2.11). The idea is to add at the beginning of the embeddings sequence, an artificial embedding with learnable parameters, that is going to pass through the transformer encoder with the whole input sequence. Then, the encoding of this class token is retrieved and used as extracted feature for classification. This approach often reaches better classification performances than averaging or taking the max., while reducing the computational costs, as the input to the classifier is not the encoding of the whole input sequence, but only the one of the class token.

III.3 Multi-feature classification

In this work, we have different representations of cerebral emboli through the Doppler IQ signal of a patient (raw signal, TFR, binary encoding, etc.): we have one modality but multiple ways of encoding its information. Therefore, it is important to exploit the complementarity of the different available representations of a single modality efficiently, in order to improve the performance of DL models.

To tackle this problem, one can get inspired from the multi-modal learning community, as each representation could be treated as a different modality. We will build on (Baltrusaitis,

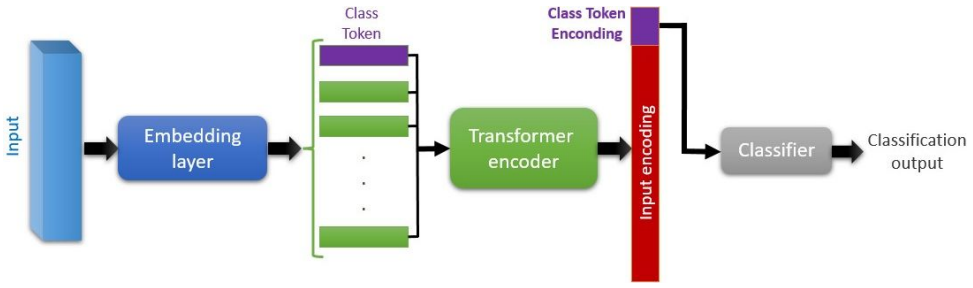


Figure 2.11: Class token principle to adapt transformer architectures for classification tasks.

Ahuja and Morency, 2017) to understand the main challenges in the field of multi-modal learning, which can be also interesting in the multi-feature learning domain.

In multi-modal machine learning we can identify five main core challenges: representation, translation, alignment, fusion and co-learning. In this work, we focus on representation and fusion. For the rest of the challenges we refer the reader to (Baltrusaitis *et al.*., 2017)

Representation: Representation in multi-modal learning refers to the way of structuring data with the aim of exploiting the complementary and redundancy of the different modalities. In simple terms, a representation of an entity is a set of features stored as as a vector or a tensor. On top of that, in multi-feature learning, one can extract different representations from one single modality, for instance the raw signal and a TFR. We can then define two main types of multi-modal/feature representations:

- **Joint representations**: It is mainly used in tasks where multi-modal/feature data is present both during training and inference. The idea is to find a representation on a common latent space of the different modalities/representations that we have access to. Different methods allow doing this, from early fusion by simple concatenation of the representations, to neural networks, AEs and deep Boltzmann machines.
- **Coordinated representations**: It is mainly used when only one modality is available at test time. In these kinds of representations, the different unimodal signals are processed separately, but similarity between the representations is enforced by some constraints. Techniques allowing to do this go from Canonical Correlation Analysis to DNNs.

Fusion: The idea of multi-modal/feature fusion is to integrate information from multiple modalities/representations with the goal of predicting an outcome (e.g., class or continuous value). It can be closely linked with multi-modal/feature representation, but they are not exactly the same because they do not (necessarily) want to achieve the same goal. There are mainly two categories of approaches: model-agnostic and model-based. We focus on model-agnostic methods as they are independent of the used model. In this category, we can identify four main approaches:

- **Early fusion** approaches: The features of the entities of the different modalities/representations are fused just after they are extracted (often by simply concatenation or multiplication). It allows exploiting the correlation and interaction between low level features of each modality, and it only requires one model.
- **Late fusion** approaches: Performs the fusion after each modality/representation has made the task (classification, for instance). The fusing mechanism can be averaging, voting schemes, weighting based on channel noise, signal variance, learned models, etc.

The main advantage is that if one modality/representation is missing, it is quite simple to do the prediction. However, this method ignores the correlation and interaction between low level features of each modality/representation.

- **Intermediate fusion**: The features of the different modalities/representations are fused at an intermediate level of a model, after encoding but before classification. As early fusion, it allows exploiting the correlation and interaction between low level features.
- **Hybrid fusion** approaches: Mix between early and late fusion, it takes the advantages of both methods. One way of doing this is by having one model per modality, each one using features extracted from each modality, and having another model using a combination (early fusion) of these features; then the output of the three models can be fused to generate the final output (late fusion).

Fusion is one of the main core challenges of multi-modal/feature learning that we are interested in. Indeed, in our application, we want to find a way to combine information of different representations of a signal (such as the raw signal or the TFR). This approach if often used for video action recognition and audio event classification, to combine images and audio signals (Hori, Hori, Wichern, Wang, Lee, Cherian and Marks, 2018; Boes and Van Hamme, 2019; Ortega, Senoussaoui, Granger, Pedersoli, Cardinal and Koerich, 2019; Akbari *et al.*, 2021). (Hori *et al.*, 2018) did video description generation by combining audio and video using a RNN architecture with an attention fusion mechanism in the intermediate layers. (Boes and Van Hamme, 2019) extracted mel-spectrograms from the raw audio and images from the video using a pre-trained VGG model. Then, the extracted features are projected in a space of fixed dimension before being fed to the transformer encoder and decoder (one modality is the input for the encoder and the other one of the decoder). (Ortega *et al.*, 2019) performed emotion recognition based on three modalities: video, audio and text. They combined by concatenation the features extracted from each modality by the means of a DNN to obtain a joint representation. (Akbari *et al.*, 2021) used a FC layer to extract features from each modality (video, audio and text), and then passed them to a transformer encoder. Afterwards, the extracted embeddings are projected in different common spaces with different granularities to do classification.

What is more, motivated by the advantages of using multiple modalities to solve a classification tasks, different works have exploited these methods using different representations of a single modality instead of different modalities, as it is often easier to obtain[1].

Computer vision: Different features can be extracted from an image, and then combined to enhance the performances of different models (Wang, Zhang, Liu, Choo and Huang, 2017; Mao, Li, Ma, Zhang, Zhou and Wang, 2020; Zhu and Jiang, 2020; Tiong, Kim and Ro, 2019).

(Wang *et al.*, 2017) did hyperspectral image classification by decomposing them into principal components using principal component analysis (PCA). This allowed them to obtain two type of images: one mainly representing the spectral characteristics of the image and composed of the concatenation of the different principal components obtained before, and another one which translates the spatial properties, composed of the first K filtered principal components (obtained using a guided filter). These two images are then fused by concatenation, and the result is fed to a sparse AE to do classification.

On the same principle, (Zhu and Jiang, 2020) performed face recognition by fusing two types of features extracted from the images: global ones using 2D PCA, and local ones through local binary patterns.

[1]For instance, for cerebral emboli classification, MRI can be a helpful modality to TCD, but it is expensive and harder to obtain.

(Mao *et al.*., 2020) extract three new images from one single image by selecting the three color components of the latter using iterative RELIEF. Then, they were fed into a multi-path CNN to extract new features, which are then fused by concatenation. Finally, these last features are projected using PCA, and the result is fed to a support vector machine (SVM) classifier, to do object detection.

Signal processing: In the signal processing community, multi-feature learning is often based on the combination of TFR and other handcrafted features (Kim and Lee, 2019; Jin, Yang, Zhao, Luo and Woo, 2020; Feng, Feng, Li, Hou and Liu, 2020; Liu, 2021; Ertugrul, Acar, Aldemir and Öztekin, 2021; Chen, Cheng, Wang, Lu, Xv, Liu and Zhu, 2021; Yao, Gao, Zhang and Ma, 2021; Ahmad, Tabassum, Guan and Khan, 2021).

(Kim and Lee, 2019) performed power signal analysis by fusing (by concatenation) different TFRs of the raw signal (spectrogram, mel-spectrogram and MFCC), and feeding them to an LSTM model.

(Jin, Yang, Zhao, Luo and Woo, 2020) did multi-feature and multi-modal emotion recognition by passing through an LSTM different MFCCs extracted from audio samples representing distinct emotions. The obtained features were then fused by weighted concatenation and projected using an FC layer into an intermediate latent space. In parallel, behavioral features were extracted from signals coming from wearable devices, and passed through a DNN, with an output latent space of the same dimension as the one handling the audio features. Finally, the two obtained representations were fused by weighted concatenation to get a joint-representation used for classification.

(Feng *et al.*., 2020) performed well-testing (classification) by concatenating features extracted from the raw signal (wavelet packet decomposition-approximate entropy, gradient and its extreme value, fast Fourier transform coefficients, empirical mode decomposition-approximate entropy) to obtain a joint-representation which was used to extract new features through a deep Boltzmann machine. Then, these features were purified using the maximum information coefficient and used for classification with an SVM.

(Ertugrul *et al.*., 2021) did heartbeat classification using 12-lead ECG signals. Because these signals have multiple channels, they were able to create a gray-scale image (of 12 rows) by concatenating the channels. Then, from this image 4 different features were extracted: gray level co-occurence matrix, texture energy measure, histogram of oriented gradients and Gabor wavelet transform. Finally, each feature is passed to a (distinct) randomized neural network to do classification.

At last, (Ahmad *et al.*., 2021) also used ECG signals to classify heartbeats. As the previous work, they also extracted different images from the raw signal: gramian angular field, recurrent plot and Markov transition field. These images were then fused, and the result used for classification based on two different strategies: multimodal image fusion and multimodal feature fusion. The former, is an early fusion approach where the images are fused before passing them through an AlexNet model for classification. The latter is an intermediate fusion approach where features are first extracted using an AlexNet model, then fused to create a final feature for classification through an SVM. This allowed them to achieve state-of-the-art performances on two heartbeat categorization dataset PTB (Bousseljot, Kreiseler and Schnabel, 1995) and MIT-BIH (Moody and Mark, 2001).

IV Model compression

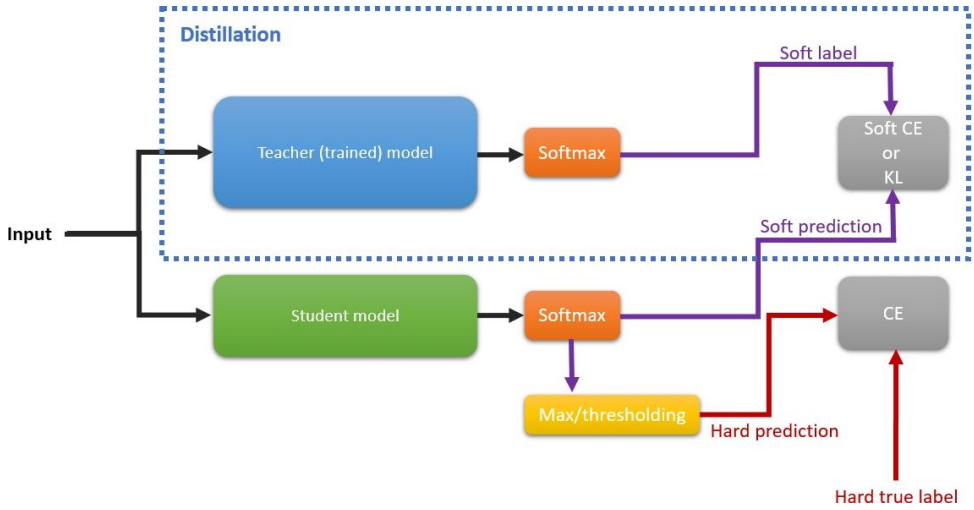


Figure 2.12: Knowledge distillation principle.

As seen in the previous sections, DNNs such as CNNs and transformer models have reached state-of-the-art performances in several tasks, going from computer vision, NLP, and signal processing. However, these methods tend to be energy greedy and often require important computation and memory resources. This is an important point to consider in the medical field, where more and more DL models are being used (Piccialli, Somma, Giampaolo, Cuomo and Fortino, 2021), with limited computation and energy resources. If deep learning models want to be used with these embedded devices, some efforts have to be made to develop more energy and computation efficient models.

We are going to build on (Gholami, Kim, Zhen, Yao, Mahoney and Keutzer, 2022) to present the main concepts and subfields of model compression, and then we are going to focus on two approaches: model pruning and model quantization.

IV.1 General overview

We can categorize the different model compression methods in five (non-exclusive) families: pruning, quantization, knowledge distillation, efficient NN design, and hardware-adapted NN architectures. The two first families will be detailed later in sections IV.2 and IV.3, whereas the last three families are going to do a briefly overviewed hereafter.

Knowledge distillation: This technique was introduced for neural networks by (Hinton, Vinyals and Dean, 2015), allowing to reduce their size. The main idea is to train a smaller network, called the student, using a larger pre-trained model, called the teacher. To do this, the student model is trained to match the soft-predictions of the teacher model, instead of only using the hard labels of the samples (see Figure 2.12), which can achieved owing to the soft CE, or the KL divergence. Even though this technique is interesting, it has some difficulties achieving by itself important compression rates without an important degradation of the classification performances compared to other families of methods (such as pruning or quantization). Moreover, when this method is applied alone, it often gives models with floating-point weights, which are difficult to adapt to efficient arithmetic operations (capable of reducing latency and energy consumption) without specialized hardware.

Efficient NN design: Another way to obtain lighter models, is to use efficient components such as dilated convolutions (Yu and Koltun, 2016), depth-wise convolutions (Howard, Zhu,

Chen, Kalenichenko, Wang, Weyand, Andreetto and Adam, 2017), or efficient decomposition of the existent components through methods such as singular value decomposition (Kanjilal, Dey and Banerjee, 1993). Moreover, more global mechanisms have been introduced to design more efficient architectures, such as residual connections (He *et al.*., 2016) and inception modules (Gao *et al.*., 2019). The main inconvenient with this manual approach is that they are time-consuming as different combinations are possible, and it is difficult to find the optimal combination for a given task. This drawback can be partially tackled using neural architecture search techniques where the optimal architecture is searched under some constraints (memory requirements, size, depth, etc.) (Elsken, Metzen and Hutter, 2019). Nevertheless, this approach still requires important amounts of computation and energy resources to find the efficient models.

Hardware-adapted NN architecutres: DL models can be optimized using specialized hardware, allowing to efficiently perform the different operations needed for inference. Some works propose to design hardware-aware models (Yao, Dong, Zheng, Gholami, Yu, Tan, Wang, Huang, Wang, Mahoney and Keutzer, 2021), taking into account the hardware where they will be deployed as well as other constraints. Other works focus on designing specialized hardware where some operations can be performed efficiently, for instance using systolic arrays (Bosio, O'Connor, Traiola, Echavarria, Teich, Hanif, Shafique, Hamdioui, Deveautour, Girard, Virazel and Bertels, 2021) as in TPUs (*Quantifying the performance of the TPU, our first machine learning chip*, n.d.). These approaches have the advantage that high optimization can be done, which allow important reduction in inference times as well as energy consumption. However, they are time-consuming and more difficult to generalize to generic architectures.

IV.2 Model pruning

IV.2.A General overview

Model pruning is another family of methods allowing to reduce the memory and energy consumption needed by DL models. The main idea is to zero a subset of the parameters of a model based on different criterion. This can be done as NN models are over-parametrized and redundant, where often an important number of parameters are close to zero (Ji, Jain, Ferdman, Milder, Schwartz and Balasubramanian, 2021). Because of this, pruning can act as a regularizer, improving the generalization of the obtained models (Hoefler, Alistarh, Ben-Nun, Dryden and Peste, 2022). Additionally, it has the advantage that it does not require an important modification of the model architecture or the training strategy, contrary to other model compression methods.

Furthermore, pruning can be performed in two ways: structured if entire blocks are removed (for instance as group of a layer), and unstructured if all the weights following some criterion are removed, whether they are in the same structure or not (weights can be zeroed even if not in the same layer or convolutional filter). Classical unstructured approaches consist in removing weights using the L1 or L2 norms (Han, Pool, Tran and Dally, 2015), or second order derivatives (Hassibi, Stork and Wolff, 1993). More formally, if we have a model $\mathcal{M}$ composed of l layers $L_1, ..., L_l$, such that (without loss of generality), for all $k \in [1, l], n_k, m_k, \in \mathbb{N}^* L_k \in \mathbb{R}^{n_k \times m_k}$, a simple pruning strategy can be based on thresholds:

$$\forall k \in [1, l], i \in [1, n_k], j \in [1, m_k], \tilde{L}_k^{ij} = \begin{cases} 0 \text{ if } ||L_k^{ij}|| \leq \tau \\ L_k^{ij} \text{ otherwise} \end{cases} \tag{2.18}$$

where $\tilde{L}_k$ is the pruned version of L_k, and τ is a pre-defined threshold. What is more, to avoid using pre-defined thresholds and to have more control on the amount of weights to remove,

these approaches propose to remove a pre-defined percentage of weights having the smallest norm.

More sophisticated methods exists, based on more complex strategies. (Zhu and Gupta, 2018) propose gradual pruning during training by starting with an initial sparsity value and then increasing it progressively over the iterations to reach a final desired sparsity value (so sparsity is done gradually over the iterations).

(Mariet and Sra, 2016) proposed to use determinantal point processes (DPPs) to determinate a subset of neurons in a layer, and then fuse the non-selected neurons (the redundant ones) into the selected subset. To do this, an initial kernel matrix is computed for a layer L_k based on their activations, and then scaled to obtain the final DPP kernel matrix. Then, one can sample from this matrix to obtain the important neurons of L_k. Therefore, if some neurons are not sampled by the DPP, then those neurons are considered as redundant and can be removed from L_k. However, to avoid changing the inputs of the next layer L_{k+1}, one cannot just discard the unselected neurons. To solve this, the information of the unselected neurons are fused with the sampled neurons through a reweighting process.

Following the same idea of the importance of neurons, (Luo, Wu and Lin, 2017) proposed a structured pruning approach for CNNs, by carefully selecting the layers/filters to prune, based on the statistics of the (next) layer's parameters. The idea is that, if we have a convolutional layer L_i, followed by another convolutional layer L_{i+1}, the input of layer L_{i+1} is generated by the filters of layer L_i. Therefore, if using only a subset of filters of L_i, one can generate an input for L_{i+1} capable of accurately estimating the true output of L_{i+1} (as if no filter were removed), then the filters that are not in this subset can be removed.

What is more, because the threshold operator often used in pruning is not non-differentiable, it is difficult to automatically tune the pruning hyperparameters during training. Some approaches try to solve this by using different methods to optimize pruning, such as reinforcement learning (RL) (He, Lin, Liu, Wang, Li and Han, 2018), genetic algorithms (Xu, Zhang, An, Liu, Liu and Wang, 2021), or differentiable pruning functions (Manessi *et al.*., 2017). (He *et al.*., 2018) used reinforcement learning to do model compression automatically without the need of any human effort. The idea is to process layer by layer a pre-trained model using RL, where the RL agent inputs the embedding of a layer, and outputs a sparsity ratio (the reward is a function of the accuracy and the number of floating operations per second, FLOPS). After compression of the layer with the given sparsity ratio, it goes to the next layer. (Xu *et al.*., 2021) follow a similar principle using genetic algorithms to find layer by layer a good sparsity ratio.

IV.2.B Related work

By the same token, (Manessi *et al.*., 2017) proposed a differentiable pruning function with symmetric thresholds, allowing direct optimization of its hyperparameters using gradient descent. In fact, the method proposes a regularizer enforcing pruning, using the this function. The first term is a regularization with respect to the learned threshold for pruning (indirect influence of the weights), and the second term is the classical weight decay regularization (direct influence on the weights). Even though these two regularizers are important, we focus here on the pruning function, as it is at the core of the method (see Figure 2.13). The proposed pruning function is defined as follows:

$$\begin{aligned} \forall x, t, \alpha \in \mathbb{R}, p(x;t,\alpha) = & [ReLU(x-t) + t \times \sigma(\alpha \times (x-t)] \\ & + [-ReLU(-x-t) - t \times \sigma(\alpha \times (-x-t))] \end{aligned} \tag{2.19}$$

where t is the pruning threshold, α is a parameter controlling the slope of the linear parts of the function (pruning speed), ReLU corresponds to the rectified linear unit function

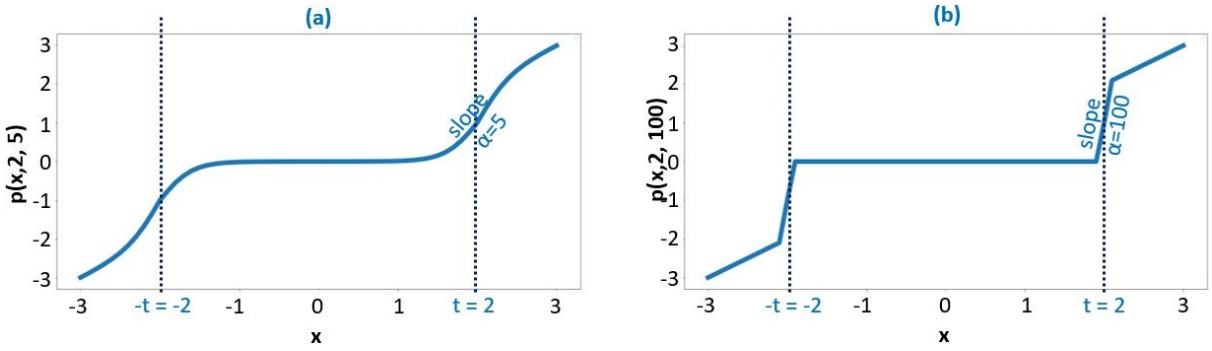


Figure 2.13: Example of the differentiable pruning function proposed in (Manessi *et al.*., 2017). (a) Pruning function with $t = 2$ and $\alpha = 5$. (b) Pruning function with $t = 2$ and $\alpha = 100$. One can note that higher values of α give a more aggressive pruning. Note that, besides the case $x = 0$, the value of the pruning function between t and $-t$ is not exactly zero, but close to zero, converging to zero when α tends to $+\infty$. However, one can define a threshold to determinate which values can be considered as zero, for instance, all the values smaller to 10^{-3} as in (Manessi *et al.*., 2017).

$ReLU : x \rightarrow max(0, x)$, and σ to the sigmoid function $\sigma : x \rightarrow \frac{e^x}{e^x+1}$. This function has the advantage of being weakly differentiable with respect to x and t.

IV.3 Model quantization

Quantization is the process allowing to reduce the precision of a floating point number, for instance from 32 or 64 bits to 16 bits or fewer. This can be beneficial from different points of views. First, from a memory perspective, reducing the precision of the weights of a model reduce the memory needed to store them and to load them into RAM. Second, from a computation point of view, some precisions allow more efficient computation, for instance, if using integers, or dyadic numbers, more efficient arithmetic operations can be used. Following the two previous points, quantization can also help from an energy perspective, as reducing the memory required to load models can help decrease the number of memory and processor transfers (data transfers), and more efficient arithmetic operations tend to consume less energy than classical floating point operations.

IV.3.A General overview

Because of the previous motivations, different works have focused on the application of quantization to deep learning models (Gholami *et al.*., 2022). Let us suppose that we have a full precision (FP) weight tensor W and a quantization operator Q giving the quantized weight $W_Q = Q(W)$. The quantization procedure can be characterized by the clipping range $[a, b]$ (interval where the values of the FP tensor W should be, if not they are clipped), the calibration (clipping range search), and scaling factor S. Therefore, a quantization process can be of different natures, based on the definition of these seven concepts (Gholami *et al.*., 2022):

- Uniformity: the clipping range can be discretized into equally distant values or not.
- Symmetry: based on the symmetry of the clipping range (symmetric if $a = -b$).
- Static/dynamic: based on when the clipping range is computed (static if it is before inference, dynamic if it is during inference).

- Granularity (see Figure 2.14): based on at what level/component the quantization is done. It can be done channel-wise for CNNs, using subgroups of weights in the different layers (Prato, Charlaix and Rezagholizadeh, 2020), or at different levels in depth and width of the weights' tensors based on the quantization error (Xu, Wang, Zhou, Lin and Xiong, 2018).

- Training aware: based on its relationship with the training procedure. If quantization is done during training, then it is quantized aware training (QAT). If it is done after training, then it can be post-training quantization (PTQ) or zero-shot quantization.

- Stochasticity: the quantization operator can be based on probability measures: $Q(W) = W_Q$ with probability p and $Q(W) = W_{Q'}$ with probability $1-p$. It can also be deterministic if the quantization operator always gives the same quantized output for a fixed input.

- Simulation: most quantization methods are simulated as, even though quantization is done, the computations are done in full-precision without taking advantage of quantization acceleration (for instance when using integer-only quantization, efficient arithmetic operations can be used).

The first approaches applying quantization to DL were based on matrix factorization (Gong, Liu, Yang and Bourdev, 2014; Kim and Lee, 2019) or weight sharing and/or clustering (Han, Mao and Dally, 2016; Ullrich, Meeds and Welling, 2017; Dubey, Chatterjee and Ahuja, 2018; Lan, Chen, Goodman, Gimpel, Sharma and Soricut, 2020). The former can be achieved through singular value decomposition or vector quantization, whereas the latter can be implemented through, k-means clustering of the weights (with or without identification of the important convolutional filters) (Han *et al.*., 2016; Dubey *et al.*., 2018), soft weight-sharing based on Gaussian mixture models (Ullrich *et al.*., 2017), or cross-layer parameter sharing (Lan *et al.*., 2020).

Furthermore, as seen before, knowledge distillation can be used for model compression. However, it tends to give low compression rates for small performance drops. Some works have proposed to combine quantization with knowledge distillation to improve the compression rates of knowledge distillation while keeping close performances to the full precision model (Polino, Pascanu and Alistarh, 2018; Zhang, Hou, Yin, Shang, Chen, Jiang and Liu, 2020; Sun, Cheng, Gan and Liu, 2019; Bai, Zhang, Hou, Shang, Jin, Jiang, Liu, Lyu and King, 2021). Indeed, knowledge distillation can directly be used to compensate the performance drop due to quantization, without reducing the size of the network nor modifying its architecture (Zhang *et al.*., 2020). Likewise, other methods use it to reduce the size of the model while doing quantization (Polino *et al.*., 2018) at the same time. (Bai *et al.*., 2021) propose binary quantization for BERT models using QAT and PTQ. First, they start by training with QAT a half-sized ternary BERT. Then, they initialize the weights of a full-size Binary BERT using a ternary weight splitting operator. Finally, they applied knowledge distillation to recover from the performance drop due to quantization.

Over and above that, most QAT methods are hard to optimize because of the non-differentiable nature of quantization operators. Therefore, often the straight through estimator (STE) (Yin, Lyu, Zhang, Osher, Qi and Xin, 2019) is used to approximate gradients (Zhang, Yang, Ye and Hua, 2018; Bhalgat, Lee, Nagel, Blankevoort and Kwak, 2020; Zhu, Han, Mao and Dally, 2017; Zhou, Ni, Zhou, Wen, Wu and Zou, 2016). (Zhang, Yang, Ye and Hua, 2018) proposed to use learnable quantizers with the objective of minimizing the quantization error. The idea is to quantize the FP weights to its nearest integer representation, with the advantage that an integer can be seen as a combination of power of 2 with binary linear coefficients, allowing efficient (bit-wise) operations. Then, the quantizer operator can be seen as a learnable FP basis and a binary vector ($\{0,1\}^B$ or $\{-1,1\}^B$), with B the number of bits used to encode the nearest

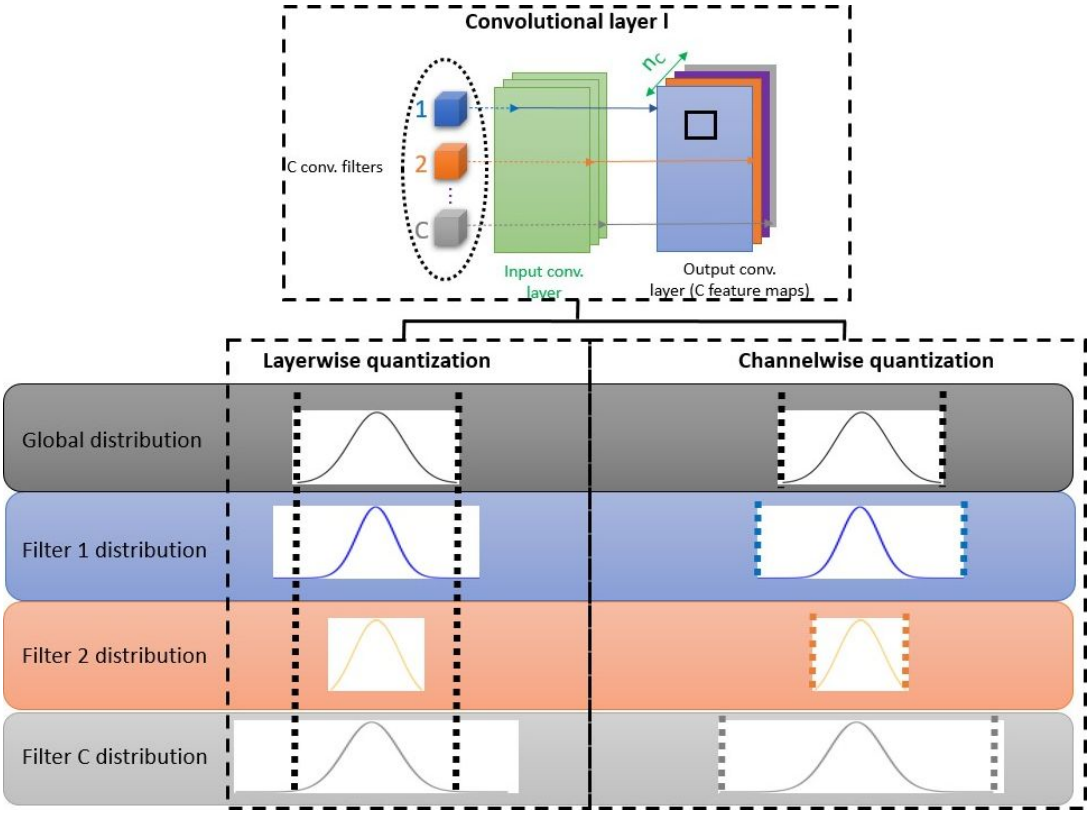


Figure 2.14: Different granularities to apply quantization from (Gholami *et al.*., 2022). In the bottom figure, the first row corresponds to the global distribution of all the weights in the convolutional layer (of all the filters). The remaining rows corresponds to the distribution of each filter. The main difference between the two approaches is the way the clipping range is determined. For layerwise quantization, the clipping range is defined using the distribution of all the weights of the layer (two black dotted lines in the left column). For channelwise quantization, we have one clipping range per filter in the convolutional layer (colored dotted lines in the right column). We can note that the number of channels of the output feature map is equal to the number of convolutional filters in the convolutional layer, hence the name channelwise quantization when quantization is done filterwise.

integer of the FP weight. (Bhalgat *et al.*, 2020) proposed LSQ+, an asymmetric quantization method for skewed distributions with negative values, and a trainable scale and offset (with STE used to approximate their gradients). The main drawback of this approach is that the final performances of the model are very sensitive to the initialization of the scale factor and the offset. To solve this, the authors proposed to initialize them in two different ways. For the weights' quantization, the scale factor is initialized using the weights' statistics. For the activations, the scale and offset are obtained by minimizing the Frobenius norm between the quantized and full precision activations. By the same token, other approaches focus on the reformulation of the quantization problem as a differentiable one using non-linear functions (Yang, Shen, Xing, Tian, Li, Deng, Huang and Hua, 2019). In fact, the quantized value of a weight/activation can be computed as an affine sum (with several scale factors and one bias) of unit step functions. However, as the unit step function is not smooth, the authors proposed to replace it by a sigmoid function, which is differentiable. In addition, they add a temperature hyperparameter to the sigmoid function, controlling the gap between the step unit function and the sigmoid.

On the other hand, to take advantage of efficient arithmetic operations (Jacob, Kligys, Chen, Zhu, Tang, Howard, Adam and Kalenichenko, 2018), it is important to know whether quantization is simulated or not. Some recent works propose to use integer-only quantization, in order to profit from arithmetic operations (Jacob *et al.*, 2018; Zafrir, Boudoukh, Izsak and Wasserblat, 2019; Kim, Gholami, Yao, Mahoney and Keutzer, 2021) to reduce inference time and energy consumption, simultaneously to memory requirements. (Zafrir *et al.*, 2019) proposed a QAT method to fine-tune transformer (BERT) models, compressing them and reducing the inference times thanks to integer operations. To do this, they quantized weights and activations using symmetric uniform quantization with 8 bits (gradients approximated with STE), with a scaling factor determined statically or dynamically during inference. Additionally, they adopted a mixed precision strategy, where the embedding and FC layers were quantized to 8 bits, whereas the biases to 32 bits. (Kim *et al.*, 2021) proposed a similar approach allowing to quantize transformer-based models (BERT) using integer only arithmetic during inference, reducing latency and energy consumption. The used quantization is relatively simple, as it is an uniform, symmetric and static quantization approach. The main difficulty comes when quantizing activations, as most activation functions are non-linear, which are difficult to quantize. To solve this, the authors proposed a polynomial approximation of non-linear functions, which has the advantage of being computable using integer-only arithmetic. Using this approach, the authors proposed a 2 degree polynomial approximation of the Gaussian error linear unit (GELU) and Softmax activation functions, with an approximation error of the order of the order of 10^{-3}.

IV.3.B Related work

Performance drop is almost inherent to quantization as information about the original weights is lost during the process. One solution to mitigate this problem is using different precisions at different levels/layers of the model (Dong, Yao, Gholami, Mahoney and Keutzer, 2019; Dong, Yao, Arfeen, Gholami, Mahoney and Keutzer, 2020; Yao, Dong, Zheng, Gholami, Yu, Tan, Wang, Huang, Wang, Mahoney and Keutzer, 2021; Shen, Dong, Ye, Ma, Yao, Gholami, Mahoney and Keutzer, 2019; Gholami *et al.*, 2022). These mixed quantization approaches can be costly, as the number of possible precisions combinations increases exponentially with the number of layers. To reduce the exploration cost, some works have proposed metrics allowing to measure the sensibility of a layer to quantization (Dong *et al.*, 2019, 2020). Indeed, using second order derivatives (Hessian), one can quantify the flatness of the loss landscape and therefore know if we are in an irregular (sensible) or regular (robust) zones. More formally, if we have a model

$\mathcal{M}$ composed of l layers with weights W_1, ..., W_l, and a supervised loss function $\mathcal{L}$, then we can define the following second order metric measuring the sensitivity to quantization for an entire model (Dong *et al.*., 2020):

$$\Omega = \sum_{k=1}^{l} \overline{Tr}(H_k) \times ||W_k - Q(W_k)||_2^2 \tag{2.20}$$

where for all $k \in [1, l]$, H_k is the Hessian of the k^{th} layer, $\overline{Tr}(H_k)$ is the mean Hessian trace[1], and Q is a quantization operator. More interestingly, under some assumptions, (Dong *et al.*., 2020) proved that for all $k \in [1, l]$, $\overline{Tr}(H_k) \times ||W_k - Q(W_k)||_2^2$ is a local measure indicating the sensitivity of the k^{th} layer to quantization. These assumptions are the following. First, the model must be twice differentiable as the Hessian has to be computed, and it has to have converged to a local minimum (with respect to $\mathcal{L}$, as the first order derivative (gradient) should be zero and the Hessian positive semi-definite, which facilitates the Hessian approximation and computation using Taylor approximation). Second, the third order term of the Taylor series approximation should be close to zero. Third, if we denote for all $k \in [1, l]$, $W_k^{Q*} = Q^*(W_k)$ the optimal quantization of W_k obtained after QAT, then $\Delta W_k^* = W_k^{Q*} - W_k$ can be decomposed in a base of the orthonormal eigenvectors of the respective Hessian H_k (positive semi-definite assumption), with coefficients that depend on the chosen quantization precision of that layer. The idea of HAWQ-v2 (Dong *et al.*., 2020) is to try different mixed-precision combinations for the different layers, compute Ω, sort the results by values of Ω, and then choose the configuration with the smallest value of Ω. An improvement of this method was recently done to take into account hardware constraints during quantization (Yao, Dong, Zheng, Gholami, Yu, Tan, Wang, Huang, Wang, Mahoney and Keutzer, 2021). What is more, other teams have shown that, for pre-trained models, near-optimal low-bit solutions exist close to the full-precision solutions (McKinstry, Esser, Appuswamy, Bablani, Arthur, Yildiz and Modha, 2019), allowing to effectively do quantization without important drop of performances. Based on this idea, adaptive quantization methods have been proposed, where the number of bits used to encode the weights can be adapted to the resources and states (low battery for example) of the hardware or the device (Jin, Yang and Liao, 2020).

Furthermore, to obtain even higher compression rates and reduce the number of bits used to store the weights, extreme quantization approaches have been developed, using less than 4 bits to encode them. This has been achieved through binary networks (Zhou *et al.*., 2016; Rastegari, Ordonez, Redmon and Farhadi, 2016; Lin, Zhao and Pan, 2017; Bai *et al.*., 2021), ternary models (Zhu *et al.*., 2017; Zhang *et al.*., 2020; Xu *et al.*., 2018; Hou and Kwok, 2018), or mixed-precision quantization (Shen *et al.*., 2019; Dong *et al.*., 2020). Most of these binary or ternary networks, work with binary or ternary weights' matrices, having (0 or 1) or (-1, 0, or 1) values multiplied by unique full-precision learnable coefficients (or scaling factors). The main difference between methods is the heuristics used to quantize the weights, which often depends on manually selected thresholds, or thresholds based on the weights' statistics. In ternary networks (Li, Liu, Wang, Zhang and Yan, 2022), weights are quantized using a single symmetric threshold t and one scaling factor λ. Let us define the following ternarization operator, depending on a threshold t:

$$\forall x \in \mathbb{R}, Q_t(x) = \begin{cases} -1 \text{ if } x < -t \\ 0 \text{ if } x \in [-t, t] \\ 1 \text{ if } x > t \end{cases} \tag{2.21}$$

Then, the ternarization w_t of a FP weight w coming from a weight matrix W is defined as follows: $w_t = Q_t^{\lambda}(w) = \lambda \times Q_t(w)$. The two quantization parameters t and λ are obtained by

[1]Indeed, the Hessian changes based on the input, therefore, it is computed for different inputs and then averaged.

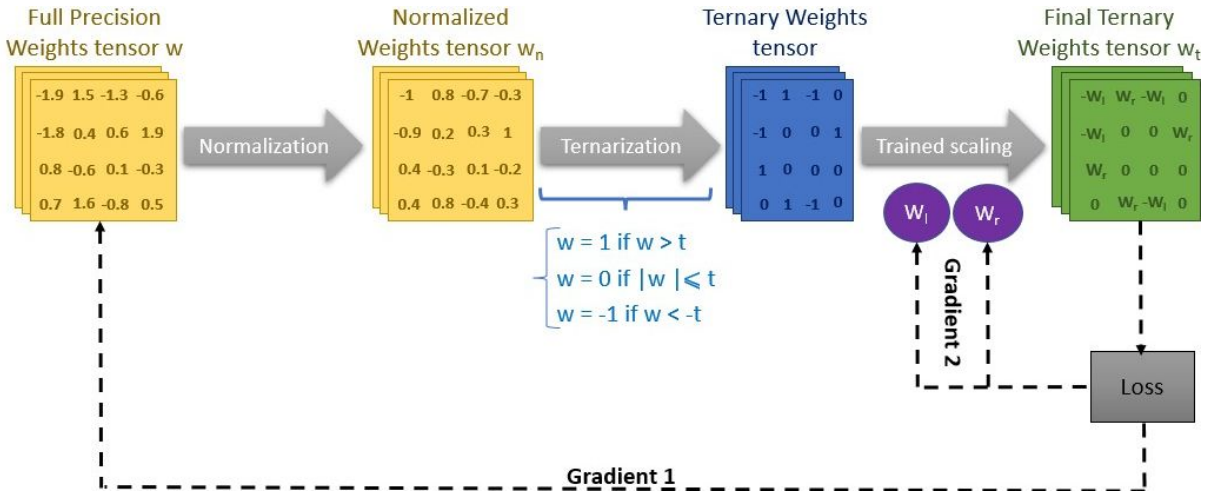


Figure 2.15: Overview of trained ternary quantization (TTQ) from (Zhu *et al.*., 2017)

solving an optimization problem where the ternarized weights approximate the FP ones, based on t and λ:

$$t^*, \lambda^* = argmin_{t \geq 0, \lambda \geq 0}(W - \lambda \times Q_t(W)) \tag{2.22}$$

The authors proposed an approximation of these two parameters to alleviate computation, based on the statistics of the weight matrix W: $t \approx 0.75 \times \mu(W)$ and $\lambda = \mu(\{w > t, w \in W\})$ where μ is the average operator. By the same token, (Zhu *et al.*., 2017), improved the previous method by introducing two trainable asymmetric scaling factors λ, γ and changing the ternarization heuristics (see Figure 2.15) corresponding to the way that the threshold t is computed: $t = \tau \times \max(W)$, where τ is a hyperparameter controlling the sparsity of the quantized weight matrix W. Therefore, the new quantization operator for a weight w of a matrix weight W is defined as follows:

$$Q_t^{\lambda,\gamma}(w) = \begin{cases} \lambda \text{ if } w < -t \\ 0 \text{ if } w \in [-t, t] \\ \gamma \text{ if } w > t \end{cases} \tag{2.23}$$

This approach allows decreasing the classification performance drop, while keeping a good compression rate.

Moreover, one important drawback of extreme quantization is the significant-created noise during training because of STE approximation of the gradients, which makes optimization difficult. (Fan, Stock*, , Graham, Grave, Gribonval, Jegou and Joulin, 2020) propose to solve this by selecting a random subset of weights to quantize instead of quantizing all the weights, allowing to keep some gradients without error, thus improving the gradient flow.

Finally, quantization can be combined with other model compression techniques to further improve the compression rates without important performance drop. Therefore, some approaches have proposed to combine quantization and pruning, as they are complementary (Han *et al.*., 2016; Park, Xu and Brick, 2018; Tung and Mori, 2020; Ullrich *et al.*., 2017). This can be done by a sequential application of the both techniques (Han *et al.*., 2016; Park *et al.*., 2018), using Bayesian optimization techniques (Tung and Mori, 2020), or soft-weight sharing (Ullrich *et al.*., 2017). We focus on Deep Compression (Han *et al.*., 2016), as it is one of the state-of-the-art approaches, and it can be easily adapted using newer and more efficient quantization and pruning methods. The method is composed of three components. The first one is network pruning, where small weights are removed based on a threshold. Once this is done, the model is re-trained to compensate the performance drop because of pruning. The second component

is trained quantization using weight sharing. In this part, weights are clustered during training using k-means, then taking the value of the respective centroid. Finally, to further compress the model, lossless Huffman encoding is applied to the final obtained weights.

V Conclusion

In this chapter we have made an overview of the main fields related to my work: cerebral and emboli detection and classification, data annotation, signal classification, and model compression.

In the following three chapters, we are going to detail the approaches that we developped to solve the limitations of some state-of-the-art methods presented in the current chapter.

3 | Data Annotation

Chapter based on our published conference paper (Vindas, Roux, Guépié, Almar and Delachartre, 2021) in the International Ultrasonics Symposium (IUS) 2021, and journal paper (Vindas, Guépié, Almar, Roux and Delachartre, 2022) in Medical Image Analysis (MEDIA) 2022.

As seen in chapters 1 and 2, data annotation is a particularly difficult and costly (but crucial) task in the medical field. In our case, as we work with portable TCD data, we have a huge amount of data as the recording of a patient can go from 30 to 180 minutes, having on average 14 HITS per minute. Nevertheless, the majority of the acquired data is not labeled, which can limit the training of deep learning models. To be able to quickly label an important quantity of data from a small number of labeled samples, different approaches have been proposed such as self-training (Rosenberg *et al.*., 2005), generative models (Kingma *et al.*., 2014), or label propagation (Zhu and Ghahramani, 2002; Weston *et al.*., 2012; Benato *et al.*., 2018, 2021). However, these methods have their limitations, which will be detailed in section I. Then, in section II we will present our proposed approach allowing to solve these limitations, and in section III we will validate it through a series of experiments.

Contents

I Motivation

I.1 Limitations of previous methods

The previous semi-automatic data annotation methods have achieved great annotation performances while reducing the annotation time. They are mainly based on the structure assumption, which states that samples living in the same structure (or manifold) are likely to have the same label.

The first limitation of these approaches is strongly related to this assumption. Indeed, in order for this assumption to be valid, the extracted features as well as the used dimensionality reduction techniques should represent the space/manifold in which the original samples (the images or signals) live, and where the structure assumption is verified. The above-mentioned works do not measure the quality of the reduced working latent space, which can lead to important annotation errors because the structure assumption is not verified in that latent space. In fact, if the working reduced latent space is strongly modified with respect to the original space, then samples from different classes can be close to each other, which disrupts neighborhood labeling using labeled samples.

The second limitation is related to annotation errors. Indeed, annotation errors are inherent to semi-automatic data annotation methods, specially when annotation is difficult for human annotators. These errors translate in noise in the labels, and come from different sources. First, if the human expert made some errors annotating the small amount of labeled samples needed to label the rest of the samples, then these errors will be propagated to the rest of the samples. Second, if the semi-automatic annotation method is applied iteratively, errors made by the method itself will be propagated too over the iterations. The previously mentioned approaches do not directly take into account these errors when using the semi-automatically labeled datasets for classification purposes.

Finally, most of the semi-automatic data annotation approaches are designed for classical machine learning classifiers (SVM, OPF-semi, random forest, etc.), and not evaluated in DL models which are known to be highly sensitive to label noise (Karimi, Dou, Warfield and Gholipour, 2020).

I.2 Objectives and contributions

We want to overcome the above-mentioned limitations by proposing a semi-automatic annotation and classification approach where numerous samples can be automatically labeled from a small amount of manually labeled ones. The obtained dataset can then be used to train DL models for classification under a noisy labels context. Our method is characterized by the following contributions:

- Novel approach for semi-automatic data annotation based on local quality metrics, allowing to control the annotation error.
- Selection strategy of the 2D projection used for automatic label propagation.
- Improvement of DL models trained on the obtained (semi-automatically labeled) datasets, thanks to noise-robust loss functions.

II Proposed method

In this section, we shall to detail our proposed semi-automatic data annotation method based on feature space projection, and local quality metrics.

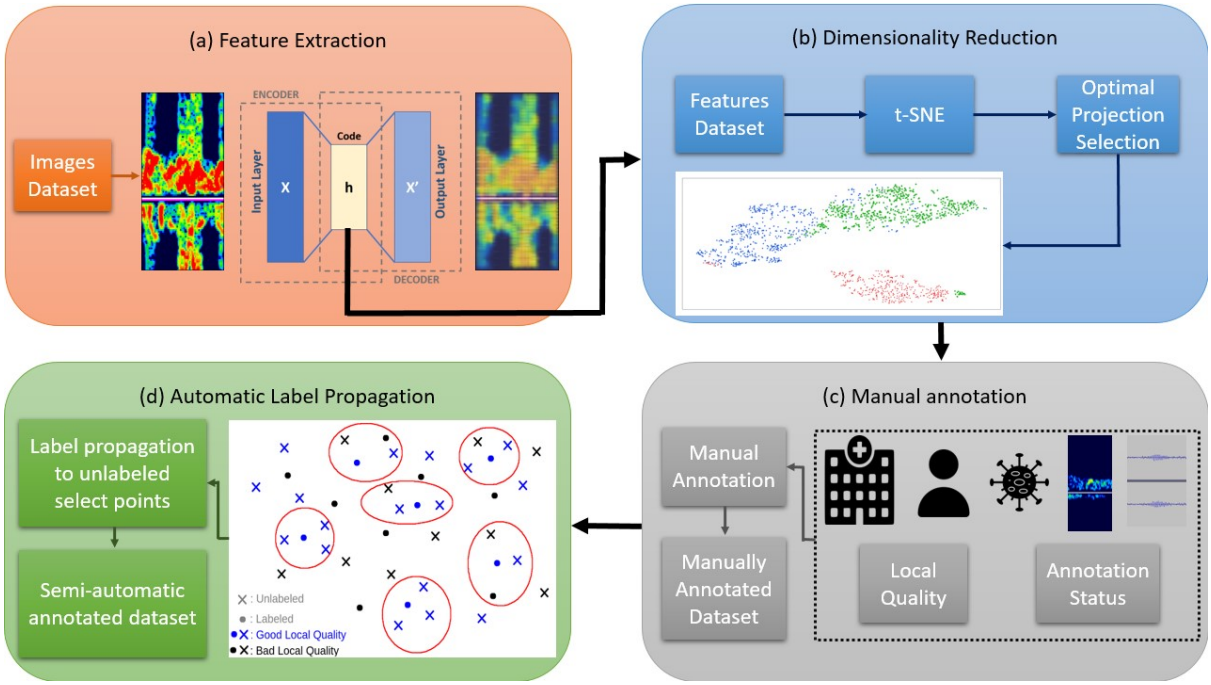


Figure 3.1: Global pipeline of our proposed label propagation approach. It is composed of four steps: (a) feature extraction through unsupervised learning, (b) dimensionality reduction through t-SNE, (c) manual annotation using the 2D reduced latent space, and (d) automatic label propagation from the labeled samples to the unlabeled ones, based on local quality metrics.

II.1 General overview and assumptions

Let us suppose that we have a dataset $\mathcal{D} = \mathcal{X} \times \mathcal{Y} = \mathcal{A} \cup \mathcal{U}$ with C classes composed of a small number A of labeled samples $\mathcal{A} = \mathcal{X}^A \times \mathcal{Y}^A = \{(X_1, y_1), ..., (X_A, y_A)\}$ ($|\mathcal{A}| = A$ is the cardinal of $\mathcal{A}$), and a large number of unlabeled samples $\mathcal{U} = \mathcal{X}^U \times \mathcal{Y}^U = \{(X_{A+1}, y_{A+1}), ..., (X_{A+U}, y_{A+U})\}$ ($|\mathcal{U}| = U$), with for all $i \in [1, A+U]$, $X_i \in \mathbb{R}^{d_I}$, $y_i \in [0,1]^C$ and $||y_i||_1 = 1$. The set of labels $\mathcal{Y}^U$ is unknown, so our main objective is to determine them using the samples in $\mathcal{A}$. To do so, we make the following three assumptions: **structure assumption** (Chapelle *et al.*., 2009), **preservation of the local structure** during dimensionality reduction, and **annotation space coverage**. The first assumption has been explained in the previous section I, and it mainly states that samples living in the same structure, are likely to have the same label. The second assumption asserts that, the reduced latent space obtained after applying a dimensionality reduction technique, should have a similar local structure to the original latent space, i.e., the neighborhoods of the samples should be globally preserved. The last assumption says that, the few initial manually labeled samples (samples in $\mathcal{A}$) should cover as much as possible the whole annotation space, so there should not be isolated regions without labeled samples (which allows reducing annotation errors during label propagation).

Furthermore, our global approach is composed of two main steps: (1) semi-automatic label propagation and (2) noisy-labels classification using noise-tolerant loss functions. The first step, (1), is subdivided in four parts (see figure 3.1):

- **Feature extraction**: We start by extracting features from the original input data using a DL unsupervised learning technique (auto-encoder), to bypass time-consuming feature handcrafting. For this step, all the samples in $\mathcal{D}$ are used, as the label information is not

necessary.

- **Dimensionality reduction**: Because of the high dimensionality of the latent space obtained in the previous step, it is necessary to further reduce its dimension. Indeed, (Benato *et al.*., 2021) showed that working on a reduced and interactive latent space allows more efficient manual and automatic labeling. In this step, several reduced latent spaces are generated, and the best one for semi-automatic label propagation is selected with a strategy detailed in the section II.3.

- **Manual annotation**: Our method relies on a small number of labeled samples to label a larger number of unlabeled ones. Because of this, a manual annotation step is necessary. This can be done using the original input data space (images), or using the 2D reduced latent space obtained in the previous step, with the help of the original data associated to each 2D sample. Moreover, it is important to take into account the **annotation space coverage** assumption when doing this process, as our method strongly relies on it. Therefore, manual annotation using the 2D reduce latent space is recommended.

- **Automatic label propagation**: At last, the obtained 2D reduced latent space is used to propagate the labels from the labeled samples to the unlabeled ones, based on local quality metrics. The use of local quality metrics allows ensuring the second assumption regarding the preservation of the local structure after dimensionality reduction. What is more, this step allows creating a high quality and richer training set (i.e., increase the size of $\mathcal{A}$) with a controlled annotation error and reduced annotation effort.

Finally, once the semi-automatically labeled dataset is obtained, DL models can be trained on it, using noise-tolerant loss functions to compensate the noise introduced in the labels by automatic label propagation.

II.2 Feature extraction

To avoid time-consuming manual design of features, we propose to use an unsupervised deep learning model to automatically extract features from the input data. This also has the advantage to extract data-specific feature from the inputs, being easier to use on different datasets than handcrafted features.

Moreover, the model that we propose to use is an auto-encoder (AE), denoted $AE = D \circ \mathcal{E}$, composed of two parts: an encoder $\mathcal{E}$ extracting features from the input data on a latent feature space $\mathcal{T} = \mathcal{E}(\mathcal{X}) \subset \mathbb{R}^{d_T}$ (with dimension $d_T >> 2$), and a decoder D which uses these feature to reconstruct the original samples. Thus, the objective of AE is to minimize the reconstruction error between $\mathcal{X}$ and its approximation $\hat{\mathcal{X}} = D(\mathcal{T})$.

The principle of AEs is generic and can be adapted to different types of data (e.g., images, volumes, time-series, text, ...), but in our case we use a convolutional AE as, in this chapter, we will directly work with images or TFRs which can be interpreted as images.

Ultimately, as our main objective is to annotate the largest number of samples in $\mathcal{U} \subset \mathcal{D}$ using $\mathcal{A} \subset \mathcal{D}$ (and not reconstruct samples from its encoded representations), we are going to use all the available samples in $\mathcal{D}$ to train the models.

II.3 Dimensionality reduction

Even if the dimension of the targeted feature space, d_T, is smaller than the dimension of the original space, d_I, the previous step allows doing a first dimensionality reduction step while doing automatic feature extraction. However, this dimension remains too large for manual

annotation and efficient automatic label propagation. Indeed, (Benato *et al.*, 2021) showed that working on a reduced low dimensional space allows doing a more efficient manual and automatic annotation. This is understandable for two main reasons. First, for human experts it is easier and faster to work in a 2D space where all the samples are visualizable at once, rather than in a high-dimensional space where only one sample (e.g., image) can be visualized at a time, and where no global or local structure is discernible. Second, doing automatic label propagation is more computationally and energy efficient in low dimensional spaces. This last point is important, as we want to reduce the annotation times, so fast automatic label propagation is necessary.

Furthermore, dimensionality reduction can be obtained using different techniques such as PCA (Jolliffe and Cadima, 2016), ISOMAP (Tenenbaum, de Silva and Langford, 2000), t-SNE (Maaten and Hinton, 2008), or UMAP (McInnes, Healy and Melville, 2020). These techniques often depends on different hyperparameters, and the obtained projection can drastically change based on the chosen ones (this will be illustrated in subsection III.4.B, figure 3.6). Therefore, it is important to define a criterion allowing to identify the best projection for the desired task, namely automatic label propagation. To do so, we propose to use the silhouette score (Rousseeuw, 1987), which allows measuring the compactness of each class cluster (*i.e.*, set of samples belonging to the same class), as well as their distances with respect to the other class clusters. This is interesting for neighborhood-based label propagation strategies, because we want to increase the inter-cluster distances and reduce the intra-cluster distances, as an important part of annotation errors come from the label propagation at the boundaries of two clusters corresponding to different classes.

More formally, let us assume that we have a dimensionality reduction technique P_γ with hyperparameters $\gamma \in \mathbb{R}^h$, allowing to reduce the dimension d_T of $\mathcal{T}$ to d_P, obtaining a final reduced latent space $\mathcal{F} = P_\gamma(\mathcal{T}) = \{P_\gamma(\mathcal{E}(X_1)), ..., P_\gamma(\mathcal{E}(X_A)), P_\gamma(\mathcal{E}(X_{A+1})), ..., P_\gamma(\mathcal{E}(X_{A+U}))\}$. Let us denote for all $p \in [1, C], C_p = \{j \in [1, A] \ / \ y_j = p\}$ the set of indices of the samples of class p. The silhouette score, S, is then defined as follows:

$$S(P_\gamma) = \frac{1}{A} \sum_{k=1}^{A} s(k) \tag{3.1}$$

where

$$\forall p \in [1, C], \forall k \in C_p, s(k) = \begin{cases} \frac{\mu_{inter}(k) - \mu_{intra}(k)}{max(\mu_{inter}(k), \mu_{intra}(k))} \text{ if } |C_p| \geq 2 \\ 0 \text{ else} \end{cases} \tag{3.2}$$

and where $\mu_{inter}(k)$ is the smallest mean distance between the labeled sample $P_\gamma(\mathcal{E}(X_k))$ and all the labeled samples for the other classes (i.e. smallest mean inter-cluster distance), whereas $\mu_{intra}(k)$ is the mean distance between the labeled sample $P_\gamma(\mathcal{E}(X_k))$ and all the labeled samples of the same class (i.e. mean intra-cluster distance). In simple terms, for all $p \in [1, C]$ and $k \in C_p$, the silhouette score compares the similarity of the k^{th} sample between the samples of its own class and the samples of the other classes. Note that, to compute S, we need the labels of the samples, so only the embeddings of the labeled samples $P_\gamma(\mathcal{E}(X_1)), ..., P_\gamma(\mathcal{E}(X_A))$ are used.

Moreover, to select the best projection for automatic label propagation using this metric, one can solve the following optimization problem:

$$\arg\max_{\gamma \in \mathbb{R}^h} S(P_\gamma) \tag{3.3}$$

For computational reasons, we relax the previous optimization problem, by limiting the possible hyperparameters to a smaller finite set $\mathcal{H} = \{h_i, \forall i \in [1, m], h_i \in \mathbb{R}^h\} \subset \mathbb{R}^h$ of m possible

combinations:

$$\arg\max_{\gamma \in \mathcal{H}} S(P_\gamma) \tag{3.4}$$

This problem can be simply solved by computing all the projections $P_{\gamma_1}, ..., P_{\gamma_m}$, computing their respective silhouette scores, and then choosing the one with the highest value. Subsequently, we denote as P_γ the best projection selected with our proposed strategy, for automatic label propagation.

At last, in our approach, we used t-SNE following the recommendations of (Benato *et al.*., 2018, 2021). This choice is justified by the second assumption of our method (preservation of the local structure): we are more interested in the local structure (neighborhood) of the feature spaces than in their global one, so methods focusing on this property are preferred over more global methods such as PCA or UMAP. What is more, t-SNE has three hyperparameters (perplexity, learning rate and early exaggeration) which have to be tuned to obtain the desired 2D reduced latent space. This is done by combining a grid search with the abovementioned projection selection strategy.

II.4 Automatic label propagation

The core of our method is the automatic label propagation step. This step is based on the concept of the local quality metric, lq, introduced in (Lueks *et al.*., 2011), and detailed in Chapter 2. We propose to use this metric to guide a k-nearest neighbor (KNN) label propagation strategy, allowing to select from which labeled samples we are going to propagate the labels, and to which unlabeled samples we are going to do it.

More formally, let us suppose that the previous steps were done, and that we have a suitable working latent space for label propagation obtained by a fine-tuned dimensionality reduction technique P_γ. As these steps were done, we suppose that the structure assumption is verified in the high-dimensional space (the one of the AE), and in the low-dimensional one (the one obtained by P_γ). The local quality of sample $i \in [1, A+U]$, denoted as $lq(P_\gamma(\mathcal{E}(X_i)), k_s, k_t) = Q_A^i(k_s, k_t)$, measures the neighborhood modification of the projected sample $P_\gamma(\mathcal{E}(X_i))$ with respect to the encoded representation by the AE, $\mathcal{E}(X_i)$. Let us recall that k_s controls the size of the neighborhood that it is used to compare the samples in the high and low dimensional manifolds[1], whereas k_t controls the tolerated rank errors[2] (in other words, rank errors below k_t will not be considered as errors).

Furthermore, following the notations of (Zhu and Ghahramani, 2002), let us denote as $Y \in \mathbb{R}^{(A+U)\times N}$ the label matrix, where the first A rows correspond to the labeled samples and the last U rows correspond to the unlabeled samples. As we work with probabilistic labels, the element of the i^{th} row and the p^{th} column, Y_{ip}, represents the probability that the sample i is from class p. Moreover, let us denote as $T^{K,\tau} \in R^{(A+U,A+U)}$ a probabilistic transition matrix, where $T_{ij}^{K,\tau}$ is the probability to pass from sample i to sample j. This probability depends on the size K of the neighborhood $\mathcal{V}_K(P_\gamma(u))$ used to search for labeled neighbors of an unlabeled sample $u \in \mathcal{U}$ (K must not be confused with k_s, which is used to compute the local quality of the $P_\gamma(X_i)$ for $i \in [1, A+U]$), and τ, the threshold used to determine whether the local quality of a point is considered acceptable or not (τ must not be mistaken for k_s, which is used to compute the local quality). We define $T^{K,\tau}$ based on the nearest-neighbors method and the local quality:

$$\forall i, j \in [1, A+U], T_{ij}^{K,\tau} = \begin{cases} 1 \text{ if } (i, j \in [1, A] \wedge i = j) \vee (i, j \in \mathcal{P}_{K,\tau}) \vee (i = j \wedge i \in C_\tau) \\ 0 \text{ else} \end{cases} \tag{3.5}$$

[1]The higher k_s, the more demanding the global quality (and not the local one) as we ask for less rank errors in larger neighborhoods.

[2]The higher k_t, the more errors are tolerated. This implicates that the local quality becomes less informative.

with

- $\mathcal{P}_{K,\tau} = \{i \in [A+1, A+U], j \in [1, A] / P_\gamma(X_i) \in \mathcal{V}_K(P_\gamma(X_j)), lq(P_\gamma(X_i), k_s, k_t) > \tau, lq(P_\gamma(X_j), k_s, k_t) > \tau, \forall f \in \mathcal{V}_K(P_\gamma(X_i)), lq(P_\gamma(X_j), k_s, k_t) > lq(f, k_s, k_t)\}$, where
  - $P_\gamma(X_i) \in \mathcal{V}_K(P_\gamma(X_j))$ means that the embedded representation of the unlabeled sample i is in the K-neighborhood of the embedded representation of the labeled sample j;
  - $lq(P_\gamma(X_i), k_s, k_t) > \tau, lq(P_\gamma(X_j), k_s, k_t) > \tau$ means that the local quality of samples X_i and X_j are greater than the defined threshold τ;
  - $\forall f \in \mathcal{V}_K(P_\gamma(X_i)), lq(P_\gamma(X_j), k_s, k_t) > lq(f, k_s, k_t)$ means that the embedded representation of the labeled sample X_j is the one that has the best local quality in the K-neighborhood of the unlabeled sample X_i.
- $C_\tau = \{i \in [A+1, A+U] / lq(P_\gamma(X_i), k_s k, k_t) < \tau\}$ is a set that contains all of the unlabeled samples with a local quality score smaller than the defined threshold τ. These samples will not be taken into account for label propagation.

The set $\mathcal{P}_{K,\tau}$ allows label propagation from the labeled samples to their unlabeled neighbors based on a local quality criterion, while the set C_τ avoids labeling samples that do not respect this criterion. We can now define our label propagation algorithm as in (Zhu and Ghahramani, 2002):

- Propagate the labels from the good local quality labeled samples to the good local quality unlabeled samples: $Y \leftarrow T^{K,\tau} \times Y$;
- Row normalize Y (by construction of $T^{K,\tau}$, Y is row-normalized)[1];
- Update $T^{K,\tau}$ by considering adding the new labeled samples to $\mathcal{A}$;
- Repeat the process until there are no more samples to label (or until some number of iterations is reached).

Now, we can directly compare our proposed approach with the one of (Benato *et al.*., 2018, 2021). Indeed, in our method, the transition matrix T is computed through KNN and a local quality criterion, whereas in (Benato *et al.*., 2021) the transition matrix T is computed using Laplacian SVM and OPF-semi.

Finally, our algorithm converges as it will finish when there are no more samples to label, or when there are no more unlabeled samples with local quality greater than the established threshold. The final algorithm of our method is presented in Algorithm 1.

II.5 Classification with noisy labels

The final step of our approach consists in classification. Indeed, our main objective is to classify HITS between artifacts, gaseous emboli, and solid emboli, but we only dispose of a small limited number of labeled samples. The previous steps allow us to get a larger labeled samples of HITS with reduce annotation effort. However, the obtained semi-automatically labeled datasets are by nature noisy, mainly because of automatic label propagation, which can disrupt the learning process of DL models. Therefore, to efficiently train DL models on these datasets without decreasing their final performances, one need to take into account this noise

[1]As its name indicate it, row-normalization is done with respect to each row. This step is done to still have a label probability interpretation of Y, if not, the sum of the elements in one row can be smaller or greater than 1.

Algorithm 1: Local quality with KNN (LQ-KNN) label propagation algorithm.

Input: $\mathcal{D} = \mathcal{A} \cup \mathcal{U}$, k_s, k_t, K, τ
Output: New labeled dataset $\tilde{\mathcal{D}}$
Initialization: $\mathcal{P}_{K,\tau}$ and $T^{K,\tau}$ using their definition.
Iterations:

- Extract features of ALL of the samples using an auto-encoder model.
- Dimensionality reduction of the previous representations:
  - Apply t-SNE with grid search;
  - Select the best projection P_γ using the silhouette score;
  - Obtain the embedded representations $P_\gamma(X_1), ..., P_\gamma(X_i A + U)$ of the samples using P_γ;
  - Compute the local quality $lq(., k_s, k_t)$ of each sample;
  - Sort the representations obtained by decreasing the local quality.
- Propagate the labels using the local quality of the embedded representations:

while $\mathcal{P}_{K,\tau} \neq \emptyset$ **do**
  $Y \leftarrow T^{K,\tau} \times Y$;
  Row normalize Y;
  Update $\mathcal{A}$, $\mathcal{U}$, $T^{K,\tau}$ and $\mathcal{P}_{K,\tau}$;
end

- Define $\tilde{\mathcal{D}} = \mathcal{A}$

during the training process. As seen in Chapter 2, several techniques allow learning under a noisy-labels context, but here we propose to use noise-tolerant loss functions, as they can easily adapt to new datasets/architectures, do not need pre-training nor the knowledge of the noise rate, and allow using all the available labeled samples.

More precisely, we propose to use the generalized cross-entropy (GCE) (Zhang and Sabuncu, 2018) which presents interesting noise tolerance properties while still having a good convergence rate, thanks to the hyperparameter q. Following the recommendations of (Zhang and Sabuncu, 2018), we fix $q = 0.7$ as it allows getting a good compromise between convergence speed and noise tolerance.

III Method evaluation

To validate our method, we proposed different experiments evaluating the main aspects of our method: automatic label propagation, projection selection, and classification under noisy labels. To do so, we perform four experiments that will be detailed in the following sections, using three different datasets.

III.1 Datasets

We used three datasets for the different conducted experiments: two publicly available ones, MNIST (LeCun and Cortes, 2010) and OrganCMNIST (Yang, Shi, Wei, Liu, Zhao, Ke, Pfister and Ni, 2021; Bilic, Christ *et al.*., 2019), and one private HITS dataset.

III.1.A MNIST and OrganCMNIST

MNIST: The MNIST dataset is a well-known machine learning dataset composed of 28×28 grayscale images of handwritten digits, distributed in 10 classes. The original dataset is composed of 60 000 training labeled samples, and 10 000 testing labeled samples. However, for computation and energy consumption reasons, we only used a subset of this dataset, composed of 15 000 randomly sampled training samples (25% of the original training dataset), and 10 000 testing samples (all the available testing samples).

OrganCMNIST: We also validate our approach on a publicly available dataset, OrganCMNIST, composed of 28×28 computed tomography images of 11 different organs. The dataset includes 15 392 training labeled samples, and 8 268 testing labeled samples. All the available samples (training and testing) were used to train and evaluate our method.

III.1.B HITS

We apply our approach to a more realistic and difficult dataset, composed of HITS samples extracted from different subjects coming from different healthcare centers. This dataset, detailed hereafter, is by nature minimally labeled, with potential manual annotation errors.

Data acquisition: TCD recordings of a duration between 30 and 180 min were performed on 51subjects (see table 3.1 for the details about the population) from 11 different healthcare centers (in France, Switzerland, Belgium, England and The Netherlands) using two Atys Medical devices (TCD-X Holter or WAKIe R3) with an insonation frequency of 1.5 MHz. The recordings were done under heterogeneous conditions as the subjects can have different pathologies (carotid stenosis, patent foramen ovale or none), come from different care units (neurovascular

Table 3.1: HITS subject's population characteristics computed with the available information. F stands for female, M for male, and U for unknown.

Sex	Number	Median Age	Range Age	Mean n° HITS/min
F	25	73.5	24-91	13.15
M	20	62.5	21-88	14.27
U	7	74.5	71-78	12.99
All	51	68.5	21-91	13.56

or cardiovascular) were injected with different contrast agents (sonovue, iodine-containing contrast agent, or none), and underwent different procedures (TAVI, AFA, or none). What is more, according to the recommendations to monitor the MCA for emboli detection, the following acquisition information was used:

- Pulse repetition frequency: 4.4-6.2 kHz;
- Transmitted ultrasound frequency: 1.5 MHz;
- Insonation depth: 45 – 55 mm;
- Sample volume: 8 – 10 mm^3.

Data pre-processing: From the abovementioned recordings, Atys medical data management software (ADMS), was used to compute their log-scale scale spectrogram, detect HITS using the criteria in Chapter 1, and export the obtained HITS into raw signals (WAV files) and TFRs (PNG images files). In a nutshell, we used a high-pass 4^{th} order Butterworth digital filter with a cutting frequency of $150Hz$, a detection threshold for the HITS of $7dB$, a gain of $6dB$, and no noise reduction. For the spectrogram computation, the length of the windowed signal after padding with zeros[1] was of $n_{fft} = 128$, the overlap size of $n_{overlap} = 8$, and a Blackman window was used. This resulted in 68 492 total HITS, among which 1 545 were manually labeled according to four classes: 569 solid emboli, 569 gaseous emboli, 403 artifacts, and 4 unknown. We refer the reader to appendix 2 for the details about the HITS distribution per subject.

Final datasets: We distinguish two datasets: (1) the fully labeled one composed of 1 541 labeled ones[2], henceforth referred as the HITS-small dataset, and (2) the partially labeled one, from now on referred as the HITS-large dataset, composed of 68 492 HITS, where only 1 541 are labeled (the same as the HITS-small dataset).

III.2 Baselines

Automatic data annotation: In all the label propagation experiments, we compared three models: LQ-KNN (our proposed approach based on local quality metrics and KNN), Std-KNN (standard KNN without local quality metrics), and OPF-semi (Amorim *et al.*., 2014) which is commonly used for data annotation. To implement the OPF-semi models, we used the python library OPFython (De Rosa and Papa, 2021), whereas for the first two, we used Scikit-Learn (Buitinck, Louppe, Blondel, Pedregosa, Mueller, Grisel, Niculae, Prettenhofer, Gramfort, Grobler, Layton, VanderPlas, Joly, Holt and Varoquaux, 2013).

[1]Padding with zeros is only done if necessary.

[2]We do not use the *unknown* HITS (coming from only 39 patients), as we only have four of none expert annotator was able to determine their true class.

Classification under noisy labels: We trained different CNNs on the different datasets, using two different loss functions: the CE (non-noise-tolerant), and the GCE (noise-tolerant). The different models were trained on a clean non-noisy dataset, as well as on artificially (symmetric) noisy datasets, and semi-automatically labeled datasets.

III.3 Evaluation metrics and strategy

To validate our method, we have to consider two perspectives: automatic data annotation through label propagation, and classification under noisy-labels. Therefore, for each task, we need to select the correct evaluation metrics and strategies.

III.3.A Automatic data annotation

Evaluation metrics: To measure the performance of the different automatic data annotation methods, we use two metrics: the annotation accuracy α, and the percentage of labeled samples p_A. The metrics are defined as follows:

$$\alpha = \frac{|\mathcal{G}|}{|\tilde{\mathcal{A}}|} \text{ and } p_A = \frac{|\tilde{\mathcal{A}}|}{|\mathcal{U}|}$$

where $\tilde{\mathcal{A}}$ is the set of all the newly automatically labeled samples, $\mathcal{G}$ is the set of correctly automatically labeled samples from $\tilde{\mathcal{A}}$ and $|.|$ is the cardinal operator.

Evaluation strategy: For all the fully labeled datasets, we consider that only 10% of the samples are annotated i.e. $\mathcal{A}$ is composed of 10% of the available labeled samples (randomly selected), and $\mathcal{U}$ is composed of the remaining ones[1]. Therefore, for the selected projection P_B, we used only a part of the labeled samples (10%), and we propagate the labels from these samples to the rest of the samples on the 2D reduced latent space obtained by P_B.

Moreover, to compute the local quality of the samples, we fixed $k_s = 10$ and $k_t = 10$. Three reasons motivate this choice. First, we observe experimentally that the values of k_s and k_t do not have an important influence on the annotation accuracy, as long as their values are not too large (i.e., less than 50); this will be further discussed in subsection III.5.E. Second, we want to have good local qualities in small neighborhoods in order to do a better label propagation. Third, as abovementioned, large values of k_s and k_t can lead to misleading values of local quality, where high values do not accurately translate the preservation of the local structure.

At last, for statistical purposes, each label propagation experiment was repeated 50 times. On top of that, for the MNIST dataset, 10 different auto-encoder models were trained, allowing to get tighter statistical results.

III.3.B Classification

Evaluation metrics: To evaluate the classification performances of the different trained classifiers, we used two metrics: the accuracy and the Matthew's correlation coefficient (MCC). The first metric is used for the MNIST and OrganCMNIST datasets, whereas the both are used for the HITS datasets. Indeed, to facilitate comparison with other works, accuracy is preferred for the MNIST and OrganCMNIST datasets, as it is commonly used. For the HITS datasets, the MCC is a good metric as it translated better the classification performances under imbalanced classes, as it will be the case for the semi-automatically labeled finally HITS-large dataset.

[1]This allows us to evaluate the annotation performance of automatic label propagation, as we know the true label of the automatically labeled samples.

Table 3.2: Training parameters of the auto-encoders used in experiment 1. MSE stands for mean squared error loss.

Dataset	Epochs	Batch Size	Learning rate	Optimizer	Weight Decay	Loss function
MNIST	50	32	5e-2	1e-5	Adamax	MSE
OrganCMNIST			5e-5	1e-7		
HITS			5e-3	1e-2		

Evaluation strategy: Two different evaluation strategies were used, based on the dataset:

- **HITS datasets**: We used a leave-one-subject-out based strategy. First, label propagation is used to automatically label a part of the unlabeled samples. For the HITS-small dataset, only 10% of the samples are considered as labeled, whereas for the HITS-large dataset, all the labeled samples are used as this dataset is partially labeled. Second, different train/test splits are created by taking as test samples the manually labeled samples of a fixed subject and as train samples all the (manually and automatically) labeled samples of the rest of the subjects. In this way, we get 39 or 51train/test splits. Finally, different models are trained using these splits, and the process is repeated 10 times for the HITS-large dataset and 20 times for the HITS-small dataset.
- **MNIST and OrganCMNIST**: We used 50 times repeated holdout as evaluation method. The final training set was obtained by considering only 10% of the training samples as labeled, and then propagating their labels to the rest of the training samples. This was not done for the test set, which remained untouched and composed only of the manually labeled testing samples (10000 for MNIST and 8268 for OrganCMNIST).

III.4 Experimental setup and results

We conducted four series of experiments to validate the different components of our approach. Hereafter, we are going to detail the experimental setups, as well as the obtained results for each experiment.

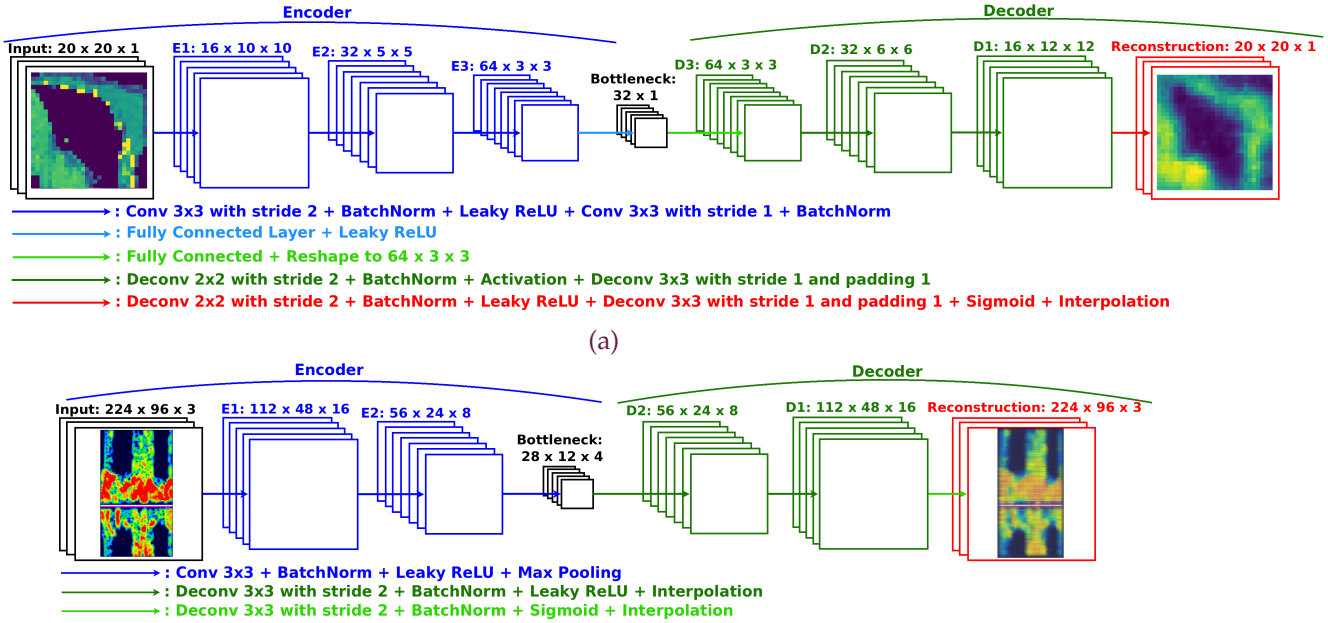
III.4.A Experiment 1: Automatic data annotation evaluation

Objective: The objective of this experiment is to test our automatic label propagation method on three different datasets: the MNIST subset, OrgnaCMNIST, and HITS-small.

Experimental setup: We used two different AE architectures, which can be found in figure 3.2 with the parameters established in table 3.2. What is more, to find the projection to use, three t-SNE parameters were fine-tuned using grid search (learning rate, perplexity, and early exaggeration), and the explored values can be found in table 3.3. At last, the parameters that we used for label propagation for each dataset can be found in table 3.4.

Results: The results of this first experiment can be found in table 3.5 and figures 3.3, 3.4, and 3.5. Several interesting points can be highlighted.

Firstly, from table 3.5 we can see that LQ-KNN outperforms OPF-semi in all the tested datasets, and this, with annotation times reduced by a factor of at least 10^2. In fact, even though LQ-KNN (nor Std-KNN) is not able to annotate all the available samples as OPF-semi, it is able to annotate more than 96% of the unlabeled samples with annotation accuracies greater than



(a)

(b)

Figure 3.2: Auto-encoder architectures used. (a) Architecture for the MNIST and OrganCMNIST datasets. (b) Architecture for the HITS dataset.

Table 3.3: Parameters for the grid search in experiment 1. As the HITS and OrganCMNIST datasets are more complex than the MNIST dataset, a more complete grid search is needed to find the projections to keep.

Dataset	Perplexity	Early Exaggeration	Learning rate
MNIST	[10, 30, 50]	[50, 250, 500]	[10, 100, 1000]
OrganCMNIST HITS	[5, 10, 15, 20, 25, 30, 35, 40, 45, 50]	[5, 10, 25, 50, 75, 100, 200, 500]	[10, 50, 100, 500, 1000]

82% against 78% for OPF-semi. What is more, compared to the baseline Std-KNN, LQ-KNN is able to annotate more samples with comparable accuracies. For instance, for the HITS dataset, LQ-KNN annotates around 98.50% of the available samples with an annotation accuracy of 82.67%, against 95.99% and 82.12% respectively for Std-KNN. A similar phenomenon can be observed for the other datasets.

Secondly, from Figures 3.3, and 3.5 we can see that, the proportion of finally labeled samples, p_A, as well as the annotation accuracy, α, are dependent from the chosen neighborhood for KNN label propagation, K. Actually, the smaller K, the higher α but the smaller p_A; inversely, the higher K, the higher p_A but the smaller α. By the same token, there is a dependency between τ, the threshold allowing to define good quality samples, and α and p_A. Indeed, smaller values of τ gives higher values of p_A but smaller ones of α, whereas increasing τ allows to increase α but decreases p_A.

Thirdly, from figure 3.3 we can identify two main label propagation regimes. The first regime, designed as 'dynamic regime', corresponds to small values of K, and is characterized by a decreasing annotation accuracy but an increasing number of labeled samples. The second regime, referred as the 'permanent' regime, corresponds to higher values of K, and is characterized by a plateau for α and p_A.

Table 3.4: Parameters used for label propagation in experiment 1. As all the tested datasets are fully manually labeled, we randomly select 10% of the samples and consider them as labeled ($\mathcal{A}$), whereas the rest are considered as unlabeled ($\mathcal{U}$). Then, we propagate the labels from the labeled samples to some of the unlabeled ones using one of the propagation methods. The experiments using the HITS-small dataset were repeated 50 times. The experiments using the MNIST dataset were repeated 50 times for 10 different auto-encoders (500 repetitions in total), except for OPF-semi, where we did 20 repetitions for each auto-encoder. K corresponds to the size of the neighborhood used to search for labeled neighbors for an unlabeled sample.

Exp. name	Dataset	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{U}\rvert$	Propagation	K	τ	Repetitions
Std-KNN	MNIST	1496	13504	Std-KNN	$1 \leq K \leq 20$	-	500
	HITS-small	152	1393			-	50
	OrganCMNIST	1534	13858			-	50
LQ-KNN-τ	MNIST	1496	13504	LQ-KNN		$0.1 \leq \tau \leq 0.5$	500
	HITS-small	152	1393				50
	OrganCMNIST	1534	13858				50
OPF-semi	MNIST	1496	13504	OPF-semi	-	-	200
	HITS-small	152	1393		-	-	50
	OrganCMNIST	1534	13858		-	-	20

Finally, we examine the importance of the label propagation order. To do this, we fix the used projection where label propagation is done, and then propagate the labels from the labeled samples to the unlabeled ones, using two strategies: (1) sorting the samples by decreasing local quality values, and (2) without sorting the samples. Figure 3.4 indicates that strategy (1) achieves higher annotation accuracies than strategy (2). On top of that, the annotation accuracy performance gap becomes more important when the neighborhood size used for label propagation, K, increases.

III.4.B Experiment 2: Validation of the projection selection strategy

Objective: The objective of this experiment is to validate our projection selection strategy, used for efficient label propagation.

Experimental setup: We used the same parameters as in the previous experiment, but we only used the HITS-small dataset. From the obtained t-SNE projections, for each dataset, we selected the best and worst projections in terms of silhouette scores. This gives us two projections per dataset, as seen in figure 3.6 for the HITS (silhouette scores of 0.54 and -0.23 for the best and worst projections, respectively). Then, we propagate the labels as in experiment 1, using these two projections.

Results: Results can be found in table 3.6. We observe that, for all values of K, and for both KNN-based label propagation methods, the best annotation accuracies are obtained when the selected projection is used, and this with an important margin with respect to the worst projection. Additionally, we can note that the number of labeled samples is often higher when the best projection is used. What is more, we can also note that, even when we use the worst projection, LQ-KNN tends to give higher annotation accuracies than Std-KNN, at the expense of a lower number of labeled samples.

Table 3.5: Experiment 1: Label propagation results. $\mathcal{A}$ corresponds to the set of initially (manually) labeled samples, $\mathcal{U}$ corresponds to the set of initially unlabeled samples, τ corresponds to the local quality threshold defining if a sample is of good quality, K corresponds to the size of the neighborhood used to search for labeled neighbors for an unlabeled sample, α is the annotation accuracy, p_A the final percentage of labeled samples, and the annotation time corresponds to the one needed to label one sample in ms. Our proposed approach LQ-KNN outperforms OPF-semi (Amorim *et al.*., 2014) and the baseline Std-KNN, at the expense of a smaller number of labeled samples. Additionally, LQ-KNN and Std-KNN are faster than OPF-semi by a factor of at least 10^2.

Dataset	Propagation method	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{U}\rvert$	τ	K	α	p_A	Annotation time (ms)
MNIST	Std-KNN	1496	13504	-	5	91.83 ± 1.47	95.39 ± 1.05	$(30.98 \pm 5.84) \times 10^{-3}$
				-	10	90.74 ± 1.45	99.43 ± 0.23	$(\mathbf{28.78 \pm 5.13}) \times \mathbf{10^{-3}}$
	LQ-KNN			0.1	5	**93.12 ± 1.36**	93.88 ± 0.66	$(59.10 \pm 12.35) \times 10^{-3}$
					10	92.66 ± 1.30	98.16 ± 0.42	$(50.48 \pm 11.32) \times 10^{-3}$
	OPF-semi			-	-	82.32 ± 6.17	**100.0 ± 0.0**	102.71 ± 17.52
OrganCMNIST	Std-KNN	1534	13858	-	5	81.87 ± 0.76	90.26 ± 2.64	$(26.33 \pm 2.65) \times 10^{-3}$
				-	10	79.86 ± 0.67	99.00 ± 0.20	$(\mathbf{23.41 \pm 1.98}) \times \mathbf{10^{-3}}$
	LQ-KNN			0.1	5	**84.46 ± 0.57**	85.62 ± 1.99	$(53.00 \pm 7.47) \times 10^{-3}$
					10	82.73 ± 0.44	96.24 ± 1.09	$(44.36 \pm 5.69) \times 10^{-3}$
	OPF-semi			-	-	75.22 ± 4.48	**100.0 ± 0.0**	86.52 ± 0.51
HITS-small	Std-KNN	152	1393	-	5	82.12 ± 2.37	95.99 ± 1.70	$(10.39 \pm 0.20) \times 10^{-2}$
				-	10	81.36 ± 1.81	99.58 ± 0.63	$(\mathbf{10.04 \pm 0.18}) \times \mathbf{10^{-2}}$
	LQ-KNN			0.1	5	**82.84 ± 2.12**	94.48 ± 1.72	$(16.87 \pm 0.48) \times 10^{-3}$
					10	82.67 ± 2.02	98.50 ± 0.80	$(16.13 \pm 0.35) \times 10^{-2}$
	OPF-semi			-	-	78.40 ± 13.44	**100.0 ± 0.0**	9.48 ± 1.1

Table 3.6: Experiment 2. Label propagation for the HITS-small dataset using the best and worst selected 2D projections according to the silhouette scores. The best selected projection allows automatic annotation of more samples with higher annotation accuracy. S, silhouette score; $\mathcal{A}$, set of initially (manually) labeled samples; $\mathcal{U}$, set of initially unlabeled samples; τ, local quality threshold that defines if a sample is considered as of good quality; α, annotation accuracy; p_A, proportion of labeled samples.

K	Propagation method	Projection	S	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{U}\rvert$	τ	α	p_A
5	**Std-KNN**	**Best**	**0.53 ± 0.05**	**152**	**1393**	**-**	**82.11 ± 2.37**	95.99 ± 1.70
		Worst	−0.26 ± 0.07			-	58.43 ± 8.95	**98.03 ± 1.22**
	LQ-KNN	**Best**	**0.53 ± 0.04**			**0.1**	**82.84 ± 2.12**	**94.47 ± 1.72**
		Worst	−0.25 ± 0.09			0.1	70.87 ± 7.18	68.57 ± 10.16
15	**Std-KNN**	**Best**	**0.53 ± 0.05**			**-**	**80.31 ± 2.03**	**99.97 ± 0.07**
		Worst	−0.26 ± 0.07			-	56.44 ± 9.83	99.66 ± 0.20
	LQ-KNN	**Best**	**0.53 ± 0.04**			**0.1**	**82.82 ± 1.96**	**98.84 ± 0.67**
		Worst	−0.25 ± 0.09			0.1	66.16 ± 8.76	79.92 ± 7.531

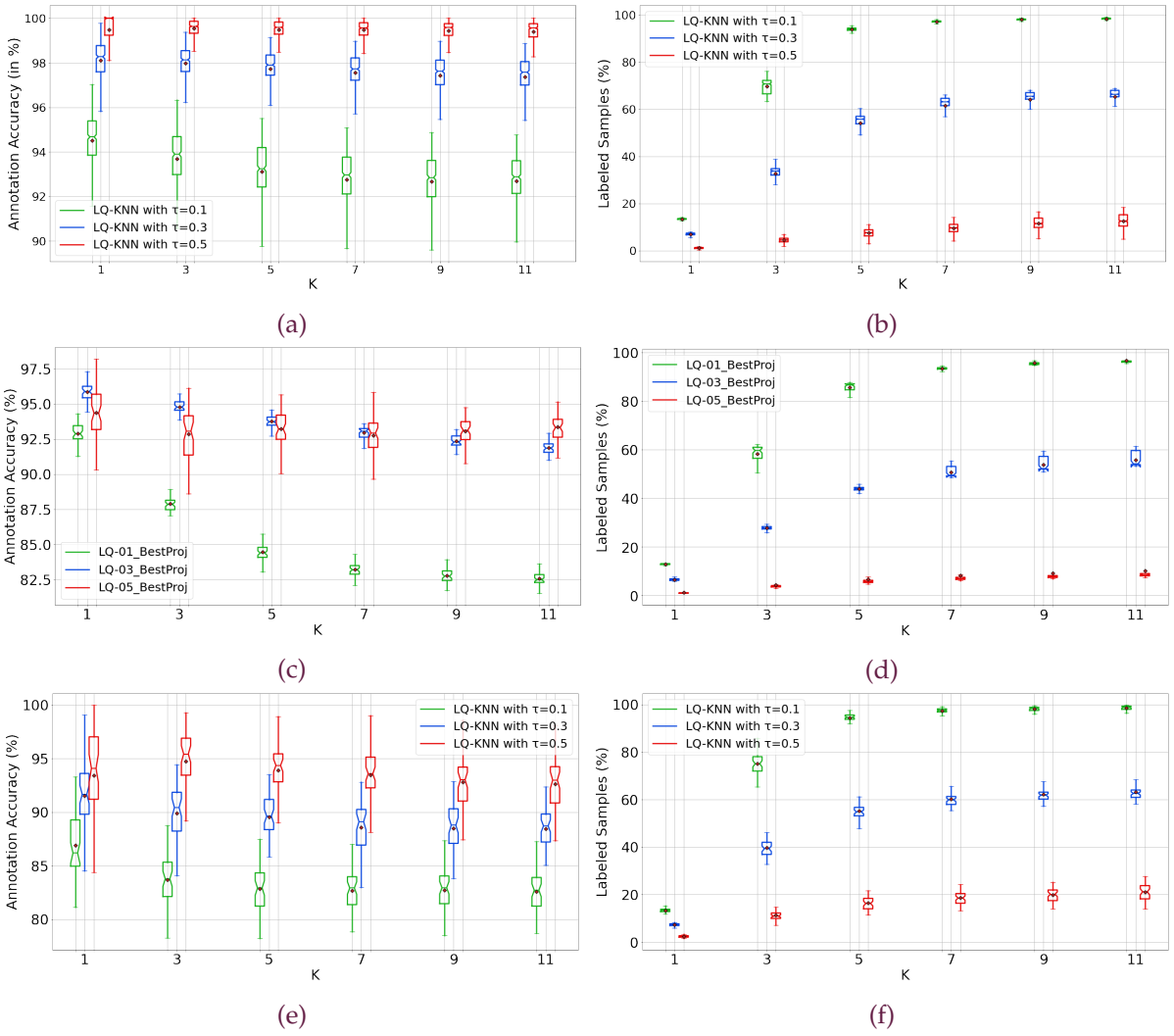
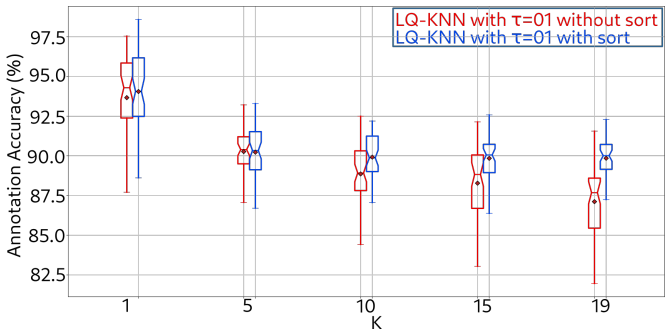


Figure 3.3: Experiment 1: Label propagation hyperparameter study. (a) MNIST dataset annotation accuracy. (b) MNIST dataset labeled samples (in %). (c) OrganCMNIST dataset annotation accuracy (in %). (d) OrganCMNIST dataset labeled samples (in %). (e) HITS-small dataset annotation accuracy. (f) HITS-small dataset labeled samples (in %). τ corresponds to the threshold used to define good local-quality samples. For LQ-KNN: purple curves, $\tau = 0.1$; green curves, $\tau = 0.3$; blue curves, $\tau = 0.5$. The proportion (%) of unlabeled samples that were labeled by the methods converges with K, hence we show here the results for $K \leq 11$.

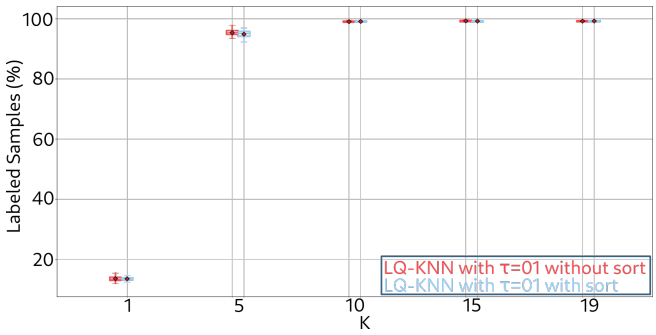
III.4.C Experiment 3: Classification on a dataset with known label noise

Objective: This experiment has two objectives. Firstly, we want to measure the classification performances changes, when using a semi-automatically (noisy) labeled dataset obtained with our approach. Secondly, we want to confirm if a noise-tolerant loss function can compensate for the noise introduced in the labels by automatic label propagation.

Experimental setup: We trained a CNN classifier (see figure 3.7) with different parameters (see table 3.8) on different datasets (see table 3.7). In order to work on a controlled environment, in this experiment we only work using fully manually labeled datasets, where only 10% of the samples are considered as labeled. Therefore, we can accurately estimate the noise rate, ζ,



(a)



(b)

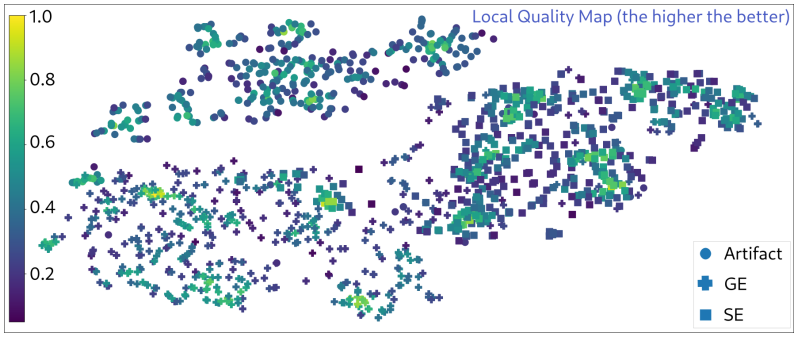
Figure 3.4: Experiment 1: Evaluation of the importance of the propagation order on the HITS-small. (a) Annotation accuracy (in %). (b) Labeled samples (in %). For LQ-KNN with $\tau = 0.1$: blue curves, starting by labeling the higher local quality samples (the samples are sorted by decreasing local qualities); red curves, without taking into account the propagation order (the samples are not sorted by decreasing local qualities).

of the obtained semi-automatically labeled dataset using the results of experiment 1. Thus, we estimate the noise rate created by a label propagation method as $\zeta_{est} = 1 - \alpha$, with α the annotation accuracy of the method. Indeed, as seen in Chapter 2, in order for GCE to be relatively robust against noisy labels, the noise rate should verify $\zeta < \frac{C-1}{C}$, with C the number of classes in the dataset.

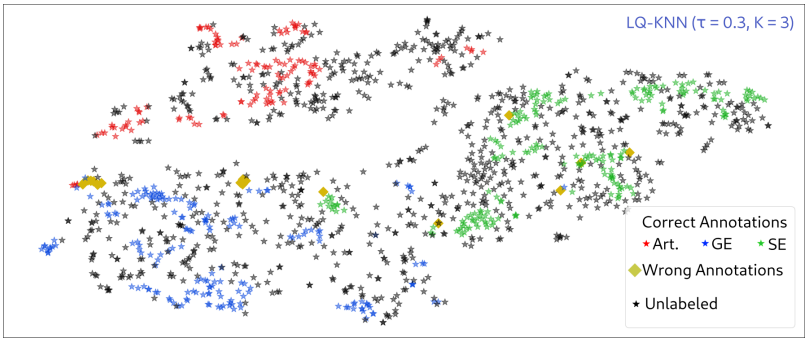
Results: The results of this experiment can be found in figures 3.8 (MNIST), 3.9 (OrganCMNIST), 3.10, 3.11, 3.12, and 3.13 (HITS-small).

On the one hand, if we focus on the OrganCMNIST results (figure 3.9), three main phenomena can be observed. First, we notice that, for a fixed loss function, the best classification performances are obtained when we train the classifier on an LQ-KNN dataset, achieving a global accuracy of 75.76% with OrganCMNIST LQ-KNN and GCE opposed to 74.58% with OrganCMNIST Std-KNN and GCE, or 70.62% without label propagation and GCE. Secondly, we can note that the robust loss function has an important impact on the final classification performances, being able to increase the accuracy by more than 2% when noise is present. Finally, we can see that even when we do not use a robust loss function (*i.e.*, when we use CE), our label propagation method provides models with better classification performances (72.73%) than the baseline OrganCMNIST Std-KNN (71.71%). A similar trend is observed in figure 3.8 for the MNIST dataset.

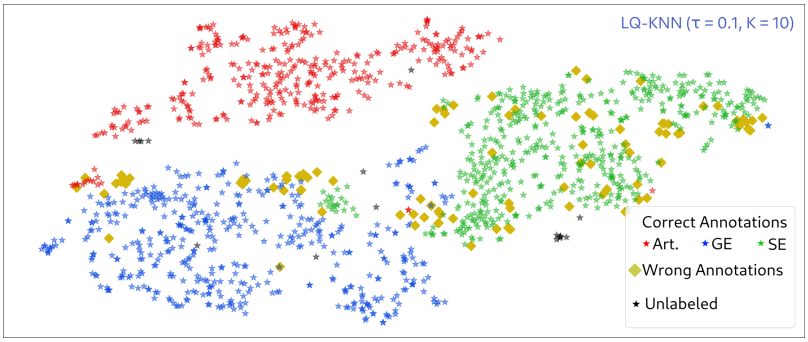
On the other hand, from the classification results of the HITS-small dataset (figures 3.10, 3.11) we can extract some information. First, for all the classes, both label propagation meth-



(a)



(b)

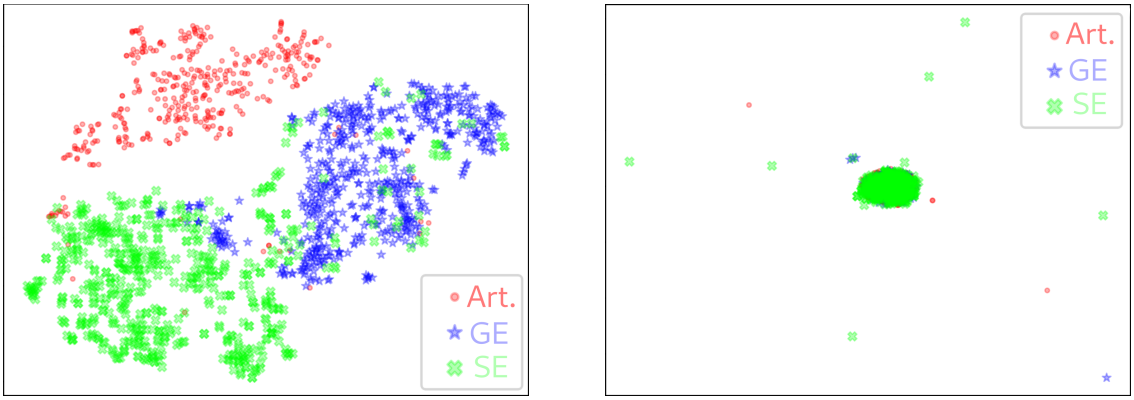


(c)

Figure 3.5: Experiment 1: Label propagation for the HITS-small dataset using LQ-KNN with $K = 3$ and $\tau = 0.3$, and with $K = 10$ and $\tau = 0.1$. (a) Local quality map of the automatically labeled samples. (b) LQ-KNN results with $K = 3$ and $\tau = 0.3$ (36.32% of labeled samples, with 96.44% accuracy). (c) LQ-KNN results with $K = 10$ and $\tau = 0.1$ (98.85% of labeled samples, with 92.52% accuracy). The diamonds correspond to the wrongly labeled samples. The manually labeled samples are not shown here, for clarity. We note that the most of the errors are located at the boundaries between two clusters of different classes.

ods (LQ-KNN and Std-KNN) allow improving classification performances of the model with respect to the model trained using less labeled samples (but no noise). In fact, in terms of MCC, the model trained on HITS-small LQ-KNN-K10 with CE outperforms the one without propagation by a margin of 5.68%, whereas when GCE is used, this margin increases to 9.71%.

Secondly, for a fixed loss function, classifiers trained on LQ-KNN based datasets with enough samples (*i.e.*, $K = 10$) outperform the ones trained on Std-KNN based datasets, in terms of MCC and class accuracy. Actually, regardless of the loss functions, HITS-small LQ-KNN-K10 outperforms HITS-small Std-KNN-K10 in terms of solid emboli accuracy by a margin of over



(a) (b)

Figure 3.6: Experiment 2: Examples of best and worst 2D chosen projections of the HITS-small dataset (1545 samples) obtained with respect to the silhouette scores. (a) Best projection (silhouette score, 0.54 ± 0.05). (b) Worst projection (silhouette score, −0.23 ± 0.09). The best selected projection gives more distinct clusters per class than the worst. Art., artifact; GE, gaseous emboli; SE, solid emboli.

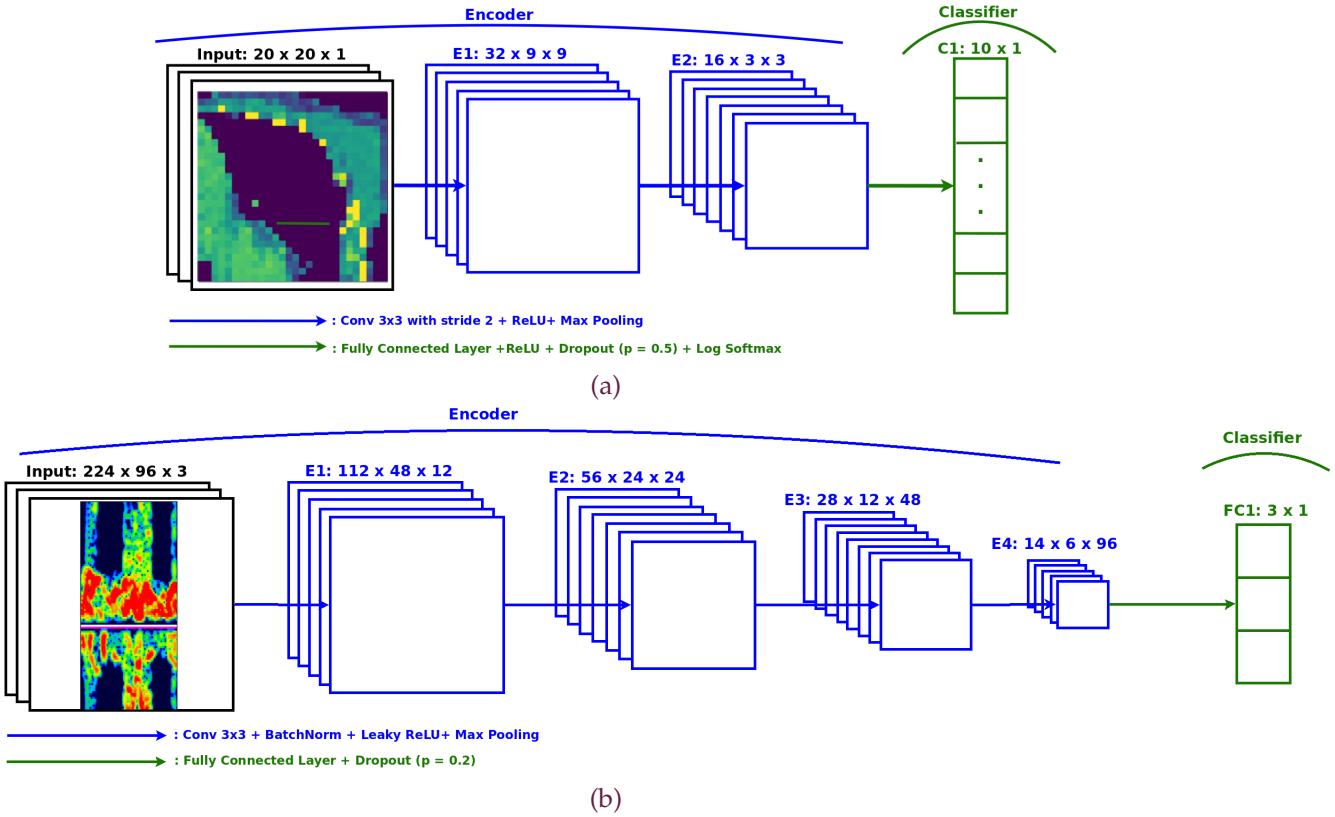


Figure 3.7: Convolutional neural network architectures used for classification for the different datasets. (a) Architecture for the MNIST dataset. (b) Architecture for the HITS datasets.

3.5%. This is particularly interesting in our case because solid emboli are the most dangerous ones as they can cause ischemic stroke. On top of that, a model trained on HITS-small LQ-KNN-K10 using GCE achieve similar performances than models trained on fully manually labeled datasets (HITS-small Whole).

Table 3.7: Different datasets used to train the models in experiment 3. The Core dataset corresponds to the dataset used as the basis to apply label propagation. Only 10% of the samples are considered as labeled ($\mathcal{A}$), and the rest are unlabeled ($\mathcal{U}$). K corresponds to the neighborhood that we consider to propagate the labels, and τ corresponds to the local quality threshold that defines if a sample is considered of good quality. None, no labeled propagation used to obtain the final dataset. ζ_{max} corresponds to the maximum tolerated noise rate, whereas ζ_{est} corresponds to the estimated noise rate, defined as $\zeta_{max} = 1 - \frac{1}{C}$ and $\zeta_{est} = 1 - \alpha$, where C is the number of classes of the dataset, and α is the annotation accuracy of the label propagation method.

Dataset	Core dataset	ζ_{max}	Prop. method	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{U}\rvert$	# automatically labeled samples	ζ_{est}	K	τ
MNIST No propagation	MNIST	90	None	1496	13504	-	-	-	-
MNIST Std-KNN			Std-KNN			13426 ± 31	9.83 ± 1.45	10	-
MNIST LQ-KNN			LQ-KNN			13256 ± 56	7.34 ± 1.30	10	0.1
OrganCMNIST No propagation	OrganCMNIST	91	None	1534	13858	-	-	-	-
OrganCMNIST Std-KNN			Std-KNN			13720 ± 28	18.13 ± 0.76	10	-
OrganCMNIST LQ-KNN			LQ-KNN			13336 ± 151	17.27 ± 0.44	10	0.1
HITS-small Whole	HITS-small	67	None	1545	0	-	-	-	-
HITS-small No propagation			None	152	1393	-	-	-	-
HITS-small Std-KNN-K2			Std-KNN			591 ± 42	15.05 ± 2.61	2	-
HITS-small Std-KNN-K10						1387 ± 8.7	18.64 ± 1.81	10	-
HITS-small LQ-KNN-K3			LQ-KNN			554 ± 63	10.12 ± 2.77	3	0.3
HITS-small LQ-KNN-K4						700 ± 54	10.35 ± 2.36	4	0.3
HITS-small LQ-KNN-K10						1372 ± 11	17.33 ± 2.02	10	0.1

Table 3.8: Training parameters used in experiment 3. q represents the GCE hyperparameter of the GCE controlling the trade-off between convergence speed and robustness to label-noise. The higher the value of q, the more robust GCE but the lower the convergence speed; the smaller the value of q the faster the convergence but the smaller the robustness to label-noise.

Dataset	Epochs	Batch Size	Learning rate	Optimizer	Weight Decay	Loss function	q
MNIST	100	32	7e-3	1e-7	Adamax	CE	-
						GCE	0.7
OrganCMNIST	150		7e-3	1e-5		CE	-
						GCE	0.5
HITS-small	50		2e-2 1e-3	1e-7		CE	-
						GCE	0.7

Finally, if we focus only on datasets obtained by automatically labeling less than 50% of the unlabeled samples (*i.e.*, HITS-small Std-KNN-K2 and HITS-small LQ-KNN-K4 datasets in table 3.7), we see that LQ-KNN labeled datasets allow obtaining models that outperform those trained on Std-KNN datasets, regardless the loss function. Interestingly, we can see that even though HITS-small LQ-KNN-K3 has fewer labeled samples than HITS-small Std-KNN-K2, it provides better classification performances, when nonrobust CE loss function is used. Indeed, the classifier trained on HITS-small LQ-KNN-K3 with CE outperforms that trained on HITS-small Std-KNN-K2 with CE by a margin of 1.90% in terms of MCC. However, this is not observed when robust loss function GCE is used.

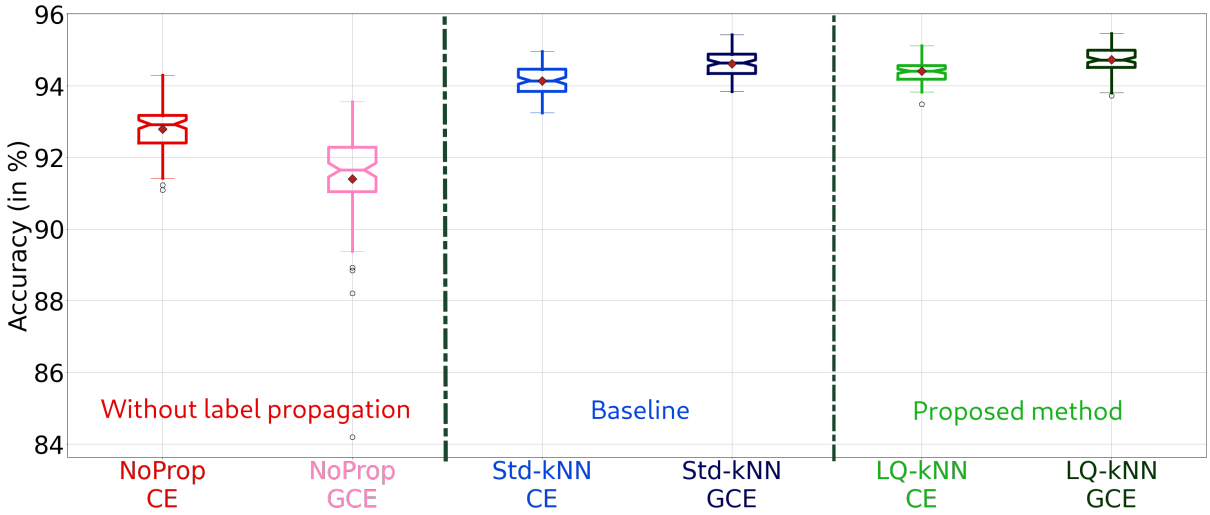


Figure 3.8: Experiment 3: MNIST accuracy results. The best performing classification model is the one trained on the semi-automatically labeled dataset obtained with LQ-KNN, using a robust loss function. When using nonrobust loss functions, the best performing classifier is also the one trained with the dataset obtained using LQ-KNN.

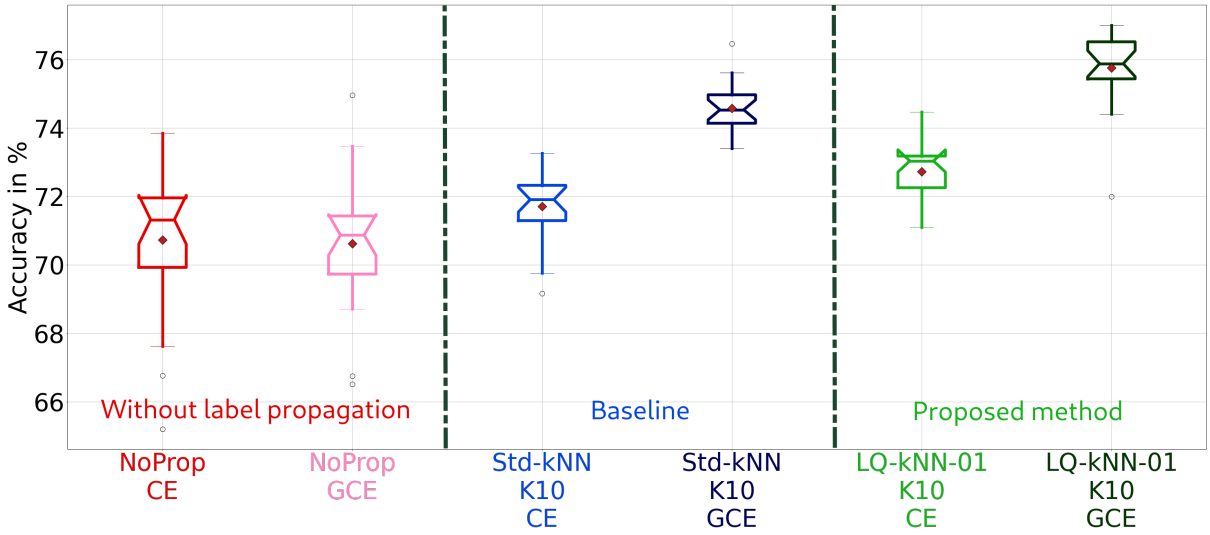


Figure 3.9: Experiment 3: OrganCMNIST accuracy results. The best performing classification model is the one trained on the semi-automatically labeled dataset obtained with LQ-KNN, using a robust loss function. When using nonrobust loss functions, the best performing classifier is also the one trained with the dataset obtained using LQ-KNN.

III.4.D Experiment 4: Classification on a semi-automatically labeled HITS-large dataset with unknown label noise

Objective: The objective of this experiment is to study the behavior of our pipeline (semi-automatic annotation followed by robust classification) on a real large medical HITS dataset, where most of the samples are not labeled (less than 3%).

Experimental setup: We created different dataset by applying LQ-KNN and Std-KNN to HITS-large, propagating the labels of the 1 545 labeled samples to a part of the unlabeled

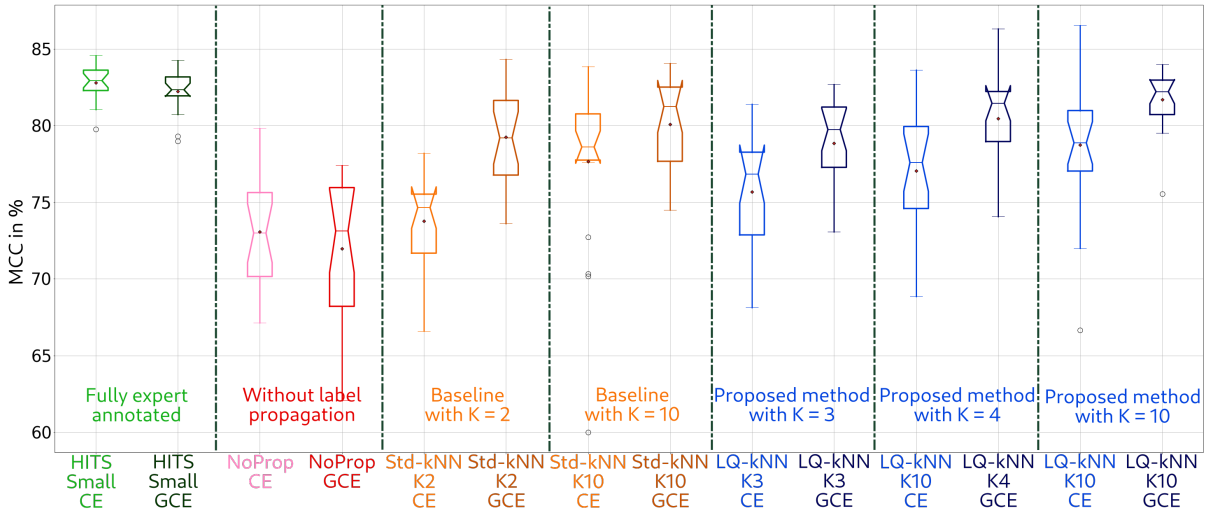


Figure 3.10: Experiment 3: HITS-small MCC results. Both label propagation methods increase the classification performances of the trained models, with similar performances to a model trained with a fully manually labeled dataset. Our proposed method, LQ-KNN, globally outperforms the baseline model.

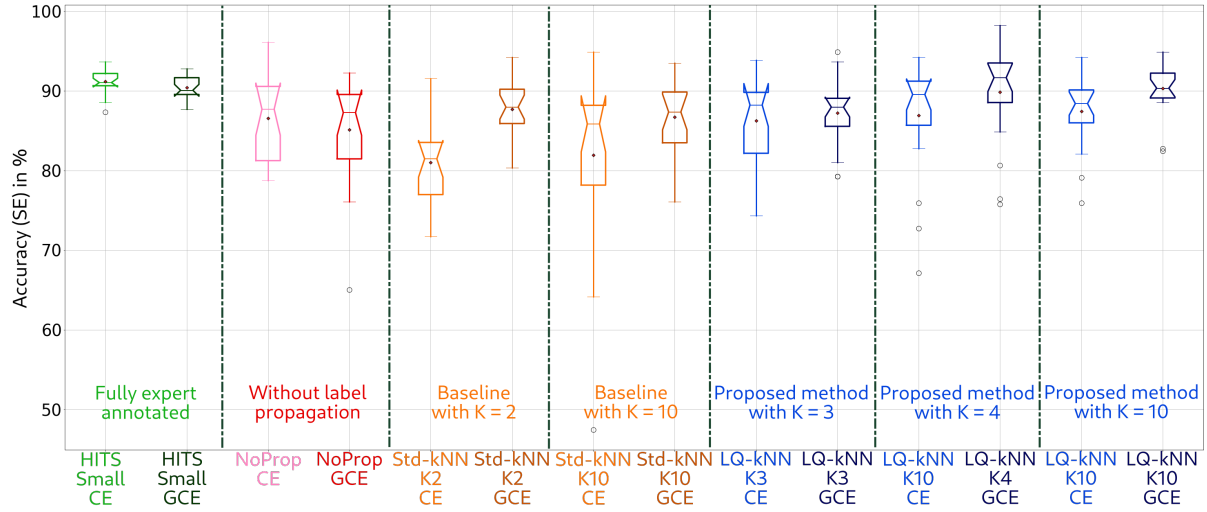


Figure 3.11: Experiment 3: HITS-small solid emboli accuracy results. Our proposed method, LQ-KNN, allows creating datasets where models can learn almost as well as on fully manually labeled datasets.

samples with unknown labels (see table 3.9). Based on the results of the previous experiments, we make the assumption that the noise rate stays smaller than the maximum tolerated noise rate, $\zeta_{max} = 67\%$. Moreover, to avoid having imbalanced classes in those datasets, we apply undersampling in order to have as many samples per class as the minority class, which always is the solid emboli one. Once we get these datasets, we train different classifiers on them, under the same conditions as in the previous experiment. The training of these models was done using two loss functions (CE and GCE), a learning rate of $1e-3$, a batch size of 32, a weight decay of $1e-7$, and 50 epochs.

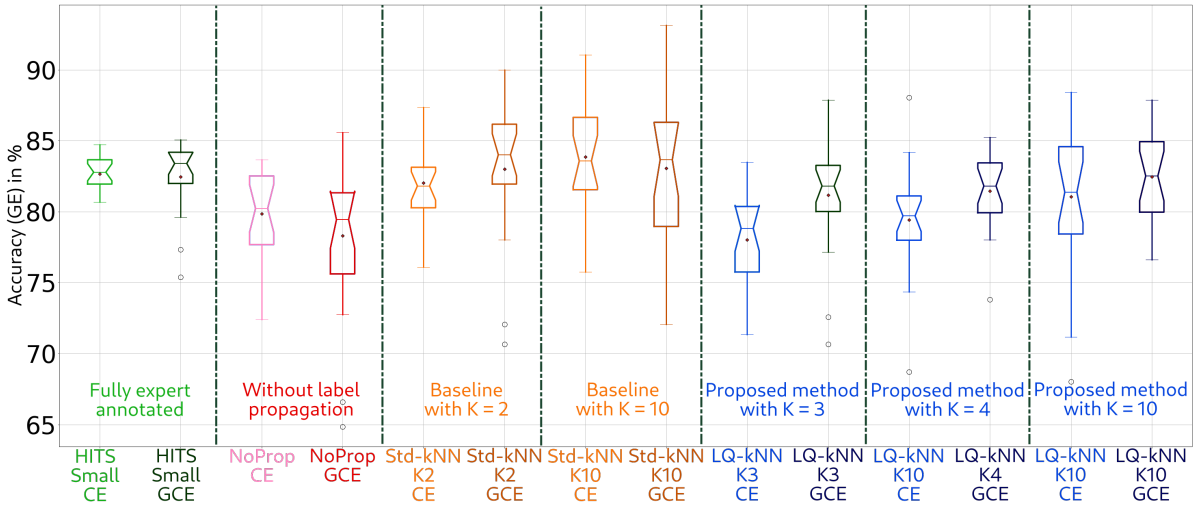


Figure 3.12: Experiment 3: HITS-small gaseous emboli accuracy results. Our proposed method, LQ-KNN, allows creating datasets where models can learn almost as well as on fully manually labeled datasets.

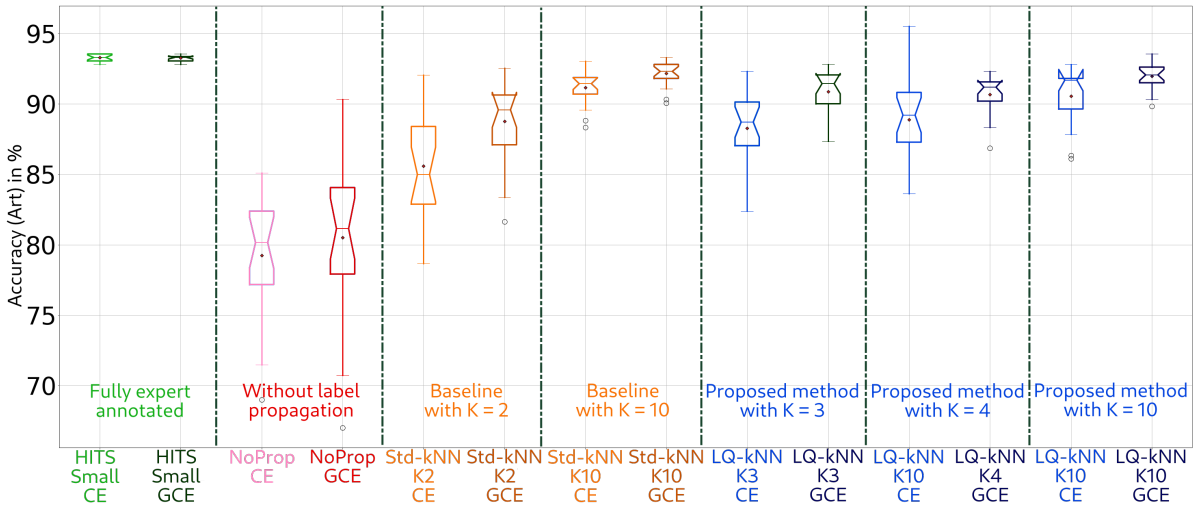


Figure 3.13: Experiment 3: HITS-small artifacts' accuracy results. Our proposed method, LQ-KNN, allows creating datasets where models can learn almost as well as on fully manually labeled datasets.

Table 3.9: Datasets used in experiment 4 without controlled label noise. In this experiment we study the behavior of our method on a real semi-automatically labeled dataset, without controlled noise.

Dataset	Prop. method	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{U}\rvert$	# of automatically labeled samples	Samples per class	K	τ
HITS-large Std-KNN	Std-KNN	1545	66947	13653	4551	10	-
HITS-large LQ-KNN	LQ-KNN			14970	4990		0.1

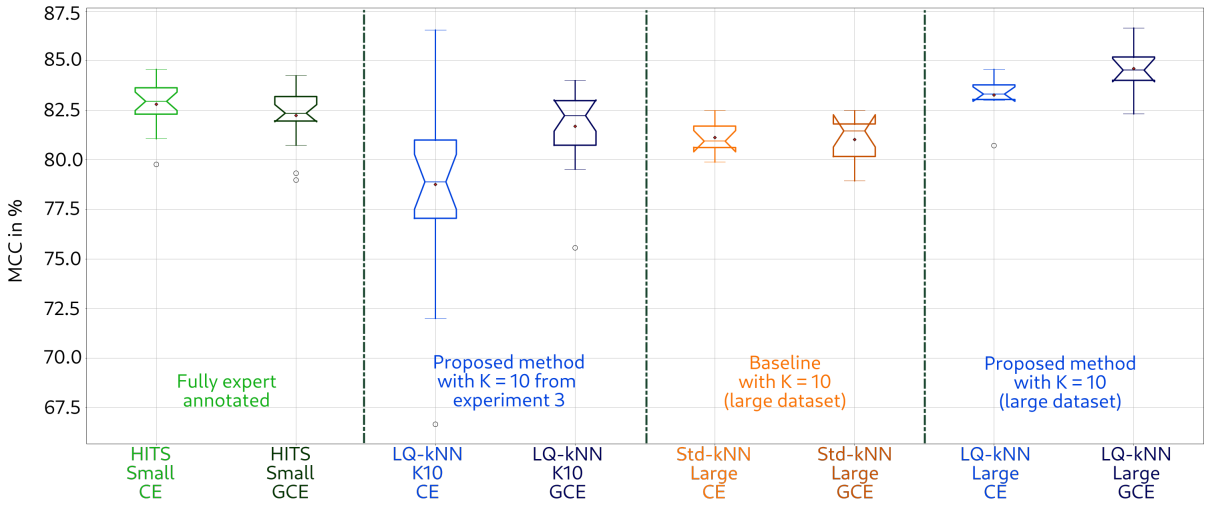


Figure 3.14: Experiment 4: HITS-large MCC results. We compare different semi-automatically labeled samples, obtained using all the available manually samples of HITS-large (1 545). The number of finally labeled samples depend on the used label propagation method used (however, the classes are always balanced).

Results: Figures 3.14 and 3.15 show the results of this experiment. Three main points can be highlighted. Firstly, we can observe that, regardless of the loss function, classifiers trained on datasets obtained by applying LQ-KNN on HITS-large (two last columns of figures 3.14 and 3.15) outperform the ones trained on all the other datasets (including those of experiment 3). More precisely, if we focus on HITS-large LQ-KNN, the model trained on this dataset outperforms the one trained on HITS-large Std-KNN by a MCC margin greater than 2%. A similar trend is observed for the class accuracies.

Secondly, if we compare the per class accuracies (figures 3.15, 3.16, 3.17) and the results of the previous experiments, we note that the models trained on HITS-large Std-KNN have higher solid emboli performance, at the expense of smaller artifacts' and gaseous emboli performances, while the variability is generally reduced when using more samples. Similarly, the performances of the model trained on HITS-large LQ-KNN have higher solid and gaseous emboli performances, but smaller the artifacts' performance, while the variability is still reduced.

At last, the best performing model of this experiment is the one trained on HITS-large LQ-KNN with GCE loss function, which outperforms the best performing model of experiment 3 trained on HITS-small, a fully manually labeled dataset and without label noise. Nevertheless, even when using a nonrobust loss function such as CE, classifiers trained on HITS-large LQ-KNN and HITS-small achieve similar performances, although HITS-large LQ-KNN is a larger dataset. More interestingly, for both loss functions, HITS-large LQ-KNN trained models achieve higher solid emboli accuracy compared to models trained on HITS-small. However, this phenomenon comes at the expense of an important lowering of the artifacts' accuracy. Nonetheless, when we use robust loss functions to compensate for the LQ-KNN label noise (mainly between the gaseous emboli and solid emboli classes), the obtained model also outperforms the HITS-small trained models for the gaseous emboli class (see figure 3.16).

III.5 Discussion

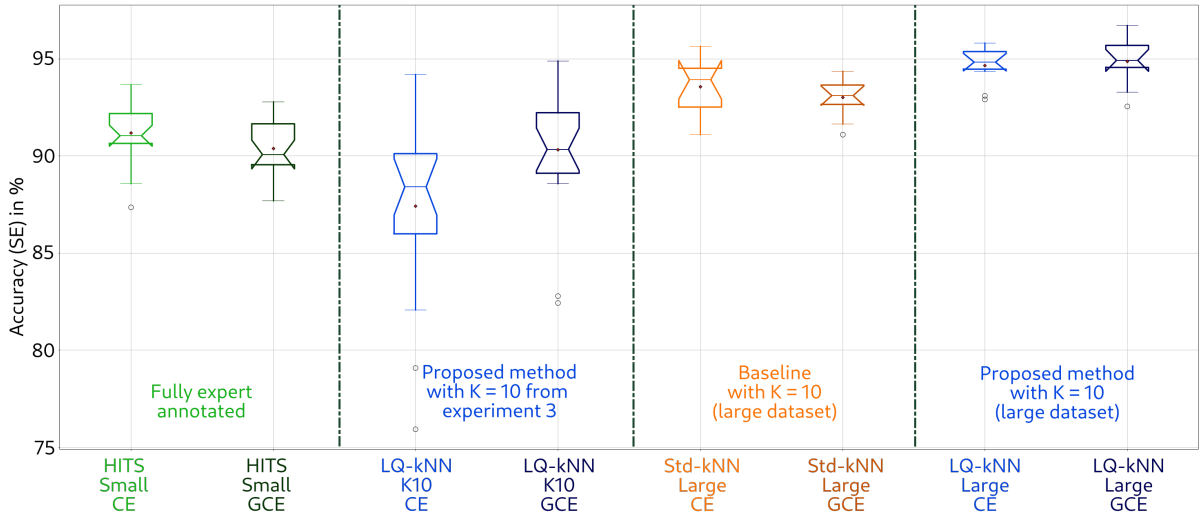


Figure 3.15: Experiment 4: HITS-large solid emboli accuracy results. We compare different semi-automatically labeled datasets, obtained using all the available manually samples of HITS-large (1 545). The number of finally labeled samples depend on the used label propagation method used (however, the classes are always balanced).

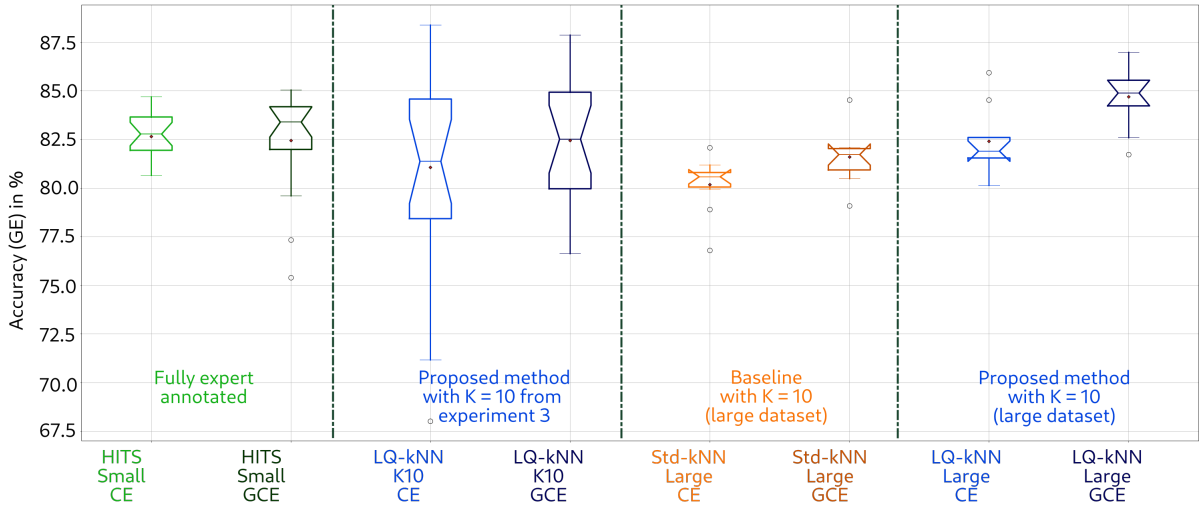


Figure 3.16: Experiment 4: HITS-large gaseous emboli accuracy results. We compare different semi-automatically labeled datasets, obtained using all the available manually samples of HITS-large (1 545). The number of finally labeled samples depend on the used label propagation method used (however, the classes are always balanced).

III.5.A Experiment 1: Automatic data annotation evaluation

This first experiment corroborates that our proposed automatic label propagation method, LQ-KNN, achieves comparable performance to other state-of-the-art methods such as OPF-semi, which is often used for data annotation. This translates in higher annotation accuracies, at the expense of a small reduction in the number of labeled samples, with 1.5% to 3.8% less labeled samples for LQ-KNN compared to OPF-semi in all the tested datasets. What is more, LQ-KNN is 2 to 3 orders of magnitude faster than OPF-semi, a non-negligible difference, specially when working with large datasets. This is an understandable observation, as data annotation with

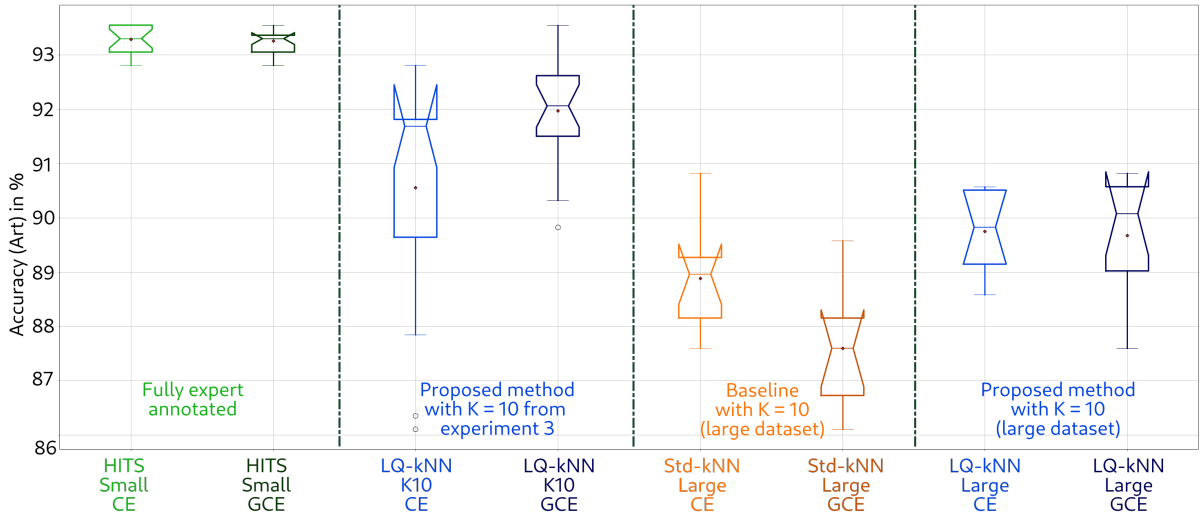


Figure 3.17: Experiment 4:HITS-large artifacts' accuracy results. We compare different semi-automatically labeled datasets, obtained using all the available manually samples of HITS-large (1 545). The number of finally labeled samples depend on the used label propagation method used (however, the classes are always balanced).

OPF-semi is time-consuming because it implies training a classifier with all the labeled and unlabeled samples, and then predicting the labels of the unlabeled samples with the trained classifiers. Furthermore, the results of this experiment also confirm that our approach LQ-KNN with $\tau = 0.1$ behaves better than the baseline Std-KNN, showing the interest of local quality for label propagation. This phenomenon becomes more apparent when the value of τ increases, achieving very high annotation accuracies for a small proportion of labeled samples, with $\alpha = 99.34\%$ and $p_A = 15.51$ for $\tau = 0.5$ and $K = 20$, on the MNIST dataset.

By the same token, the results show that the annotation performances can vary from one dataset to another one. Indeed, they are significantly smaller on the HITS-small and OrganCMNIST datasets, for all the label propagation methods. If we take the example of our dataset of interest, the HITS-small one, we can explain this observation on the basis that gaseous and solid emboli can be mistaken in some cases (even for expert human annotators), and this translates in overlapping clusters in the reduced annotation latent space. Because of this, when we propagate labels at the boundary between the solid and gaseous emboli clusters, we are going to make more annotation errors. Therefore, if these boundaries between overlapping clusters are reduced (as in the MNIST dataset), the annotation errors too, hence the annotation accuracies tend to increase. More interestingly, as the annotation accuracies show it, using the local quality of the projection to select the samples for label propagation allows a more cautious annotation, so samples that were wrongly projected at the boundary between two clusters are not used for label propagation.

Moreover, the results of this experiment also reveal one important advantage of LQ-KNN: the control of the annotation error thanks to the local quality threshold, τ, and the neighborhood K considered for label propagation. Indeed, these hyperparameters can reduce the annotation errors at the expense of the number of labeled samples. We can directly visualize this by comparing different set of hyperparameters as in figure 3.5. In this figure, we have a clear example where the annotation errors (and the number of labeled samples) are reduced by reducing the values τ and K. On top of that, this figure also shows that most of the annotation errors are located between the boundaries of clusters of different classes or in low local quality

zones. Therefore, if we increase the local quality threshold, incorrectly projected labeled and unlabeled samples are not going to be used for label propagation, reducing the number of propagated errors.

Furthermore, we observe that, globally, when the annotation accuracy increases, the proportion of labeled samples tends to decrease, which is why higher values of τ give higher annotation accuracies but fewer newly labeled samples. Indeed, the success of our method strongly depends on the structure assumption (Chapelle *et al.*, 2009). Thanks to the use of the local quality criterion (Lueks *et al.*, 2011), we can guarantee that, if the structure assumption is (partially) verified in the feature space obtained by the AE model, then it should be (partially) verified in the 2D reduced annotation latent space obtained by t-SNE. Therefore, the higher the values of the threshold τ, the more demanding we are in terms of structure assumption guarantees, so we reduce the samples that can be annotated to only 'good samples', thus labeling fewer samples. On the contrary, the smaller the values of τ, the more changes we tolerate in the projected space, allowing to annotate more samples. Into the bargain, we can see that choosing good values for K and τ is difficult, and it is application-dependent: if the quality of the labels is crucial for the application, higher values of τ should be favored (*e.g.*, $0.3 \leq \tau \leq 0.7$) and/or smaller values of K (*e.g.*, $K \leq 5$), paying with fewer labeled samples, whereas if the quality of the labels is not crucial and the number of samples is, higher values of K should be favored (*e.g.*, $K > 5$) and/or smaller values of τ (*e.g.*, $\tau \leq 0.3$).

Finally, one last interesting point is the importance of the propagation order. In fact, in our method, an annotation order is established based on the local quality values: higher local quality samples are labeled first than the lower quality ones. The rationale behind this is that high local quality samples are stronger representatives of the local space where they are found; therefore, it is more likely for the samples located in that zone to have the same label as the highest local quality neighbors, rather than the lowest ones. This was confirmed by this experiment, especially for high values of K, whereas for small values, there is no important difference. This indicates that the labeling order is very important for applications that need to consider large neighborhoods for label propagation, for instance if we want a quicker annotation.

III.5.B Experiment 2: Validation of the projection selection strategy

This experiment validate our projection selection strategy for automatic label propagation purposes. It is clear that, when selecting the best projection with our strategy, we obtain significantly better annotation performance than when the worst selected projection is used. This can be explained on the basis that our method strongly relies on the structure assumption, therefore it is important that the selected lower dimensional working space follows a similar local structure than the high-dimensional original one. Thanks to the silhouette score, our proposed selection strategy selects the best projection for this objective (*i.e.*, highest score), keeping the samples of a same class/cluster in the same structure. What is more, this experiment also showed the cautiousness of LQ-KNN with respect to Std-KNN, as it allows labeling fewer samples with higher accuracies, by selecting only the reliable samples based on the local quality. This result is particularly interesting, as it shows that LQ-KNN is more robust than Std-KNN against 2D projection degrading the latent space structure.

III.5.C Experiment 3: Classification on a dataset with known label noise

This experiment confirms the interest of our method for classification purposes. Indeed, the obtained semi-automatically labeled datasets can be used to train classifiers. The results suggest that, for the three tested datasets, training classifiers on datasets obtained with LQ-KNN (*i.e.*,

$K = 10$, $\tau = 0.1$) allows yielding better classification performances than models trained on reduced datasets without label propagation, or datasets obtained with Std-KNN.

What is more, this experiment also highlights the importance of robust loss functions when using semi-automatically noisy labeled data. Indeed, these types of loss functions are able to partially compensate the introduced annotation error by the automatic label propagation method. That is why, when we use this type of functions, LQ-KNN and Std-KNN achieve similar performance. As a matter of fact, both label propagation methods introduce similar levels of noise (for the chosen hyperparameters, LQ-KNN is slightly better), so noise-tolerant loss functions can compensate this difference. Nevertheless, when this annotation error difference increases, the robust loss function is not able to totally compensate the gap, allowing the models trained on LQ-KNN based datasets to achieve better classification results, as it is shown in the OrganCMNIST results.

At last, this experiment showed that applying our proposed automatic label propagation approach on a real medical dataset can be beneficial for classification purposes. Indeed, regardless of the loss function, the obtained datasets allow the trained models to improve their classification performances with respect to the datasets having limited labeled samples. In addition, these models can achieve similar performances than fully manually annotated datasets. Moreover, the proportion of labeled samples by the method also play an important role as, when we propagate the labels to less than 50% of the available samples, LQ-KNN outperforms Std-KNN. However, when the proportion of labeled samples increases, both method behave similarly, even if LQ-KNN based datasets allow training classifiers with better mean performances, especially when noise-tolerant loss functions are used.

III.5.D Experiment 4: Classification on a semi-automatically labeled HITS-large dataset with unknown label noise

This last experiment confirms that our approach is stable when using large-scale datasets, permitting to obtain a large semi-automatically labeled dataset, allowing to train classifiers with better generalization capabilities. Models trained on HITS-large LQ-KNN with both loss functions, outperform the models trained using the other datasets, including those of experiment 3. More interestingly, in terms of MCC and emboli accuracies, when using a noise-tolerant loss function, our method is able to outperform the models trained with CE/GCE losses on the HITS-small fully manually (but reduced) dataset, as it partially compensates the noise introduced in the labels by our approach. Nevertheless, this is not the case for the artifact accuracy. Our hypothesis is that, in the HITS-large dataset, the obtained 2D reduced working latent space do not verify the structure assumption anymore. There are two potential explanations for this: (1) the original HITS image space does not verify the structure assumption, or (2) the AE high-dimensional space does not (which could be partially solved using more complex architectures). Because of this, when doing label propagation, we introduce unexpectedly high noise in the labels of the artifact class (besides the one already present in the gaseous and solid emboli classes), no longer verifying the noise-tolerant assumption ($\zeta < \frac{C-1}{C} \approx 67\%$, for the HITS dataset), therefore, the noise-tolerant loss function is not able to compensate for this noise anymore. Nevertheless, our LQ-KNN approach is more stable, as its artifact accuracy is higher than the one obtained with Std-KNN, and its gaseous emboli accuracy remains similar (or even better) to the HITS-small one.

III.5.E Choice of k_s and k_t

An important aspect of our method that we have not discussed so far, is the choice of $k_t = 10$ and $k_s = 10$ for all the experiments. Indeed, these parameters are important for the computation

of the global and local quality measures. As seen in Chapter 2, k_t controls the tolerated errors in terms of rank modification, whereas k_s controls the size of the neighborhood that is going to be used to evaluate the structure preservation through ranks modifications. As shown in (Lueks *et al.*, 2011) and figure 3.18, when k_t and k_s increase, the global quality varies smoothly, as between two consecutive values of k_t and k_s, its variation is small. When k_t and k_s increase, we tolerate more errors and we expand the neighborhood considered to measure these errors, therefore the measured global quality with respect to these parameters is higher. However, these high values of the global quality are not necessarily representative. In fact, in those cases, what the global quality means is that, for the chosen tolerated errors and for the chosen size of the neighborhood to measure them (both are high), we can consider that the structure of the high and low dimensional manifold are similar. This shows that the value of the global and local qualities should be interpreted with the chosen values of k_t and k_s and not alone. Thus, smaller values of these parameters, means that we are more demanding in terms of errors, and that the used neighborhood to measure these errors is small, which is why we obtain lower global quality values (dimensionality reduction techniques always modify the global and local structure of the high dimensional manifold as it reduces the number of degrees of freedom).

At last, in our case, as we are interested in accurate automatic label propagation, it is important to have meaningful and demanding values of the global and local qualities to assess as best as possible the preservation of the local structure assumption mentioned in section II. Therefore, we follow the recommendations of (Lueks *et al.*, 2011) by first focusing on the error tolerance k_t and then to choosing the value of k_s, allowing a good selection of the samples to benefit for label propagation.

III.5.F Limitations

During the study of our approach, we can identify five main limitations.

The first one concerns the validation framework. Indeed, we limited our study to three datasets, and one type of classifier (CNN). To strengthen the results of our study, it can be interesting to study our approach using other types of datasets and models.

The second limitation regards the used feature extraction model. We used a simple AE model, whereas more complex models and architectures can be used (variational AE, siamese AE, etc.). Therefore, the influence of different types of models should be quantitatively measured.

Thirdly, in terms of development time, our projection selection strategy can be expensive to compute and only takes advantage of the labeled samples, which can lead to sub-optimal projections for label propagation. Indeed, if all the labeled samples are not representative of the true structure of the different classes' manifolds, then we can select biased projections. Semi-supervised learning metrics and approaches can be introduced to better exploit both, labeled and unlabeled samples, and reduce the development time.

Fourthly, the choice of the automatic label propagation methods (k_t, k_s, K, and τ) is not trivial. In fact, even if the results showed that the choice $k_s = 10$, $k_t = 10$, $K = 10$ and $\tau = 0.1$ tend to give good results for different datasets, more efficient hyperparameter selection strategies can be developed for this task.

Finally, for classification under noisy-labels coming from the semi-automatic annotation method, we only tested one robust loss function, GCE, but other loss functions can be used such as symmetric CE (Wang *et al.*, 2019), or other strategies can be adopted such as robust regularization, robust architectures, loss adjustment, etc. (Song *et al.*, 2022).

IV Conclusion

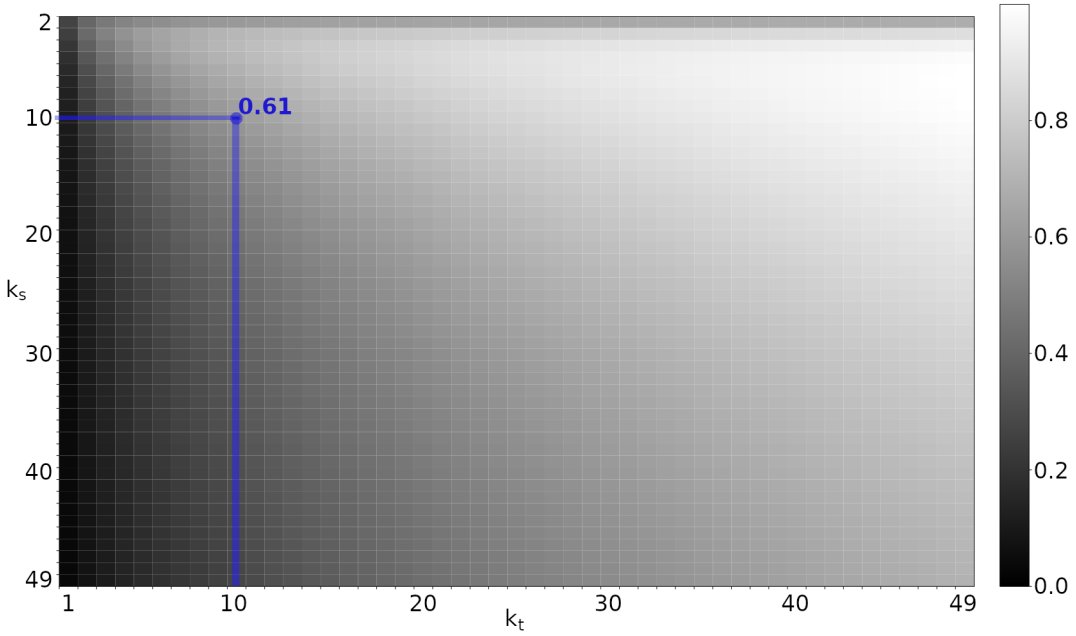


Figure 3.18: Normalized global quality matrix (Lueks *et al.*., 2011) of a selected HITS-small projection. The x-axis shows the values of k_t and the y-axis the values of k_s. For smaller values of k_s and larger values of k_t, we have higher global quality, because we take a smaller neighborhood for the computation and we tolerate more errors. The global quality decreases when the error tolerance k_t decreases, because we are more demanding on the preservation of the neighborhood of size k_s of each sample. For $k_s = 10$ and $k_t = 10$, we have a global quality of 0.61 (as indicated). (Lueks *et al.*., 2011) recommend to first focus on the error tolerance k_t and then to choose the value of k_s.

In this chapter we proposed our semi-automatic data annotation method, LQ-KNN, allowing to annotate an important number of unlabeled samples, from a small number of labeled samples, and this with controlled annotation error. Our method is based on feature extraction using an AE model, dimensionality reduction through t-SNE, projection selection based on the silhouette score, and local structure preservation measurements using local quality metrics. Moreover, as annotation errors are inherent to semi-automatic data annotation, we propose to partially compensate them through noise-tolerant loss functions, when training classifiers on these datasets.

What is more, our approach was validated through extensive experiments, giving interesting results. Firstly, our automatic label propagation results outperform the baseline, Std-KNN, and other state-of-the-art methods such as OPF-semi. Secondly, we observed that, thanks to the choice of the hyperparameters K and τ, we are able to control the annotation error, which can be beneficial if the targeted application is very sensitive to annotation errors. Thirdly, the use of robust loss functions such as GCE was proven beneficial to train the classifiers on the obtained semi-automatically noisy labeled datasets, allowing to achieve similar performances to a fully manually labeled dataset. This point is particularly interesting, as automatic label propagation is much faster than manual annotation (0.2 ms per samples against 8 s per sample for an expert annotator). Finally, even though it was not exhaustively tested, our method was designed to be general and applicable to different datasets and models, the only components that should be adapted are the feature extraction block, and the classification block.

At last, a semi-automatic data annotation tool was developed using the proposed method,

allowing to quickly and interactively label HITS[1]. In addition, once the HITS labeled, they can be manually verified before being introduced in the final database.

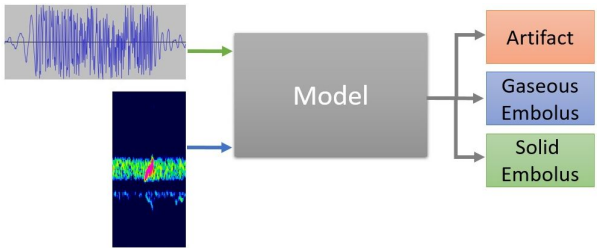
In the following chapters, we are going to develop more complex architectures for HITS classification, allowing to exploit both, the temporal and spectral information of the signals, as well as model compression methods to reduce the memory, computation, and energy requirements of the developed models.

[1]This annotation is done using soft labels and not hard labels. This means that the label of a sample is composed of membership scores for each class, and note one single label.

4 | Medical Signal Classification

Chapter based on the following papers: conference paper (Vindas, Guepie, Almar, Roux and Delachartre, 2022) in Machine Learning for Healthcare (MLHC) 2022; conference paper (Vindas, Roux, Guépié, Almar and Delachartre, 2023a) at EUSIPCO 2023; journal paper (Vindas, Roux, Guépié, Almar and Delachartre, 2023b) in Pattern Recognition.

In chapters 1 and 2 we saw that the distinction between artifact and emboli is important to help clinician with patient management. More interestingly, separating solid from gaseous emboli can enhance patient treatment as well as medical and clinical studies. Different classical signal processing and machine learning techniques have been used to detect and classify HITS between artifacts and emboli (Markus and Punter, 2005; Keunen *et al.*., 2008; Serbes and Aydin, 2014; Sombune *et al.*., 2018; Imaduddin *et al.*., 2019; Guepie *et al.*., 2018). Other works used deep learning techniques (CNNs) to classify HITS (Sombune *et al.*., 2017; Tafsast *et al.*., 2018), without surpassing classical approaches, but showing promising results. However, these methods do not make a distinction between solid and gaseous emboli, and the only one that does it (Tafsast *et al.*., 2018) used *in vitro* TCD data. To cope with the limitations of previous works (which will be detailed in section I.1) we propose generic multi-feature classification approaches, presented in section II, and validated in section III.



Contents

I Motivation

I.1 Limitations of previous methods

The previous methods have achieved great performances when it comes to HITS detection and classification between artifact and emboli. These methods mainly try to extract features translating the emboli detection criteria of the 1998 consensus (Ringelstein *et al.*., 1998).

The first limitation of these approaches is that, almost none of them make a distinction between solid and gaseous emboli. Indeed, either they use dual-frequency (non-portable) TCD devices, using different frequencies to separate solid from gaseous emboli, or they work on (non-portable) TCD data acquired *in vitro*. This is an important limitation because, as seen in Chapter 1, the causes and treatments for solid and gaseous emboli are not exactly the same.

A second important limitation is that, the majority of these approaches have been designed and tested using data acquired using non-portable TCD devices, which are less prone to artifacts. In fact, because of the free movement of the subjects and the heterogeneous recording conditions, portable TCD devices tend to give an important number of artifacts coming from different sources (subject's voice, robotized probe movement, electrical noise, etc.). However, the main advantage of portable devices is that they allow long-duration monitoring, which increases the chance of detecting emboli, which are rare events. Therefore, it is important to be able to develop robust methods against these types of devices.

Furthermore, the previously mentioned methods are mainly based on signal processing and statistical based handcrafted features, which are time-consuming, and not necessarily optimal for the task that we want to solve. Additionally, these features depend on the data acquisition system, as well as on the final used classification model. Therefore, they are less robust against new types of TCD data (for instance non-portable to portable TCD acquired data, different insonation frequencies, new types of HITS, etc.).

At last, even though it is not the main scope of this work, an important limitation of previous works is that they are not easily adaptable to new types of data (for instance EEG or ECG data). Often, specific methods are developed for the specific task, and not tested on a variety of different medical signal data types to prove and validate its generalizability.

I.2 Objectives and contributions

We suggest solving the previously mentioned limitations by proposing a multi-feature classification framework, where both the temporal and spectral characteristics of the medical signals are used to do the final classification. The proposed approach achieve state-of-the-art performances on several medical signal classification datasets. Our main contributions can be summarized as follows:

- Novel hybrid CNN-transformer models, exploiting the complementarity between the temporal and spectral characteristics of the input (Doppler) medical signal.
- Guided and regularized training of an intermediate fusion approach, improving generalization while partially handling imbalanced datasets and label-noise.
- Late-fusion mechanisms, based on learnable and interpretable attention weights.

II Proposed method

In this section, we are going to detail the different multi-feature learning approaches that we developed to classify medical signals. In particular, our main objective is to classify HITS,

between solid emboli (SE), gaseous emboli (GE), and artifacts (Art.), using *in vivo* TCD data acquired with portable devices (TCD-X from Atys Medical).

II.1 General overview and assumptions

The methods detailed in this section were designed to be general and applicable to different types of data, with different representations, and different model architectures. However, as our main interest is medical signal classification, we are going to focus on (medical) temporal-dependent signals, and we are going to use a hybrid CNN-transformer model, where a 2D CNN model exploits the spectral information (taking as input a TFR), and a 1D CNN-transformer model exploits the temporal information (taking as input the raw signal).

Our method relies on four main assumptions. The first one is that the raw signal and an associated TFR share complementary information. Indeed, even though the TFR is obtained from the raw signal, it will mainly focus on the frequency content information, whereas the raw signal better translates the temporal dependencies between the samples of a signal. The second assumption is closely related to the previous one. Indeed, we suppose that temporal information is important, as the context before or after an event (e.g. a HITS) can help to its classification. Moreover, we also make the assumption that the length of the studied signals is variable, as the acquisition and pre-processing parameters can change (for instance, different sampling frequencies can be used). At last, in order to efficiently process these signals with the proposed models, we suppose that the number of samples per signal is short, smaller than 1 600. This is a reasonable hypothesis as we work with short duration signals (HITS with duration smaller than 250 ms) and/or small sampling frequencies (around 4.4 kHz for the HITS, and 100-200 Hz for the other medical signals).

Furthermore, we propose to use different methods to classify medical signals, with a special interest on TCD HITS:

- **Single feature models**: We propose to use two different types of models, based on the used input representation. For the TFR, we propose to use a VGG like vanilla 2D CNN, where the input is processed as an image. For the raw signal, we use a 1D CNN-transformer, where a 1D CNN encoder allows extracting features from the input signal, which will then be passed as tokens to a transformer encoder and classifier.
- **Late-fusion model**: Hybrid model based on interpretable late attention weights, combining the classification outputs of the two previous single feature models.
- **Intermediate-fusion models**: End-to-end joint-trainable hybrid model, taking as input both the raw signal and a TFR. The encoders, used to extract features from each input representation, are based on the 2D CNN and 1D CNN-transformer single feature models. Fusion is done at an intermediate level, between the latent spaces of the two encoder models. This obtained common latent space is then used for final classification. What is more, the training of each encoder is guided through iterated losses, and the common fused latent space is regularized by adapting DEC to a supervised learning context.

In the following sections, each of these models will be detailed. But first, let us define some general notations for this chapter. Let us suppose that we have a fully labeled dataset $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$, composed of N labeled samples $\mathcal{X} = \{X_i\}_{i \in [1,N]}$ divided into C classes ($\mathcal{Y} = \{\mathbf{y}_i / i \in [1,N], \mathbf{y}_i \in \{0,1\}^C\}$). Moreover, we assume that each input sample $X_i \in \mathcal{X}$, with $i \in [1,N]$, is composed of two representations, a raw signal $X_i^{RS} \in \mathbb{R}^{L \times V}$ (composed of L samples and V channels), and a TFR $X_i^{TFR} \in \mathbb{R}^{F \times M}$ (composed of F frequency bins, and M time bins). At last, we suppose that each classification model $\mathcal{M}$ can be written under the form $\mathcal{M} = C \circ \mathcal{E}$, where C is a classifier and $\mathcal{E}$ is an encoder/feature extractor.

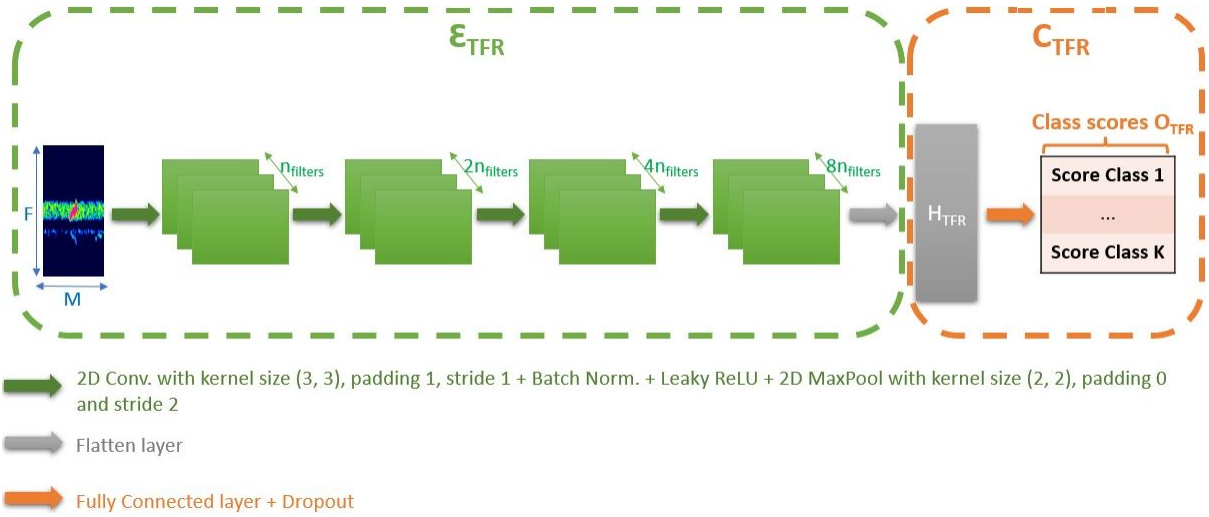


Figure 4.1: Used 2D CNN architecture, taking as input a TFR (considered as grayscale or RGB image).

II.2 Single feature models

II.2.A Time-frequency 2D CNN

To extract features from the TFRs, we use a vanilla VGG like 2D CNN architecture, where each TFR is processed as an image (with multiple channels or a single channel). The global architecture of the model can be found in figure 4.1. The encoder of the model, denoted $\mathcal{E}_{TFR}$, is composed of four convolutional blocks and a flatten operation. Each convolutional block has four components (applied sequentially): convolutional layer, batch normalization, leaky ReLU activation function, and pooling layer (average or max). The flatten operation is applied to convert the output into a hidden feature vector, $\mathbf{H}_{TFR} = \mathcal{E}_{TFR}(X_{TFR})$. Then, the obtained feature vector is passed thought a classifier, C_{TFR}, composed of a fully connected layer with dropout, obtaining classification scores $\mathbf{O}_{TFR} = C_{TFR}(\mathcal{E}_{TFR}(X_{TFR})) \in \mathbb{R}^{C\times 1}$. Additionally, this feature vector can be used for intermediate fusion, as it will be detailed in section II.4.

II.2.B Raw signal 1D CNN-transformer

To extract features from the raw signals, we use a 1D CNN-transformer inspired from (Natarajan *et al.*., 2020; Dosovitskiy *et al.*., 2021), and presented in figure 4.2.

The first part, the encoding one (denoted as $\mathcal{E}_{RS}$) is composed of two blocs. The first block is a 1D CNN encoder, where a series of 1D convolutional blocks are applied to the input signal (with overlapping convolutional kernels), in order to extract embeddings from the input raw signal. Indeed, at the output of the last convolutional block, embedding vectors (corresponding to local patches of the raw signal) are formed by taking all the channel components of each position in the encoded signal. This gives a sequence of embedding vectors, where a learnable class token vector is added at the beginning. Then, this sequence is passed through a second block, a transformer encoder, where the positional (temporal) information is taken into account thanks to the sinusoidal positional embeddings, and the attention mechanism. The final embedding, $\mathbf{H}_{RS} = \mathcal{E}_{RS}(X_{RS})$ is then obtained by taking the first vector of the signal encoding, which corresponds to the class token embedding.

The second part, the classification one (denoted as C_{RS}), uses the previously obtained encoding of the raw signal, and passes it through a series of layer normalization and FC blocks,

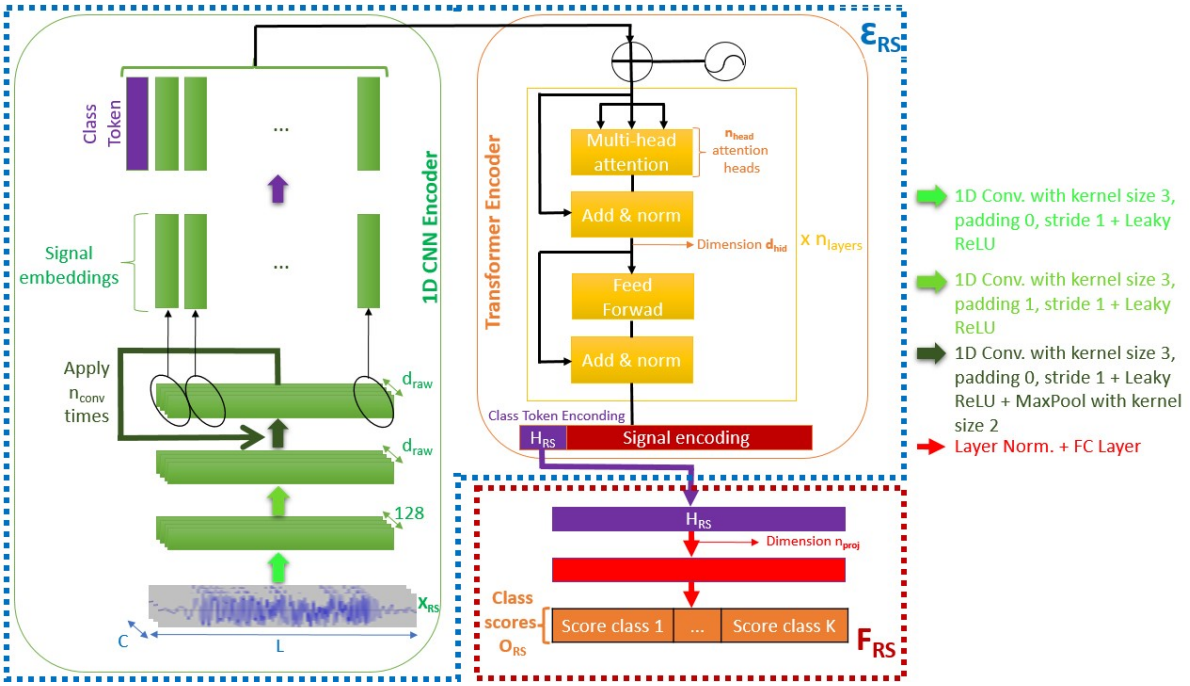


Figure 4.2: Used 1D CNN-transformer architecture, taking as input the (multi-channel) raw signal.

giving the classification scores $\mathbf{O}_{RS} = C_{RS}(\mathcal{E}_{RS}(X_{RS})) \in \mathbb{R}^{C\times 1}$. We can see that the main difference with respect to (Natarajan *et al.*, 2020) is that classification is done using a class token with a single feature, instead of using multiple heterogeneous features, and global average pooling at the last block of the transformer model. At last, as for the previous single feature model, $\mathbf{H}_{RS}$ can be used for intermediate fusion (section II.4).

II.3 Late fusion approach

II.3.A Principle

The core idea of our late fusion approach is to combine the classification outputs of two models taking different representations of a same input, in our case the raw signal and a TFR (see figure 4.3). This allows to take into account different representations for the final classification, while still allowing to do classification if one representation of the input is not available. Moreover, for expert annotators, one representation can be more useful than another one (even though they are complementary), based on the sample. Therefore, to simulate this behavior, we introduce learnable attention weights, indicating for each class and each input representation, its importance with respect to the final classification output.

II.3.B Late fusion attention weights

Let us denote by $\mathbf{W}_{RS}, \mathbf{W}_{TFR} \in \mathbb{R}^{C\times 1}$ two attention weight vectors associated to the raw signal and the TFR outputs, $\mathbf{O}_{RS}, \mathbf{O}_{TFR} \in \mathbb{R}^{C\times 1}$, respectively. By making the assumption that the two single feature models are pre-trained, we compute the final classification multi-feature scores, $\mathbf{O}_{LF} \in \mathbf{R}^{C\times 1}$, as follows:

$$\mathbf{O}_{LF} = \mathbf{W}_{RS} \odot \mathbf{O}_{RS} + \mathbf{W}_{TFR} \odot \mathbf{O}_{TFR} \tag{4.1}$$

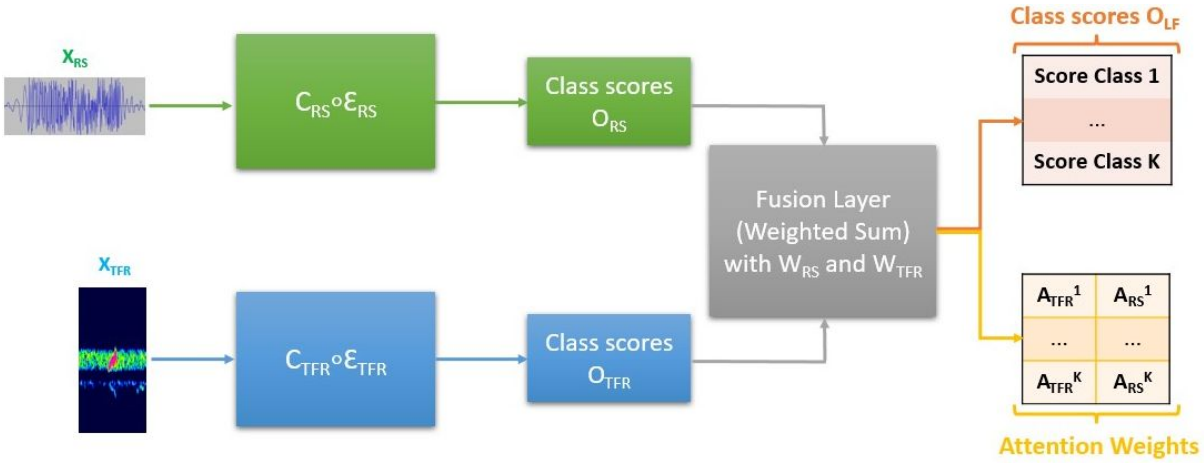


Figure 4.3: Global pipeline of our late fusion approach. We suppose that the single feature classification models are pre-trained. Moreover, we use learnable attention weights to simulate the behavior of expert annotators, where different representations do not have the same importance for the final decision.

where $\odot$ represents the Hadamard product, and the weights vectors $\mathbf{W}_{RS}$ and $\mathbf{W}_{TFR}$ are learned using gradient descent.

Furthermore, to obtain more interpretable weights, once the training is finished, we convert these attention weights into scores by applying a softmax function, obtaining $\mathbf{A}_{RS}$ and $\mathbf{A}_{TFR}$:

$$\mathbf{A}_{RS} = softmax(\mathbf{W}_{RS}) \tag{4.2}$$

$$\mathbf{A}_{TFR} = softmax(\mathbf{W}_{TFR}) \tag{4.3}$$

Under this form, the i^{th} element, $\mathbf{A}_{RS}^i$ of $\mathbf{A}_{RS}$, corresponds to the contribution of the raw signal for the final score prediction of class i. Likewise, $\mathbf{A}_{TFR}^i$ indicates the contribution of the TFR to the prediction of class i.

II.4 Intermediate fusion

In this section, we will introduce the intermediate fusion approach that we adopt for multi-feature learning, where the fused outputs are the ones of the encoders of the single feature models, namely $\mathbf{H}_{RS} = \mathcal{E}_{RS}(X_{RS})$ and $\mathbf{H}_{TFR} = \mathcal{E}_{TFR}(X_{TFR})$.

II.4.A Simple intermediate fusion

The first intermediate fusion approach consists in three simple intermediate fusion strategies: sum, weighted sum, and concatenation. The idea is to get a joint representation embedding space from the two single feature latent spaces obtained by each single feature encoder.

To allow the sum of the encodings of each input representation, we cannot directly use $\mathbf{H}_{RS}$ nor $\mathbf{H}_{TFR}$ since they do not necessarily have the same dimension. Therefore, before fusion, we used two FC layers (with layer normalization), denoted $\mathcal{FC}_{RS}$ and $\mathcal{FC}_{TFR}$, allowing to project $\mathbf{H}_{RS}$ and $\mathbf{H}_{TFR}$ into spaces of same dimension, d_{com}, obtaining two new embeddings, $\tilde{\mathbf{H}}_{RS}, \tilde{\mathbf{H}}_{TFR} \in \mathbb{R}^{d_{com}}$:

$$\tilde{\mathbf{H}}_{RS} = LN(\mathcal{FC}_{RS}(\mathbf{H}_{RS}))$$

$$\tilde{\mathbf{H}}_{TFR} = LN(\mathcal{FC}_{TFR}(\mathbf{H}_{TFR}))$$

where LN is the layer normalization operator.

Moreover, thanks to these embeddings, we can obtain a joint representation in a common fused latent space, using a fusion function, f:

$$\mathbf{H}_{fus} = f(\tilde{\mathbf{H}}_{RS}, \tilde{\mathbf{H}}_{TFR}) \tag{4.4}$$

This fusion function depends on the selected fusion strategy. We propose to use three different strategies

- Concatenation:

$$\begin{aligned} f : \mathbb{R}^{d_{com}} \times \mathbb{R}^{d_{com}} &\rightarrow \mathbb{R}^{2d_{com}} \\ (\mathbf{H}_1, \mathbf{H}_2) &\mapsto \mathbf{H}_1 \oplus \mathbf{H}_2 \end{aligned}$$

  with $\oplus$ the concatenation operator.

- Sum:

$$\begin{aligned} f : \mathbb{R}^{d_{com}} \times \mathbb{R}^{d_{com}} &\rightarrow \mathbb{R}^{d_{com}} \\ (\mathbf{H}_1, \mathbf{H}_2) &\mapsto \mathbf{H}_1 + \mathbf{H}_2 \end{aligned}$$

- Weighted sum:

$$\begin{aligned} f_{w_1,w_2} : \mathbb{R}^{d_{com}} \times \mathbb{R}^{d_{com}} &\rightarrow \mathbb{R}^{d_{com}} \\ (\mathbf{H}_1, \mathbf{H}_2) &\mapsto w_1 \times \mathbf{H_1} + w_2 \times \mathbf{H_2} \end{aligned}$$

  where $w_1, w_2 \in [0, 1]$ are two learnable parameters such that $w_1 + w_2 = 1$, indicating the global importance of each input representation to obtain the final fused representation (used for classification).

II.4.B Guided and regularized intermediate fusion

Global pipeline: In order to facilitate joint-training of the different representations' encoders, $\mathcal{E}_{RS}$ and $\mathcal{E}_{TFR}$, and regularize the final fused common latent space, we propose to use two strategies (see figure 4.4). For the former, we guide the joint-training of the encoders thanks to two iterated losses, one per input representation, allowing classification even if one representation is not available (this will be explained in the next paragraph). For the latter, we propose to adapt unsupervised DEC to a semi-supervised context, where the number of clusters to use is defined as the number of classes to classify, C.

Guided training: Guided training is obtained using two intermediate classifiers, one per input representation. Hence, during training of the whole model, we also train single feature vanilla classifiers, taking as input the encodings of each input representation[1], $\mathbf{H}_{RS}$ and $\mathbf{H}_{TFR}$. In fact, these classifiers correspond to the classification parts of the single feature models introduced in section II.2, denoted as C_{RS} and C_{TFR}. This will force the encoder of each representation to generate intermediate (single feature) latent spaces discriminative enough to do classification by themselves (thus classification with a single representation is possible)[2]. Moreover, let us denote by $\tilde{\mathbf{y}}_{\mathbf{RS}} = C_{RS}(\tilde{\mathbf{H}}_{\mathbf{RS}})$ and $\tilde{\mathbf{y}}_{\mathbf{TFR}} = C_{TFR}(\tilde{\mathbf{H}}_{\mathbf{TFR}})$ the intermediate classification output of the raw signal and the TFR, respectively. The iterated losses are defined as the cross entropy

[1] This is why our approach allows performing classification even if one representation is missing.

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

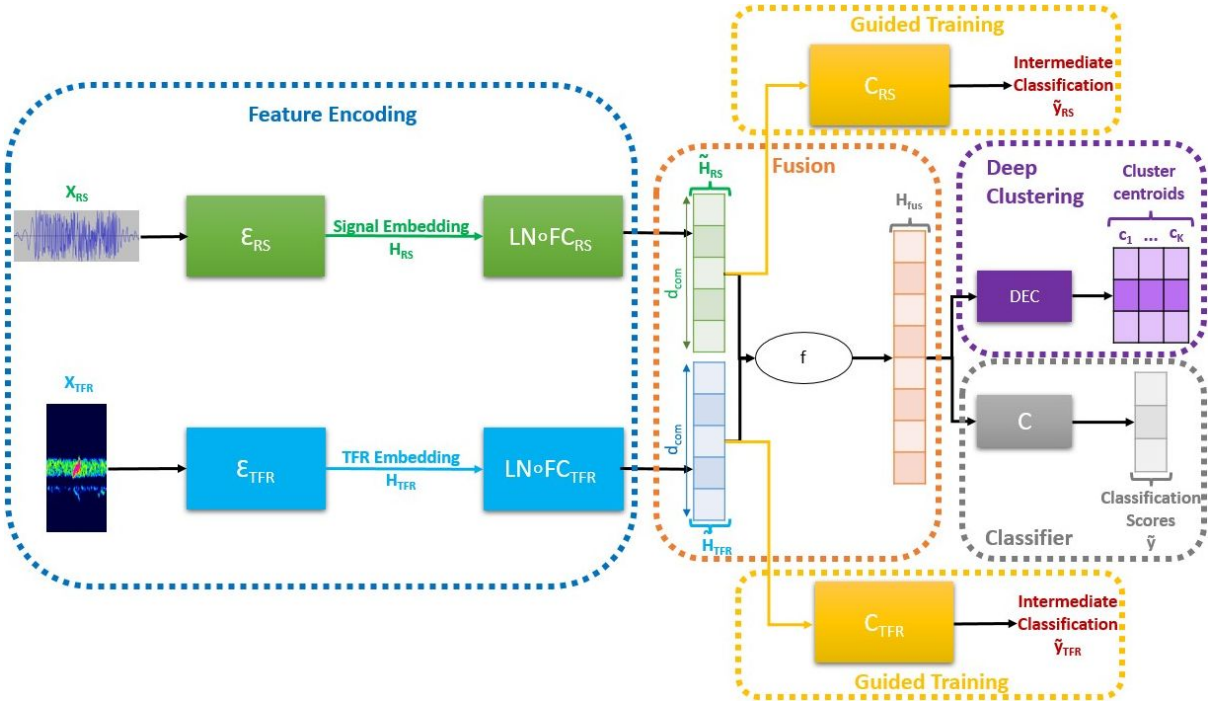


Figure 4.4: Global pipeline of our guided and regularized intermediate fusion approach. The intermediate representation, $\mathbf{H}_{fus}$, used for classification is guided by using two iterated losses, and regularized thanks to semi-supervised DEC.

(CE) loss between the intermediate classification outputs, and the true labels of the samples denoted as $\mathbf{y}$:

$$\mathcal{L}_{RS} = \mathcal{L}_{CE}(\tilde{\mathbf{y}}_{\mathbf{RS}}, \mathbf{y}) \tag{4.5}$$

$$\mathcal{L}_{TFR} = \mathcal{L}_{CE}(\tilde{\mathbf{y}}_{\mathbf{TFR}}, \mathbf{y}) \tag{4.6}$$

Semi-supervised DEC regularization: The last module of our intermediate fusion approach consists in semi-supervised DEC regularization of the joint representation embedding space. The rationale behind this is threefold. First, we want to enforce clustering in the final fused latent space in order to get a more discriminant space, improving the generalization capabilities of the obtained model. Second, we want to partially handle imbalanced datasets, as DEC is relatively robust against this. Finally, we want to be able to handle noisy labeled datasets (as ours HITS dataset are noisy), which can be done by applying a clustering technique which is independent of the labels (noise tolerant).

Moreover, in Chapter 2, we saw that the idea behind DEC is to form clusters in the latent space obtained by a DL encoder model, where the centroids of each cluster are initialized with k-means, and then learned through the training of the model. What is more, because of the formulation of DEC, it contributes to the update of the weights of the encoder model through the iterations.

More particularly, in our case, let us denote as $\mathbf{H}^1_{fus}$, ..., $\mathbf{H}^N_{fus}$ the generic fused embeddings obtained from the inputs samples (X^1_{RS}, X^1_{TFR}), ..., (X^N_{RS}, X^N_{TFR}), after the encoding and fusion modules. We propose to apply DEC to these samples (joint latent space) using C clusters (same as the number of classes). Thus, if we denote as $\mathbf{c}_1$, ..., $\mathbf{c}_C$ the centroids of the C clusters, DEC

will jointly optimize these centroids and the weights of the encoder models, $\mathcal{E}_{RS}$ and $\mathcal{E}_{TFR}$, as the fused representations depends on them.

On top of that, contrary to the original unsupervised DEC method which needs pre-training, we propose to avoid this by introducing a hyperparameter e_{init}, corresponding to the epoch from which DEC will be activated. Indeed, our DEC regularization approach is not applied from the beginning to avoid a random k-means intialization of the centroids: at the beginning of the training process the joint latent space is not yet structured which makes the unsupervised clustering difficult. Therefore, our proposed semi-supervised DEC regularization term can be defined as follows:

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

with e_c the current epoch number, $E = \{e \in \mathbb{N} / e \geq e_{init}\}$, and $\mathcal{L}_{DEC}$ is the DEC loss defined in Chapter 2.

Furthermore, the influence of e_{init} will be discussed in the experimental validation, but in a nutshell, we should choose e_{init} such that, from that epoch, the joint embedded space is relatively clustered, allowing a good initialization of the DEC centroids through k-means.

Final loss function: Our main goal is to do supervised classification using two different representations of a signal, namely the raw signal and a TFR. Therefore, we optimize a CE loss function, $\mathcal{L}_{CE}$, between the final predicted class scores, $\tilde{\mathbf{y}}$, and the true labels scores, $\mathbf{y}$. Indeed, the predicted labels scores are obtained by feeding a classifier C with the joint representation $\mathbf{H}_{fus} = f(\tilde{\mathbf{H}}_{RS}, \tilde{\mathbf{H}}_{TFR})$. The final loss function is then defined as follows:

$$\mathcal{L} = \mathcal{L}_{CE} + \alpha \times \mathcal{L}_{TFR} + \beta \times \mathcal{L}_{RS} + \gamma \times \tilde{\mathcal{L}}_{DEC} \tag{4.7}$$

where α, β, and γ are hyperparameters defining the importance of each term in the final loss. It is important to note that all the terms have an influence on the weights of the single feature encoders:

- $\tilde{\mathbf{y}} = C(f(LN(\mathcal{FC}_{RS}(\mathcal{E}_{RS}(X_{RS}))), LN(\mathcal{FC}_{TFR}(\mathcal{E}_{TFR}(X_{TFR})))))$ used in general CE loss, $\mathcal{L}_{CE}$.
- $\tilde{\mathbf{y}}_{\mathbf{RS}} = C_{RS}(LN(\mathcal{FC}_{RS}(\mathcal{E}_{RS}(X_{RS}))))$ used in the raw signal iterated loss, $\mathcal{L}_{RS}$.
- $\tilde{\mathbf{y}}_{\mathbf{TFR}} = C_{TFR}(LN(\mathcal{FC}_{TFR}(\mathcal{E}_{TFR}(X_{TFR}))))$ used in the TFR iterated loss, $\mathcal{L}_{TFR}$.
- $\mathbf{H}_{fus} = f(LN(\mathcal{FC}_{RS}(\mathcal{E}_{RS}(X_{RS}))), LN(\mathcal{FC}_{TFR}(\mathcal{E}_{TFR}(X_{TFR}))))$ used in the semi-supervised DEC loss, $\tilde{\mathcal{L}}_{DEC}$.

III Method evaluation

We validate different aspects of our approach through a series of five experiments. We mainly study the advantage of using multiple features, the influence of guided training, as well as a detailed study of the different components of the proposed semi-supervised DEC regularization.

III.1 Datasets

We used three medical signal datasets to evaluate our method: a HITS (TCD), PTB (ECG), and ESR (EEG).

Table 4.1: HITS-sada dataset train/test splits.

Split	No. patients	Total	SE	GE	Art.
Train	40	7 264	456	610	6 198
Test	11	1 421	240	392	789

III.1.A HITS

As our main interest is emboli classification, we used the same HITS as the ones described in the previous chapter (see Chapter 3). From these HITS, different datasets were created.

HITS-small-I: It corresponds to an improvement of the HITS-small dataset of the previous chapter (Chapter 3), and it is composed of 1 680 HITS from 51patients, distributed in three classes: 608 solid emboli, 616 gaseous emboli, and 456 artifacts. From these HITS, 1 541 are the same as the HITS-small dataset (39 patients), and are used for training and validation, whereas the remaining 139 are used for testing (coming from 13 new patients). It is important to note that the new 139 HITS do not follow the same distribution as the 1 541 used for training and validation, as their extraction was performed with a more recent version of ADMS, where the HBR computation was modified. What is more, in the 1 541 training and validation samples, the maximal observed length of a Doppler signal in terms of samples was of 1 400, whereas in the new 139 HITS, there are signals with up to 1600 samples (which also translates in more points used to compute the TFR).

HITS-sada: This dataset corresponds to a large HITS dataset that was semi-automatically labeled with the method introduced in the previous Chapter 3. All the semi-automatic data annotation (sada) procedure was done with $k_s = 10$ and $k_t = 10$ (see Chapter 3), and it was composed of several steps. First, we started by automatically propagating the labels from the 1 541 manually labeled HITS from other unlabeled samples using $K = 10$ and $\tau = 0.5$. Once there were no more samples to label, manual annotation was done using the 2D reduced latent space, with the objective of labeling zones without labeled samples (to ensure the annotation space coverage assumption), and boundaries between clusters of different classes. Then, automatic label propagation was re-applied, and this step was repeated several times. At last, to improve the quality of the dataset and reduce label-noise, all the solid emboli samples and a majority of the gaseous ones were carefully and manually verified, whereas the artifacts samples were rapidly verified. Thanks to this procedure, we obtained a dataset composed of 8 685 HITS from 51patients, distributed in three classes: 696 solid emboli, 1 002 gaseous emboli, and 6 987 artifacts (see Table 4.1). It is important to note that the test dataset used here is not the same as HITS-small-I, as the distribution of HITS per class and per subject is not the same, so no direct comparison can be made between the results obtained with these two datasets. For this dataset, we used 84% of the samples for training (7 295 HITS), and 16% for testing (1 390 HITS).

III.1.B PTB (ECG) dataset

General description: Since the HITS datasets are private ones, we also evaluate our approach on publicly available medical signal datasets. The first one is a Physionet (Goldberger, Amaral, Glass, Hausdorff, Ivanov, Mark, Mietus, Moody, Peng and Stanley, 2000) dataset for myocardial infarction identification, called PTB (Bousseljot *et al.*., 1995). This dataset is composed of 14 552 ECG lead-II recordings, with a sample frequency of 125 Hz. The heartbeats are distributed into two classes, normal (10 506) and abnormal (4 046), which are imbalanced.

Pre-processing: We used the standardized and pre-processed version of the dataset from (Kachuee, Fazeli and Sarrafzadeh, 2018)[1], where the signals are segmented into heartbeats (of around 1 s), denoised and normalized. On top of that, we computed a log-scaled spectrogram (TFR) using $n_{fft} = 32$, $n_{overlap} = 4$, and a Blackman window of length n_{fft}[2].

Data splitting: We split the dataset into three subsets: train (64%), validation (16%), and test (20%)[3]. The hyperparameters selection and study were performed using the validation split. However, for a fair comparison with (Ahmad *et al.*., 2021), the final model was trained by regrouping the train and validation subsets, in order to use 80% of the samples for training, and 20% for testing.

A supplementary larger ECG dataset, MIT-BIH, was also used to validate our late fusion approach, but for the sake of space, we do not present the results here. We refer the reader to (Vindas, Guepie, Almar, Roux and Delachartre, 2022) for further details.

III.1.C Epileptic seizure recognition (EEG) ESR dataset

To further validate our approach on a different medical classification task, we used the epileptic recognition dataset (ESR) from the UCI repository (Andrzejak, Lehnertz, Mormann, Rieke, David and Elger, 2002).

General description: The ESR dataset is composed of EEG signals with a sample frequency of 178 Hz, showing the cerebral activity of 500 subjects under different circumstances during 23.6 s: (1) seizure activity, (2) recording in a brain tumor area, (3) recording in a healthy brain area, (4) the patient had their eyes closed, and (5) the patient had their eyes opened. For more information, we refer the reader to (Andrzejak *et al.*., 2002).

Pre-processing: We used the pre-processed and publicly available version found at `https://www.kaggle.com/datasets/harunshimanto/epileptic-seizure-recognition`. In this version, the EEG signals were segmented into 23 segments of duration around 1 s, giving a total of 11 500 samples, distributed in the abovementioned five classes which are equally distributed (2 300 samples per class). However, as most works, we focus on binary classification where the first class is composed of the seizure activity samples (2 300), and the second one of the non seizure activity samples (9 200), giving an imbalanced dataset. Moreover, to compute the TFR, we used the same strategy as for the PTB dataset; we compute a log-scale spectrogram with $n_{fft} = 32$, $n_{overlap} = 4$, and a Blackman window of length n_{fft}.

Data splitting: We split the dataset into two subsets: train (90%) and test (10%)[3]. As no special hyperparameter tuning was performed, we do not used a validation set. On top of that, we follow the same splitting strategy as (Xu, Ren, Chen and Che, 2020), where no validation set was used.

[1]We use the public available versions found in `https://www.kaggle.com/datasets/shayanfazeli/heartbeat`

[2]The choice of these parameters was motivated by a trade-off between model performance, model complexity, and available training data. Indeed, lower values of n_{fft} reduce the performance of the model while reducing the number of parameters. Higher values of n_{fft} increase performance (up to some threshold value) while increasing the number of parameters.

[3]Because of the structure of the dataset, we were not able to obtain a subject-wise split.

III.2 Baselines

HITS: For these datasets, we use as baselines the single feature models presented in section II.2, and we compare their performances with respect to our late and intermediate fusion approaches.

Heartbeat categorization dataset: For the PTB dataset, we take as reference the single feature models presented in section II.2, as well as the multi-feature model of (Ahmad *et al.*., 2021), detailed in Chapter 2.

Epileptic seizure recognition dataset: For this dataset, we also use the single feature models of section II.2, in addition to the single feature models of (Xu *et al.*., 2020; Hilal, Albraikan, Dhahbi, Nour, Mohamed, Motwakel, Zamani and Rizwanullah, 2022), based on 1D CNN-LSTMs and sparse AE, respectively.

III.3 Evaluation metrics and strategy

All the models/methods were evaluated using a train/test split, and repeated 10 times for statistical purposes. For the HITS and PTB dataset, we have an additional validation test, used to choose and fine-tune the different hyperparameters.

Moreover, for the different models, we compute several metrics to evaluate their performances from different perspectives. For the classification point of view, we used the MCC, macro-averaged F1-score and the accuracy. At last, to measure the computation complexity of the models, we used the number of parameters of the models, as well as the number of multiplications and additions (mult-adds).

III.4 Experimental setup and results

Different experiments were performed to study our proposed methods. Hereafter we are going to detail the experimental setup and the obtained results.

For most of the experiments, we fixed the training and models' architectures, and these can be found in Tables 4.2 and 4.3:

- 1D-CNN-trans: single feature model taking as input the raw signal
- 2D CNN single feature model taking as input the log-scale spectrogram.
- Late Fusion: multifeature model, taking as input both representations, and using late learnable attention weights, where models 1D CNN-trans and 2D CNN are used to obtain the classification outputs to fuse.
- MIF-N: joint-trained multifeature intermediate fusion (MIF) model with no (N) guiding nor DEC regularization.
- MIF-GR: joint-trained guided and regularized (GR) MIF model.

Additionally, all the 2D CNN models were trained with Adam optimizer, whereas Noam was used for models using a transformer architecture. The batch size was of 32, except for the late fusion model, where the batch size was fixed to 16.

III.4.A Experiment 1: Single feature vs multi-feature models

Objective: The objective of this experiment is to compare multi-feature and single-feature models, without special guiding or regularization.

Table 4.2: Training parameters of the different models. α, β, and γ correspond to the importance of $\mathcal{L}_{TFR}$, $\mathcal{L}_{RS}$, and $\tilde{\mathcal{L}}_{DEC}$, respectively. *Cat* stands for concatenation and *Weight. Sum* for weighted sum. Two versions of our joint-trained model are evaluated: one without regularization and guided training, named *MIF-N*, and another with guiding and the proposed regularization, named *MIF-GR*. The 1D CNN-transformer and 2D CNN models are single feature models, taking as input the raw signal and a TFR, respectively. The other models are late fusion multifeature models, taking as input both the raw signal and a TFR.

Dataset	Model	Epochs	Learning rate	Weight decay	α	β	γ	e_{init}	Fusion
HITS-small-I	1D CNN-trans	150	0.07	$1e^{-7}$	-	-	-	-	-
	2D CNN	50	0.001	$1e^{-7}$	-	-	-	-	-
	Late Fusion	15	0.01	$1e^{-8}$	-	-	-	-	
	MIF-N	150	0.3	$1e^{-7}$	-	-	-	-	Cat.
					-	-	-	-	Weight. Sum
	MIF-GR				0.01	0.1	0.01	50	Cat
					0.001	1	0.1	50	Weight. Sum
PTB	1D CNN-trans	150	0.1	$1e^{-7}$	-	-	-	-	-
	2D CNN	50	0.0001	$1e^{-7}$	-	-	-	-	-
	Late Fusion	15	0.01	$1e^{-8}$	-	-	-	-	
	MIF-N	150	0.3	$1e^{-7}$	-	-	-	-	Cat.
					-	-	-	-	Weight. Sum
	MIF-GR				0.01	1	0.0001	50	Cat.
					0.01	0.1	0.1	50	Weight. Sum
ESR	1D CNN-trans	100	0.3	0.0001	-	-	-	-	-
	2D CNN	100	0.001	0.00001	-	-	-	-	-
	Late Fusion	15	0.001	$1e^{-8}$	-	-	-	-	
	MIF-N	200	0.3	0.0001	-	-	-	-	Cat.
					-	-	-	-	Weight. Sum
	MIF-GR				0.01	1	0.0001	50	Cat.
									Weight. Sum

Table 4.3: Architecture parameters of the models based on the used dataset. We refer the reader to figures 4.1 and 4.2 for the definition of the model parameters. Models 1D CNN-trans, 2D CNN, MIF-N, and MIF-GR are the same as those in Table 4.2. Model Late Fusion from Table 4.2 is not presented here since the base models of Late Fusion are 1D CNN-trans and 2D CNN models.

Dataset	Model	n_{head}	d_{hid}	n_{layers}	$p_{dropout}$	n_{proj}	d_{raw}	n_{conv}	$n_{filters}$	d_{com}	Pool
HITS-small-I	1D CNN-trans	8	64	8	0.1	10	128	2	-	-	-
	2D CNN	-	-	-	0.2	-	-	-	64	-	Max
	MIF-N	4	64	4	0.1	10	128	2	64	64	Max
	MIF-GR										
PTB	1D CNN-trans	8	64	8	0.1	10	128	4	-	-	-
	2D CNN	-	-	-	0.2	-	-	-	64	-	Max
	MIF-N	4	64	4	0.1	10	128	2	64	64	Max
	MIF-GR										
ESR	1D CNN-trans	4	8	4	0.3	4	64	2	-	-	-
	2D CNN	-	-	-	0.2	-	-	-	64	-	Max
	MIF-N	4	8	4	0.3	4	64	2	64	64	Max
	MIF-GR										

Experimental setup: We trained single-feature and multi-feature models 1D CNN-trans, 2D CNN, Late Fusion, and MIF-N, presented in Tables 4.2 and 4.3 with the indicated parameters, on three different medical signal datasets, HITS-small-I, PTB, and ESR.

Results: Results can be found in Table 4.4. Several points can be highlighted.

First, we can note that, for all the datasets, the best performing models that we proposed are always multi-feature ones. More particularly, our late fusion approach always outperform the single feature models, on the three datasets, with MCC, F1-Score, and accuracy improvements up to 8.77%, 7.07%, and 6.19%. However, this is not observed for the intermediate fusion approaches on the PTB and ESR datasets, where the performance tend to be similar to the worst performing single feature model.

Second, our proposed late fusion approach achieves state-of-the-art performances on HITS-small-I and PTB datasets compared to other multi-feature methods, with an improvement up to 3.41%, 1.83%, and 2.09% in terms of MCC, F1-Score, and accuracy, respectively. What is more, on the ESR dataset, although this same approach is generic (i.e. not designed specifically for this ESR dataset), it achieves the second-best performances, with an MCC, F1-Score, and accuracy performances gaps of 1.64%, 0.18%, and 0.23%, respectively, compared to (Hilal *et al.*, 2022; Xu *et al.*, 2020).

Third, compared to the manual classification performances on the PTB dataset (Makimoto, Höckmann, Lin, Glöckner, Gerguri, Clasen, Schmidt, Assadi-Schmidt, Bejinariu, Müller, Angendohr, Babady, Brinkmeyer, Makimoto and Kelm, 2020), all the models give better classification performances, in terms of F1-Score and accuracy, with an improvement up to 29.22% and 32.2% (respectively), for the late fusion model, with a great reduction in the variability.

Finally, we observe that multi-feature models often have more parameters than single-feature models, using more operations to obtain the final classification. However, the number of parameters for our proposed approaches, specially the late fusion, remains of the same order of magnitude as the largest single feature model.

Table 4.4: Experiment 1. Test classification performances of different single and multi-feature models on three medical datasets: HITS-small-I, PTB, and ESR. The results confirm the appeal and adaptability of multi-feature approaches as it can outperform the single-feature models, 1D CNN-trans and 2D CNN, on the three datasets. The proposed late fusion approach achieves state-of-the-art performance on two datasets, the HITS-small-I and PTB datasets, and an excellent performance on the ESR dataset. The number of multiplications and additions (mult-adds) is given in billions (G).

Dataset	Model	Representation	Fusion method	MCC	F1-Score	Accuracy	No. Parameters	No. mult-adds (G)
HITS-small-I	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	Both	Weight. Sum	**87.94 ± 2.60**	**91.44 ± 1.91**	**91.80 ± 1.83**	2 448 072	1.40
	MIF-N		Cat	84.53 ± 1.58	89.61 ± 1.06	89.71 ± 1.02	4 833 727	1.40
	MIF-N		Weight. Sum	85.93 ± 1.21	90.56 ± 0.78	90.58 ± 0.82	4 876 233	1.40
PTB	Manual classification Makimoto *et al.*. (2020)	-	-	-	70 ± 7	67 ± 7	-	-
	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*
	Ahmad *et al.*. (2021)	GAF MTF RP	Weight. Sum	-	98	99.2	9 259 427	-
	Late Fusion	Both	Weight. Sum	**98.45 ± 0.49**	**99.22 ± 0.25**	**99.38 ± 0.20**	2 321 594	0.089
	MIF-N		Cat	97.11 ± 0.43	98.6 ± 0.22	98.84 ± 0.18	2 128 820	0.236
	MIF-N		Weight. Sum	97.29 ± 0.50	98.64 ± 0.25	98.91 ± 0.20	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*
	Hilal *et al.*. (2022)	Raw signal		**99.09**	**98.89**	98.67	-	-
	Xu *et al.*. (2020)			–	98.59	**99.39**	-	-
	Late Fusion	Both	Weight. Sum	*97.45 ± 1.49*	*98.71 ± 0.77*	*99.16 ± 0.51*	1 665 724	0.070
	MIF-N		Cat	93.40 ± 1.32	96.67 ± 0.68	97.89 ± 0.45	1 801 590	0.123
	MIF-N		Weight. Sum	93.01 ± 2.22	96.45 ± 1.22	97.77 ± 0.69	1 803 456	0.123

Table 4.5: Experiment 2: Test classification performances of different multi-feature models on three medical datasets: HITS-small-I, PTB, and ESR. The results confirm the appeal and adaptability of guiding and regularization while jointly training end-to-end models as it improves the performances of the intermediate fusion models, on the three datasets. Moreover, our proposed guided and regularized hybrid model 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	Representation	Fusion method	MCC	F1-Score	Accuracy	No. Parameters	No. mult-adds (G)
HITS-small-I	Late Fusion	Both	Weight. Sum	87.94 ± 2.60	91.44 ± 1.91	91.80 ± 1.83	**2 448 072**	1.40
	MIF-N		Cat	84.53 ± 1.58	89.61 ± 1.06	89.71 ± 1.02	4 833 727	
	MIF-N		Weight. Sum	85.93 ± 1.21	90.56 ± 0.78	90.58 ± 0.82	4 876 233	
	MIF-GR		Cat	**91.89 ± 2.64**	**94.31 ± 1.66**	**94.53 ± 1.74**	*4 833 727*	
	MIF-GR		Weight. Sum	*88.28 ± 6.91*	*91.69 ± 4.81*	*92.01 ± 4.73*	4 876 233	
PTB	Ahmad *et al.*. (2021)	GAF MTF RP	Weight. Sum	-	98	99.2	9 259 427	-
	Late Fusion	Both	Weight. Sum	98.45 ± 0.49	99.22 ± 0.25	99.38 ± 0.20	2 321 594	**0.089**
	MIF-N		Cat	97.11 ± 0.43	98.6 ± 0.22	98.84 ± 0.18	2 128 820	*0.236*
	MIF-N		Weight. Sum	97.29 ± 0.50	98.64 ± 0.25	98.91 ± 0.20	2 130 366	
	MIF-GR		Cat	**99.28 ± 0.11**	**99.64 ± 0.05**	**99.71 ± 0.04**	**2 128 820**	
	MIF-GR		Weight. Sum	*99.18 ± 0.25*	*99.59 ± 0.13*	*99.67 ± 0.10*	2 130 366	
ESR	Hilal *et al.*. (2022)	Raw signal	-	**99.09**	**98.89**	98.67	-	-
	Xu *et al.*. (2020)			–	98.59	**99.39**	-	-
	Late Fusion	Both	Weight. Sum	*97.45 ± 1.49*	*98.71 ± 0.77*	*99.16 ± 0.51*	**1 665 724**	**0.070**
	MIF-N		Cat	93.40 ± 1.32	96.67 ± 0.68	97.89 ± 0.45	1 801 590	*0.123*
	MIF-N		Weight. Sum	93.01 ± 2.22	96.45 ± 1.22	97.77 ± 0.69	1 803 456	
	MIF-GR		Cat	96.51 ± 0.46	98.25 ± 0.23	98.88 ± 0.15	*1 801 590*	
	MIF-GR		Weight. Sum	96.85 ± 0.70	98.42 ± 0.35	98.98 ± 0.23	1 803 456	

III.4.B Experiment 2: Advantage of guided and regularized end-to-end training

Objective: The objective of this experiment is to study the influence of guiding and regularization while jointly training end-to-end models.

Experimental setup: We trained (intermediate and late fusion) multi-feature models Late Fusion, MIF-N, and MIF-GR presented in Tables 4.2 and 4.3 with the indicated parameters, on three different medical signal datasets, HITS-small-I, PTB, and ESR.

Results: Results can be found in Table 4.5. Several interesting points can be noted.

First, for all the datasets and all the models, our proposed guiding and regularization strategy allow improving the performances of the MIF models, with a margin up to 7.36% in terms of MCC (same behavior for the F1-Score and the accuracy). More interestingly, this strategy allows obtaining MIF models outperforming the single feature models, on the three datasets, by a margin up to 4.80% in terms of MCC. Additionally, the obtained models with our proposed guiding and regularization strategy are more stable than models without it (late fusion and single feature models) with a variability that can be reduced by a factor up to 10.

Second, MIF-GR models achieve state-of-the-art performances on the HITS and PTB datasets. Indeed, on the HITS dataset, the best performing model is MIF-GR with concatenation, outperforming the late fusion approach by a margin of 3.95% in terms of MCC. Similarly, in the PTB dataset, the MIF-GR model with concatenation outperforms the late fusion approach by a margin of 0.83% in terms of MCC, and (Ahmad *et al.*., 2021) by a margin of 0.51% in terms of

accuracy. However, this is not observed in the ESR dataset, where MIF-GR are the third best performing models in terms of accuracy, outperformed by (Xu *et al.*., 2020) and the late fusion approach, by a margin of at least 0.51%.

At last, as in the previous experiment, we can note that multi-feature models tend to have more parameters than the single-feature ones. Nevertheless, the order of magnitude of these parameters remains similar to single feature models. Additionally, MIF models can have a similar (or even less) number of parameters than the late fusion approach, with a comparable number of operations (mult-adds). What is more, our proposed guiding and regularization strategy does not increase the number of parameters, as it does not introduce new parameters needed for inference. Indeed, the vanilla classifiers used to guide the training of each feature encoder are only necessary during training, and not during inference, except if trying to predict with only one representation (in case the other one is missing).

III.4.C Experiment 3: Influence of guided training

Objective: The objective of this experiment is to study the influence of guided training on the final classification performances of MIF models.

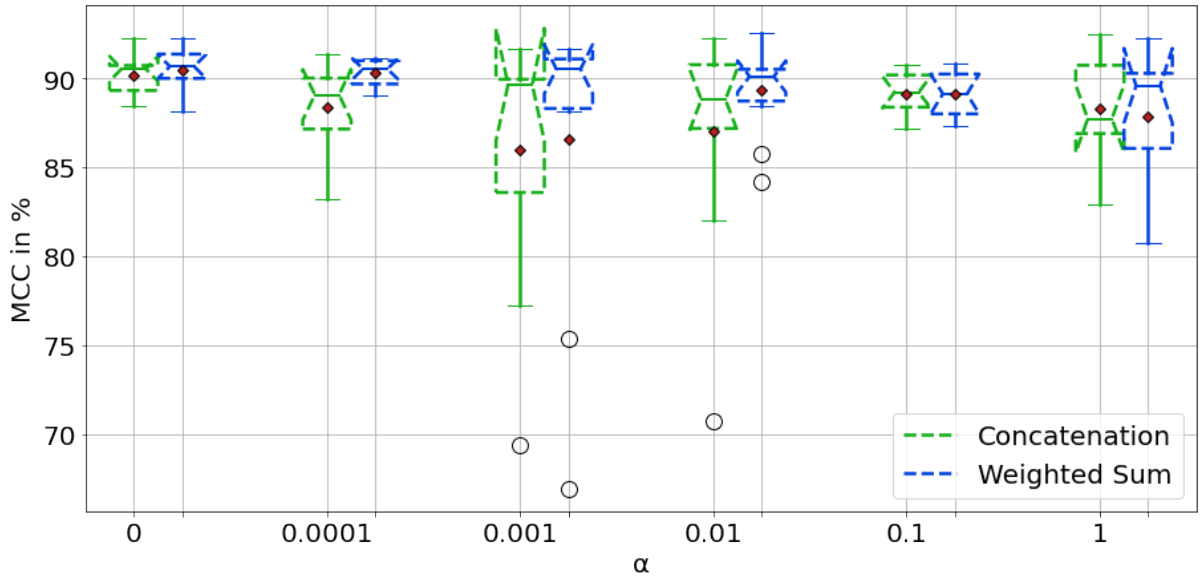
Experimental setup: To this end, we trained model MIF-GR from Table 4.3 without semi-supervised DEC regularization ($\gamma = 0$), and introducing the guiding at different levels (see equation 4.7) in the model's architecture: (1) in the TFR encoder's latent space only ($\alpha > 0$ and $\beta = 0$), (2) in the raw signal encoder's latent space ($\alpha = 0$ and $\beta > 0$), (3) and in both encoder's latent spaces ($\alpha > 0$ and $\beta > 0$). On top of that, we varied the values of α and β in the range $\{0, 0.0001, 0.001, 0.01, 0.1, 1\}$, and we used the HITS and PTB datasets for sub-experiments (1) and (2), whereas for experiment (3) we only used the HITS dataset due to energy and resources limitations.

Results: The results can be found in figures 4.5, 4.6, and 4.7. Several observations can be done.

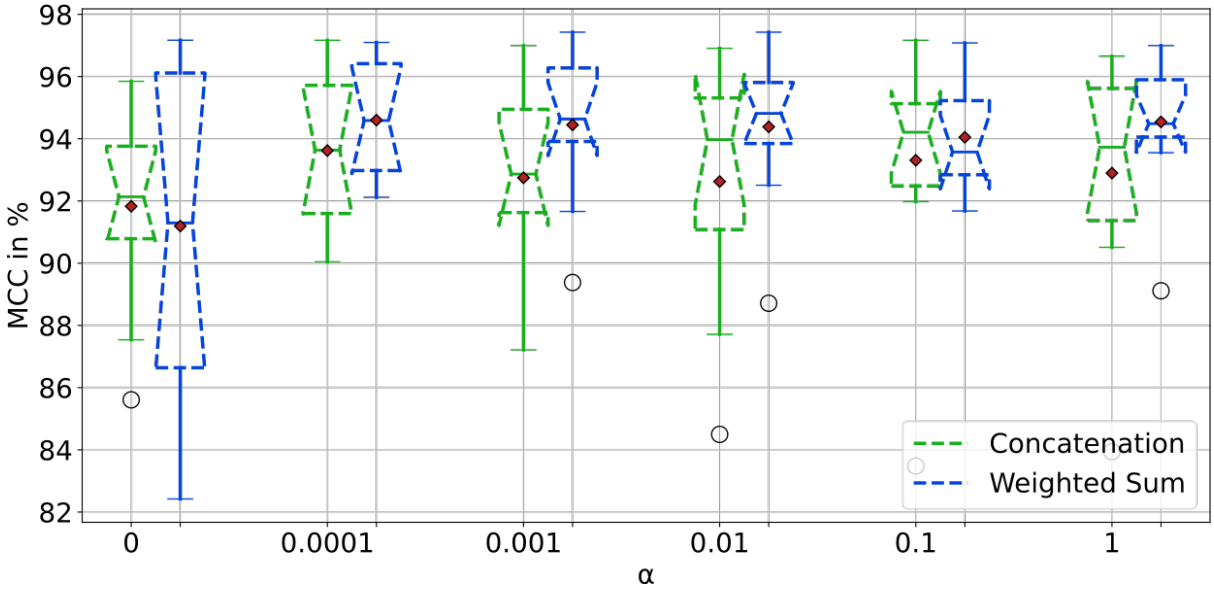
First, from figure 4.5 we can see that, on both datasets, guiding the training of the TFR encoder does not allow important improvements on the classification performances. Indeed, on the HITS dataset, for $\alpha = 1e^{-4}$, MIF with weighted sum yields an MCC of $90.28 \pm 0.77\%$ when guiding is applied, whereas the unguided model achieves $90.49 \pm 1.21\%$. On the other hand, on the PTB dataset we observe a higher performance gap, with $93.31 \pm 3.65\%$ MCC for MIF guided and with concatenation ($\alpha = 0.01$), whereas for the unguided counterpart we obtain $91.83 \pm 3.13\%$. These results are relatively stable for different values of α.

Second, we can see the importance of the raw signal encoder's guiding (see figure 4.6): it has a favorable effect on the classification performances of the model. Indeed, on both datasets, the presence of guiding tends to improve the classification performances, with MCC improvements greater than 0.84% and 6.68% on the HITS and PTB datasets, respectively. More interestingly, it seems that what is important is the presence of raw signal encoder's guiding rather than its amplitude (value of β), as different values of β achieve similar results on both datasets (and more particularly in the PTB dataset).

Finally, from figure 4.7 we observe that guiding both encoders has also a beneficial effect on the model performances. Globally, for $\alpha \leq \beta$, there is a performance improvement allowing to outperform the unguided MIF models. Moreover, when we fix the importance of $\mathcal{L}_{TFR}$ (value of α) and we increase the one of $\mathcal{L}_{RS}$ (value of β), the MCC tends to increase. As well, when we do the opposite (fix β and vary α), the classification performances do not necessarily increase, but it remains relatively stable, specially for large values of β.



(a) HITS-small-I dataset



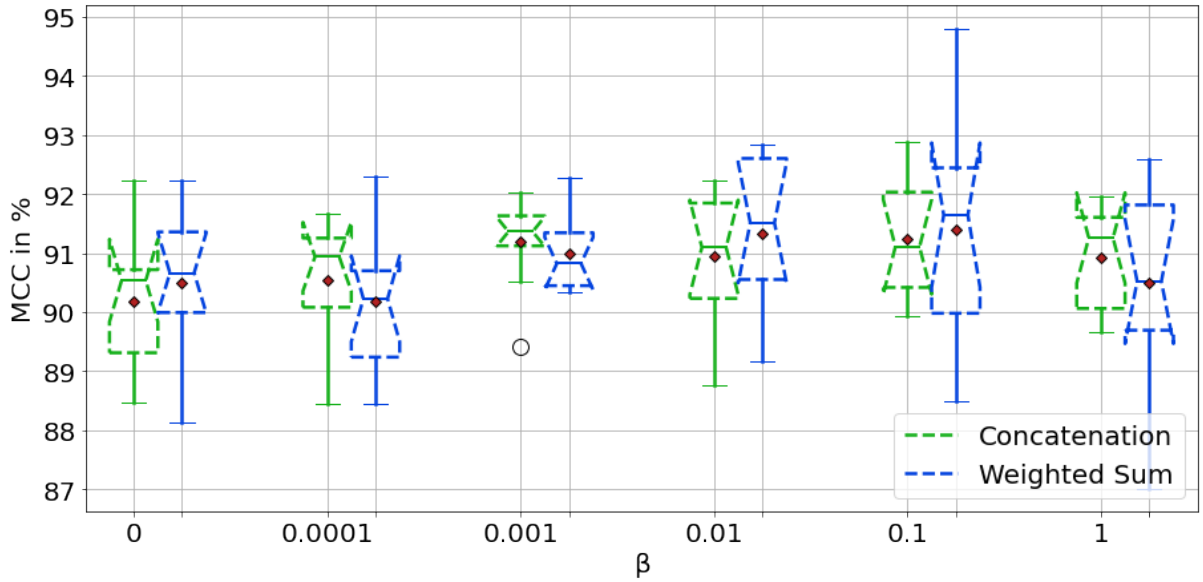
(b) PTB dataset

Figure 4.5: Experiment 3. Validation classification performances (MCC) of two MIF 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 performances of the model considerably with respect to the unregularized model.

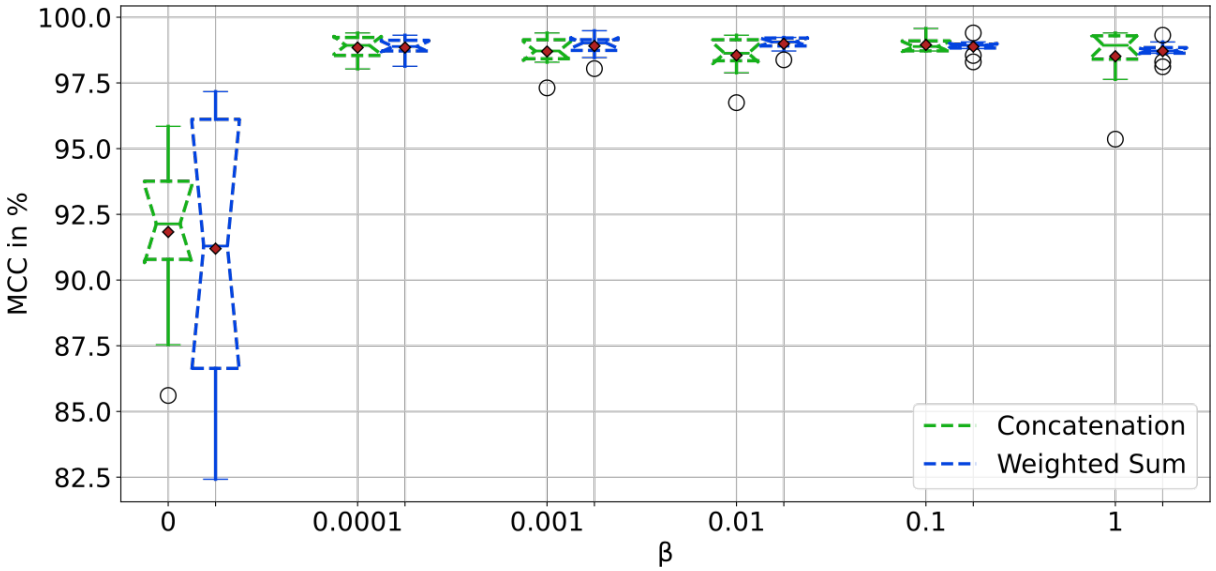
III.4.D Experiment 4: Influence of DEC regularization

Objective: The objective of this experiment is to study the influence of our semi-supervised DEC regularization on the model performances.

Experimental setup: To do this, we also trained model MIF-GR from Table 4.3 on the HITS-small-I and PTB datasets, but this time we removed guiding ($\alpha = \beta = 0$) and kept the regulariza-



(a) HITS-small-I dataset



(b) PTB dataset

Figure 4.6: Experiment 3. Validation classification performances (MCC) of two MIF 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 $\mathcal{L}_{RS}$. We observe that guiding the training of the 1D CNN-transformer encoder can considerably increase the classification performances of the model with respect to the unregularized model, especially in the PTB dataset (imbalanced dataset).

tion ($\gamma > 0$). On top of that, we varied the values of γ in the range $\{0, 0.0001, 0.001, 0.01, 0.1, 1\}$.

Results: Results can be found in figure 4.8. Some interesting points can be highlighted.

First, we can see that, even if the HITS-small-I dataset is balanced, semi-supervised DEC regularization do not damage the classification performances, but they are not globally im-

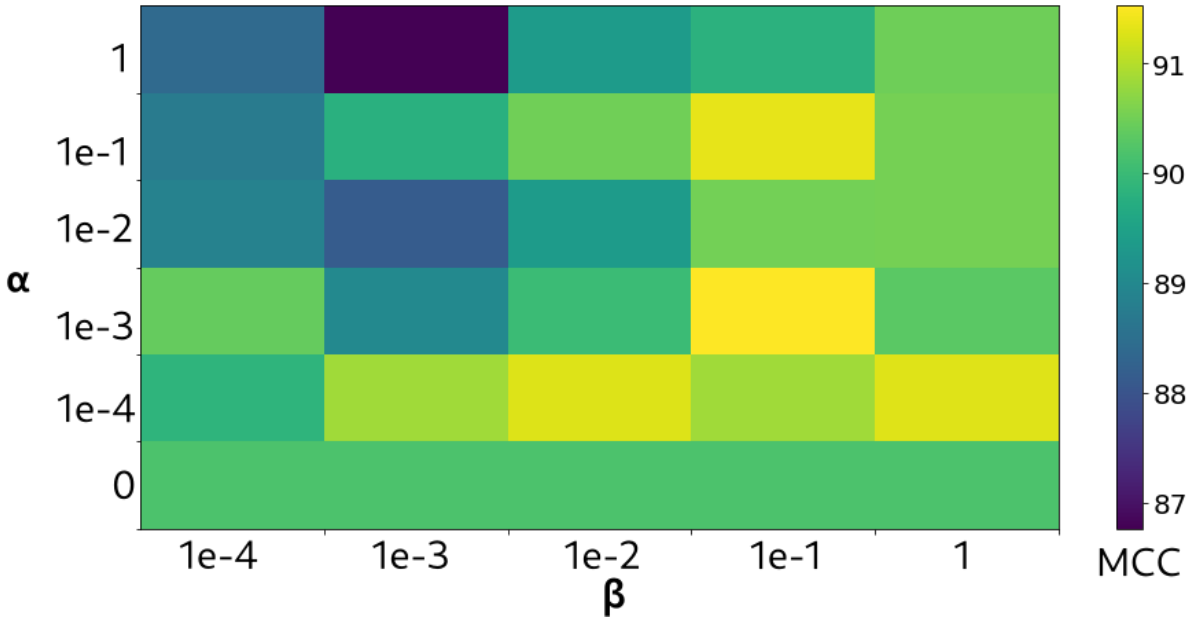


Figure 4.7: Experiment 3. Guided training on the two latent spaces (2D CNN TFR encoder and 1D CNN-transformer raw signal encoder) for the MIF model with concatenation on the HITS dataset. α and β correspond to the importance of $\mathcal{L}_{TFR}$ and $\mathcal{L}_{RS}$, respectively. We see that the 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.

proved either[1]. Indeed, semi-supervised DEC regularization allows achieving similar or better classification performances than when it is not applied, with 91.09 ± 1.09 ($\gamma = 0.001$) and 90.45 ± 1.43 ($\gamma = 0.1$) MCC against 90.18 ± 1.03 and 90.49 ± 1.21 MCC for the concatenation and weighted sum fusion strategies, respectively.

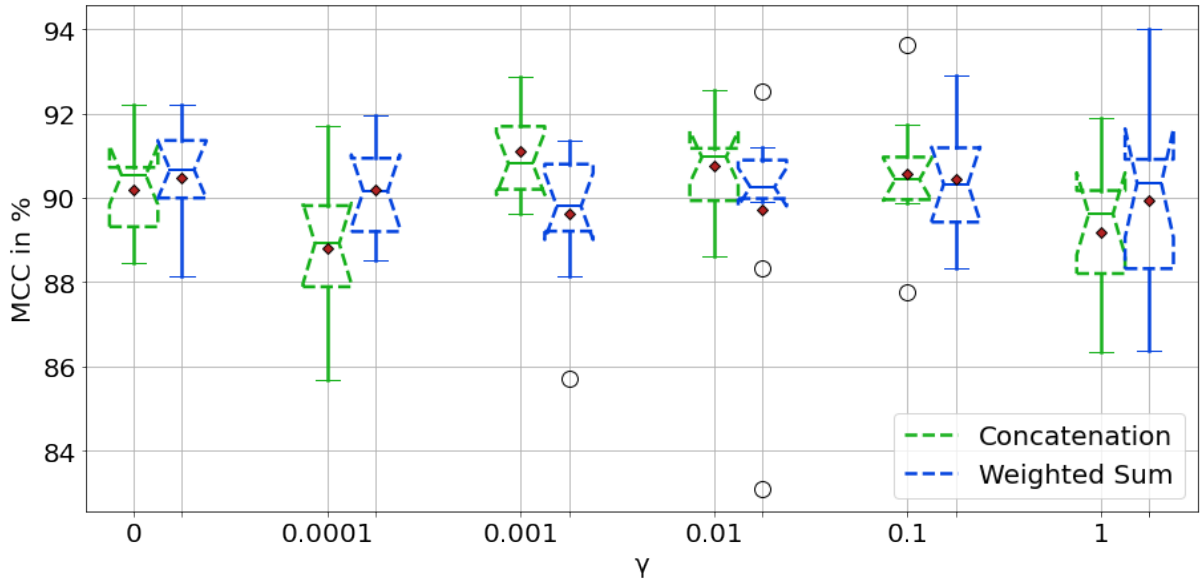
Second, the results show the interest of our semi-supervised DEC regularization to partially handle imbalanced datasets[2]. Indeed, the HITS dataset is balanced, contrary to PTB dataset (imbalance ratio of 2.6). Therefore, as expected, DEC regularization has a higher influence on the PTB trained models than those trained on the HITS-small-I dataset. More interestingly, when the importance of $\hat{\mathcal{L}}_{DEC}$ increases (higher values of γ), the classification performances tend to increase too, while its variability decreases. In that way, we obtain important performances gaps with, $95.07 \pm 3.07\%$ and $95.64 \pm 1.52\%$ MCC against $91.83 \pm 3.12\%$ and 91.19 ± 5.09 for the unregularized concatenation and weighted sum strategies models, respectively.

III.5 Discussion

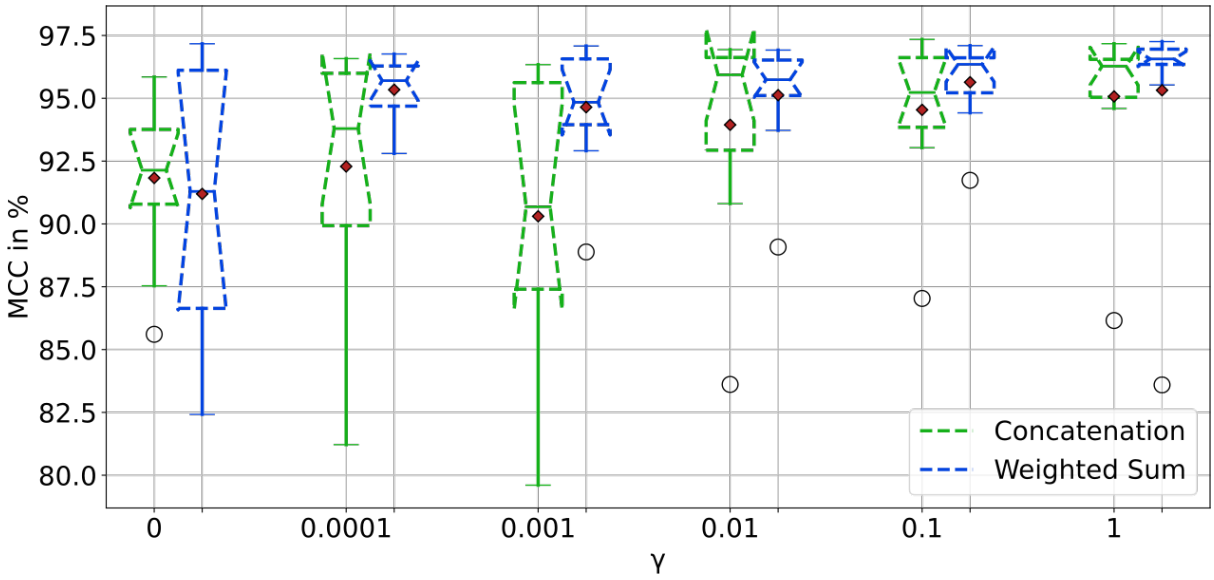
III.5.A Experiment 1: Single feature vs multi-feature models

The results of this first experiment confirm the interest of using multiple features of a same modality to enhance the classification performances of a model, which was validated using three medical signal datasets. Indeed, even if only one modality is used, different representations can have complementary information, therefore combining them can have a beneficial effect. In our case, the raw signal focuses in the temporal context as well as the amplitude

[1]This was further studied in our work (Vindas, Roux, Guépié, Almar and Delachartre, 2023a)
[2]This was further studied in our work (Vindas, Roux, Guépié, Almar and Delachartre, 2023a)



(a) HITS-small-I dataset



(b) PTB dataset

Figure 4.8: Experiment 4. Validation classification performances (MCC) of two MIF 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 $\tilde{\mathcal{L}}_{DEC}$.

information, whereas the TFR focuses on the spectral characteristics of the sample. To further analyze this, we can observe the learned attention weights per representation and per class in Table 4.6. If we focus on the HITS-small-I dataset we can see that, for the artifact class, the raw signal is as important as the TFR for the final decision of the model. However, for the gaseous and solid emboli classes, the TFR is more important. This is coherent with the emboli classification procedure used by expert annotators, where the TFR is primarily used to differentiate artifacts from emboli, and solid emboli from gaseous emboli; it is mainly in case of doubt that the expert listens to the raw Doppler signal.

Furthermore, we tested different multi-feature approaches on different datasets, and our

Table 4.6: Experiment 1. Learned late fusion attention weights (median and mean absolute deviation) for the late fusion model on the HITS-small-I and PTB datasets.

Dataset	Class	TFR	Raw Signal
HITS-small-I	Artifact	0.46 ± 0.29	0.54 ± 0.29
	Gaseous emboli	0.65 ± 0.17	0.35 ± 0.17
	Solid emboli	0.71 ± 0.15	0.29 ± 0.15
PTB	Normal	0.49 ± 0.12	0.51 ± 0.12
	Abnormal	0.18 ± 0.10	0.82 ± 0.10

proposed late fusion approach outperforms all the single feature models on the three tested datasets, showing its generality. Indeed, these three datasets correspond to different classification tasks, and our late fusion approach showed the same behavior with great performances and small variability (more stable results) on all the datasets. This can be explained by the fact that, contrary to other multi-feature models such as (Ahmad *et al.*, 2021), we exploit two common and simple representations in signal processing, namely the raw signal, and the TFR, instead of designing specific representations for one particular task. Additionally, even if our method was applied for medical signal classification, we designed it to be general, and easily adaptable to other types of inputs and/or models.

By the same token, the late fusion approach achieves state-of-the-art performances on the HITS and PTB datasets. However, we cannot easily compare our approaches with those of (Ahmad *et al.*, 2021), as neither the MCC nor the variability of the classification metrics are given in their publication. Indeed, the PTB dataset is an imbalanced dataset, so the MCC is better suited to compare classifiers than the accuracy or the F1-Score (Chicco and Jurman, 2020). Additionally, the variability is important as in some cases one would prefer smaller mean performances but more stable results (low variability), than higher mean performances with high variability. As well, even though our approaches do not reach state-of-the-art performances on the ESR dataset, our late fusion approach achieve excellent performances. Nevertheless, the comparison with other state-of-the-art models such as (Xu *et al.*, 2020; Hilal *et al.*, 2022) is not straightforward as the evaluation strategy/sets[1] are not the same, and these works did not give the standard deviation of their results. In fact, in the ESR dataset, it is very difficult to create a subject-wise train/test split or cross-validation folds, as subjects do not always have samples from all the classes, and their quantity vary from one subject to another one. This can lead to overconfident results when evaluating the models. Likewise, the state-of-the-art methods to which we compare were specifically designed to perform the task of the targeted dataset, whereas our approaches are more generic.

On top of that, we observed that the classification performances on the HITS-small-I dataset are considerably lower than the ones obtained on the PTB or ESR dataset. This can be explained by different reasons. Firstly, the size of the HITS-small-I dataset is smaller than the other two datasets, with around 500 samples per classes, against at least 5000 for PTB and at least 2300 for the ESR dataset. Secondly, the available temporal context is around 4 times smaller for the HITS-small-I dataset compared to the PTB and ESR datasets. Indeed, for PTB and ESR datasets, the samples represent events of at least 1 s, whereas for the HITS-small-I dataset they represent events of around 0.250 s (less than one cardiac cycle). Thirdly, the HITS classification task is more complex, specially when identifying solid emboli from gaseous emboli, as even expert

[1](Xu *et al.*, 2020) used a 80/20% train/test split, whereas (Hilal *et al.*, 2022) used 10-fold cross-validation.

annotators and clinicians have difficulties identifying them.

What is more, even if the multi-feature approaches tend to have a larger number of parameters and mult-adds compared to single feature models, they do not explode, and the order of magnitude remains similar. Globally, we can observe that the number of parameters of the multi-feature models as well as the number of mult-adds, is roughly the sum of the same quantities for each single feature encoder composing the multi-feature model. Additionally, it is important to note that, compared to the multi-feature model of (Ahmad *et al.*, 2021), our models have at least 4 times fewer parameters, with similar or even better classification performances. The size of all the models is smaller than 35 MB, and their mean inference time is smaller than 1 s (using Intel (R) Xeon (R) CPU E5-2650L v3 @ 1.80GHz and no GPU).

Finally, even if the late fusion approach achieves better performances than the intermediate fusion MIF approaches, it has two main drawbacks. First, the models are longer to train, as two different single feature models have to be pre-trained (preferably on the same data) before training the attention weights. This drawback can be partially solved if some computational resources are available, as the models can be trained in parallel. Second, late fusion is harder to optimize as we have two different sets of hyperparameters (one per single feature model), which can take some time to properly fine-tune.

III.5.B Experiment 2: Advantage of end-to-end training

The results of this experiment confirm the interest of end-to-end multifeature jointly trained models, specially when guiding and regularization is done. Indeed, thanks to guiding, we have more clustered intermediate latent spaces, specially for the raw signal, whereas semi-supervised DEC regularization enhance clustering in the fused common latent space that is used for final classification. To reveal this, for the HITS-small-I and PTB datasets, we projected the different latent spaces of MIF-GR and MIF-N on a 2D plane, using UMAP (McInnes *et al.*, 2020), and for each obtained set of 2D points, we computed the silhouette score (figures 4.9 and 4.10). From the one hand, these figures show that, regarding the TFR guided and unguided latent spaces, there is no obvious difference between them, with silhouette scores of 0.21 ± 0.05 and 0.19 ± 0.04 on the HITS-small-I dataset, respectively. On the other hand, if we focus on the raw signal guided and unguided latent spaces, we observe a more striking difference, with an important gap between silhouette scores, with 0.38 ± 0.04 and -0.03 ± 0.01 on the HITS-small-I dataset, respectively. Likewise, the difference between the DEC regularized and unregularized fused common latent spaces is not that evident, as we achieve respective silhouette scores of 0.36 ± 0.11 and 0.39 ± 0.05 on the HITS-small-I dataset.

Furthermore, the guided and regularized MIF-GR model performs comparably and even better than state-of-the-art models, including the late fusion approach of the previous experiment. This can be explained by the fact that, as the late-fusion approach, MIF-GR is able to exploit the complementarity of the different representations thanks to the guided and regularized joint-training. As we saw in the previous experiment, the two chosen representations are generic and complementary, as the raw signal focuses on the temporal and amplitude information, whereas the TFR focuses on the spectral information. Therefore, with our approach, each encoder is able to (partially) compensate for the weaknesses of the other one, as in the late-fusion approach. Nevertheless, the results of Table 4.5 also show that, if no guiding nor DEC regularization is applied, then the individual and common latent spaces are not discriminative enough, thus giving similar performances than some single-feature models. Additionally, as for the late-fusion approach, our MIF-GR models are not able to outperform the state-of-the art methods on the ESR dataset (which can be justified by the same points mentioned in the discussion of experiment 1), but the behavior is the same as in the other datasets: guiding and DEC regularization improve the classification performances with respect to MIF-N models.

Moreover, we can note that, on the three datasets, if guiding and regularization are not applied, the MIF models can give worst results than single feature models. In fact, jointly training multi-feature models makes the learning of each single feature encoder more difficult, thus penalizing the final classification performances. Therefore, this shows that guiding and regularization are important components of our approach.

At last, we can compare the different models in terms of number of parameters and mult-adds (in billions, G), to better analyze them. First, we notice that, globally for all the datasets, with respect to late-fusion approach, the number of parameters and mult-adds is not considerably increased as they remain of the same order of magnitude. This can be justified by the fact that the core of our method are not the used single-feature encoders, but rather the joint-training strategy assisted by the guiding and the semi-supervised DEC regularization. Because of this, the number of parameters remains of the same order of magnitude as the single-feature models. As well, this is important as we want to keep this multi-feature models nearly as fast and as lite as single-feature models for practical usage. By the same token, if we compare our approach to the one of (Ahmad *et al.*., 2021) on the PTB dataset, our MIF-GR approach not only outperforms it in terms of classification, but it also has at least 4.3 fewer parameters, giving lighter models.

III.5.C Experiment 3: Influence of guided training

The results of this experiment showed the interest of guided training to enhance the classification performances of joint-trained MIF models, on the HITS and PTB datasets.

First, we can see that, for the tested medical signal datasets, the guiding of the raw signal's encoder is crucial, as it is the one allowing to increase the most the classification performances. As we mention it before, jointly training two different encoders with different inputs makes the learning process harder, therefore guiding can help to ease this problem. The UMAP 2D projections in figures 4.9 and 4.10 show that guiding allows to learn more discriminative features for the raw signal's encoder latent space. On the other hand, when no guiding is done, we can see that the obtained features are not discriminative at all (translated by silhouette scores close to 0), which has a negative impact on the obtained common representation, and therefore in the final classification performances.

Second, the results in this experiment allow us to have some guidelines about an strategy to choose α, β, and γ. Indeed, when the importance of $\mathcal{L}_{TFR}$ is higher than the importance of $\mathcal{L}_{RS}$ (i.e., $\alpha > \beta$), the classification performances tend to decrease. This is consistent with the results in figures 4.5, 4.6, 4.9, and 4.10, as they show that the raw signal's encoder latent space is harder to learn, whereas the one of the TFR's encoder is easier. Therefore, giving too much importance to $\mathcal{L}_{TFR}$ at the expense of $\mathcal{L}_{RS}$ avoids important gains in classification performances, as we guide the simple representation (TFR) and not enough that difficult one (raw signal). Thus, we recommend choosing α and β such that $\alpha \leq \beta$, since the guiding of the raw signal's encoder is more important than the one of the TFR's encoder in a joint training context.

Finally, linked to the previous point, the results in figures 4.5 and 4.10 showed that guiding training of the TFR's encoder is not crucial, as it does not have a significant negative nor positive impact on the classification performances of the model. On the contrary, the guiding of the raw signal's encoder training is important. Therefore, guiding the training of both encoders can mostly be beneficial for the model, specially for other types of models and representations. Indeed, here we focus on three medical signal datasets, two type of models, and two type of features, but if the models, features or dataset characteristics change, the guiding behavior may be the opposite, so keeping this guiding for all the encoders (of the used representations) allows being more robust to model, representations, or dataset changes. Additionally, as guiding enforces each single-feature encoder to learn discriminative features

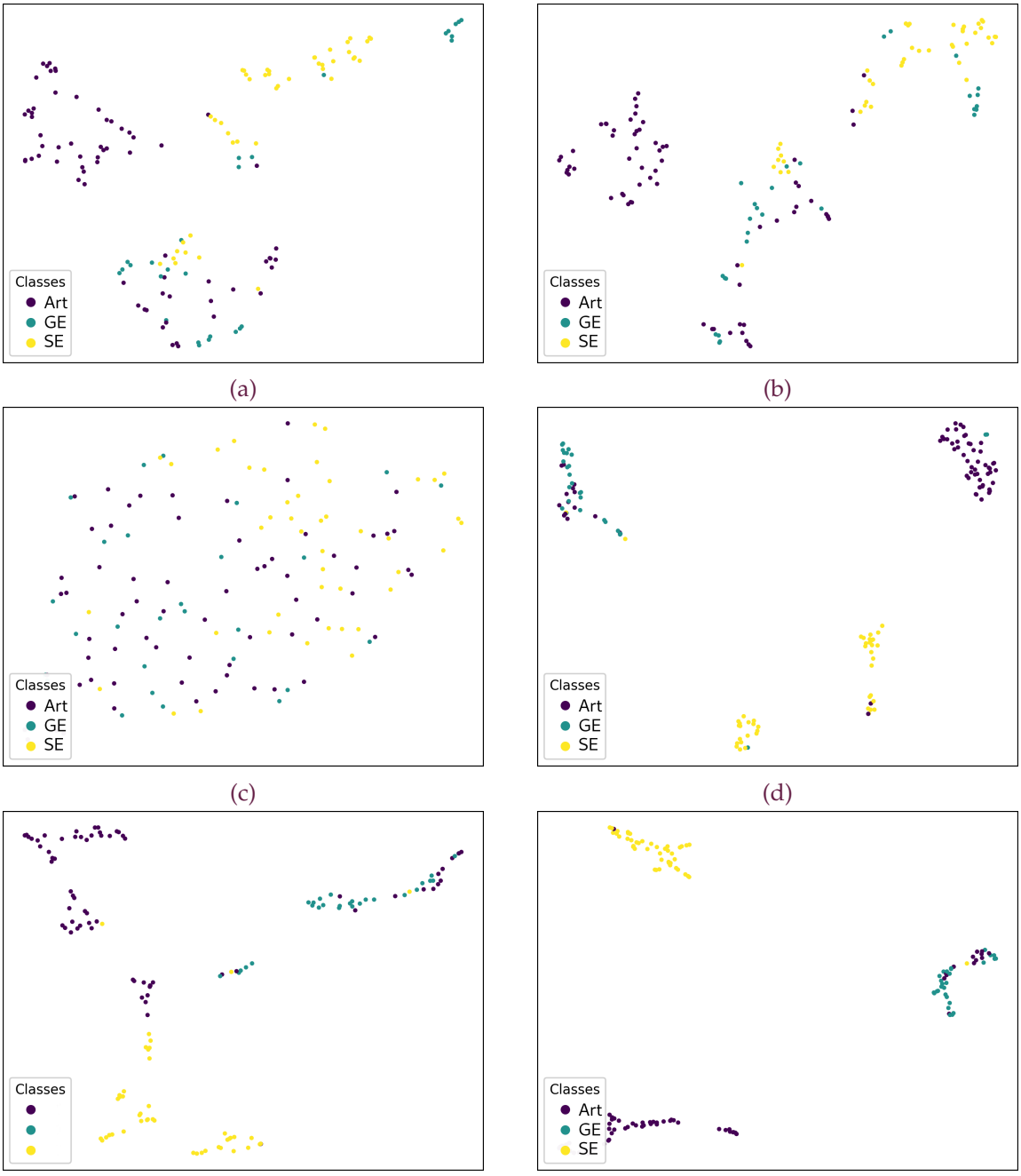


Figure 4.9: Experiment 1. Test embeddings of the MIF-GR and MIF-N models on the HITS-small-I dataset. (a) TFR encoder's latent space without regularization, (b) TFR encoder's latent space with regularization, (c) raw signal encoder's latent space without regularization, (d) raw signal encoder's latent 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 $e_{init} = 50$.

(which was shown through the UMAP 2D projections and the associated silhouette scores greater than 0.2), it allows being more robust against missing features, as one can deactivate the encoder of the missing feature during inference, and still perform classification.

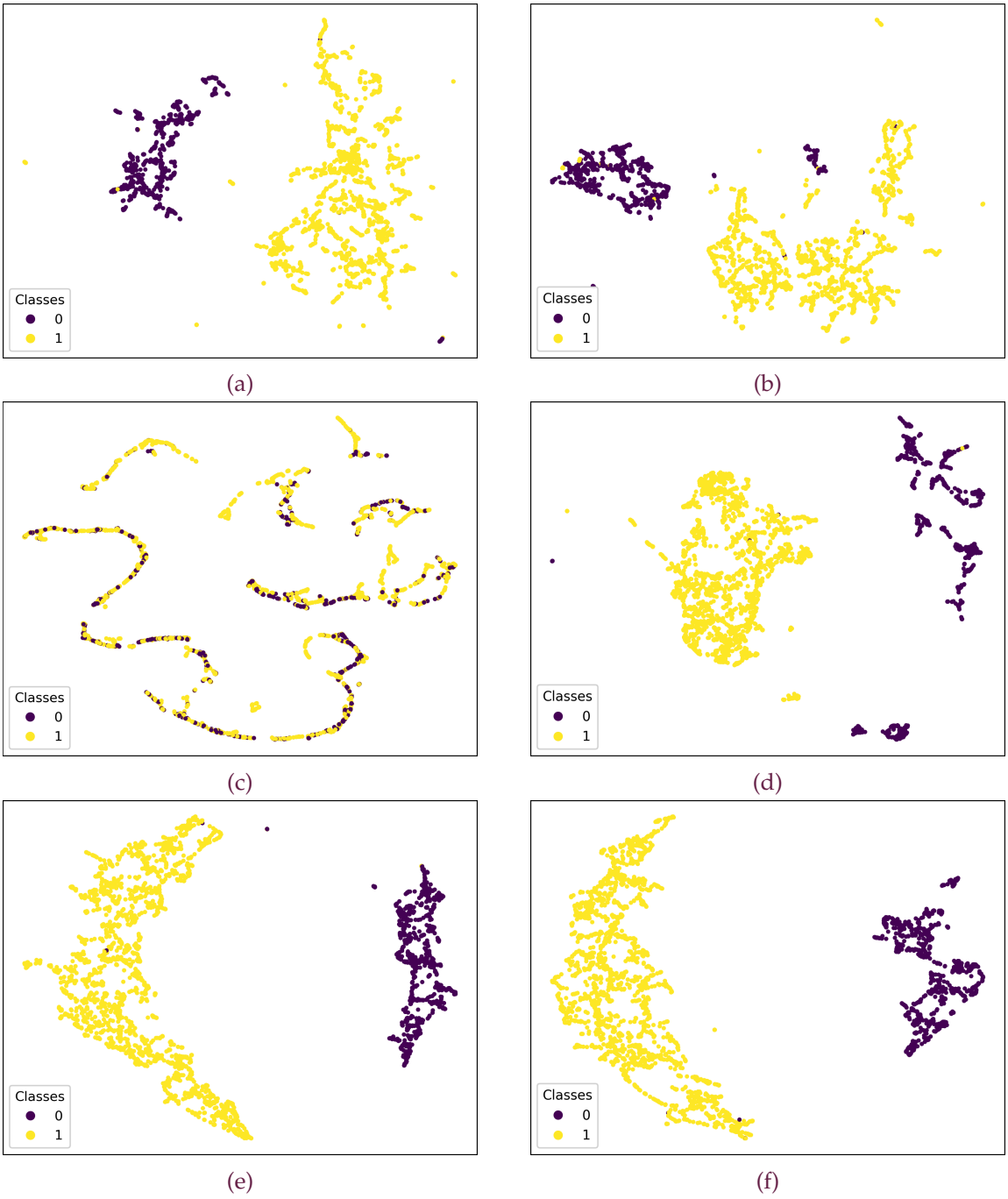


Figure 4.10: Experiment 1. Test embeddings of the MIF-GR and MIF-N models on the PTB dataset. (a) TFR encoder's latent space without regularization, (b) TFR encoder's latent space with regularization, (c) raw signal encoder's latent space without regularization, (d) raw signal encoder's latent space with regularization, (e) fused common latent space without regularization, (f) fused common latent space with regularization. For the regularized model we used $\alpha = 0.01$, $\beta = 1.0$, $\gamma = 0.0001$, and $e_{init} = 50$.

III.5.D Experiment 4: Influence of DEC regularization

The results of this experiment revealed the interest of semi-supervised DEC regularization, specially when using imbalanced datasets such as PTB.

First, the results showed that the proposed semi-supervised DEC regularization help dealing with real-life imbalanced datasets, while improving its generalization capabilities with respect to non regularized models. Indeed, in figures 4.8 we can see that DEC allows improving the classification performances, specially for the PTB dataset, which is imbalanced: the higher the importance of $\tilde{\mathcal{L}}_{DEC}$, the higher the mean classification performances and the smaller the variability. Our hypothesis is that, because of semi-supevised DEC does not use the label information (only the number of classes), it will enforce clustering regardless of the number of samples per cluster (i.e. class)[1]. Additionally, in our work (Vindas, Roux, Guépié, Almar and Delachartre, 2023a), we showed that this DEC regularization can also be used in single-feature models under the presence of label-noise, outperforming other imbalanced handling methods such as oversampling and class weights.

Second, linked to the last point, the obtained models with semi-supervised DEC regularization yield smaller variability, thus being more stable. In addition, with the good choice of hyperparameters[2], one can be able to improve the classification performances with respect to unregularized models, even for balanced datasets. This can also be justified by the more clustered fused common latent space obtained thanks to DEC, which allows reducing the number of errors because of the samples that are at the boundaries between clusters for different models (thus the variability).

At last, the results showed that, in the worst-case scenario, semi-supervised DEC regularization does not significantly degrade the classification performances of the models. This is an important point as we combine this regularization with guided training. Indeed, even if no particular hyperparameter optimization is done, the behavior of our proposed DEC regularization will remain similar under the presence of guided training, specially for imbalanced datasets such as PTB.

III.5.E Limitations

The proposed multi-feature learning methods showed to be a promising approach to enhance the classification performances of medical signal classification tasks, thanks to the exploitation of the complementarity between different representations of the same initial input. However, some limitations can be highlighted.

First, for the late-fusion approach, the main drawback is its training and optimization difficulty. Indeed, our proposed late-fusion approach is not end-to-end trainable, as it requires to first train two independent single-feature classification models, and then train the attention weights doing late fusion. The optimization and fine-tuning of each single-feature model can be time-consuming, therefore, if possible, end-to-end trainable models are preferred.

Second, our guided and regularized jointly trained intermediate fusion model (MIF-GR) overcomes the previous limitation, by proposing end-to-end trainable models, which are easier to train and optimize. However, these models have an important drawback, regarding the several introduced hyperparameters. Indeed, for the MIF-GR model, we introduced four hyperparameters (α, β, γ, and e_{init}), which have to be carefully fine-tuned to obtain optimal performances. Even though we give some guidelines on how to choose them, more extensive experiments should be carried out in order to better understand their impact, as well as to validate the generality of our method.

Third, linked to the generality of our method, we design MIF-GR to be generic, meaning that it can be used with different representations (not only raw signals and TFRs) and types of

[1] In the original DEC paper (Xie *et al.*., 2016) it was shown that, under a fully unsupervised context, DEC is relatively robust against data imbalance.

[2] We studied the influence of e_{init} in (Vindas, Roux, Guépié, Almar and Delachartre, 2023a)

Table 4.7: Combination of semi-automatic data annotation and MIF-GR: training parameters of the different models on HITS-sada. α, β, and γ correspond to the importance of $\mathcal{L}_{TFR}$, $\mathcal{L}_{RS}$, and $\tilde{\mathcal{L}}_{DEC}$, respectively. *Cat* stands for concatenation. 1D CNN-trans and 2D CNN are single-feature models, whereas MIF-GR is multifeature.

Model	Epochs	Learning rate	Weight decay	α	β	γ	e_{init}	Fusion
1D CNN-trans	75	0.04	$1e^{-5}$	-	-	-	-	-
2D CNN	100	$1e^{-4}$		-	-	-	-	-
MIF-GR	75	0.05	$1e^{-7}$	0.001	0.1	0.001	1	Cat

models (not only 2D CNNs or transformer-based models). However, in this work, we did not tested other types of representations or architectures.

Finally, we focused on medical signal classification datasets, as it is our main interest. Nevertheless, more studies can be carried out on datasets of other natures (such as environmental sound recognition, voice recognition, music classification, etc.) in order to further validate the generality of our approach.

IV Combination of semi-automatic data annotation and MIF-GR

In the previous chapter, we presented a semi-automatic data annotation method based on feature space projection and local quality metrics in order to ease the difficult and time-consuming annotation of HITS. Here, we applied this method to HITS data in order to obtain the semi-automatically labeled dataset, HITS-sada, which was partially verified by an expert. Therefore, we propose to train a MIF-GR model on this dataset, and compare its performances with single feature models 1D CNN-trans and 2D CNN.

To do this, we trained the 1D CNN-trans, 2D CNN, and MIF-GR with concatenation models on the HITS-sada dataset, using the GCE loss[1], a batch size of 32, and the training parameters of Table 4.7. Moreover, the models' architectures and hyperparameters were the same as the ones of section III, Table 4.3.

The results can be found in Table 4.8, where diverse points can be noted. First, we see that, from a classification perspective, the best performing model in all the metrics is our MIF-GR model, outperforming the 1D CNN-trans and 2D CNN models by 1.61% and 3.32% in terms of MCC respectively, while reducing the variability. These results are coherent, as our MIF-GR approach was designed to handle noisy-labeled and imbalanced datasets. There are two main reasons explaining this: guiding (with GCE) and semi-supervised DEC regularization. Indeed, the noise-tolerant loss function GCE is applied also in the intermediate classifiers C_{RS} and C_{TFR} used for guiding, allowing to obtain intermediate representations, $\mathbf{H}_{RS}$ and $\mathbf{H}_{TFR}$, relatively robust against noisy-labels, thus improving the quality and robustness of the fused common representation $\mathbf{H}_{fus}$ used for final classification. In addition to this, GCE is also applied in the final classifier C, reinforcing the robustness of the fused common representation. What is more, our semi-supervised DEC regularization makes the model more robust against noisy-labeled and imbalanced dataset. In fact, this regularization does not depend on the labels themselves, but only on the number of classes, so the noise in the labels does not have any influence on

[1]The impact of GCE with respect to CE in noisy-labeled datasets was studied in the previous chapter, therefore it will not be studied here.

Table 4.8: Combination of semi-automatic data annotation and MIF-GR: test classification performances of different single and multi-feature models on HITS-sada. The results confirm the appeal and adaptability of our guided and regularized multifeature intermediate fusion approach for real-life noisy-labeled datasets. The number of multiplications and additions (mult-adds) is given in billions (G).

Model	MCC	F1-Score	Accuracy	No. Parameters	No. mult-adds (G)
1D CNN-trans	85.74 ± 1.16	88.96 ± 0.78	91.35 ± 0.77	766 271	0.173
2D CNN	84.03 ± 1.20	86.81 ± 1.50	90.68 ± 1.12	1 681 923	1.23
MIF-GR	**87.35 ± 0.85**	**89.41 ± 0.64**	**92.59 ± 0.50**	4 833 727	1.40

it. In top of that, DEC is relatively robust against imbalanced datasets in an unsupervised (Xie *et al.*., 2016) and semi-supervised context (Vindas, Roux, Guépié, Almar and Delachartre, 2023a) because it only uses the learned hidden representations for clustering, and not the label information, partially preventing the fused common latent space from being biased by the label information.

Second, we observe that, contrary to experiment 1, the 1D CNN-trans model performs relatively well, getting closer to the MIF-GR model, as well as outperforming the 2D CNN model. On top of that, in terms of number of parameters, the 1D CNN-trans is the best model, with 2.2 and 6.3 times fewer parameters compared to the 2D CNN and MIF-GR models (a similar behavior can be observed for the number of mult-adds). This can be explained by the fact that classical transformer models are data hungry, often requiring large training datasets to generalize well. In this case, HITS-sada is around 5 times larger than HITS-small-I, which is more adapted to train transformer models from scratch without pre-training. Furthermore, this 1D CNN-trans model is the one having the smaller number of parameters and mult-adds, making it well suited for resource-limited applications. What is more, even though MIF-GR is a multifeature model whereas the 2D CNN is a single feature model, the former has only 2.9 times more parameters and requires only 1.1 times more mult-adds. Thus, a good trade-off between classification performances and resource requirements is our proposed MIF-GR model, as it is the best classification model in the used dataset, while being relatively reasonable in terms of number of parameters and number of operations.

V Interest from an industrial (Atys medical) perspective

Our work takes place within the context of the CAREMB project, where we work closely with Atys medical to improve cerebral emboli detection and classification.

As it was discussed in Chapter 2, before starting my PhD thesis, cerebral emboli detection and classification was principally done with non-portable TCD devices, and classification was mainly done between artifacts and emboli, without distinction between solid or gaseous (the latter were in some cases in the artifact class, and in other cases in the solid emboli class).

Thanks to our work, we pushed further cerebral emboli classification using portable TCD devices, and allowing the classification of HITS between artifacts, gaseous emboli, and solid emboli. In this chapter, we proposed different single and multi-feature models doing this, and which can be used for research purposes. Indeed, in collaboration with Atys medical, we developed an ADMS plugin, allowing to use the different proposed models of this chapter, to

refine the first classification proposed by ADMS. However, the choice of the model to use in an industrial context is not trivial, as there is a trade-off between inference time and classification performances.

To further investigate this, we perform inference time measurements (in ms) for the different models of experiment 2 on the HITS-small-I dataset, and the results can be found in Table 4.9. The measurements were performed without GPU, on a computer with 64 Gb of RAM memory, and a Intel(R) Core(TM) i7-6700HQ CPU. Several points can be discussed. Firstly, we can note the fastest inference time model is the 2D CNN, with 97 ms to predict the class of one single sample. This is interesting from an industrial application perspective, as patients often have thousands of HITS, so a quick analysis is needed for a practical use. With this inference time, one can process 1000 HITS in 97 s (1.5 minutes). Secondly, we can see that the MIF-GR model with concatenation achieve the second-best inference time, being able to classify 1000 HITS in 223 s (4 minutes), while having the highest classification performances. Interestingly, the MIF-GR with concatenation model which is multi-feature, is faster than the 1D CNN-transformer model, which is single-feature. This can be explained by the fact that the 1D CNN-transformer encoder of the MIF-GR with concatenation model is lighter than the single feature 1D CNN-transformer model, the former having a lower inference time. Therefore, even if the MIF-GR with concatenation model has more parameters than the 1D CNN-transformer model, it is able to process data faster, as the majority of the parameters comes from the 2D CNN encoder, which is relatively fast. Thirdly, we observe that, when we take into account the data loading, an important part of the final inference time comes from the data loading itself. For instance, for the 2D CNN classifier, the data loading represents around one third of the total inference time. Therefore, for real applications, processing 1000 HITS takes at least 149 s ($\approx$ 2.5 minutes), which is still reasonable. This result is interesting as optimizing data loading in the ADMS plugin that we developed could reduce processing time. Fourthly, we observe some interesting results. For the artifact class, the classification performances are similar for the different models, achieving up to 96% accuracy. For the gaseous and solid emboli classes, the proposed MIF-GR with concatenation model is the best one, outperforming the other models by a margin of 2.55% (gaseous emboli) and 26.15% (solid emboli) accuracy. This behavior is coherent with figure 4.9 where we observe more separable clusters in the fused common space than in the single-feature latent spaces. In addition, we can see that the worst performing class is the gaseous emboli one, where we are not able to reach performances superior to 90%. Thus, an important research axis could be focused on improving these performances. At last, we recommend using the MIF-GR with concatenation model for applications needing an accurate classification, whereas the 2D CNN can be used for time-sensitive applications.

VI Conclusion

In this chapter we proposed two main approaches for multi-feature signal classification, based on late and intermediate fusion, allowing to considerably increase the classification performances compared to other methods/models. The late fusion approach is based on the training of two independent single-feature classifiers, for which their classification outputs are combined using learnable attention weights. The intermediate fusion approaches follow a similar strategy, but this time the intermediate representations of each single-feature model are fused, and the obtained common representation regularized through semi-supervised DEC. On top of that, the classification part of each single-feature model is kept during training, to guide the training of each single-feature encoder.

Furthermore, we validate and study our proposed approach through extensive experimentation, yielding state-of-the-art results. Firstly, both late and guided-regularized intermediate

Table 4.9: Average inference times (in *ms*) and class accuracies of the different single and multi-feature models of experiments 1 and 2, for one sample. The measurements were performed using the test set of HITS-small-I, and repeated 5 times. Bold values are the best ones, and italic ones are the second-best ones. *Cat.* stands for concatenation. The 2D CNN single feature model is the fastest inference time model, whereas MIF-GR Cat. is the one with the best classification performances.

Model	Inference time	Inference with data loading	MCC	Artifact Accuracy	Gaseous Emboli Accuracy	Solid Emboli Accuracy
1D CNN-trans	297 ± 4	354 ± 4	79.17 ± 6.64	93.02 ± 7.74	86.81 ± 4.23	71.03 ± 24.68
2D CNN	**97 ± 1**	**149 ± 2**	87.09 ± 4.31	95.28 ± 4.87	87.45 ± 2.22	90.51 ± 6.98
Late Fusion	391 ± 4	446 ± 4	*87.94 ± 2.60*	96.04 ± 5.93	87.66 ± 2.48	91.28 ± 5.98
MIF-GR Cat.	*223 ± 2*	*278 ± 3*	**91.89 ± 2.64**	**96.60 ± 4.29**	**89.36 ± 1.00**	**97.18 ± 2.42**

fusion models achieve excellent and state-of-the-art performances on the tested datasets. Secondly, for the intermediate fusion models, we showed that guiding and regularization are crucial as they allow more discriminative intermediate representations, as well as a more clustered fused common latent space. Particularly, if guiding and regularization are not applied, the intermediate fusion approaches can yield worse performances than some single-feature models. Thirdly, our model is relatively robust against semi-automatically labeled, noisy-labels and imbalanced datasets, outperforming single feature models under the same training conditions with a noise-tolerant loss function such as GCE. Finally, even if it was not directly tested during experimentation, our method is intended to be generic and usable in other contexts, where the datasets, the representations, or the used architectures are not the same as the ones used in our experiments.

In the following chapter, we are going to work on model compression methods, allowing to reduce the memory, computation, and energy requirements of the different models developed in the current and the previous chapter.

5 | Model Compression

In the previous chapter (Chapter 4) we developed multi-feature models to push further the HITS classification performances. These models can be trained on noisy-labeled datasets as the ones obtained with our semi-automatic data annotation method presented in Chapter 3, further improving their classification performances. However, these models are often memory and energy greedy, which makes difficult their use in practical situations. To overcome this difficulty, model compression techniques such as pruning (Hoefler *et al.*., 2022) or quantization (Liang, Glossner, Wang, Shi and Zhang, 2021; Gholami *et al.*., 2022) can be used. As seen in Chapter 2, some works have proposed to (directly or indirectly) combine pruning and quantization to take advantage of both techniques, while keeping reasonable classification performances (Han *et al.*., 2016; Park and Yoo, 2020; Tung and Mori, 2020; Rastegari *et al.*., 2016; Zhu *et al.*., 2017). However, to our knowledge, not many works have tried to directly take into account pruning during extreme quantization (using less than 4 bits to encode the quantized weights).

In this chapter, we propose a new quantization heuristic to directly take into account asymmetric pruning when doing trained ternary quantization (TTQ). This allows us to compress the models developed in the previous chapters to reduce their memory requirements and energy consumption.

Contents

I Motivation

I.1 Limitations of previous methods

Several methods have done model compression on different types of deep learning models as seen in Chapter 2. Here, we focus on pruning and quantization methods as they can be combined relatively easily, and allow great compression and energy consumption gains, while keeping reasonable classification performances for practical use.

The first limitation of previous methods is that, to our knowledge, they often do not directly take into account pruning while doing extreme quantization (such as trained binary or ternary quantization). Indeed, in trained extreme quantization, some heuristics are defined to decide which quantized value will be associated to each weight, but pruning is not done unless 0 is part of the quantized values (as for TTQ). If 0 is part of these values, then pruning is implicitly done through the quantization heuristic, but often it is difficult to control it without modification of the latter.

Another important limitation, related to the last one, is the difficulty of controlling the trade-off between classification and compression performances. Indeed, even though the sparsity obtained through pruning can have a regularization effect on the trained models, very high sparsity rates (i.e. aggressive pruning) can bring important losses of information, reducing considerably the classification performances of the models. As the sparsity rate is difficult to control in current extreme quantization methods (because of the proposed heuristics), controlling this trade-off is not an evident task.

Furthermore, the symmetry of the quantization heuristics and the quantized weights can also be considered as a limitation. Indeed, TTQ (Zhu *et al.*, 2017) highlighted this for the scaling factors used by ternary weight networks (Li and Liu, 2016) which were symmetric. They proposed to improve the quantization procedure by using asymmetric learnable scaling factors. However, regarding the quantization heuristic, positive and negative weights are considered as having the same importance, as quantization is done based using symmetric thresholds. The same rationale can be used to improve the quantization heuristic and process the negative and positive weights differently.

Finally, even though different compression metrics are used in the previous works to measure the trade-off between model compression and classification performances, few works take into account the sparsity, or use energy consumption based metrics. This is important to take into account as sparsity can reduce energy consumption, by limiting the number of operations to perform, as well as the data transfers to memory. Therefore, to accurately compare compressed models, it is necessary to take into account sparsity in the compression rates, and energy consumption metrics. Likewise, hardware-independent energy consumption metrics should be used to ease this comparison.

I.2 Objectives and contributions

We suggest solving the previous limitations with a new quantization heuristic based on asymmetric pruning done using the weights' statistics. We parametrize our quantization heuristic with two hyperparameters, allowing to better control the trade-off between compression and classification performances. This allows us to increase the sparsity rates, while keeping the quantized weights in reduced precision (2 bits), and reasonable classification performances. Our proposed approach outperforms TTQ (Zhu *et al.*, 2017) in terms of trade-off between compression, energy consumption, and classification performances. Our main contributions can be summarized as follows:

- A new extreme quantization heuristics, based on the weights' statistics of the layers of the model to quantize.
- Asymmetric pruning before ternarization, giving a better trade-off between compression and classification performances.
- Asymmetric parametrization of the sparsity rate of the layers to quantize using two hyperparameters, allowing to control the abovementioned trade-off.
- A new hardware-independent energy consumption metric, taking into account the sparsity of the models' weights, the number of bits needed to encode them, the number of mult-adds performed by the model, as well as the number of data transfers to memory.

II Proposed method

In this section, we are going to detail the ternarization heuristic that we propose in order to incorporate asymmetric pruning during the quantization procedure, as well as the compression and energy consumption metrics that we use to compare the different models. Our approach can be used for different types of models, including the ones developed in the previous chapter.

In this chapter, some notations may differ with respect to previous chapters. This is to facilitate the explanation of the proposed quantization method, and the comparison with similar works.

II.1 General overview and assumptions

Let us suppose that we have a pre-trained twice-differentiable full precision (FP) model $\mathcal{M}_{FP}$ which has reached a local minimum, and composed of p layers $L_1, ..., L_p$. Without loss of generality, we suppose that $\mathcal{M}_{FP}$ can be written under the form $\mathcal{M}_{FP} = L_1 \circ L_2 \circ ... \circ L_p$, where for all $i \in [1, p]$, $L_i \in \mathbf{R}^{n_i \times m_i}$ (n_i and m_i are the dimensions of layer L_i). Our aim is to prune and ternarize $\mathcal{M}_{FP}$ using a quantization method Q, to obtain a compressed and sparse model $\mathcal{M}_C = \tilde{L}_1 \circ \tilde{L}_2 \circ ... \circ \tilde{L}_p$, where for all $i \in [1, p]$, $\tilde{L}_i$ is the i^{th} layer of the compressed model, defined as:

$$\tilde{L}_i = \begin{cases} Q(L_i) \text{ if } i \in I_Q \\ L_i \text{ else} \end{cases}$$

where I_Q is the set of indices of the layers that are going to be quantized, and Q is applied element-wise to L_i.

Moreover, our approach is composed of two main components (see figure 5.1): quantization layer selection and asymmetric TTQ (aTTQ).

- **Quantization layer selection:** To avoid important classification performances drop, we select the layers that are going to be quantized, I_Q, using the Hessian based metric introduced in Chapter 2 (equation 2.20).
- **Asymmetric TTQ (aTTQ):** Done through a new quantization heuristic where the pruning thresholds are asymmetric, and computed using the mean and standard deviation of the weights of each layer to quantize, parametrized by two hyperparameters, t_{min} and t_{max}. A general overview of this step is presented in figure 5.2.

Furthermore, our approach relies on several assumptions. First, in order to be able to select the layer to quantize using the Hessian based metric of (Dong *et al.*, 2020), we need to have a twice-differentiable pre-trained model which has reached a local minimum (in addition to the

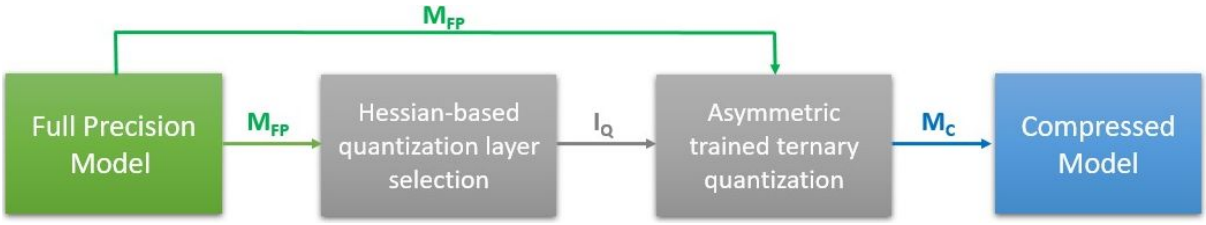


Figure 5.1: Main components of our proposed extreme quantization approach: layer quantization selection and asymmetric TTQ (aTTQ). The layers to be quantized are obtained using the Hessian-based metric of (Dong *et al.*., 2020). Asymmetric trained ternary quantization is based on a new quantization heuristic where the quantization thresholds are computed based on the weights statistics of a given layer.

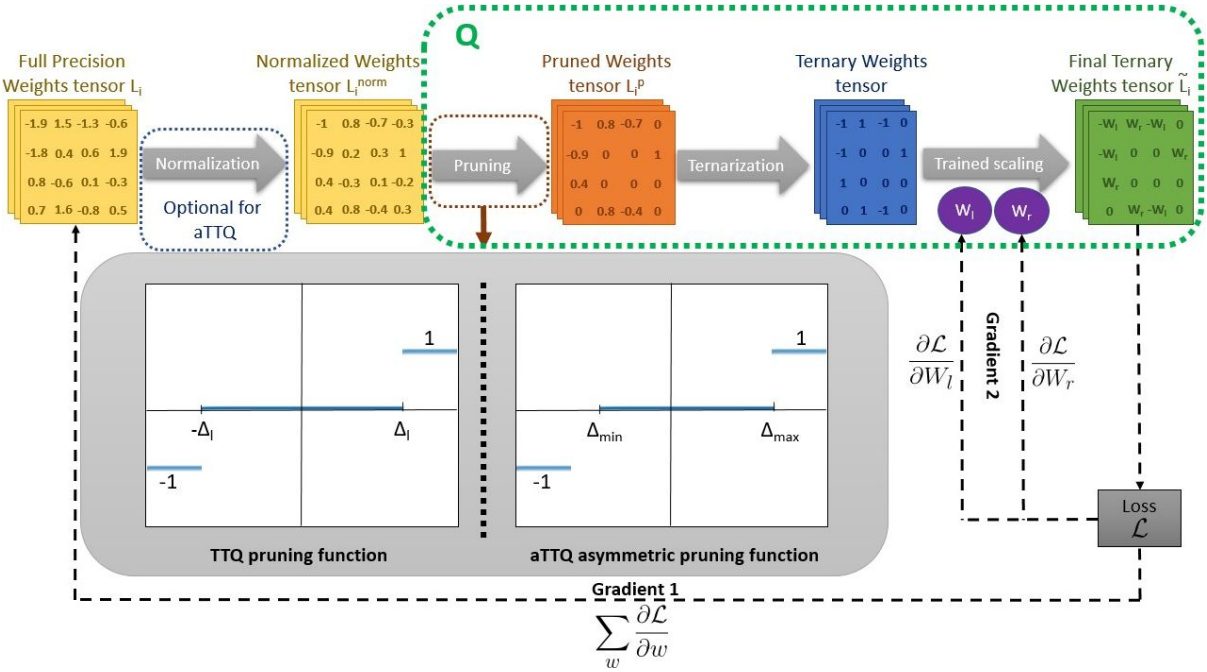


Figure 5.2: Proposed asymmetric TTQ (aTTQ) method. The main difference with respect to TTQ lies on the asymmetric pruning mechanism, which is done using two asymmetric thresholds, Δ_{min} and Δ_{max} instead of one symmetric threshold $\Delta_{min} = -\Delta_{max} = -\Delta_L$. Additionally, these thresholds are computed using the mean and standard deviation of the weights to quantize, instead of the max. The normalization step, always used in TTQ, is optional in our approach.

other two assumptions introduced in Chapter 2). Second, as (Zhu *et al.*., 2017) does it for the learnable scaling factors, we make the assumption that asymmetry in the pruning thresholds before quantization can be beneficial for the final compression and classification performances of the quantized model. The rationale behind this is that it enlarges the quantized model search space used during training, which allows searching for models having a better trade-off between compression, energy consumption, and classification performances[1]. Third, to simplify the energy consumption computation, we make three assumptions: (1) only the nonzero parameters are counted to determine the number of mult-adds, (2) multiplications and additions have the same energy cost, the one of a 32 floating-point multiplication ($3.7pJ$ according to (Horowitz, 2014)) and (3) only the nonzero weights are transferred to RAM memory, and all transfers are done by blocks of 32 bits (4 bytes). The main reason we make assumption 2 is that multiplications are more expensive than additions, and separating multiplications from additions in current versions of *torch* and *torchinfo* is not straightforward.

II.2 Asymmetric weights statistics based pruning

We propose a novel asymmetric quantization heuristic for trained ternary quantization, based on the weights' statistics of the layers to quantize, allowing a better control of the sparsity rates (see figure 5.2). Contrary to TTQ, the quantization thresholds are not the same for the positive and the negative weights (asymmetry), and they do not depend on the maximum absolute value of the weights' tensors, but rather on their mean and standard deviation:

$$\forall i \in I_Q, \forall k \in [1, n_i], j \in [1, m_j], Q(L_i^{k,j}) = \begin{cases} W_i^l \text{ if } L_i^{k,j} < \Delta_{min}^i \\ 0 \text{ if } L_i^{k,j} \in [\Delta_{min}^i, \Delta_{max}^i] \\ W_i^r \text{ if } L_i^{k,j} > \Delta_{max}^i \end{cases} \tag{5.1}$$

where $L_i^{k,j} \in \mathbb{R}$ is the weight of layer L_i at row k and column j, $W_i^l, W_i^r \in \mathbb{R}$ are learnable scaling parameters (the superscript l stands for "left" as the r stands for "right") for layer L_i, $\Delta_{min}^i = \mu_i + t_{min} \times \sigma_{min}$ and $\Delta_{max}^i = \mu_i + t_{max} \times \sigma_i$ are the quantization thresholds for layer L_i, μ_i and σ_i are the mean and standard deviation of the weights of layer L_i, and t_{min} and t_{max} are two hyperparameters allowing to control the sparsity of the quantized weights (same for all the layers), and following the constraint $t_{min} \leq t_{max}$. This constraint is necessary because, if $t_{min} > t_{max}$, then $\Delta_{min}^i > \Delta_{max}^i$, thus $\forall i \in I_Q, \forall k \in [1, n_i], j \in [1, m_j], Q(L_i^{k,j}) \neq 0$, avoiding any pruning. On the contrary, the lower t_{min} with respect to t_{max}, the larger the range $[\Delta_{min}^i, \Delta_{max}^i]$, resulting in sparser models.

Furthermore, as for TTQ, we used the straight forward estimator to compute the gradients of the loss $\mathcal{L}$ to optimize for the layers to quantize:

$$\frac{\partial \mathcal{L}}{\partial L_i^{k,j}} = \begin{cases} W_l \times \frac{\partial L}{\partial \tilde{L}_i^{k,j}} \text{ if } L_i^{k,j} < \Delta_{min}^i \\ 0 \text{ if } L_i^{k,j} \in [\Delta_{min}^i, \Delta_{max}^i] \\ W_r \times \frac{\partial L}{\partial \tilde{L}_i^{k,j}} \text{ if } L_i^{k,j} > \Delta_{max}^i \end{cases} \tag{5.2}$$

where $L_i^{k,j}$ and $\tilde{L}_i^{k,j}$ are the full-precision and ternarized weights of a given layer to quantize. Thanks to this, aTTQ can be optimized through gradient descent.

[1]Note that the fact that we have two asymmetric thresholds do not have an influence during inference as they are only used when quantizing the model during training.

II.3 Layer selection

As we are doing extreme quantization (2 bits to store the ternary weights plus 32 bits for each of the two the full precision scaling factors, per quantized layer), we need to select the layers that can be quantized without considerably degrading the classification performances of the model. To do this, we use the Hessian-based metric introduced in (Dong *et al.*., 2020), allowing to quantify the curvature of the loss landscape, and therefore the less sensitive layers for quantization. The logic behind this metric is that, layers with irregular loss landscapes (i.e. high values of the trace of the Hessian matrix) are more sensitive to quantization, as they can easily get out from the reached local minimum during pre-training of the full-precision model. Therefore, we consider that a layer can be quantized if the Hessian-based metric $\mathcal{H}$, is smaller than a certain threshold τ:

$$I_Q(\mathcal{M}) = \{i / i \in [1, p] \text{ and } \mathcal{H}(L_i) \leq \tau \text{ and } L_i \in \mathcal{M}\} \tag{5.3}$$

The threshold τ is a hyperparameter to tune as it depends on the application and the model. It is important to note that, this threshold can be defined globally (for all the model), or locally, for blocks of a given model. Note that one can also define this threshold based on the number of layers that we would like to quantize (for instance, choose τ to quantize the 10 layers with the smallest Hessian-based metric, i.e. such that $|I_Q(\mathcal{M})| = 10$).

II.4 Model compression metrics

We propose to compare the quantized and full precision models using several metrics evaluating different aspects: sparsity, compression, and energy consumption. We suppose that we have the four following functions:

- *nbits*: counts the number of bits required to store the (nonzero) weights of the model according to the COO encoding of sparse tensors[1].

- *nqw*: for a full-precision model $\mathcal{M}_{FP}$, it counts the total number of weights that can be quantized. It can be defined as follows:

$$nqw(\mathcal{M}_{FP}) = |\{(i, j, k) / i \in I_Q(\mathcal{M}_{FP}), j \in [1, n_i], k \in [1, m_i]\}|$$

- *nzqw*: for a quantized model, it counts the number of quantized weights having a value of 0[2]. It can be defined as follows:

$$nzqw(\mathcal{M}_C) = |\{(i, j, k) / i \in I_Q, j \in [1, n_i], k \in [1, m_i], \tilde{L}_i^{j,k} = 0\}|$$

- *nnzw*: for a given model $\mathcal{M}$, it counts the number of nonzero weights. It can be defined as follows:

$$nnzw(\mathcal{M}) = |\{(i, j, k) / i \in [1, p], j \in [1, n_i], k \in [1, m_i], L_i^{j,k} \neq 0\}|$$

[1]The coordinate list or COO encoding of sparse tensors consists in a list of tuples indicating the position of the nonzero values in the tensor, as well as its value.

[2]If a weight was already 0 in a layer that is not in I_Q, it is counted in *nzqw*.

II.4.A Sparsity

Different quantization methods can do implicit or explicit pruning during their procedure, which can have an impact on the memory requirements of the model during inference. Therefore, we propose to quantify the sparsity obtained by a quantization/pruning method, by measuring the sparsity rate over the quantized weights, $SRQW$, and defined as follows:

$$SRQW(\mathcal{M}_{FP}, \mathcal{M}_C) = \frac{nzqw(\mathcal{M}_C)}{nqw(\mathcal{M}_{FP})} \tag{5.4}$$

where higher values of $SRQW$ indicates sparser quantized models.

II.4.B Compression

The compression performance of a quantization method Q can be determined by measuring the gain in terms of number of bits required to store the compressed model $\mathcal{M}_C = Q(\mathcal{M}_{FP})$ with respect to the full precision model $\mathcal{M}_{FP}$. Thus, we denote as CR the compression rate, and CR_G the compression rate gain, defined as follows:

$$CR(\mathcal{M}_{FP}, \mathcal{M}_C) = \frac{nbits(\mathcal{M}_C)}{nbits(\mathcal{M}_{FP})} \tag{5.5}$$

$$CR_G(\mathcal{M}_{FP}, \mathcal{M}_C) = 1 - CR(\mathcal{M}_{FP}, \mathcal{M}_C) \tag{5.6}$$

where smaller values of CR and higher values of CR_G indicate more compressed models. It is important to note that these compression metrics take into account the sparsity of the weights' tensors, as they are stored using the sparse COO encoding.

Moreover, as not all the layers of the models are quantized, we can distinguish two types of compression rates gains, one for the whole mode (quantized and unquantized layers), denoted as CR_G^T, and one focusing only on the quantized layers, denoted as CR_G^Q.

II.5 Energy consumption

Energy consumption during inference of neural network model can come from different sources, but we focus on the two most expensive ones: the number of mult-adds and the number of data transfers to RAM memory. Indeed, the idea is to propose an energy consumption metric hardware-independent, which does not require manual measurements for two main reasons. First, it is difficult to make hardware-dependent manual measurements to fairly compare different models as they are easily disturbed by other parallel tasks (operative system tasks, browser, execution of other apps/programs, etc.). Second, it is possible to design specific hardware taking advantage of the architecture and characteristics of a given model (for instance the quantized and sparse weights), which can significantly reduce energy consumption (which raises fairness comparison issues).

Therefore, we propose a new energy consumption metric, based on the order of magnitudes of mult-adds given in (Horowitz, 2014), and the ones of RAM data transfers given in (Molka, Hackenberg, Schöne and Müller, 2010). With this metric, one can obtain orders of magnitudes (in Joules) of the energy consumption of a model, taking into account its sparsity and quantized weights. Our energy consumption metric relies on three assumptions:

1. Only the nonzero weights are taken into account for the mult-adds and data transfers.

2. Multiplications and additions are considered to have the same energy consumption, the one of a 32 floating-point multiplication[1], corresponding to 3.7 pJ according to Horowitz (2014).

3. For the transfers of weights to RAM memory, we only take into account the nonzero weights, and they are transferred by blocks of 32 bits (4 bytes).

We denote as EC_{MA}, EC_{DT}, and EC_T, the energy cost of mult-adds, data transfers, and total, respectively. It is important to note that, because of the used assumptions, our proposed energy consumption metrics are upper bounds of the true energy consumption of the models. Hereafter, we are going to detail how each term of the final energy consumption metric is computed.

Number of operations: First, we start by computing the energy cost of mult-adds. To do this, we star by counting the number N_{MA} of multiplications and additions needed to obtain the output of the model, for a given input. Then, using the order of magnitudes of (Horowitz, 2014), we can compute the energy cost of a model $\mathcal{M}$ in J as follows:

$$EC_{MA}(\mathcal{M}) = N_{MA} \times 3.7 \times 10^{-12} \tag{5.7}$$

where smaller values of EC_{MA} indicates smaller energy consumption with respect to the multiplications and additions.

Number of data transfers: The number of data transfers depends on the number of bits needed to store each weight (which can vary from one layer to another one), as well as on the number of nonzero weights. Moreover, for a layer L_i with $i \in [1, p]$, we denote as N^i_{SF} the number of quantization scaling factors used in that layer (0 for a full-precision layer, 2 for a ternarized one), and B_i the number of bits necessary to encode a weight of the layer (32 for a full-precision layer, and 2 for a ternarized one). Thus, the energy cost of data transfers in J for a model $\mathcal{M}$ can be defined as follows:

$$EC_{DT}(\mathcal{M}) = 10^{-9} \times \sum_{i=1}^{p} (\lceil \frac{nnzw(L_i) \times B_i}{32} \rceil + N^i_{SF}) \tag{5.8}$$

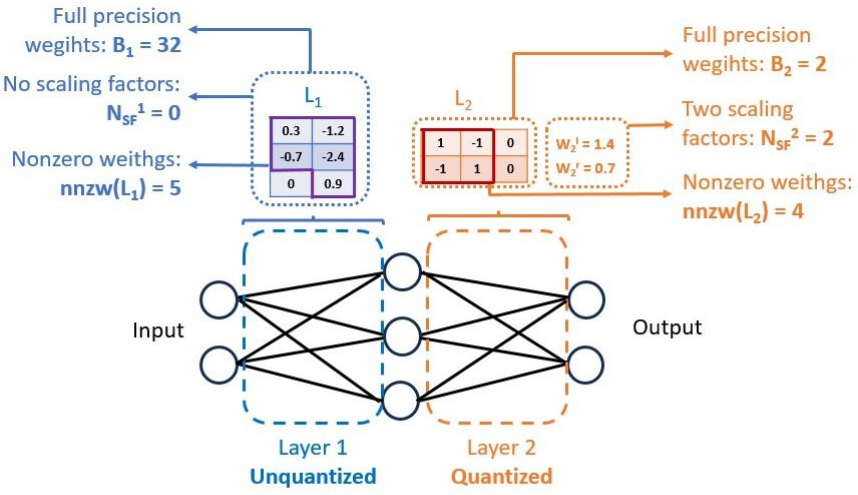
where smaller values of EC_{DT} indicates smaller energy consumption with respect to the data transfers. An example of application can be found in figure 5.3. In simple terms, this metric will count the number of 32 bits chunks in which we can distribute all the weights of the model[2]. The first term of the sum corresponds to the number of 32 bit chunks needed to encode the weights of a given layer L_i, based on the number of nonzero weights $nnzw(L_i)$ and the number of bits B_i needed to encode them. The second term in the sum corresponds to the number of 32 bit scaling factors N^i_{SF} needed in layer L_i.

Final metric: The final energy consumption metric, EC_T, measures the total energy cost needed to perform the inference of one model, and it is the sum of the two abovementioned terms:

$$EC_T(\mathcal{M}) = EC_{MA}(\mathcal{M}) + EC_{DT}(\mathcal{M}) \tag{5.9}$$

[1]This is the worst case scenario as multiplications are more expensive than additions. We do this because separating multiplications and additions in the current versions of *torch* and *torchinfo* is not straightforward.

[2]It is important to note that in this work we make the assumption that data transfers are done by chunks of 32 bits. In some particular hardwares, the data transfers can occur at other bit widths, in which case it is necessary to adapt the proposed metric with the correct bit width value.



$$EC_{DT}(\mathcal{M}) = 10^{-9} \times [(\lceil \frac{nnzw(L_1) \times B_1}{32} \rceil + N_{SF}^1) + (\lceil \frac{nnzw(L_2) \times B_2}{32} \rceil + N_{SF}^2)]$$
$$= 10^{-9} \times [(\lceil \frac{5 \times 32}{32} \rceil + 0) + (\lceil \frac{4 \times 2}{32} \rceil + 2)]$$
$$= 10^{-9} \times [(5+0) + (1+2)] = 8 \times 10^{-9} \ J$$

Figure 5.3: Example of computation of the energy consumption of data transfers for a vanilla model. We have a two layer model, with the first layer (blue) kept in full precision, and the second layer (orange) quantized. As the first layer is in full precision and it is not quantized, there are no scaling factors ($N_{SF}^1 = 0$), and the number of bits necessary to encode the weights is $B_1 = 32$. As the second layer is quantized, there are two scaling factors ($N_{SF}^2 = 2$), and the number of bits necessary to encode the weights is $B_2 = 2$.

where smaller values of EC_T indicates smaller energy consumption with respect to the multiplications, additions, and data transfers.

Finally, as we do it for the compression rates, we measure the energy consumption gain, EC_G^T of a quantized and sparse model $\mathcal{M}_C = Q(\mathcal{M}_{FP})$ with respect to its full-precision counterpart, $\mathcal{M}_{FP}$ as follows:

$$EC_G^T(\mathcal{M}_{FP}, \mathcal{M}_C) = \frac{|EC_T(\mathcal{M}_{FP}) - EC_T(\mathcal{M}_C)|}{EC_T(\mathcal{M}_{FP})} \tag{5.10}$$

where higher values of EC_G^T indicates smaller global energy consumption with respect to the full precision model.

III Method evaluation

We conduct three experiments to validate our proposed aTTQ approach. We studied: (1) the advantage of aTTQ over TTQ, (2) the influence of t_{min} and t_{max}, and (3) the influence of weight normalization.

III.1 Datasets

We used three different datasets, among which two are medical signals datasets: HITS (TCD), ESR (EEG), and MNIST.

HITS-small: It corresponds to the same HITS-small dataset as the one introduced in Chapter 3, with the same pre-processing configuration. As a reminder, the dataset is composed of 1 545 labeled samples distributed in four classes, but only the first three are used: 569 solid emboli, 569 gaseous emboli, 403 artifacts, and 4 unknown samples. Moreover, the dataset is split into two subsets subject-wise, where 63% of the samples are used for training, and 37% of them for testing.

ESR: It corresponds to the same ESR dataset as the one introduced in Chapter 3, with the same pre-processing configuration. As a reminder, the dataset is composed of 11 500 samples, distributed in 5 balanced classes. However, as most of the works, we focus on binary classification, where the first class contains the seizure activity signals (2300 samples), and the second class the non seizure activity signals (9200 samples). Moreover, as in the previous chapter, 90% of the samples are used for training, and 10% for testing.

MNIST: We used a subset of MNIST (28×28 images), composed of 10% of the training samples (random sampling), and all the available testing samples. Therefore, the dataset is composed of 10 000 training samples, and 10 000 testing samples.

III.2 Baselines

For all the datasets, we considered two baselines:

- **Full-precision** models: corresponding to the reference and allowing to measure by how much the compression, energy, and classification performances are degraded (or improved) by aTTQ.
- TTQ **quantized** models: allowing to compare our proposed aTTQ approach with another state-of-the-art ternary quantization method. The selection of the layers to quantize is the same as the one used for aTTQ.

III.3 Evaluation metrics and strategy

We used several evaluation metrics to compare the different models between each other. Our main interest is to compare the different methods from a compression and energy consumption perspective, thus, we used all the metrics introduced in subsection II.4 of this chapter.

Furthermore, we also want to compare the different models from the classification perspective, as it is important to keep good classification performances after compression, to keep the developed models usable in practice. Therefore, we use the MCC and the difference of MCC, ΔMCC, between the full-precision model and its quantized counterpart to evaluate this point:

$$\Delta MCC = MCC(\mathcal{M}_C) - MCC(\mathcal{M}_{FP})$$

III.4 Experimental setup and results

Following the results of the previous chapter (Chapter 4), we start by focusing on the compression of single feature models as they have good classification performances, with fewer parameters than multi-feature models[1]. Therefore, we used the 2D CNN and 1D CNN-transformer (1D CNN-trans) models of the previous chapter (with the same architecture parameters) for the

[1]In addition, the late fusion and intermediate fusion multi-feature models strongly depend on the single feature models, so compressing the single feature models can help to compress multi-feature models.

Table 5.1: Architecture parameters of the models to be compressed, based on the dataset used. We refer the reader to Chapter 4 for the definition of the model parameters.

Dataset	Model	n_{head}	d_{hid}	n_{layers}	$p_{dropout}$	n_{proj}	d_{raw}	n_{conv}	$n_{filters}$	d_{com}	Pool
HITS-small	1D CNN-trans	8	64	8	0.1	10	128	2	-	-	-
	2D CNN	-	-	-	0.2	-	-	-	64	-	Max
ESR	1D CNN-trans	4	8	4	0.3	4	64	2	-	-	-
	2D CNN	-	-	-	0.2	-	-	-	64	-	Max

Table 5.2: Training parameters of different models (for model compression). The last column corresponds to the percentage of weights of the model that are going to be quantized.

Dataset	Model	Quant. method	t_{min}	t_{max}	Learning rate	Epochs	No. params.	% weights to quantize
HITS	2D CNN	FP	-	-	10^{-3}	50	1 548 288	-
		TTQ	-	-	3×10^{-3}	50		92.05
		aTTQ	–4	0	10^{-4}	150		
	1D CNN-trans.	FP	-	-	7×10^{-2}	150	766 271	-
		TTQ	-	-	10^{-4}	50		14.97
		aTTQ	–2	1.5	5×10^{-5}	100		
ESR	2D CNN	FP	-	-	10^{-3}	100	1 555 842	-
		TTQ	-	-	10^{-3}	50		99.51
		aTTQ	–3	1	10^{-3}	200		
	1D CNN-trans.	FP	-	-	3×10^{-1}	100	109 942	-
		TTQ	-	-	10^{-3}	100		24.22
		aTTQ	–2	1	5×10^{-4}	100		
MNIST	2D MNIST CNN	FP	-	-	10^{-3}	70	9 840	-
		TTQ	-	-	10^{-4}	200		53.35
		aTTQ	–1	0.5	10^{-3}	200		

HITS and ESR datasets, whereas for the MNIST dataset, we used the vanilla 2D CNN MNIST from Chapter 3.

Furthermore, for all the experiments, the models' architectures hyperparameters and training parameters were fixed, and can be found in tables 5.1 and 5.2.

What is more, for all the experiments, we start by training a full precision model on the given dataset, and then this model is quantized (with aTTQ or TTQ) on the same dataset to obtain the quantized model. The layers selected for quantization (without the biases) depend on the model:

- 2D CNN (HITS and ESR): all the convolutional layers except the first one.
- 1D CNN-trans: second convolutional layer, and second linear layer of all the encoder layers of the transformer encoder.
- 2D CNN MNIST: all the convolutional layers.

We can note that the layers selected for quantization are the same for a given model on different datasets. There are two main reasons for this. First, globally, for a fixed model and

from one dataset to another one, the Hessian-based metric $\mathcal{H}$ tends to give the smaller values for the selected layers in a given block[1] (see appendix 3). Second, from appendix 3, we observe that the selected layers to quantize are not always the ones with the smallest value inside the model or inside a block. However, to simplify the experimental setup, we fixed the same layers to quantize per block, by selecting the ones having the smallest $\mathcal{H}$ value, the highest number of times (see appendix 3).

III.4.A Experiment 1: Comparison with respect to TTQ

Objective: The objective of this experiment is to compare our proposed aTTQ approach with another extreme quantization approach, TTQ.

Experimental setup: We trained the abovementioned models with the indicated parameters, and then we quantized them using aTTQ and TTQ. Additionally, TTQ was trained using weight normalization for the layers to quantize (as its original version (Zhu *et al.*., 2017)), whereas for aTTQ we did not use weight normalization.

Results: Results can be found in table 5.3 and figure 5.4 (the effect of normalization will be discussed in experiment 3). Several points can be highlighted.

First, we observe that, from a compression perspective, all the aTTQ quantized models outperform the TTQ ones by a large margin, for all the proposed metrics. If we focus on the sparsity rates of the quantized weights (figure 5.4), $SRQW$, aTTQ outperforms TTQ by a margin going from 2.7% to 86.8%. This margin is particularly important for the 1D CNN-transformer based models, where aTTQ improves the sparsity rate over a factor of 13 compared with TTQ, passing from 6.75% to 93.58%. A similar behavior is observed for CR_G^Q and CR_G^T. However, for the global compression rates (of the whole model), CR_G^T, the difference between aTTQ and TTQ is less striking, which will be discussed later on.

Second, aTTQ also outperforms TTQ in terms of energy consumption by important margins, going from 2.16% to 20.62% of energy consumption gain EC_G^T. More interestingly, for the HITS models, this margin is particularly important (larger signals and TFRs compared to the other datasets), improving by a factor of 1.88 and 4.06 for the 2D CNN and 1D CNN-trans, respectively.

Thirdly, in terms of classification performances, both aTTQ and TTQ achieve similar performances with respect to the full precision model. Indeed, the maximum observed MCC drop, ΔMCC, is obtained for the 2D CNN on the HITS datasets, where aTTQ drops the MCC with respect to the FP model of −3.70%, whereas TTQ does it by −3.02%. However, for the other models and datasets, this drop is smaller, going from −0.77 (performance drop) to +2.19 (performance gain). Indeed, in some cases, aTTQ and TTQ are able to increase the classification performances with respect to the FP model, particularly on the ESR dataset where they do it by +1.01% (aTTQ) and +1.92% (TTQ) respectively for the 1D CNN-transformer model.

At last, if we compare aTTQ and TTQ between each other from a classification performances perspective, we note that the great compression and energy consumption performances of aTTQ come at the expense of classification results. Indeed, TTQ weakly outperforms aTTQ on the majority of the datasets by a margin going from 0.68% to 2.59% in terms of MCC. However, this does not hold for the MNIST dataset, where aTTQ outperforms TTQ by a margin of 1.52%.

[1] A block inside a model can be considered as a set of layers with a given purpose. For instance, for the 2D CNN model, we have two blocks, the convolutional feature extraction one (sequence of convolutions), and the classifier (composed of FC layers). For the 1D CNN-transformer model, each transformer encoder layer can be considered as one block.

Table 5.3: Experiment 1 results (in %): comparison between aTTQ and TTQ. FP corresponds to the full-precision model (no quantization). ΔMCC corresponds to the difference between the MCC of the full precision model and the MCC of the quantized model. CR_G^T, and CR_G^Q evaluate the compression performance of each quantization method and were introduced in II.4. EC_G^T evaluates the energy consumption performance, and it was introduced in II.4.

Dataset	Model	Quant. method	Norm.	$CR_G^T \uparrow$	$CR_G^Q \uparrow$	$EC_G^T \uparrow$	MCC ↑	Δ MCC ↑
HITS	2D CNN	FP	No	-	-	-	89.84 ± 3.09	-
		TTQ	Yes	24.96 ± 2.25	27.12 ± 2.44	23.42 ± 1.30	**86.82 ± 2.29**	**−3.02**
		aTTQ		**42.98 ± 0.23**	**46.69 ± 0.25**	**44.04 ± 0.19**	86.14 ± 3.37	−3.70
	1D CNN-trans.	FP	No	-	-	-	82.64 ± 1.77	-
		TTQ	Yes	0.14 ± 0.04	0.91 ± 0.27	1.88 ± 0.03	**83.22 ± 2.36**	**+0.58**
		aTTQ	No	**13.94 ± 0.02**	**93.17 ± 0.16**	**7.64 ± 0.11**	81.66 ± 4.17	−0.98
ESR	2D CNN	FP	No	-	-	-	92.81 ± 3.53	-
		TTQ	Yes	85.61 ± 1.37	86.03 ± 1.37	76.45 ± 1.13	**95.00 ± 1.11**	**+2.19**
		aTTQ	No	**88.48 ± 0.44**	**88.91 ± 0.45**	**84.49 ± 0.33**	92.41 ± 2.22	−0.40
	1D CNN-trans.	FP	No	-	-	-	94.33 ± 1.51	-
		TTQ	Yes	11.40 ± 2.61	47.07 ± 10.79	3.21 ± 0.66	**96.25 ± 0.79**	**+1.92**
		aTTQ	No	**21.02 ± 0.15**	**86.78 ± 0.63**	**5.37 ± 0.04**	95.34 ± 0.79	+1.01
MNIST	2D MNIST CNN	FP	No	-	-	-	94.39 ± 0.46	-
		TTQ	Yes	13.86 ± 2.33	25.97 ± 4.37	2.58 ± 0.35	92.09 ± 0.89	−2.30
		aTTQ	No	**28.98 ± 1.26**	**54.32 ± 2.36**	**4.97 ± 0.22**	**93.62 ± 0.96**	**−0.77**

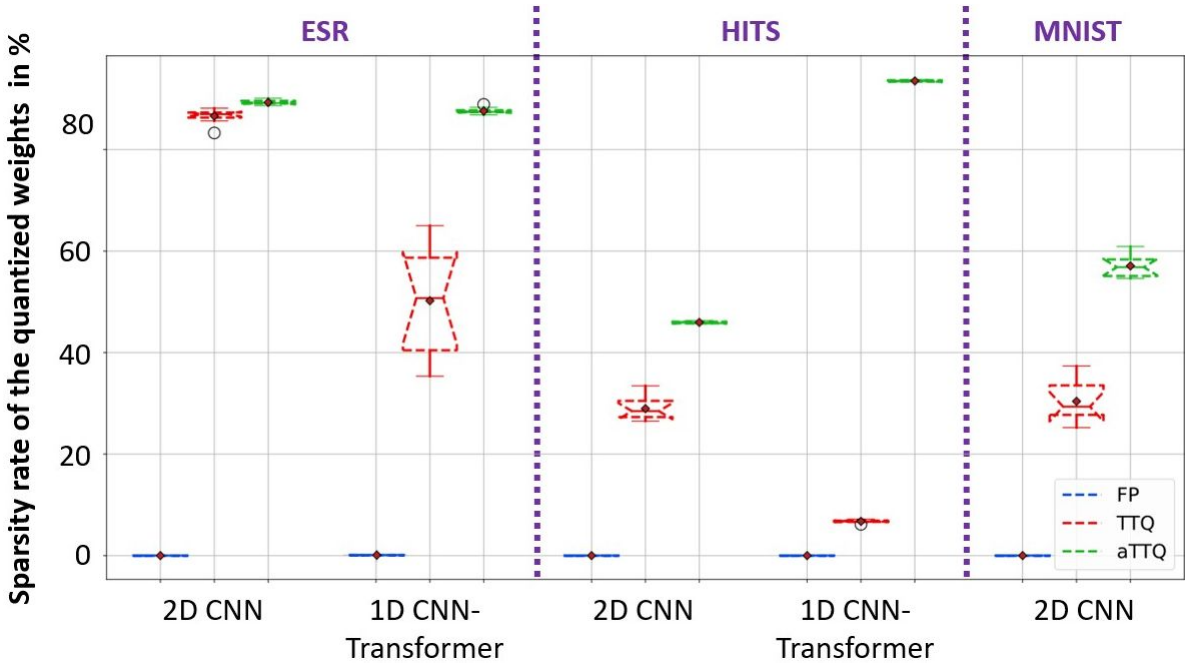


Figure 5.4: Experiment 1 results: sparsity rate of the quantized weights ($SRQW$) in %. The blue boxes correspond to the FP model. The red and green curves correspond to the TTQ and aTTQ models, respectively.

III.4.B Experiment 2: Influence of t_{min} and t_{max}

Objective: This experiment has two main objectives. First, highlight the interest of asymmetric pruning during ternary quantization with respect to the symmetric one. Second, study the influence of the two hyperparameters of our approach, t_{min} and t_{max}.

Experimental setup: To do so, we trained the 2D MNIST CNN on the MNIST dataset, and the 1D CNN-transformer on the ESR dataset, varying the values of t_{min} and t_{max} in $\{-2, -1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2\}$.

Results: Results can be found in figure 5.5. Different remarks can be done.

Firstly, we can note that the best classification performances for both models and datasets are not obtained for symmetrical values of t_{min} and t_{max} (i.e. $t_{min} = -t_{max}$), but rather for asymmetrical values (i.e. $t_{min} \neq t_{max}$) where $t_{min} \leq 0$ and $t_{max} \geq 0$. In fact, the best models acquired on the MNIST and ESR datasets are obtained for the couples $(t_{min}, t_{max}) = (-1, 0.5)$ and $(t_{min}, t_{max}) = (-2, 1)$ respectively, achieving 93.56% and 95.09% MCC, for the 2D MNIST CNN and the ESR 1D CNN-transformer, correspondingly. In addition, for both datasets and models, the best sparsity rates of the quantized weights, $SRQW$, are obtained for the same asymmetric values range of t_{min} and t_{max}[1].

Following this, we can observe the existence of trade-off between compression and classification performances. In point of fact, for both datasets, the best model in terms of compression is not the best one in terms of classification, or, in other words, the models with the highest $SRWQ$ are not the ones having the highest MCC. What is more, we can notice that, the higher the distance between t_{min} and t_{max}, the larger the sparsity rates (and subsequently the compression rates), but at some point, the classification performances start decreasing, achieving worse MCCs than closer t_{min} and t_{max} values.

Finally, related to the last remark, an interesting point to highlight is the fact that small distances between t_{min} and t_{max} giving smaller sparsity rates, do not necessarily translate in higher MCCs (i.e. better classification performances). Actually, models with sparsity rates close to 0% are more prone to have worst classification performances than models with higher sparsity rates.

III.4.C Experiment 3: Influence of weights normalization

Objective: The objective of this experiment is to study the influence of normalizing the layers to quantize (before each quantization iteration), as it is done for TTQ. For recall, in TTQ, normalization of the weights to quantize is done by the maximum of the absolute value of the weights (of a given layer), at the beginning of each quantization iteration.

Experimental setup: To do this, all the models were quantized with and without normalization of the layers to quantize. The training and model parameters were the same as the ones introduced at the beginning of this subsection.

Results: Results can be found in table 5.4. Some interesting points can be observed.

First, we can see that the classification performances with or without normalization of the weights to quantize are similar for almost all the datasets and models. However, there are some cases, such as the 2D CNNs on the ESR and MNIST datasets, where weight normalization before quantization has a substantial negative influence on the classification performances, with respective MCC gaps of 84.16% and 93.62% between the normalized and non-normalized models. In addition, in some cases normalization can weakly improve the classification performances, such as for the 2D CNN HITS model, and the 1D CNN-transformer ESR one, where we observe an MCC increase of 0.69% and 0.06%, respectively.

Ultimately, normalization does not seem to have an important effect on the compression performance of aTTQ, as for all the models and datasets, the CR_G^Q values are similar in terms

[1]Here we talk about the range of values on not the values themselves.

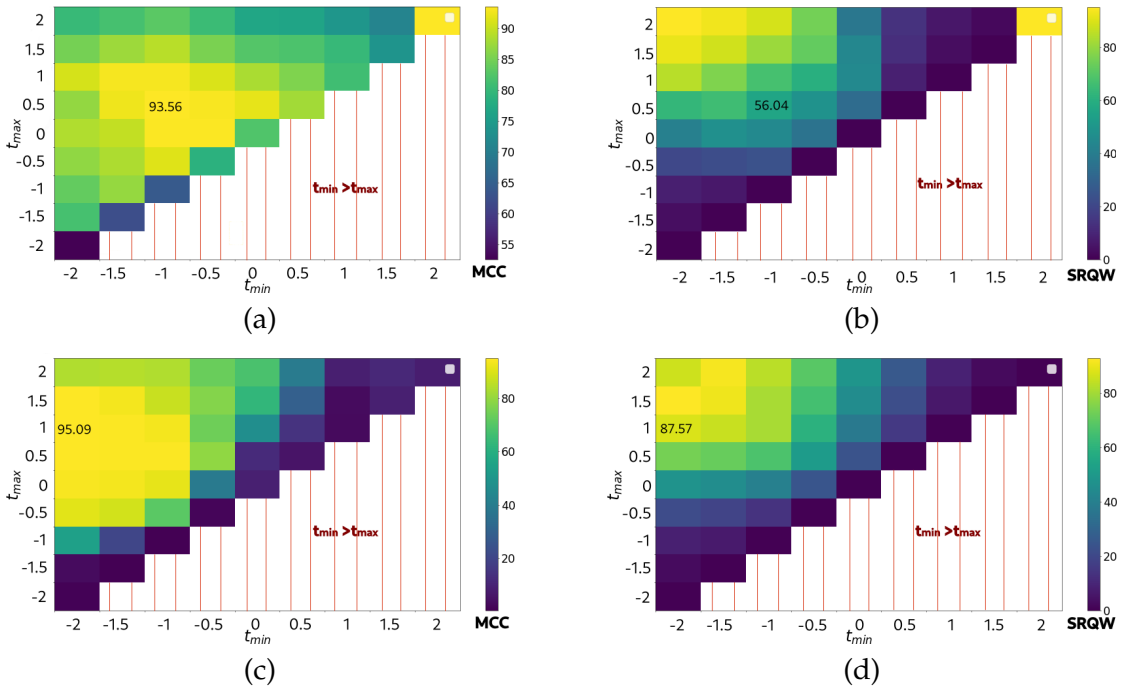


Figure 5.5: Experiment 2 results: influence of t_{min} and t_{max} (in %). (a) MCC for the 2D CNN model trained on the MNIST dataset. (b) $SRQW$ for the 2D CNN model trained on the MNIST dataset. (c) MCC for 1D CNN-transformer model trained on the ESR dataset. (d) $SRQW$ for 1D CNN-transformer model trained on the ESR dataset. The x-axis corresponds to the different tested values of t_{min} and the y-axis to the different values of t_{max}.

of means and standard deviations, with mean gaps going from 0% to 6%. Subsequently, the values of the other compression and energy consumption metrics are also alike.

III.5 Discussion

III.5.A Experiment 1: Comparison with respect to TTQ

The results of this experiment confirm that, from a compression and energy perspective, our proposed aTTQ approach is more interesting than TTQ. Indeed, aTTQ outperforms TTQ by a large margin for all the tested models and datasets, in terms of compression rates, sparsity rates, and energy consumption, at the expense of slightly smaller classification performances. However, if we focus on the per class accuracies (table 5.5) on the HITS dataset, we observe that, for the most critical class (the solid emboli one) both compressed models, aTTQ and TTQ, outperform the full precision model. For the other classes, globally, the classification performances have a slight decrease. This shows that aTTQ offers a better compression-energy-classification trade-off than TTQ. Indeed, our approach allows choosing t_{min} and t_{max} in order to better control this trade-off. This is important as, for an important number of (medical embedded) applications, we have limited memory, computation and energy resources, so even though better classification models exist, if they do not respect the hardware limitations, they are less prone to be used in practice, so we have to improve the trade-off between the different performances to have the best model respecting those criteria.

Moreover, if we observe the different compression metrics, we notice that the performance increase is not the same from one metric to another one. This is because some metrics focus only

Table 5.4: Experiment 3 results: influence of the normalization of the weights to quantize at the beginning of each iteration. The normalization column indicates if normalization is done or not.

Dataset	Model	Normalization	MCC ↑	CR_G^Q ↑
HITS	2D CNN	No	85.45 ± 3.33	46.69 ± 0.25
		Yes	**86.14 ± 3.37**	46.69 ± 0.25
	1D CNN-trans.	No	**81.66 ± 4.17**	93.11 ± 0.16
		Yes	80.45 ± 3.59	93.17 ± 0.16
ESR	2D CNN	No	**92.41 ± 2.22**	88.91 ± 0.45
		Yes	8.25 ± 13.47	88.95 ± 0.58
	1D CNN-trans.	No	95.34 ± 0.79	86.78 ± 0.63
		Yes	**95.40 ± 0.73**	86.77 ± 0.62
MNIST	2D MNIST CNN	No	**93.62 ± 0.96**	54.32 ± 2.36
		Yes	0 ± 0	60.36 ± 3.01

Table 5.5: Experiment 1: artifacts, gaseous emboli, and solid emboli accuracies of the different compressed models on the HITS-small dataset. Bold results correspond to the best ones, and italic results correspond to the second-best ones.

Model	Quant. method	Norm.	Artifact Accuracy	Gaseous Emboli Accuracy	Solid Emboli Accuracy
2D CNN	FP	No	**95.70 ± 3.01**	**94.47 ± 2.85**	85.73 ± 4.23
	TTQ	Yes	*91.23 ± 3.83*	*93.52 ± 2.33*	*86.46 ± 8.89*
	aTTQ		90.96 ± 3.16	92.33 ± 6.30	**88.41 ± 3.14**
1D CNN-trans.	FP	No	90.88 ± 4.72	**88.62 ± 5.92**	85.49 ± 3.51
	TTQ	Yes	**93.33 ± 3.73**	*86.52 ± 3.97*	**87.80 ± 5.00**
	aTTQ	No	*91.45 ± 3.66*	86.05 ± 7.82	*87.20 ± 5.51*

on the quantized layers, whereas others are more global, taking into account the whole model. Therefore, if we focus only on the quantized layers ($SRQW$ and CR_G^Q), we will have larger performances increases than if we focus on more global metrics (CR_G^T and EC_G^T). Indeed, from one model to another, we do not quantized the same proportion of weights (see last column of table 5.2), so if the proportion of weights to quantize is small (as for the 1D CNN-transformer models or the 2D CNN MNIST model), then the global compression or energy gains will be relatively small compared to the ones focusing only on the quantized layers.

Furthermore, an important point to mention is that, in some cases, quantization allows improving the classification performances with respect to the full precision models. This can be explained by three elements. Firstly, as mentioned in (Hoefler *et al.*., 2022), sparsity can act as regularization, and aTTQ and TTQ tend to give sparser models than FP models. Secondly, neural networks are highly over-parametrized with a lot of redundancies, therefore, quantization can also act as regularization because it helps to reduce over-parametrization and redundancy. Finally, aTTQ and TTQ quantize pre-trained full-precision models, where the selected layers to quantize are chosen using a Hessian-based metric, allowing to choose the layers less sensitive to quantization. Because of this, the loss landscape of the layers to quantize is relatively flat, which can help the pre-trained models to get closer to a local minimum, improving its classification performances.

Finally, from the energy consumption perspective, we observe some interesting results. Indeed, aTTQ and TTQ allow reducing the total energy consumption by important margins, specially aTTQ, for all the models and datasets. The two main reasons for this reduction are the following. First, sparsity in the weights allows reducing computation (so energy consumption) as the operations with zero-weights can be ignored. In addition, it also reduces data transfers, as zero weights do not need to be transferred to memory, but only the nonzero weights. Secondly, linked to this, reduced precision has also an important impact on the data transfer energy consumption term, EC_{DT}, as if we reduce the number of bits necessary to encode one weight, we reduce the transfers to memory (which are faster). Indeed, the transfer of one 32-bits encoded weight is equivalent to transferring sixteen 2-bits encoded nonzero weights. What is more, we can separate the total energy consumption, EC_T into the mult-adds term EC_{MA} and the data transfer term EC_{DT}, to further study the different models and quantization methods, as it is done in table 5.6. From this table, we can see that, depending on the dataset and the model, EC_{MA} or EC_{DT} can be dominant, even though data transfers are more expensive than mult-adds by a factor of 10^3. This is because, based on the size of the input of the model, the number of mult-adds can considerably increase, making the data transfer term smaller than the mult-adds term. This is the case of the HITS dataset, where the size of the inputs makes EC_{MA} the dominant term. Additionally, we can see that, with quantization, we often reduce the EC_{MA} term importance with respect to EC_{DT}, as we remove an important number of useless zero operations thanks to sparsity.

III.5.B Experiment 2: Influence of t_{min} and t_{max}

This experiment highlighted the importance of asymmetric pruning with respect to symmetric one, in addition to the importance of the choice of the hyperparameters t_{min} and t_{max} controlling the pruning thresholds.

Firstly, we observed that asymmetric thresholds, obtained using asymmetric values of t_{min} and t_{max}, improved the classification performances with respect to symmetric thresholds (where $t_{min} = -t_{max}$), and this with good sparsity rates for the quantized weights. We have two main explanations for this. First, for a neural network, there is no reason that positive and negative weights have the same importance starting from a fixed value (case of symmetric

Table 5.6: Experiment 1 energy consumption decomposition. $EC_T = EC_{MA} + EC_{DT}$ corresponds to the total energy consumption. EC_{MA} and EC^P_{MA} corresponds to the energy consumption due to the number of mult-adds, and its proportion in EC_T. EC_{DT} and EC^P_{DT} correspond to the energy consumption due to data transfers, and its proportion in EC_T. All the energy values are given in μJ, whereas the proportions are given in %.

Dataset	Model	Quant. method	Norm.	EC_{MA} ↓	EC^P_{MA}	EC_{DT} ↓	EC^P_{DT}	EC_T ↓
HITS	2D CNN	FP	No	4547 ± 0	83.15 ± 0	922 ± 0	16.85 ± 0	5469 ± 0
		TTQ	Yes	3388 ± 63	80.90 ± 1.52	800 ± 9	19.10 ± 0.21	4188 ± 71
		aTTQ		**2331 ± 10**	76.19 ± 0.32	**729 ± 1**	23.81 ± 0.03	**3060 ± 11**
	1D CNN-trans.	FP	No	1569 ± 0	37.32 ± 0	2635 ± 0	62.68 ± 0	4204 ± 0
		TTQ	Yes	1491.9 ± 1	36.17 ± 0.03	2633.1 ± 0.1	63.833 ± 0.002	4125 ± 1
		aTTQ	No	**1274.8 ± 0.4**	32.83 ± 0.01	**2608.24 ± 0.04**	67.171 ± 0.001	**3883.0 ± 0.5**
ESR	2D CNN	FP	No	230 ± 0	35.53 ± 0	417 ± 0	64.47 ± 0	647 ± 0
		TTQ	Yes	71.34 ± 2.52	46.79 ± 1.65	81.12 ± 5.00	53.21 ± 3.28	152.47 ± 7.34
		aTTQ	No	**29.80 ± 0.74**	29.68 ± 0.74	**70.62 ± 1.63**	70.32 ± 1.62	**100.42 ± 2.15**
	1D CNN-trans.	FP	No	31.64 ± 0	8.51 ± 0	339.97 ± 0	91.49 ± 0	371.61 ± 0
		TTQ	Yes	23.05 ± 1.77	6.41 ± 0.49	336.62 ± 0.68	93.59 ± 0.19	359.67 ± 2.44
		aTTQ	No	**17.51 ± 0.11**	4.98 ± 0.03	**334.13 ± 0.04**	95.02 ± 0.01	**351.64 ± 0.14**
MNIST	2D MNIST CNN	FP	No	0.56 ± 0	2.77 ± 0	19.68 ± 0	97.23 ± 0	20.24 ± 0
		TTQ	Yes	0.44 ± 0.01	2.22 ± 0.09	19.29 ± 0.05	97.78 ± 0.27	19.72 ± 0.07
		aTTQ	No	**0.30 ± 0.02**	1.58 ± 0.09	**18.94 ± 0.03**	98.42 ± 0.15	**19.24 ± 0.04**

thresholds), therefore, they should be treated differently as asymmetric thresholds do it. Second, if normalization by the maximum of the absolute value of the weights is not done, then the minimum and maximum of the weights to quantize are certainly not opposites, so we should use thresholds adapted to this situation.

Second, we noted that there is a trade-off between the compression obtained from sparsity and the classification performances. Indeed, increasing the sparsity rates of the quantized layers by increasing the gap between t_{min} and t_{max}, tends to reduce the classification performances of the compressed models. However, this reduction is not always significant, whereas the gain in sparsity is. Therefore, based on the targeted application, one could prefer higher sparsity rates in order to respect some memory and/or energy requirements, despite the classification performances decrease. But in our case, we decided to choose the values of t_{min} and t_{max} which maximize the classification performances, giving a lower priority to the sparsity rates. Nevertheless, the choice of t_{min} and t_{max} can be done differently based on the application, which is an advantage of our aTTQ approach, as our parametrization allows controlling the trade-off between compression, energy, and classification performances.

Finally, the experiment confirmed that sparsity can act as a regularizer (figure 5.5). Indeed, for a given model and dataset, there exist a range of values of t_{min} and t_{max}, where the classification performances increase along with the sparsity rates. However, for other values, the classification performances tend to decrease while having very high sparsity rates. This can be explained by the fact that, very high sparsity rates can come without an important loss of information, which cannot be compensated otherwise, therefore, considerably reducing the original performances of the pre-trained full-precision model.

III.5.C Experiment 3: Influence of weights normalization

This last experiment showed that our method is robust against weight normalization by the maximum of the absolute value of the weights (same as the one used by TTQ). Indeed, for almost all the datasets, the classification performances between normalized and non-normalized

models is similar, whereas the compression performance remains nearly the same. More interestingly, in some cases, significantly better classification performances are obtained when not applying normalization (2D CNN models on the ESR and MNIST datasets), whereas the decrease in performance when normalizing remains small for other datasets (up to 0.69% in terms of MCC). Therefore, given these results, we recommend to not apply this type of normalization when using our aTTQ approach.

III.5.D Limitations

Our quantization approach aTTQ showed promising results in terms of compression-energy-classification performances trade-off, thanks to the use of weights' statistics based asymmetric pruning. However, we identified some limitations to our method.

First, the choice of t_{min} and t_{max} is not straightforward and strongly depends on the targeted application. Indeed, even if our method allows controlling the compression-energy-classification trade-off through the choice of t_{min} and t_{max}, this trade-off depends on the application objective of the model, requiring potentially numerous experiments to find the adapted values.

Second, if we want to take advantage of the obtained compressed models from an energy consumption and inference times perspectives, specialized hardware need to be designed to make computations more efficient, as classical hardwares are not designed for this task. Indeed, as seen in Chapter 2, our approach does simulated quantization, which means that on classical hardware all the quantized weights and computations are going to be done on 32 or 64 bits precision. What is more, sparse operations between (sparse or not) tensors are not fully supported by the different deep learning libraries[1], and not straightforward to optimize for classical hardware, which also avoids fully taking advantage of the models obtained with our aTTQ quantization approach. It is because of these reasons that table 4.9 of Chapter 4 cannot be completed with the approach developed in this chapter.

Thirdly, as our aTTQ approach does extreme quantization, it does not allow quantizing all the available layers of a model without important degradation of the classification performances, which also limits the memory, energy consumption, and latency gains that we can reach.

Finally, the last limitation of our work concerns our proposed energy consumption metric EC_T, and more particularly, the mult-adds term EC_{MA}. Indeed, this term does not take into account the precision in which the nonzero weights are encoded, but only the fact that the obtained quantized tensors are sparse. Therefore, we over-estimate the energy consumption of mult-adds, as we consider all the nonzero weights encoded with 32-bits precision, which is not the case for the obtained quantized weights.

IV Overview of single feature models

As in this chapter we compressed single feature models developed in previous chapters, we can do a quick overview of the results, as it is done in table 5.7. Before this chapter, we had full precision (FP) models, and at the end, we have compressed models. We can compare them in terms of energy consumption, inference times and classification performances per class.

The different energy consumption and classification results have already been discussed in detail in section III.5. To recall, aTTQ reduces considerably the energy consumption with respect to the full precision model, while keeping close classification performances. In fact,

[1]Even tough some libraries such as PyTorch and Tensorflow have make important advances on this topic, through torch.sparse and tensorflow.sparse.

Table 5.7: Overview of single feature models on the HITS-small dataset in terms of energy consumption (in μJ), inference time (in ms), and classification performances. Bold results correspond to the best ones. We do not observe significant differences in terms of inference times between the full precision and compressed models. This is because for the compressed models, simulated quantization is used, therefore, all operations are done in full precision. To observe gains in inference time, specialized hardware or frameworks should be developed.

Model	Quant. method	Norm.	EC_G^T	Inference time	Artifact Accuracy	Gaseous Emboli Accuracy	Solid Emboli Accuracy
2D CNN	FP	No	5469 ± 0	10.26 ± 1.67	**95.70 ± 3.01**	**94.47 ± 2.85**	85.73 ± 4.23
	aTTQ	Yes	**3060 ± 11**	**10.15 ± 1.35**	90.96 ± 3.16	92.33 ± 6.30	**88.41 ± 3.14**
1D CNN-trans.	FP	No	4204 ± 0	33.81 ± 5.20	90.88 ± 4.72	**88.62 ± 5.92**	85.49 ± 3.51
	aTTQ	No	**3883 ± 1**	**32.83 ± 4.99**	**91.45 ± 3.66**	86.05 ± 7.82	**87.20 ± 5.51**

in terms of the critical class (solid emboli), models compressed with aTTQ achieve better performances.

Moreover, what is interesting to note in table 5.7 is that, the inference times between the compressed and full precision models are not significantly different. This can be explained by the fact that for inference, simulated quantization is done, meaning that all the computations are done in 32 bit precision. What is more, even if we work with sparse tensors, this is not taken into account to optimize computations. Therefore, to observe significant changes between the inference times of the compressed and full precision models, it is necessary to developed specialized hardware and/or frameworks adapted to the used quantization precision, as well as to sparse tensor operations.

V Conclusion

In this chapter we proposed a novel extreme quantization (ternary) approach based on a new quantization heuristics relying on the weights' statistics for parametrized pruning to perform ternarization. Indeed, contrary to TTQ, we propose to compute two asymmetric thresholds, one for the smaller values and one for the larger values, based on the mean and standard deviation of the weights of a given layer, parametrized by two hyperparameters, t_{min} and t_{max}. These thresholds are then used to do a preliminary ternarization of the weights to quantize, which are then scaled using two learnable parameters, one for the negative weights, and one for the positive ones.

Moreover, we validated our method through a series of experiments on three different models and datasets, obtaining promising results. Firstly, we showed that aTTQ achieves a better trade-off than TTQ, in terms of compression, energy consumption, and classification performances. Indeed, for all the models and datasets, aTTQ outperformed TTQ from a compression and energy consumption perspective, while only slightly reducing the classification performances with respect to TTQ and even the full-precision model. Secondly, we empirically proved that asymmetric thresholds for trained ternary quantization are beneficial over symmetric pruning. Particularly, our asymmetric threshold parametrization allows to better control the abovementioned trade-off, making it easier to adapt to a given application. Finally, we demonstrated that our method do not need any particular prior normalization step before extreme quantization, as it achieves similar or even significantly better results than max normalized models, when no normalization is employed.

In the following chapter, we are going to conclude about the main contributions of this

work, and we are going to present some guidelines for future work.

Conclusion and perspectives

Conclusion

In this thesis, we focused on deep learning methods for cerebral emboli classification and characterization using portable TCD data. We mainly focused on the categorization of high intensity transient signals (HITS), extracted from Doppler signals, into three main classes: solid embolus, gaseous embolus, and artifacts. The distinction between emboli and artifacts is crucial, as portable TCD devices are more prone to artifacts than the conventional ones. Moreover, emboli are dangerous as they can cause ischemic stroke, one of the leading causes of disability and death in the world. What is more, the differentiation between solid and gaseous emboli is also critical, as the treatments are not the same for both of them.

Furthermore, in the first two chapters, we saw the medical and scientific context of this work, as well as the main challenges and the state-of-the-art of the related fields: cerebral emboli classification, semi-automatic data annotation, signal classification, and model compression. Different approaches for cerebral emboli classification using TCD data exist, but they mainly focus on nonportable TCD acquired data, and use classical signal processing or machine learning techniques, without *in vivo* distinction between solid and gaseous emboli. Moreover, HITS annotation is often done manually, which is time-consuming, and few methods in the literature exist allowing to efficiently and accurately do semi-automatic data annotation with controlled label-noise. On top of that, contrary to image classification with convolutional neural networks (CNNs), in the deep learning framework, there is not yet a generic and efficient model architecture for signal classification, and the choice of the optimal representation of the signal for a given task is not trivial. What is more, the obtained deep learning models are often memory and energy greedy, making their use in practical applications limited. Therefore, model compression methods such as pruning and quantization have been developed to reduce their requirements, but they are often used sequentially, which limits the complementarity of both compression approaches.

In this work, we proposed to solve the abovementioned limitations with three main contributions. To begin with, we proposed a semi-automatic data annotation method based on feature space projection and local quality metrics, allowing to efficiently and accurately label large amounts of samples (e.g., HITS) from a small quantity of labeled samples. Our method uses auto-encoders to automatically extract features from the input samples, and then projects the obtained latent space into a 2D space for manual and automatic data annotation purposes. We propose to control the annotation error by introducing local quality metrics, allowing to indicate if the dataset local structure was preserved during projection into the 2D annotation working space. This allowed us to obtain larger labeled datasets with controlled label noise, which can then be used to train deep learning models using noise-robust loss functions, such as the generalized cross entropy loss. What is more, we used this method to semi-automatically label a larger HITS dataset, composed of 8 685 HITS (696 solid emboli, 1 002 gaseous emboli, 6

987 artifacts), which were manually verified by an expert annotator. This dataset will be used in future work to train and test the proposed models in this manuscript, as well as new models.

Second, we performed medical signal classification, based on multi-feature models, allowing to take advantage of the complementarity of different representations of a signal, namely the raw signal and a time-frequency representation (TFR), two of the most commonly used representations. Thus, we proposed two multi-feature models, one based on late-fusion with learnable attention weights, and another one based on guided and regularized intermediate fusion. Both models have two branches, one exploiting the raw signal and composed of a 1D CNN-transformer model, and another one using a TFR and composed of a 2D CNN. The late-fusion approach uses learnable attention weights per class and per representation, indicating the importance of each representation for each class. The intermediate fusion approach is jointly-trained, and guides the training of each representation's encoder with iterated losses, as well as regularizes the fused common space with semi-supervised DEC regularization, enforcing clustering in this space. The results showed that, both types of models allow increasing the classification performances with respect to single-feature models in a variety of medical signal classification tasks.

At last, we proposed to compress the previously developed single-feature models, by directly combining extreme quantization and pruning techniques. To do this, we suggest a new ternary extreme quantization heuristic, based on the statistics of the weights to quantize. Therefore, we introduced two hyperparameters controlling the trade-off between memory requirements, energy consumption, and classification performances. Our approach, aTTQ, achieved state-of-the-art compression and energy consumption performances, with slights decrease in the classification performances, comparable to the ones of TTQ, and close to the ones of full-precision models.

In the end, our different contributions are strongly linked, as our model compression method introduced in Chapter 5 can be applied to the multifeature models developed in Chapter 4, which can be trained using the datasets obtained in Chapter 3.

Perspectives

Several improvements can be done to our work, opening the doors for several perspectives.

Data annotation : Our semi-automatic data annotation method can be improved in several ways. First, we only tested our method using a simple AE taking as input TFRs of HITS. One can imagine more complex AE architectures such as VAE or diffusion models, as well as using mutli-feature autoencoders, simultaneously taking advantage of the raw signal and a TFR. Second, one can imagine adding more structure to the AE feature space using different regularization techniques such as contrastive learning or DEC, or by including the human expert in the loop (active learning). Moreover, linked to this last point, we can improve the quality of the semi-automatically labeled datasets, by proposing to human experts the opportunity to label the most difficult samples (for instance those with the smaller local quality), and/or label clusters without good local quality labeled samples.

Soft labels: Although we have not addressed this topic in this manuscript, it could be interesting to do it in the future. Indeed, our HITS datasets are labeled with soft labels (one score per class, instead of one single true class), as human annotators are not always certain about their annotations. Therefore, it could be interesting to train deep learning models to match the true soft labels probability distribution, instead of the hard-labels one, in order to better take

into account this uncertainty (Vindas, Guépié, Almar, Roux and Delachartre, 2023). This can be done using soft cross entropy loss or the Kullback-Leibler divergence as it is done for knowledge distillation, or using other types of losses such as the Jensen-Shannon divergence, or the Hellinger distance. On top of that, our semi-automatic data annotation method is completely compatible with both hard and soft labels. In addition, soft labels can be relatively robust against noisy labels, so their use for semi-automatically labeled datasets can be beneficial.

Signal classification : Our multi-feature medical signal classification approach establishes a relatively generic model for this task. However, numerous improvements can be made. First, even though our method was designed to be generic and easily adaptable to other types of data and architectures, we did not test its capabilities on datasets of very different natures. Therefore, it could be interesting to replace the encoders of each input representations using other architectures, such as ResNets or visual transformers for the TFRs, or adapted cross visual transformers for the raw signals, allowing to exploit different temporal scales of the signal. What is more, it can be interesting to validate our approach using other representations than the raw signal and/or a TFR, as well as use other datasets, not limited to medical signals. Furthermore, our multi-feature approaches can be enhanced using other types of regularization based on contrastive learning approaches with weak supervision, instead of semi-supervised DEC.

Model compression : Our asymmetrically trained ternary quantization method, aTTQ, yields a great trade-off between memory requirements, energy consumption, and classification performances. Nevertheless, this was only a first step on what we have planned. A first and direct improvement would be to apply our approach to multi-feature methods, as it was successfully applied to single-feature models, but our best performing models uses different representations of a signal and are not limited to one. On top of that, we can improve the energy consumption metric that we proposed, as the mult-adds term only takes into account the sparsity of the quantized tensors, and the precision in which the weights are encoded. What is more, it is possible to practically improve the energy consumption and latency by designing specialized hardware efficiently performing the different operations required by the model. This last point can be coupled with mixed quantization approaches, where different precisions are used at different levels of the models, in order to reduce classification performances drops, while allowing a full quantization of the model. At last, more complex ways of combining pruning and extreme ternary quantization can be proposed. For instance, one can use differentiable pruning functions, allowing to learn the pruning thresholds before quantization.

Appendices

Appendix 1: TCD-X maximum recording duration based on the pulse repetition frequency.

Table 5.8: TCD-X maximum recording's duration based on the pulse repetition frequency. The frequencies are given in kHz, and the format of the recording durations is $hh:mm:ss$, where hh are the hours, mm the minutes, and ss the seconds. The maximum recording duration depends on the pulse repetition frequency because when the latter increases, the number of recorded samples increases too (giving a better resolution of the IQ Doppler signal). Thus, as the size of the IQ Doppler signal that can be saved in the SD card is limited by Atys file management system, when the number of recorded samples increases, the maximum recording duration decreases. This is valid for the TCD-X R2, the one used in this work, but this limitation was improved on the new version, the TCD-X R3, going up to 12 hours of recording's duration.

Pulse repetition frequency (kHz)	Max. recording duration
3.7	10:08:27
4.4	08:30:01
5.5	06:47:07
6.2	05:57:54
7.4	05:04:13

Appendix 2: HITS distribution per subject.

Table 5.9: Distribution of the HITS per class and per subject (subjects 0 to 19). The HITS are classified using three classes: artifacts, solid emboli and gaseous emboli. Some HITS are classified as unknown, but they are not used to train or evaluate the classification models. Indeed, in some cases, an expert is not able to annotate a HITS. This happens particularly when a HITS can be a solid or gaseous emboli, or when there is doubt between a small intensity solid emboli and an artifact.

Subject ID	Artifacts	Solid emboli	Gaseous embolus	Unknown	Total
0	18	1	223	1	243
1	1	24	1	0	26
2	1	17	90	1	109
3	291	13	0	0	304
4	0	1	0	0	1
5	0	2	0	0	2
6	1770	0	32	0	1802
7	0	3	0	0	3
8	1	65	0	0	66
9	54	1	0	0	55
10	0	0	4	0	4
11	0	1	0	0	1
12	0	0	15	0	15
13	7	3	104	1	115
14	0	2	0	0	2
15	164	5	0	0	169
16	0	3	0	0	3
17	4	14	0	0	18
18	0	4	0	0	4
19	0	0	56	0	56

Table 5.10: Distribution of the HITS per class and per subject (subjects 20 to 38). The HITS are classified using three classes: artifacts, solid emboli and gaseous emboli. Some HITS are classified as unknown, but they are not used to train or evaluate the classification models. Indeed, in some cases, an expert is not able to annotate a HITS. This happens particularly when a HITS can be a solid or gaseous emboli, or when there is doubt between a small intensity solid emboli and an artifact.

Subject ID	Artifacts	Solid emboli	Gaseous embolus	Unknown	Total
20	0	0	7	0	7
21	0	20	0	0	20
22	1	0	0	0	1
23	0	17	0	0	17
24	0	1	0	0	1
25	0	1	0	0	1
26	0	1	0	0	1
27	1	46	7	0	54
28	304	255	1	0	560
29	0	42	277	3	322
30	0	0	7	0	7
31	0	25	0	0	25
32	4	7	1	0	12
33	260	1	0	0	261
34	42	0	0	0	42
35	0	15	0	0	15
36	18	2	0	0	20
37	0	2	4	0	6
38	74	4	32	3	113

Table 5.11: Distribution of the HITS per class and per subject (subjects 39 to 51). The HITS are classified using three classes: artifacts, solid emboli and gaseous emboli. Some HITS are classified as unknown, but they are not used to train or evaluate the classification models. Indeed, in some cases, an expert is not able to annotate a HITS. This happens particularly when a HITS can be a solid or gaseous emboli, or when there is doubt between a small intensity solid emboli and an artifact.

Subject ID	Artifacts	Solid emboli	Gaseous embolus	Unknown	Total
39	0	0	0	0	0
40	44	23	4	0	71
41	774	16	12	0	802
42	598	4	15	2	619
43	93	17	18	0	128
44	804	3	14	1	822
45	521	11	17	0	549
46	523	5	14	0	542
47	108	0	23	0	131
48	490	16	18	1	525
49	1	1	0	0	2
50	6	1	0	0	7
51	10	1	6	0	17

Appendix 3: Hessian-based metric values for the different quantized models

Table 5.12: Hessian-based metric $\mathcal{H}$ values of the 2D CNN models for layer quantization selection on the HITS, ESR, and MNIST datasets. The model is decomposed in two blocks, one for encoding composed of convolutions ($\mathcal{E}$), and one for classification composed of FC layers (C). The batch normalization layers are ignored for the sake of clarity. In bold we have the selected layers for quantization and we can see that the selected layers are the ones with the smallest value of $\mathcal{H}$ in its belonging block.

Dataset	Block	Layer	$\lvert\mathcal{H}\rvert$
HITS	$\mathcal{E}$	Conv. 1	$9.52e^{-2}$
		Conv. 2	$\mathbf{2.08e^{-3}}$
		Conv. 3	$\mathbf{2.24e^{-4}}$
		Conv. 4	$\mathbf{1.34e^{-5}}$
	C	FC	$1.27e^{-3}$
ESR	$\mathcal{E}$	Conv. 1	$1.48e^{-1}$
		Conv. 2	$\mathbf{4.35e^{-4}}$
		Conv. 3	$\mathbf{4.48e^{-5}}$
		Conv. 4	$\mathbf{1.10e^{-6}}$
	C	FC	$3.48e^{-4}$
MNIST	$\mathcal{E}$	Conv. 1	$\mathbf{7.42e^{-3}}$
		Conv. 2	$\mathbf{5.18e^{-3}}$
	C	FC 1	$5.91e^{-3}$
		FC 2	$2.99e^{-3}$

Table 5.13: Hessian-based metric $\mathcal{H}$ values of the 1D CNN-transformer models for layer quantization selection on the HITS dataset. The model is decomposed into three blocks: one 1D CNN encoder, one transformer encoder, and one classifier F_{RS}. The normalization layers, self attention projection layers, and the classifier layers are ignored for the sake of clarity. In bold we have the selected layers for quantization, and in italic the ones with the smallest value of $\mathcal{H}$ per block. The selected layer is not always the one with the smallest value of $\mathcal{H}$ per block. However, they are the ones coming the more often with the smallest value per block.

Block	**Layer**	$\lvert\mathcal{H}\rvert$
1D CNN encoder	*Conv. 1*	$6.42e^{-5}$
	Conv. 2	$\mathbf{2.39e^{-4}}$
	Conv. 3	$1.04e^{-3}$
Transformer encoder 1	*FC 1*	$1.09e^{-4}$
	FC 2	$\mathbf{2.33e^{-4}}$
Transformer encoder 2	FC 1	$1.09e^{-4}$
	FC 2	$\mathbf{7.50e^{-5}}$
Transformer encoder 3	*FC 1*	$1.22e^{-4}$
	FC 2	$\mathbf{2.14e^{-4}}$
Transformer encoder 4	*FC 1*	$8.42e^{-6}$
	FC 2	$\mathbf{6.17e^{-5}}$
Transformer encoder 5	FC 1	$4.51e^{-5}$
	FC 2	$\mathbf{1.30e^{-5}}$
Transformer encoder 6	FC 1	$1.79e^{-4}$
	FC 2	$\mathbf{5.93e^{-5}}$
Transformer encoder 7	FC 1	$2.77e^{-4}$
	FC 2	$\mathbf{1.92e^{-4}}$
Transformer encoder 8	FC 1	$5.24e^{-4}$
	FC 2	$\mathbf{2.81e^{-4}}$

Table 5.14: Hessian-based metric $\mathcal{H}$ values of the 1D CNN-transformer models for layer quantization selection on the ESR dataset. The model is decomposed into three blocks: one 1D CNN encoder, one transformer encoder, and one classifier F_{RS}. The normalization layers, self attention projection layers, and the classifier layers are ignored for the sake of clarity. In bold we have the selected layers for quantization, and in italic the ones with the smallest value of $\mathcal{H}$ per block. The selected layer is not always the one with the smallest value of $\mathcal{H}$ per block. However, they are the ones coming the more often with the smallest value per block.

Block	**Layer**	$\lvert\mathcal{H}\rvert$
1D CNN encoder	Conv. 1	$7.97e^{-3}$
	Conv. 2	$\mathbf{1.98e^{-4}}$
	Conv. 3	$2.18e^{-3}$
Transformer encoder 1	*FC 1*	$2.11e^{-5}$
	FC 2	$\mathbf{2.68e^{-5}}$
Transformer encoder 2	FC 1	$5.73e^{-4}$
	FC 2	$\mathbf{1.16e^{-4}}$
Transformer encoder 3	FC 1	$6.56e^{-5}$
	FC 2	$\mathbf{4.84e^{-6}}$
Transformer encoder 4	FC 1	$1.27e^{-5}$
	FC 2	$\mathbf{2.59e^{-6}}$

Bibliography

Abu-Omar, Y., Balacumaraswami, L., Pigott, D. W., Matthews, P. M. and Taggart, D. P. (2004). Solid and gaseous cerebral microembolization during off-pump, on-pump, and open cardiac surgery procedures, *The Journal of Thoracic and Cardiovascular Surgery* **127**(6): 1759–1765.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S0022522303018282* 15

Ackerstaff, R. G. A., Moons, K. G. M., van de Vlasakker, C. J. W., Moll, F. L., Vermeulen, F. E. E., Algra, A. and Spencer, M. P. (2000). Association of Intraoperative Transcranial Doppler Monitoring Variables With Stroke From Carotid Endarterectomy, *Stroke* **31**(8): 1817–1823.
URL: *https://www.ahajournals.org/doi/10.1161/01.STR.31.8.1817* 14

Ackerstaff, R., Jansen, C., Moll, F., Vermeulen, F., Hamerlijnck, R. and Mauser, H. (1995). The significance of microemboli detection by means of transcranial Doppler ultrasonography monitoring in carotid endarterectomy, *Journal of Vascular Surgery* **21**(6): 963–969.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S0741521495702245* 14

Aggarwal, S. K., Delahunty RN, N., Menezes, L. J., Perry, R., Wong, B., Reinthaler, M., Ozkor, M. and Mullen, M. J. (2018). Patterns of solid particle embolization during transcatheter aortic valve implantation and correlation with aortic valve calcification, *Journal of Interventional Cardiology* **31**(5): 648–654.
URL: *http://doi.wiley.com/10.1111/joic.12526* 3, 14

Ahmad, Z., Tabassum, A., Guan, L. and Khan, N. M. (2021). ECG heartbeat classification using multimodal fusion, *IEEE Access* **9**: 100615–100626. 51, 106, 107, 110, 111, 117, 118, 119

Akbari, H., Yuan, L., Qian, R., Chuang, W.-H., Chang, S.-F., Cui, Y. and Gong, B. (2021). VATT: Transformers for multimodal self-supervised learning from raw video, audio and text, Vol. 34. 46, 50

Amorim, W. P., Falcão, A. and Carvalho, M. H. (2014). Semi-supervised pattern classification using optimum-path forest, pp. 111–118. 34, 73, 78

Andrzejak, R., Lehnertz, K., Mormann, F., Rieke, C., David, P. and Elger, C. (2002). 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**: 061907. 106

Anzola, G., Magoni, M., Guindani, M., Rozzini, L. and Volta, G. D. (1999). Potential source of cerebral embolism in migraine with aura, *Neurology* **52**(8): 1622–1622.
URL: *https://n.neurology.org/content/52/8/1622* 13

Atrial Fibrillation and Stroke | National Institute of Neurological Disorders and Stroke (2023).
URL: *https://www.ninds.nih.gov/health-information/disorders/atrial-fibrillation-and-stroke* 13

Bai, H., Zhang, W., Hou, L., Shang, L., Jin, J., Jiang, X., Liu, Q., Lyu, M. and King, I. (2021). BinaryBERT: Pushing the limit of BERT quantization, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, Online, pp. 4334–4348.
URL: *https://aclanthology.org/2021.acl-long.334* 56, 59

Baltrusaitis, T., Ahuja, C. and Morency, L.-P. (2017). Multimodal machine learning: A survey and taxonomy, *IEEE Transactions on Pattern Analysis and Machine Intelligence* . 48, 49

Basic Identification Criteria of Doppler Microembolic Signals (1995). *Stroke* **26**(6): 1123–1123.
URL: *https://www.ahajournals.org/doi/10.1161/01.STR.26.6.1123* 7, 9, 23

Belkin, M., Niyogi, P. and Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, *Journal of Machine Learning Research* **7**(85): 2399–2434.
URL: *http://jmlr.org/papers/v7/belkin06a.html* 34

Benato, B. C., Gomes, J. F., Telea, A. C. and Falcão, A. X. (2021). Semi-automatic data annotation guided by feature space projection, *Pattern Recognition* **109**: 107612.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S0031320320304155* xiii, 33, 34, 35, 38, 63, 67, 68, 69, 70

Benato, B. C., Telea, A. C. and Falcao, A. X. (2018). Semi-supervised learning with interactive label propagation guided by feature space projections, *Proceedings of the 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pp. 392–399.
URL: *https://ieeexplore.ieee.org/document/8614354/* 33, 34, 63, 69, 70

Bhalgat, Y., Lee, J., Nagel, M., Blankevoort, T. and Kwak, N. (2020). LSQ+: Improving low-bit quantization through learnable offsets and better initialization, *Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 56, 58

Bilic, P., Christ, P. F. *et al.*. (2019). The liver tumor segmentation benchmark (lits), *CoRR* **abs/1901.04056**. 72

Boes, W. and Van Hamme, H. (2019). Audiovisual transformer architectures for large-scale classification and synchronization of weakly labeled audio events, *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, Association for Computing Machinery, New York, NY, USA, p. 1961–1969.
URL: *https://doi.org/10.1145/3343031.3350873* 46, 50

Bonati, L. H., Kakkos, S., Berkefeld, J., de Borst, G. J., Bulbulia, R., Halliday, A., van Herzeele, I., Koncar, I., McCabe, D. J., Lal, A., Ricco, J.-B., Ringleb, P., Taylor-Rowan, M. and Eckstein, H.-H. (2021). European Stroke Organisation guideline on endarterectomy and stenting for carotid artery stenosis, *European Stroke Journal* **6**(2).
URL: *http://journals.sagepub.com/doi/10.1177/23969873211026990* 14

Borger, M. A. and Feindel, C. M. (2002). Cerebral emboli during cardiopulmonary bypass: effect of perfusionist interventions and aortic cannulas, *The Journal of Extra-Corporeal Technology* **34**(1): 29–33. 15

Bosio, A., O'Connor, I., Traiola, M., Echavarria, J., Teich, J., Hanif, M. A., Shafique, M., Hamdioui, S., Deveautour, B., Girard, P., Virazel, A. and Bertels, K. (2021). Emerging computing

devices: Challenges and opportunities for test and reliability, *26th IEEE European Test Symposium, ETS 2021, Bruges, Belgium, May 24-28, 2021*, IEEE, pp. 1–10.
URL: *https://doi.org/10.1109/ETS50041.2021.9465409* 53

Bousseljot, R., Kreiseler, D. and Schnabel, A. (1995). Nutzung der EKG-signaldatenbank cardiodat der PTB über das internet, **40**(s1): 317–318.
URL: *https://doi.org/10.1515/bmte.1995.40.s1.317* 51, 105

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B. and Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project, *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122. 73

Carotid Artery Stenosis (2022).
URL: *https://www.hopkinsmedicine.org/health/conditions-and-diseases/carotid-artery-disease* 13

Carotid endarterectomy (2017).
URL: *https://www.nhs.uk/conditions/carotid-endarterectomy/* 14

CDC (2022). Atrial Fibrillation | cdc.gov.
URL: *https://www.cdc.gov/heartdisease/atrial_fibrillation.htm* 13

Chapelle, O., Scholkopf, B. and Zien, Eds., A. (2009). Semi-supervised learning, *IEEE Transactions on Neural Networks* **20**(3): 542–542. 33, 66, 90

Che, C., Zhang, P., Zhu, M., Qu, Y. and Jin, B. (2021). Constrained transformer network for ECG signal processing and arrhythmia classification, *BMC Medical Informatics and Decision Making* **21**. 42, 46, 47

Chen, M., Shi, X., Zhang, Y., Wu, D. and Guizani, M. (2017). Deep features learning for medical image analysis with convolutional autoencoder neural network, *IEEE Transactions on Big Data* . 34

Chen, X., Cheng, Z., Wang, S., Lu, G., Xv, G., Liu, Q. and Zhu, X. (2021). Atrial fibrillation detection based on multi-feature extraction and convolutional neural network for processing ECG signals, *Computer Methods and Programs in Biomedicine* **202**: 106009.
URL: *https://www.sciencedirect.com/science/article/pii/S0169260721000845* 51

Chicco, D. and Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation, *BMC Genomics* **21**(1): 6.
URL: *https://doi.org/10.1186/s12864-019-6413-7* 117

Chung, C. and Caplan, L. R. (2007). Chapter 45 - stroke and other neurovascular disorders, *in* C. G. Goetz (ed.), *Textbook of Clinical Neurology (Third Edition)*, third edition edn, W.B. Saunders, Philadelphia, pp. 1019–1051.
URL: *https://www.sciencedirect.com/science/article/pii/B9781416036180100451* 6

De Rosa, G. H. and Papa, J. P. (2021). OPFython: A Python implementation for optimum-path forest, *Software Impacts* p. 100113. 73

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* . 48

Dieleman, S. and Schrauwen, B. (2014). End-to-end learning for music audio, *Proceedings of the 39th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6964–6968. 42

Ding, Y., Jia, M., Miao, Q. and Cao, Y. (2022). A novel time–frequency transformer based on self–attention mechanism and its application in fault diagnosis of rolling bearings, *Mechanical Systems and Signal Processing* **168**: 108616. 46

Doersch, C., Gupta, A. and Efros, A. A. (2015). Unsupervised visual representation learning by context prediction, pp. 1422–1430. 34

Dong, Z., Yao, Z., Arfeen, D., Gholami, A., Mahoney, M. W. and Keutzer, K. (2020). Hawq-v2: Hessian aware trace-weighted quantization of neural networks, *in* H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan and H. Lin (eds), *Proceedings of the 33th International Conference on Neural Information Processing Systems*, Vol. 33, Curran Associates, Inc., pp. 18518–18529.
URL: *https://proceedings.neurips.cc/paper/2020/file/d77c703536718b95308130ff2e5cf9ee-Paper.pdf*
58, 59, 130, 131, 133

Dong, Z., Yao, Z., Gholami, A., Mahoney, M. W. and Keutzer, K. (2019). Hawq: Hessian aware quantization of neural networks with mixed-precision, *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 58

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale, *ICLR* . 48, 99

Dubey, A., Chatterjee, M. and Ahuja, N. (2018). Coreset-based neural network compression, *Proceedings of the 15th European Conference on Computer Vision (ECCV)*, Springer-Verlag, Berlin, Heidelberg, p. 469–486.
URL: *https://doi.org/10.1007/978-3-030-01234-2_28* 56

Effects of Stroke (2022).
URL: *https://www.hopkinsmedicine.org/health/conditions-and-diseases/stroke/effects-of-stroke* 6

Effects of Stroke (2023).
URL: *https://www.stroke.org/en/about-stroke/effects-of-stroke* 6

Elsken, T., Metzen, J. H. and Hutter, F. (2019). Neural architecture search: A survey, *J. Mach. Learn. Res.* **20**(1): 1997–2017. 53

Ertugrul, Ö. F., Acar, E., Aldemir, E. and Öztekin, A. (2021). Automatic diagnosis of cardiovascular disorders by sub images of the ECG signal using multi-feature extraction methods and randomized neural network, *Biomedical Signal Processing and Control* **64**: 102260. 51

Fan, A., Stock*, P., , Graham, B., Grave, E., Gribonval, R., Jegou, H. and Joulin, A. (2020). Training with quantization noise for extreme model compression, *ArXiv e-prints* . 60

Feigin, V., Nichols, E., Alam, T., Bannick, M., Beghi, E., Blake, N., Culpepper, W., Dorsey, E., Elbaz, A., Ellenbogen, R., Fisher, J., Fitzmaurice, C., Giussani, G., Glennie, L., James, S., Johnson, C., Kassebaum, N., Logroscino, G., Marin, B. and Vos, T. (2019). Global, regional, and national burden of neurological disorders, 1990-2016: a systematic analysis for the global burden of disease study 2016, *The Lancet Neurology* **18**: 459–480. 3

Feng, X., Feng, Q., Li, S., Hou, X. and Liu, S. (2020). A deep-learning-based oil-well-testing stage interpretation model integrating multi-feature extraction methods, *Energies* **13**(8).
URL: *https://www.mdpi.com/1996-1073/13/8/2042* 51

Freedman, B., Potpara, T. S. and Lip, G. Y. H. (2016). Stroke prevention in atrial fibrillation, *The Lancet* **388**(10046): 806–817.
URL: *https://www.sciencedirect.com/science/article/pii/S0140673616312570* 13

Gaier, A. and Ha, D. (2019). Weight agnostic neural networks.
URL: *http://arxiv.org/abs/1906.04358* 39

Gao, K., Zhang, Q. and Wang, H. (2019). A lightweight residual-inception convolutional neural network, *Journal of Physics: Conference Series* **1237**: 032058. 41, 53

Garcia, L. P., Carvalho, A. C. d. and Lorena, A. C. (2016). Noise detection in the meta-learning level, *Neurocomputing* **176**: 14–25.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S0925231215005482* 39

Gencer, M., Bilgin, G. and Aydin, N. (2013). Embolic Doppler ultrasound signal detection via fractional fourier transform, *Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, pp. 3050–3053.
URL: *http://ieeexplore.ieee.org/document/6610184/* 30, 31

George, M. (2017). CDC Grand Rounds: Public Health Strategies to Prevent and Treat Strokes, *MMWR. Morbidity and Mortality Weekly Report* **66**.
URL: *https://www.facebook.com/CDCMMWR* 6

Georgiadis, D., Grosset, D. G., Kelman, A., Faichney, A. and Lees, K. R. (1994). Prevalence and characteristics of intracranial microemboli signals in patients with different types of prosthetic cardiac valves., *Stroke* **25**(3): 587–592. 3

Gholami, A., Kim, S., Zhen, D., Yao, Z., Mahoney, M. and Keutzer, K. (2022). A survey of quantization methods for efficient neural network inference, *Low-Power Computer Vision*, pp. 291–326. 52, 55, 57, 58, 127

Ghosh, A., Kumar, H. and Sastry, P. S. (2017). Robust loss functions under label noise for deep neural networks, *ArXiv e-prints* .
URL: *http://arxiv.org/abs/1712.09482* 39, 40

Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K. and Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals, *Circulation* **101**(23): e215–e220. Circulation Electronic Pages: http://circ.ahajournals.org/content/101/23/e215.full PMID:1085218; doi: 10.1161/01.CIR.101.23.e215. 105

Goldberger, J. and Ben-Reuven, E. (2017). Training deep neural-networks using a noise adaptation layer, *Proceedings of the 5th International Conference on Learning Representations, ICLR*.
URL: *https://openreview.net/forum?id=H12GRgcxg* 38

Gong, Y., Chung, Y.-A. and Glass, J. (2021). AST: Audio Spectrogram Transformer, *Proceedings of the 22nd Interspeech Conference*, pp. 571–575. 46

Gong, Y., Liu, L., Yang, M. and Bourdev, L. D. (2014). Compressing deep convolutional networks using vector quantization, *ArXiv e-prints* **abs/1412.6115**. 56

Goodfellow, I., Shlens, J. and Szegedy, C. (2015). Explaining and harnessing adversarial examples, *Proceedings of the 3rd International Conference on Learning Representations, ICLR*.
URL: *http://arxiv.org/abs/1412.6572* 38

Guepie, B., Martin, M., Lacrosaz, V., Almar, M., Guibert, B. and Delachartre, P. (2018). Sequential emboli detection from ultrasound outpatient data, *IEEE Journal of Biomedical and Health Informatics* . xiii, 20, 23, 30, 31, 32, 42, 95

Guépié, B. K., Sciolla, B., Millioz, F., Almar, M. and Delachartre, P. (2017). Discrimination between emboli and artifacts for outpatient transcranial Doppler ultrasound data, *Medical & Biological Engineering & Computing* **55**(10): 1787–1797. Number: 10.
URL: *http://link.springer.com/10.1007/s11517-017-1624-z* 30, 31

Haines, D. (2018). Chapter 8 - A survey of the cerebrovascular system, *in* D. E. Haines and G. A. Mihailoff (eds), *Fundamental Neuroscience for Basic and Clinical Applications (Fifth Edition)*, fifth edition edn, Elsevier, pp. 122–137.e1.
URL: *https://www.sciencedirect.com/science/article/pii/B9780323396325000086* 6

Han, S., Mao, H. and Dally, W. J. (2016). Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding, *in* Y. Bengio and Y. LeCun (eds), *Proceedings of the 4th International Conference on Learning Representations, ICLR 2016*.
URL: *http://arxiv.org/abs/1510.00149* 56, 60, 127

Han, S., Pool, J., Tran, J. and Dally, W. J. (2015). Learning both weights and connections for efficient neural networks, *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, MIT Press, Cambridge, MA, USA, p. 1135–1143. 53

Hassibi, B., Stork, D. and Wolff, G. (1993). Optimal brain surgeon and general network pruning, *Proceedings of the IEEE 1993 International Conference on Neural Networks*, pp. 293–299 vol.1. 53

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition, *Proceedings of the 29th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. 41, 53

He, Y., Lin, J., Liu, Z., Wang, H., Li, L.-J. and Han, S. (2018). AMC: AutoML for model compression and acceleration on mobile devices, *in* V. Ferrari, M. Hebert, C. Sminchisescu and Y. Weiss (eds), *Computer Vision – ECCV 2018*, Springer International Publishing, Cham, pp. 815–832. 54

Hilal, A. M., Albraikan, A. A., Dhahbi, S., Nour, M. K., Mohamed, A., Motwakel, A., Zamani, A. S. and Rizwanullah, M. (2022). Intelligent epileptic seizure detection and classification model using optimal deep canonical sparse autoencoder, *Biology* **11**(8). 107, 109, 110, 111, 117

Hinton, G. E., Vinyals, O. and Dean, J. (2015). Distilling the knowledge in a neural network, *ArXiv* **abs/1503.02531**. 52

Hoefler, T., Alistarh, D., Ben-Nun, T., Dryden, N. and Peste, A. (2022). Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks, *Journal of Machine Learning Research* **22**(1). 53, 127, 144

Hori, C., Hori, T., Wichern, G., Wang, J., Lee, T.-Y., Cherian, A. and Marks, T. K. (2018). Multimodal attention for fusion of audio and spatiotemporal features for video description, *CVPR Workshops*. 50

Horowitz, M. (2014). 1.1 computing's energy problem (and what we can do about it), *Proceedings of the 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 10–14. 132, 134, 135

Hou, L. and Kwok, J. T. (2018). Loss-aware weight quantization of deep networks, *Proceedings of the 6th International Conference on Learning Representations*.
URL: *https://openreview.net/forum?id=BkrSv0lA-* 59

Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. 52

Iguchi, Y., Kimura, K., Kobayashi, K., Ueno, Y., Shibazaki, K. and Inoue, T. (2008). Microembolic signals at 48 hours after stroke onset contribute to new ischaemia within a week, *Journal of Neurology, Neurosurgery & Psychiatry* **79**(3): 253–259.
URL: *https://jnnp.bmj.com/lookup/doi/10.1136/jnnp.2007.123414* 6

Ikizceli, T., Donmez, H., Kahveci, S. and Kahriman, G. (2021). Ischaemic brain changes associated with catheter-based diagnostic cerebral angiography: a diffusion-weighted imaging study, *Polish Journal of Radiology* **86**(1): 481–488.
URL: *https://www.termedia.pl/doi/10.5114/pjr.2021.108793* 15

Imaduddin, S. M., LaRovere, K. L., Kussman, B. D. and Heldt, T. (2019). A time-frequency approach for cerebral embolic load monitoring, *IEEE Transactions on Biomedical Engineering* **67**(4): 1007–1018. 30, 95

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H. and Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference, *Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 58

Jansen, C., Ramos, L. M., van Heesewijk, J. P., Moll, F. L., van Gijn, J. and Ackerstaff, R. G. (1994). Impact of microembolism and hemodynamic changes in the brain during carotid endarterectomy., *Stroke* **25**(5): 992–997.
URL: *https://www.ahajournals.org/doi/10.1161/01.STR.25.5.992* 14

Jeon, H., Jung, Y., Lee, S. and Jung, Y. (2020). Area-efficient short-time fourier transform processor for time–frequency analysis of non-stationary signals, *Applied Sciences* **10**(20).
URL: *https://www.mdpi.com/2076-3417/10/20/7208* 20

Ji, T., Jain, S., Ferdman, M., Milder, P., Schwartz, H. A. and Balasubramanian, N. (2021). On the distribution, sparsity, and inference-time quantization of attention values in transformers, *ArXiv* **abs/2106.01335**. 53

Jin, J., Yang, S., Zhao, B., Luo, L. and Woo, W. L. (2020). Attention-block deep learning based features fusion in wearable social sensor for mental wellbeing evaluations, *IEEE Access* **8**. 51

Jin, Q., Yang, L. and Liao, Z. (2020). Adabits: Neural network quantization with adaptive bit-widths, *Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 59

Johnson, W., Onuma, O., Owolabi, M. and Sachdev, S. (2016). Stroke: a global response is needed, *Bulletin of the World Health Organization* **94**(9): 634–634A. Number: 9.
URL: *http://www.who.int/entity/bulletin/volumes/94/9/16-181636.pdf* 3

Jolliffe, I. T. and Cadima, J. (2016). Principal component analysis: a review and recent developments, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **374**(2065): 20150202. Number: 2065.
URL: *https://royalsocietypublishing.org/doi/10.1098/rsta.2015.0202* 68

Judge, C., Mello, S., Bradley, D. and Harbison, J. (2017). A Systematic Review of the Causes and Management of Ischaemic Stroke Caused by Nontissue Emboli, *Stroke Research and Treatment* **2017**: 1–8.
URL: *https://www.hindawi.com/journals/srt/2017/7565702/* 15

Kachuee, M., Fazeli, S. and Sarrafzadeh, M. (2018). ECG heartbeat classification: A deep transferable representation, *Proceedings of the IEEE International Conference on Healthcare Informatics (ICHI)*, IEEE, pp. 443–444. 106

Kamel, H., Okin, P. M., Elkind, M. S. and Iadecola, C. (2016). Atrial fibrillation and mechanisms of stroke, *Stroke* **47**(3): 895–900.
URL: *https://www.ahajournals.org/doi/abs/10.1161/STROKEAHA.115.012004* 13, 14

Kanjilal, P., Dey, P. and Banerjee, D. (1993). Reduced-size neural networks through singular value decomposition and subset selection, *Electronics Letters* **29**: 1516–1518. 53

Karahoca, A. and Tunga, M. A. (2015). A polynomial based algorithm for detection of embolism, *Soft Computing* **19**(1): 167–177.
URL: *http://link.springer.com/10.1007/s00500-014-1240-x* 30, 31

Karimi, D., Dou, H., Warfield, S. K. and Gholipour, A. (2020). Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis, *Medical Image Analysis* **65**: 101759.
URL: *https://www.sciencedirect.com/science/article/pii/S1361841520301237* 65

Karita, S., Wang, X., Watanabe, S., Yoshimura, T., Zhang, W., Chen, N., Hayashi, T., Hori, T., Inaguma, H., Jiang, Z., Someki, M., Soplin, N. and Yamamoto, R. (2019). A comparative study on transformer vs rnn in speech applications, *Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop*, IEEE, pp. 449–456. 44, 46

Keunen, R. W. (2022). The art of TCD embolus detection in prevention of stroke. Brain Physics Seminar. 9, 10

Keunen, R. W. M., Hoogenboezem, R., Wijnands, R., Van den Hengel, A. C. M. and Ackerstaff, R. G. A. (2008). Introduction of an embolus detection system based on analysis of the transcranial Doppler audio-signal, *Journal of Medical Engineering & Technology* **32**(4): 296–304.
URL: *http://www.tandfonline.com/doi/full/10.1080/03091900701541265* 9, 30, 95

Kim, J.-G. and Lee, B. (2019). Appliance classification by power signal analysis based on multi-feature combination multi-layer LSTM, *Energies* **12**(14).
URL: *https://www.mdpi.com/1996-1073/12/14/2804* 51, 56

Kim, S., Gholami, A., Yao, Z., Mahoney, M. W. and Keutzer, K. (2021). I-bert: Integer-only bert quantization, *Proceedings of the 38th International Conference on Machine Learning*. 58

Kingma, D. P., Rezende, D. J., Mohamed, S. and Welling, M. (2014). Semi-supervised learning with deep generative models, *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, MIT Press, Cambridge, MA, USA, pp. 3581–3589. 33, 63

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P. and Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations, *Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net.
URL: *https://openreview.net/forum?id=H1eA7AEtvS* 56

Lane, N. D., Bhattacharya, S., Georgiev, P., Forlivesi, C. and Kawsar, F. (2015). An early resource characterization of deep learning on wearables, smartphones and internet-of-things devices, *Proceedings of the 2015 International Workshop on Internet of Things towards Applications*, IoT-App '15, Association for Computing Machinery, New York, NY, USA, p. 7–12.
URL: *https://doi.org/10.1145/2820975.2820980* 26

LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database.
URL: *http://yann.lecun.com/exdb/mnist/* 72

Lee, J., Park, J., Kim, K. and Nam, J. (2017). Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms, *Proceedings of the 14th Sound Music Computing Conference*. 42

Li, F. and Liu, B. (2016). Ternary weight networks, *ArXiv* **abs/1605.04711**. 129

Li, F., Liu, B., Wang, X., Zhang, B. and Yan, J. (2022). Ternary weight networks, *ArXiv e-prints* . 59

Li, J., Mohamed, A., Zweig, G. and Gong, Y. (2015). LSTM time and frequency recurrence for automatic speech recognition, *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, IEEE, pp. 187–191.
URL: *http://ieeexplore.ieee.org/document/7404793/* 42

Li, Y., Yang, J., Song, Y., Cao, L., Luo, J. and Li, L.-J. (2017). Learning from noisy labels with distillation, *arXiv e-prints* .
URL: *http://arxiv.org/abs/1703.02391* 39

Liang, T., Glossner, J., Wang, L., Shi, S. and Zhang, X. (2021). Pruning and quantization for deep neural network acceleration: A survey, *Neurocomputing* **461**: 370–403.
URL: *https://www.sciencedirect.com/science/article/pii/S0925231221010894* 127

Lin, X., Zhao, C. and Pan, W. (2017). Towards accurate binary convolutional neural network, *in* I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett (eds), *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Vol. 30, Curran Associates, Inc.
URL: *https://proceedings.neurips.cc/paper/2017/file/b1a59b315fc9a3002ce38bbe070ec3f5-Paper.pdf* 59

Liu, Z.-M. (2021). Multi-feature fusion for specific emitter identification via deep ensemble learning, *Digital Signal Processing* **110**: 102939.
URL: *https://www.sciencedirect.com/science/article/pii/S1051200420302840* 51

Lueks, W., Mokbel, B., Biehl, M. and Hammer, B. (2011). How to evaluate dimensionality reduction? - Improving the co-ranking matrix, *arXiv:1110.3917 [cs]* .
URL: *http://arxiv.org/abs/1110.3917* xiii, xiv, 36, 37, 69, 90, 92, 93

Luo, J.-H., Wu, J. and Lin, W. (2017). Thinet: A filter level pruning method for deep neural network compression, *Proceedings of the 16th IEEE International Conference on Computer Vision (ICCV)*, pp. 5068–5076. 54

Lyon, R. F. (2017). Human and machine hearing: Extracting meaning from sound, Cambridge University Press. 21

Maaten, L. V. D. and Hinton, G. (2008). Visualizing data using t-SNE, *Journal of Machine Learning Research* **9**(86): 2579–2605.
URL: *http://jmlr.org/papers/v9/vandermaaten08a.html* 25, 34, 68

Makimoto, H., Höckmann, M., Lin, T., Glöckner, D., Gerguri, S., Clasen, L., Schmidt, J., Assadi-Schmidt, A., Bejinariu, A., Müller, P., Angendohr, S., Babady, M., Brinkmeyer, C., Makimoto, A. and Kelm, M. (2020). Performance of a convolutional neural network derived from an ECG database in recognizing myocardial infarction, *Scientific Reports* **10**(1): 8445. 109, 110

Manessi, F., Rozza, A., Bianco, S., Napoletano, P. and Schettini, R. (2017). Automated pruning for deep neural network compression, *Proceedings of the 24th International Conference on Pattern Recognition (ICPR)*, pp. 657–664.
URL: *https://api.semanticscholar.org/CorpusID:23164747* xiii, 54, 55

Mao, S., Li, Y., Ma, Y., Zhang, B., Zhou, J. and Wang, K. (2020). Automatic cucumber recognition algorithm for harvesting robots in the natural environment using deep learning and multi-feature fusion, *Computers and Electronics in Agriculture* **170**. 50, 51

Marcoff, L. and Homma, S. (2014). Embolism, cardiac and aortic, *in* M. J. Aminoff and R. B. Daroff (eds), *Encyclopedia of the Neurological Sciences (Second Edition)*, second edition edn, Academic Press, Oxford, pp. 1–7.
URL: *https://www.sciencedirect.com/science/article/pii/B9780123851574004140* 14

Mariet, Z. and Sra, S. (2016). Diversity networks: Neural network compression using determinantal point processes, *Proceedings of the 4th International Conference on Learning Representations (ICLR)*. 54

Markus, H., Israel, D., Brown, M., Loh, A., Buckenham, T. and Clifton, A. (1993a). Microscopic air embolism during cerebral angiography and strategies for its avoidance, *The Lancet* **341**(8848): 784–787. Number: 8848.
URL: *https://linkinghub.elsevier.com/retrieve/pii/014067369390561T* 3

Markus, H., Israel, D., Brown, M., Loh, A., Buckenham, T. and Clifton, A. (1993b). Microscopic air embolism during cerebral angiography and strategies for its avoidance, *The Lancet* **341**(8848): 784–787.
URL: *https://linkinghub.elsevier.com/retrieve/pii/014067369390561T* 15

Markus, H. S. and MacKinnon, A. (2005). Asymptomatic embolization detected by Doppler ultrasound predicts stroke risk in symptomatic carotid artery stenosis, *Stroke* **36**(5): 971–975.
URL: *https://www.ahajournals.org/doi/abs/10.1161/01.STR.0000162717.62684.40* 13

Markus, H. S. and Punter, M. (2005). Can Transcranial Doppler Discriminate Between Solid and Gaseous Microemboli?: Assessment of a Dual-Frequency Transducer System, *Stroke* **36**(8): 1731–1734.
URL: *https://www.ahajournals.org/doi/10.1161/01.STR.0000173399.20127.b3* 25, 30, 95

Masson, J.-B., Kovac, J., Schuler, G., Ye, J., Cheung, A., Kapadia, S., Tuzcu, M. E., Kodali, S., Leon, M. B. and Webb, J. G. (2009). Transcatheter Aortic Valve Implantation, *JACC: Cardiovascular Interventions* **2**(9): 811–820.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S1936879809005044* 14

McFee, B. and Bello, J. P. (2017). Structured training for large-vocabulary chord recognition, *Proceedings of the International Society for Music Information Retrieval Conference*. 43

McInnes, L., Healy, J. and Melville, J. (2020). UMAP: Uniform manifold approximation and projection for dimension reduction, *arXiv:1802.03426 [cs, stat]* .
URL: *http://arxiv.org/abs/1802.03426* 68, 118

McKinstry, J. L., Esser, S. K., Appuswamy, R., Bablani, D., Arthur, J. V., Yildiz, I. B. and Modha, D. S. (2019). Discovering low-precision networks close to full-precision networks for efficient inference, *Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition (EMC2-NIPS)*, pp. 6–9. 59

Meschia, J. and Brott, T. (2018). Ischaemic stroke, *European journal of neurology* **25**(1): 35–40. 3

Meyer, J. S., Mutamatsu, K. and Shirai, T. (1996). Cerebral embolism as a cause of stroke and transient ischemic attack, *Echocardiography* **13**(5): 513–518.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-8175.1996.tb00929.x* 3, 6

Mitchell, S. and Gorman, D. (2002). The pathophysiology of cerebral arterial gas embolism, *The journal of extra-corporeal technology* **34**(1): 18—23.
URL: *http://europepmc.org/abstract/MED/11911624* 15

Mohamed, A., Okhonko, D. and Zettlemoyer, L. (2019). Transformers with convolutional context for asr, *ArXiv* **abs/1904.11660**. 46

Molka, D., Hackenberg, D., Schöne, R. and Müller, M. S. (2010). Characterizing the energy consumption of data transfers and arithmetic operations on x8664 processors, *Proceedings of the International Conference on Green Computing*, pp. 123–133. 134

Moody, G. and Mark, R. (2001). The impact of the mit-bih arrhythmia database, *IEEE Engineering in Medicine and Biology Magazine* **20**(3): 45–50. 51

Naqvi, J., Yap, K. H., Ahmad, G. and Ghosh, J. (2013). Transcranial Doppler Ultrasound: A Review of the Physical Principles and Major Applications in Critical Care, *International Journal of Vascular Medicine* **2013**: 1–13.
URL: *http://www.hindawi.com/journals/ijvm/2013/629378/* 16

Natarajan, A., Chang, Y., Mariani, S., Rahman, A., Boverman, G., Vij, S. and Rubin, J. (2020). A wide and deep transformer neural network for 12-lead ECG classification, *Proceedings of the 47th 2020 Computing in Cardiology Conference*, pp. 1–4. xiii, 42, 46, 47, 99, 100

Nguyen, D., Nguyen, D. T., Zeng, R., Nguyen, T. T., Tran, S., Nguyen, T. K., Sridharan, S. and Fookes, C. (2021). Deep auto-encoders with sequential learning for multimodal dimensional emotion recognition, *IEEE Transactions on Multimedia* . 42

Nishizaki, H. and Makino, K. (2019). Signal classification using deep learning, *Proceedings of the 2019 IEEE International Conference on Sensors and Nanotechnology*, pp. 1–4. 43

Ogasawara, K., Suga, Y., Sasaki, M., Chida, K., Kobayashi, M., Yoshida, K., Otawara, Y. and Ogawa, A. (2008a). Intraoperative Microemboli and Low Middle Cerebral Artery Blood Flow Velocity Are Additive in Predicting Development of Cerebral Ischemic Events After Carotid Endarterectomy, *Stroke* **39**(11): 3088–3091.
URL: *https://www.ahajournals.org/doi/10.1161/STROKEAHA.107.511360* 10

Ogasawara, K., Suga, Y., Sasaki, M., Chida, K., Kobayashi, M., Yoshida, K., Otawara, Y. and Ogawa, A. (2008b). Intraoperative Microemboli and Low Middle Cerebral Artery Blood Flow Velocity Are Additive in Predicting Development of Cerebral Ischemic Events After Carotid Endarterectomy, *Stroke* **39**(11): 3088–3091.
URL: *https://www.ahajournals.org/doi/10.1161/STROKEAHA.107.511360* 14

Okawa, M., Saito, T., Sawada, N. and Nishizaki, H. (2019). Audio classification of bit-representation waveform, *Proceedings of the 20th Interspeech Conference*, pp. 2553–2557. 22, 23, 41, 42, 43

Organization, W. H. (2006). Neurological disorders: public health challenges, World Health Organization. 3

Ortega, J. D. S., Senoussaoui, M., Granger, E., Pedersoli, M., Cardinal, P. and Koerich, A. L. (2019). Multimodal fusion with deep neural networks for audio-video emotion recognition, *ArXiv* **abs/1907.03196**. 50

Park, H. and Yoo, C. D. (2020). Cnn-based learnable gammatone filterbank and equal-loudness normalization for environmental sound classification, *IEEE Signal Processing Letters* **27**: 411–415. 41, 127

Park, M. S., Xu, X. and Brick, C. (2018). SQuantizer: Simultaneous learning for both sparse and low-precision neural networks, *CoRR* **abs/1812.08301**.
URL: *http://arxiv.org/abs/1812.08301* 60

Patent Foramen Ovale (PFO) (2011).
URL: *https://www.heart.org/en/health-topics/congenital-heart-defects/about-congenital-heart-defects/patent-foramen-ovale-pfo* 13, 14

Patrini, G., Rozza, A., Menon, A., Nock, R. and Qu, L. (2017). Making deep neural networks robust to label noise: a loss correction approach, *arXiv e-prints* .
URL: *http://arxiv.org/abs/1609.03683* 39

Pereyra, G., Tucker, G., Chorowski, J., Kaiser, and Hinton, G. (2017). Regularizing neural networks by penalizing confident output distributions, *arXiv e-prints* .
URL: *http://arxiv.org/abs/1701.06548* 38

Piccialli, F., Somma, V. D., Giampaolo, F., Cuomo, S. and Fortino, G. (2021). A survey on deep learning in medicine: Why, how and when?, *Information Fusion* **66**: 111–137.
URL: *https://www.sciencedirect.com/science/article/pii/S1566253520303651* 52

Polino, A., Pascanu, R. and Alistarh, D. (2018). Model compression via distillation and quantization, *ArXiv e-prints* . 56

Prato, G., Charlaix, E. and Rezagholizadeh, M. (2020). Fully quantized transformer for machine translation, *Findings of the Association for Computational Linguistics: EMNLP 2020*, Association for Computational Linguistics, Online, pp. 1–14.
URL: *https://aclanthology.org/2020.findings-emnlp.1* 56

Pu, J., Panagakis, Y. and Pantic, M. (2021). Learning separable time-frequency filterbanks for audio classification, *Proceedings of the 46th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3000–3004. 41, 42

Purkayastha, S. and Sorond, F. (2013). Transcranial Doppler Ultrasound: Technique and Application, *Seminars in Neurology* **32**(04): 411–420.
URL: *http://www.thieme-connect.de/DOI/DOI?10.1055/s-0032-1331812* 16, 17

Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S.-Y. and Sainath, T. (2019). Deep learning for audio signal processing, *IEEE Journal of Selected Topics in Signal Processing* **13**(2): 206–219. 41, 42

Qu, Y., Liu, C., Zhang, K., Xiao, K., Jin, B. and Xiong, H. (2021). Diagnostic sparse connectivity networks with regularization template, *IEEE Transactions on Knowledge and Data Engineering* . 47

Quantifying the performance of the TPU, our first machine learning chip (n.d.).
URL: *https://cloud.google.com/blog/products/gcp/quantifying-the-performance-of-the-tpu-our-first-machine-learning-chip* 53

Rastegari, M., Ordonez, V., Redmon, J. and Farhadi, A. (2016). Xnor-net: Imagenet classification using binary convolutional neural networks, *in* B. Leibe, J. Matas, N. Sebe and M. Welling (eds), *Proceedings of the 14th European Conference on Computer Vision (ECCV)*, Springer International Publishing, Cham, pp. 525–542. 59, 127

Rawat, W. and Wang, Z. (2017). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review, *Neural Computation* **29**(9): 2352–2449.
URL: *https://doi.org/10.1162/neco_a_00990* 40

Reed, S. E., Lee, H., Anguelov, D., Szegedy, C., Erhan, D. and Rabinovich, A. (2015). Training deep neural networks on noisy labels with bootstrapping, *in* Y. Bengio and Y. LeCun (eds), *Proceedings of the 3rd International Conference on Learning Representations, ICLR*.
URL: *http://arxiv.org/abs/1412.6596* 39

Ringelstein, E. B., Droste, D. W., Babikian, V. L., Evans, D. H., Grosset, D. G., Kaps, M., Markus, H. S., Russell, D. and Siebler, M. (1998). Consensus on microembolus detection by TCD, *Stroke* **29**: 725–729. 3, 7, 23, 97

Rosenberg, C., Hebert, M. and Schneiderman, H. (2005). Semi-supervised self-training of object detection models, *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, IEEE, pp. 29–36.
URL: *http://ieeexplore.ieee.org/document/4129456/* 33, 63

Rosenkranz, M., Fiehler, J., Niesen, W., Waiblinger, C., Eckert, B., Wittkugel, O., Kucinski, T., Röther, J., Zeumer, H., Weiller, C. and Sliwka, U. (2006). The amount of solid cerebral microemboli during carotid stenting does not relate to the frequency of silent ischemic lesions, *American Journal of Neuroradiology* **27**(1): 157–161.
URL: *http://www.ajnr.org/content/27/1/157* 3, 6

Roth, E. J. (2011). Cerebral embolism, *in* J. S. Kreutzer, J. DeLuca and B. Caplan (eds), *Encyclopedia of Clinical Neuropsychology*, Springer, New York, NY, pp. 530–531.
URL: *https://doi.org/10.1007/978-0-387-79948-3_2166* 6

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics* **20**: 53–65.
URL: *https://linkinghub.elsevier.com/retrieve/pii/0377042787901257* 68

Russell, D. and Brucher, R. (2002). Online Automatic Discrimination Between Solid and Gaseous Cerebral Microemboli With the First Multifrequency Transcranial Doppler, *Stroke* **33**(8): 1975–1980.
URL: *https://www.ahajournals.org/doi/10.1161/01.STR.0000022809.46400.4B* 25

Sainath, T. N. and Li, B. (2016). Modeling time-frequency patterns with LSTM vs. convolutional architectures for LVCSR tasks, *Proceedings of the 17th Interspeech Conference*, pp. 813–817.
URL: *http://www.isca-speech.org/archive/Interspeech_2016/abstracts/0084.html* 43

Schminke, U., Ries, S., Daffertshofer, M., Staedt, U. and Hennerici, M. (1995). Patent foramen ovale: A potential source of cerebral embolism?, *Cerebrovascular Diseases* **5**(2): 133–138.
URL: *https://www.karger.com/DOI/10.1159/000107838* 13, 15

Schuchlenz, H. W., Weihs, W., Horner, S. and Quehenberger, F. (2000). The association between the diameter of a patent foramen ovale and the risk of embolic cerebrovascular events, *The American Journal of Medicine* **109**(6): 456–462.
URL: *https://www.sciencedirect.com/science/article/pii/S0002934300005301* 14

Seera, M., Lim, C. P., Tan, K. S. and Liew, W. S. (2017). Classification of transcranial Doppler signals using individual and ensemble recurrent neural networks, *Neurocomputing* **249**: 337–344.
URL: *https://www.sciencedirect.com/science/article/pii/S0925231217305684* 31

Serbes, G. and Aydin, N. (2014). Denoising performance of modified dual-tree complex wavelet transform for processing quadrature embolic Doppler signals, *Medical & Biological Engineering & Computing* **52**(1): 29–43. Number: 1.
URL: *http://link.springer.com/10.1007/s11517-013-1114-x* 30, 31, 95

Serena, J., Jimenez-Nieto, M., Silva, Y. and Castellanos, M. (2010). Patent foramen ovale in cerebral infarction, *Current Cardiology Reviews* **6**(3): 162–174. Number: 3.
URL: *http://www.eurekaselect.com/openurl/content.php?genre=articleissn=1573-403Xvolume=6issue=3spage=162* 3

Serena, J., nez Nieto, M., Silva, Y. and Castellanos, M. (2010). Patent foramen ovale in cerebral infarction, *Current Cardiology Reviews* **6**(3): 162–174. 13

Sharan, R., Xiong, H. and Berkovsky, S. (2021). Benchmarking audio signal representation techniques for classification with convolutional neural networks, *Sensors* **21**: 3434. 41, 42

Shen, S., Dong, Z., Ye, J., Ma, L., Yao, Z., Gholami, A., Mahoney, M. W. and Keutzer, K. (2019). Q-bert: Hessian based ultra low precision quantization of bert, *Proceedings of the AAAI Conference on Artificial Intelligence*. 58, 59

Siebler, M., Kleinschmidt, A., Sitzer, M., Steinmetz, H. and Freund, H. J. (1994). Cerebral microembolism in symptomatic and asymptomatic high-grade internal carotid artery stenosis, *Neurology* **44**(4): 615–618. 13

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition, *Proceedings of the International Conference on Learning Representations*. 41

Sindhwani, V., Niyogi, P. and Belkin, M. (2005). Beyond the point cloud: from transductive to semi-supervised learning, *Proceedings of the 22nd International Conference on Machine learning*. 34

Sitzer, M., Müller, W., Siebler, M., Hort, W., Kniemeyer, H.-W., Jäncke, L. and Steinmetz, H. (1995). Plaque ulceration and lumen thrombus are the main sources of cerebral microemboli in high-grade internal carotid artery stenosis, *Stroke* **26**(7): 1231–1233.
URL: *https://www.ahajournals.org/doi/abs/10.1161/01.STR.26.7.1231* 13

Sombune, P., Phienphanich, P., Muengtaweepongsa, S., Ruamthanthong, A. and Tantibundhit, C. (2016). Automated embolic signal detection using adaptive gain control and classification using anfis, *Proceedings of the 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 3825–3828. 30

Sombune, P., Phienphanich, P., Phuechpanpaisal, S., Muengtaweepongsa, S., Ruamthanthong, A., Chazal, P. D. and Tantibundhit, C. (2018). Automated cerebral emboli detection using adaptive threshold and adaptive neuro-fuzzy inference system, *IEEE Access* **6**: 55361–55371. 30, 95

Sombune, P., Phienphanich, P., Phuechpanpaisal, S., Muengtaweepongsa, S., Ruamthanthong, A. and Tantibundhit, C. (2017). Automated embolic signal detection using deep convolutional neural network, *Proceedings of the 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, pp. 3365–3368.
URL: *https://ieeexplore.ieee.org/document/8037577/* xiii, 31, 32, 33, 42, 95

Song, H., Kim, M., Park, D., Shin, Y. and Lee, J.-G. (2022). Learning from noisy labels with deep neural networks: A survey, *IEEE Transactions on Neural Networks and Learning Systems* . 38, 92

Spencer, M. P. (1992). Detection of cerebral arterial emboli, *Transcranial Doppler* pp. 215–230. 3

Spencer, M. P., Thomas, G. I., Nicholls, S. C. and Sauvage, L. R. (1990). Detection of middle cerebral artery emboli during carotid endarterectomy using transcranial Doppler ultrasonography., *Stroke* **21**(3): 415–423.
URL: *https://www.ahajournals.org/doi/10.1161/01.STR.21.3.415* 14

Stirling, J., Muramatsu, K. and Shirai, T. (1996). Cerebral Embolism as a Cause of Stroke and Transient Ischemic Attack, *Echocardiography* **13**(5): 513–518. 13

Stork, J. L., Kimura, K., Levi, C. R., Chambers, B. R., Abbott, A. L. and Donnan, G. A. (2002). Source of microembolic signals in patients with high-grade carotid stenosis, *Stroke* **33**(8): 2014–2018.
URL: *https://www.ahajournals.org/doi/abs/10.1161/01.STR.0000021002.17394.7F* 6, 13

Sun, S., Cheng, Y., Gan, Z. and Liu, J. (2019). Patient knowledge distillation for bert model compression, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 56

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015). Going deeper with convolutions, *Proceedings of the 28th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9. 41

Szeto, W. Y., Augoustides, J. G., Desai, N. D., Moeller, P., McGarvey, M. L., Walsh, E., Bannan, A., Herrmann, H. C. and Bavaria, J. E. (2011). Cerebral Embolic Exposure During Transfemoral and Transapical Transcatheter Aortic Valve Replacement, *Journal of Cardiac Surgery* **26**(4): 348–354.
URL: *https://onlinelibrary.wiley.com/doi/10.1111/j.1540-8191.2011.01265.x* 14

Tafsast, A., Ferroudji, K., Hadjili, M. L., Bouakaz, A. and Benoudjit, N. (2018). Automatic microemboli characterization using convolutional neural networks and radio frequency signals, *Proceedings of the International Conference on Communications and Electrical Engineering (ICCEE)*, IEEE, pp. 1–4.
URL: *https://ieeexplore.ieee.org/document/8634521/* 31, 32, 95

Telman, G., Kouperberg, E., Sprecher, E., Goldsher, D. and Yarnitsky, D. (2005). Distribution of artificial cerebral microemboli in stroke patients with patent foramen ovale, *Neurological Research* **27**(1): 109–111. 14

Tenenbaum, J. B., de Silva, V. and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction, *Science (New York, N.Y.)* **290**(5500): 2319–2323. 68

Tiong, L., Kim, S. T. and Ro, Y. (2019). Implementation of multimodal biometric recognition via multi-feature deep learning networks and feature fusion, *Multimedia Tools and Applications* **78**. 50

Tjandra, A., Liu, C., Zhang, F., Zhang, X., Wang, Y., Synnaeve, G., Nakamura, S. and Zweig, G. (2020). Deja-vu: Double feature presentation and iterated loss in deep transformer networks, *Proceedings of the 45 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6899–6903. 46

Tsao, C. W., Aday, A. W., Almarzooq, Z. I., Alonso, A., Beaton, A. Z., Bittencourt, M. S., Boehme, A. K., Buxton, A. E., Carson, A. P., Commodore-Mensah, Y., Elkind, M. S. V., Evenson, K. R., Eze-Nliam, C., Ferguson, J. F., Generoso, G., Ho, J. E., Kalani, R., Khan, S. S., Kissela, B. M., Knutson, K. L., Levine, D. A., Lewis, T. T., Liu, J., Loop, M. S., Ma, J., Mussolino, M. E., Navaneethan, S. D., Perak, A. M., Poudel, R., Rezk-Hanna, M., Roth, G. A., Schroeder, E. B., Shah, S. H., Thacker, E. L., VanWagner, L. B., Virani, S. S., Voecks, J. H., Wang, N. Y., Yaffe, K. and Martin, S. S. (2022). Heart Disease and Stroke Statistics-2022 Update: A Report From the American Heart Association, *Circulation* **145**(8): e153–e639. 13

Tung, F. and Mori, G. (2020). Deep neural network compression by in-parallel pruning-quantization, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **42**(3): 568–579. 60, 127

Types of Stroke (2022).
URL: *https://www.hopkinsmedicine.org/health/conditions-and-diseases/stroke/types-of-stroke* 6

Ullrich, K., Meeds, E. and Welling, M. (2017). Soft weight-sharing for neural network compression, *Proceedings of the International Conference on Learning Representations*.
URL: *https://openreview.net/forum?id=HJGwcKclx* 56, 60

Uğuz, H. (2012). A hybrid system based on information gain and principal component analysis for the classification of transcranial Doppler signals, *Computer Methods and Programs in Biomedicine* **107**(3): 598–609.
URL: *https://www.sciencedirect.com/science/article/pii/S0169260711000782* 30

Van Mieghem, N. M., El Faquir, N., Rahhab, Z., Rodríguez-Olivares, R., Wilschut, J., Ouhlous, M., Galema, T. W., Geleijnse, M. L., Kappetein, A.-P., Schipper, M. E. and de Jaegere, P. P. (2015). Incidence and Predictors of Debris Embolizing to the Brain During Transcatheter Aortic Valve Implantation, *JACC: Cardiovascular Interventions* **8**(5): 718–724.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S1936879815002277* 14

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u. and Polosukhin, I. (2017). Attention is all you need, *in* I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett (eds), *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Vol. 30, Curran Associates, Inc.
URL: *https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf* 43, 44

Viedma-Guiard, E., Guidoux, C., Amarenco, P. and Meseguer, E. (2021). Aortic Sources of Embolism, *Frontiers in Neurology* **11**: 606663.
URL: *https://www.frontiersin.org/articles/10.3389/fneur.2020.606663/full* 13

Viguier, A., Pavy le Traon, A., Massabuau, P., Valton, L. and Larrue, V. (2001). Asymptomatic cerebral embolic signals in patients with acute cerebral ischaemia and severe aortic arch atherosclerosis, *Journal of Neurology* **248**(9): 768–771.
URL: *https://doi.org/10.1007/s004150170092* 13

Vindas, Y., Guepie, B. K., Almar, M., Roux, E. and Delachartre, P. (2022). An hybrid cnn-transformer model based on multi-feature extraction and attention fusion mechanism for cerebral emboli classification, *in* Z. Lipton, R. Ranganath, M. Sendak, M. Sjoding and S. Yeung (eds), *Proceedings of the 7th Machine Learning for Healthcare Conference*, Vol. 182 of *Proceedings of Machine Learning Research*, PMLR, pp. 270–296.
URL: *https://proceedings.mlr.press/v182/vindas22a.html* 95, 106

Vindas, Y., Guépié, B. K., Almar, M., Roux, E. and Delachartre, P. (2022). Semi-automatic data annotation based on feature-space projection and local quality metrics: an application to cerebral emboli characterization, *Medical Image Analysis* p. 102437. 63

Vindas, Y., Guépié, B. K., Almar, M., Roux, E. and Delachartre, P. (2023). Soft-labels noise tolerant loss functions for transcranial Doppler ultrasound signal classification, *2023 IEEE International Ultrasonics Symposium (IUS)*, pp. 1–4. 151

Vindas, Y., Roux, E., Guépié, B. K., Almar, M. and Delachartre, P. (2021). Semi-supervised annotation of transcranial Doppler ultrasound micro-embolic data, *2021 IEEE International Ultrasonics Symposium (IUS)*, pp. 1–4. 63

Vindas, Y., Roux, E., Guépié, B. K., Almar, M. and Delachartre, P. (2023a). Deep embedded clustering regularization for supervised imbalanced cerebral emboli classification using transcranial Doppler ultrasound, *Proceedings of the 31st European Signal Processing Conference (EUSIPCO)*, pp. 1165–1169. 95, 115, 122, 124

Vindas, Y., Roux, E., Guépié, B. K., Almar, M. and Delachartre, P. (2023b). Guided deep embedded clustering regularization for multifeature medical signal classification. Submitted to the Pattern Recognition (2023), minor revision. 95

Wang, L., Zhang, J., Liu, P., Choo, K.-K. R. and Huang, F. (2017). Spectral–spatial multi-feature-based deep learning for hyperspectral remote sensing image classification, *Soft Computing* **21**. 50

Wang, R., Liu, T. and Tao, D. (2018). Multiclass learning with partially corrupted labels, *IEEE Transactions on Neural Networks and Learning Systems* **29**(6): 2568–2580.
URL: *https://ieeexplore.ieee.org/document/7929355/* 39

Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J. and Bailey, J. (2019). Symmetric cross entropy for robust learning with noisy labels, *Proceedings of the 17th IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 322–330. 40, 92

Wasimuddin, M., Elleithy, K., Abuzneid, A.-S., Faezipour, M. and Abuzaghleh, O. (2020). Stages-based ECG signal analysis from traditional signal processing to machine learning approaches: A survey, *IEEE Access* **8**: 177782–177803. 42

Weir, A. (2016). Channel characterisation and modelling for transcranial Doppler ultrasound, Heriot-Watt University.
URL: *https://books.google.fr/books?id=5MVkwAEACAAJ* 17

Weston, J., Ratle, F., Mobahi, H. and Collobert, R. (2012). Deep learning via semi-supervised embedding, *in* G. Montavon, G. B. Orr and K.-R. Müller (eds), *Neural Networks: Tricks of the Trade: Second Edition*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 639–655.
URL: *https://doi.org/10.1007/978-3-642-35289-8_34* 33, 34, 63

Xiao, T., Xia, T., Yang, Y., Huang, C. and Wang, X. (2015). Learning from massive noisy labeled data for image classification, *Proceedings of the 28th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2691–2699. 38

Xie, J., Girshick, R. and Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis, *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, ICML'16, JMLR.org, p. 478–487. 47, 48, 122, 124

Xu, G., Ren, T., Chen, Y. and Che, W. (2020). A one-dimensional cnn-lstm model for epileptic seizure recognition using EEG signal analysis, *Frontiers in Neuroscience* **14**. 106, 107, 109, 110, 111, 112, 117

Xu, K., Zhang, D., An, J., Liu, L., Liu, L. and Wang, D. (2021). Genexp: Multi-objective pruning for deep neural network based on genetic algorithm, *Neurocomputing* **451**: 81–94.
URL: *https://www.sciencedirect.com/science/article/pii/S092523122100549X* 54

Xu, Y., Wang, Y., Zhou, A., Lin, W. and Xiong, H. (2018). Deep neural network compression with single and multiple level quantization, *Proceedings of the 32nd AAAI Conference on Artificial Intelligence and 30th Innovative Applications of Artificial Intelligence Conference and 8th AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18, AAAI Press. 56, 59

Yang, J., Shen, X., Xing, J., Tian, X., Li, H., Deng, B., Huang, J. and Hua, X.-s. (2019). Quantization networks, *Proceedings of the 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 58

Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H. and Ni, B. (2021). Medmnist v2: A large-scale lightweight benchmark for 2d and 3d biomedical image classification, *arXiv preprint arXiv:2110.14795* . 72

Yao, T., Gao, F., Zhang, Q. and Ma, Y. (2021). Multi-feature gait recognition with dnn based on semg signals, *Mathematical Biosciences and Engineering* **18**: 3521–3542. 51

Yao, Z., Dong, Z., Zheng, Z., Gholami, A., Yu, J., Tan, E., Wang, L., Huang, Q., Wang, Y., Mahoney, M. and Keutzer, K. (2021). Hawq-v3: Dyadic neural network quantization, *in* M. Meila and T. Zhang (eds), *Proceedings of the 38th International Conference on Machine Learning*, Vol. 139 of *Proceedings of Machine Learning Research*, PMLR, pp. 11875–11886.
URL: *https://proceedings.mlr.press/v139/yao21a.html* 53, 58, 59

Yeh, C.-F., Mahadeokar, J., Kalgaonkar, K., Wang, Y., Le, D., Jain, M., Schubert, K., Fuegen, C. and Seltzer, M. L. (2019). Transformer-transducer: End-to-end speech recognition with self-attention, *ArXiv* **abs/1910.12977**. 41, 42, 46

Yin, P., Lyu, J., Zhang, S., Osher, S. J., Qi, Y. and Xin, J. (2019). Understanding straight-through estimator in training activation quantized neural nets, *Proceedings of the International Conference on Learning Representations*.
URL: *https://openreview.net/forum?id=Skh4jRcKQ* 56

Yu, F. and Koltun, V. (2016). Multi-scale context aggregation by dilated convolutions, *in* Y. Bengio and Y. LeCun (eds), *Proceedings of the 4th International Conference on Learning Representations, ICLR*.
URL: *http://arxiv.org/abs/1511.07122* 41, 52

Zafrir, O., Boudoukh, G., Izsak, P. and Wasserblat, M. (2019). Q8BERT: quantized 8bit BERT, *Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition*, IEEE, pp. 36–39.
URL: *https://doi.org/10.1109/EMC2-NIPS53020.2019.00016* 58

Zavala, J. A., Amarrenco, P., Davis, S. M., Jones, E. F., Young, D., Macleod, M. R., Horky, L. L. and Donnan, G. A. (2006). Aortic arch atheroma, *International Journal of Stroke* **1**(2): 74–80. 13

Zhang, D., Yang, J., Ye, D. and Hua, G. (2018). LQ-Nets: Learned quantization for highly accurate and compact deep neural networks, *Proceedings of the 15th European Conference on Computer Vision*, Springer-Verlag, Berlin, Heidelberg, p. 373–390.
URL: *https://doi.org/10.1007/978-3-030-01237-3_23* 56

Zhang, H., Cisse, M., Dauphin, Y. N. and Lopez-Paz, D. (2018). Mixup: Beyond empirical risk minimization, *Proceedings of the 6th International Conference on Learning Representations, ICLR*.
URL: *https://openreview.net/forum?id=r1Ddp1-Rb* 38, 39

Zhang, R., Isola, P., Efros, A. A., Shechtman, E. and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric, *Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 586–595.
URL: *https://ieeexplore.ieee.org/document/8578166/* 34

Zhang, W., Hou, L., Yin, Y., Shang, L., Chen, X., Jiang, X. and Liu, Q. (2020). Ternarybert: Distillation-aware ultra-low bit bert, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 56, 59

Zhang, Z. and Sabuncu, M. R. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels, *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, Curran Associates Inc., Red Hook, NY, USA, p. 8792–8802. 40, 72

Zhou, S., Ni, Z., Zhou, X., Wen, H., Wu, Y. and Zou, Y. (2016). DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients, *arXiv e-prints* **abs/1606.06160**.
URL: *http://arxiv.org/abs/1606.06160* 56, 59

Zhu, C., Han, S., Mao, H. and Dally, W. J. (2017). Trained ternary quantization, *Proceedings of the 5th International Conference on Learning Representations*.
URL: *https://openreview.net/forum?id=S1_pAu9xl* 56, 59, 60, 127, 129, 132, 139

Zhu, M. and Gupta, S. (2018). To prune, or not to prune: Exploring the efficacy of pruning for model compression, *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*, OpenReview.net.
URL: *https://openreview.net/forum?id=Sy1iIDkPM* 54

Zhu, X. and Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation, *Technical Report CMU-CALD-02-107*, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. 33, 34, 63, 69, 70

Zhu, Y. and Jiang, Y. (2020). Optimization of face recognition algorithm based on deep learning multi feature fusion driven by big data, *Image and Vision Computing* **104**: 104023.
URL: *https://www.sciencedirect.com/science/article/pii/S0262885620301554* 50

Publications and conferences participation

Journal publications

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 79 (2022) 102437. URL: https://doi.org/10.1016/j.media.2022.102437

Y. Vindas, E. Roux, B. K. Guépié, M. Almar and P. Delachartre, Guided deep embedded clustering regularization for multifeature medical signal classification, Pattern Recognition (2023), Pattern Recognition, vol. 143, novembre 2023, p. 109812. ScienceDirect, https://doi.org/10.1016/j.patcog.2023.109812.

Conference participation

With proceeding

Y. Vindas, E. Roux, B. K. Guépié, M. Almar and P. Delachartre, "Semi-supervised annotation of Transcranial Doppler ultrasound micro-embolic data," 2021 IEEE International Ultrasonics Symposium (IUS), Xi'an, China, 2021, pp. 1-4, doi: 10.1109/IUS52206.2021.9593847. URL: https://ieeexplore.ieee.org/document/9593847

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: Z. Lipton,595R. Ranganath, M. Sendak, M. Sjoding, S. Yeung (Eds.), Proceedings of the 7th Machine Learning for Healthcare Conference, volume 182 of *Proceedings of Machine Learning Research*, PMLR, 2022, pp. 270–296. URL: https://proceedings.mlr.press/v182/vindas22a.html

Y. Vindas, E. Roux, B. K. Guépié, M. Almar and P. Delachartre. Deep Embedded Clustering regularization for supervised imbalanced cerebral emboli classification using transcranial Doppler ultrasound. *2023 31th European Signal Processing Conference (EUSIPCO)*.

Y. Vindas, B. K. Guépié, M. Almar, E. Roux, P. Delachartre, "Soft-labels noise tolerant loss functions for transcranial Doppler ultrasound signal classification,". 2023 IEEE International Ultrasonics Symposium (IUS), Montreal, Canada, 2023.

Without proceeding

Y. Vindas, B. K. Guépié, M. Almar, E. Roux, P. Delachartre. Classification multi-representation d'emboles cerebraux a partir d'un dispositif de Doppler transcranien. *1er Colloque Français d'Intelligence Artificielle en Imagerie Biomédicale (IABM 2023)*.

FOLIO ADMINISTRATIF

THESE DE L'UNIVERSITE DE LYON OPEREE AU SEIN DE L'INSA LYON

NOM : VINDAS-YASSINE DATE de SOUTENANCE : 12/10/2023
(avec précision du nom de jeune fille, le cas échéant)

Prénoms : Yamil Emilio

TITRE : Weakly-supervised learning for emboli characterization with Transcranial Doppler (TCD) monitoring

NATURE : Doctorat Numéro d'ordre : 2023ISAL0064

Ecole doctorale : N°160 Électronique, Électrotechnique, Automatique

Spécialité : Traitement du Signal et de l'Image

RESUME : This thesis focuses on the classification and characterization of high intensity transient signals coming from portable transcranial Doppler (TCD) ultrasound devices. The main objective is to help clinicians identify solid and gaseous emboli from artifacts generated during TCD monitoring sessions. In fact, emboli are solid or gaseous particles that can circulate in the cerebral arteries, sometimes blocking them and causing ischemic stroke. However, the identification and classification of HITS between solid embolus, gaseous embolus, and artifacts is not evident and require important expert knowledge. Because of this, even clinicians can have some trouble differentiating these different types of HITS, which makes the treatment of patients difficult. Therefore, its detection, classification and characterization are key factors to improve patient management in healthcare centers.
In this work, we propose deep learning models capable of doing an accurate classification between solid embolus, gaseous embolus, and artifacts, with limited memory and energy consumption. Our approach combines semi-automatic data annotation, with multi-feature learning and model compression techniques. We evaluate the different components of our approach using several medical *in vivo* datasets, besides HITS classification.
Our method proved to be effective, with great classification results, and low memory and energy consumption on several medical signal classification tasks. More precisely, for HITS classification, to our knowledge, our work is the only one proposing an *in vivo* classification of portable TCD HITS between solid embolus, gaseous embolus, and artifacts.
The main contributions of this work are the following. Firstly, we proposed a semi-automatic data annotation method based on local quality metrics with controlled annotation error, allowing to quickly label a large dataset, using a small number of labeled samples. Secondly, we propose a hybrid guided and regularized multi-feature classification model allowing to accurately classify HITS, simultaneously taking advantage of the raw Doppler signal, and its time-frequency representation. Finally, we proposed new model compression techniques based on pruning and extreme quantization, allowing to reduce the memory requirements of the trained models, as well as the energy consumption.
Finally, as we worked in close cooperation with Atys Medical, manufacturer of portable TCD devices, we were able to incorporate our developed models into their data management software. Even though validation is still needed, we hope that the models and methods developed in this work can help clinicians with their patient management.

MOTS-CLÉS : Doppler transcrânien, Imagerie médicale, apprentissage profond, traitement du signal, apprentissage semi-supervisé, annotation semi-automatique, classification, compression de modèles

Laboratoire (s) de recherche : Centre de Recherche en Acquisition et Traitement de l'Image pour la Santé (CREATIS)

Directeur de thèse : Philippe Delachartre

Président de jury : Danilo MANDIC

Composition du jury :
- Su RUAN
- François ROUSSEAU
- Danilo MANDIC
- Philippe DELACHARTRE
- Emmanuel ROUX
- Blaise Kévin GUÉPIÉ