CITY, UNIVERSITY OF LONDON

DOCTORAL THESIS

Enhancing the Generalization of Convolutional Neural Networks for Speech Emotion Recognition

---

Author:

Eric GUIZZO

Supervisor:

Dr. Tillman WEYDE

A thesis submitted in fulfillment of the requirements

for the degree of Doctor of Philosophy

in the

Department of Computer Science

December 6, 2023

Declaration of Authorship

I, Eric GUIZZO, hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text. This dissertation contains less than 65,000 words including appendices, bibliography, footnotes, tables and equations and has less than 150 figures. I grant powers of discretion to the City, University of London librarian to allow the dissertation to be copied in whole or in part without further reference to myself (the author). This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgment

Eric Guizzo

December 6, 2023

Abstract

Human-machine interaction is rapidly gaining significance in our daily lives. While speech recognition has achieved near-human performance in recent years, the intricate details embedded in speech extend beyond the mere arrangement of words. Speech Emotion Recognition (SER) is therefore acquiring a growing role in this field by decoding not only the linguistic content but also the emotional nuances of human spoken communication and enabling therefore a more exhaustive comprehension of the information conveyed by speech signals.

Despite the success that neural networks have already achieved in this task, SER is still challenging due to the variability of emotional expression, especially in real-world scenarios where generalization to unseen speakers and contexts is required. In addition, the high resource demand of SER models, combined with the scarcity of emotion-labelled data, hinder the development and application of effective solutions in this field.

In this thesis, we present multiple approaches to overcome the aforementioned difficulties. We first introduce a multiple-time-scale (MTS) convolutional neural network architecture to create flexibility towards temporal variations when analyzing time-frequency representations of audio data. We show that resilience to speed fluctuations is relevant in SER tasks, since emotion is expressed through complex spectral patterns that can exhibit significant local dilation and compression on the time axis depending on speaker and context. The results indicate that the use of MTS consistently improves

the generalization of networks of different capacity and depth, compared to standard convolution.

In a second stage, we propose a more general approach to discourage unwanted sensitivity towards specific target properties in CNNs, introducing the novel concept of anti-transfer learning. While transfer learning assumes that the learning process for a target task will benefit from re-using representations learned for another task, anti-transfer avoids the learning of representations that have been learned for an orthogonal task, i.e., one that is not relevant and potentially confounding for the target task, such as speaker identity and speech content for emotion recognition. In anti-transfer learning we penalize similarity between activations of a network being trained and another network previously trained on an orthogonal task. This leads to better generalization and provides a degree of control over correlations that are spurious or undesirable. We show that anti-transfer actually leads to the intended invariance to the orthogonal task and to more appropriate feature maps for the target task at hand. Anti-transfer creates a computation and memory cost at training time, but it enables enables the reuse of pre-trained models.

In order to avoid the high resource demand of SER models in general and anti-transfer learning specifically, we propose RH-emo, a novel semi-supervised architecture aimed at extracting quaternion embeddings from real-valued monoaural spectrograms, enabling the use of quaternion-valued networks for SER tasks. RH-emo is a hybrid real/quaternion autoencoder network that consists of a real-valued encoder in parallel to a real-valued emotion classifier and a quaternion-valued decoder. We show that the use of RH-emo, combined with quaternion convolutional neural networks provides a consistent improvement in SER tasks, while requiring far fewer trainable parameters and therefore substantially reducing the resource demand of SER models.

Finally, we apply anti-transfer learning to quaternion-valued neural networks fed with RH-emo embeddings. We demonstrate that the combination of the two approaches maintains the disentanglement properties of anti-transfer, while using a reduced amount of memory, computation, and training time, making this a suitable approach for SER scenarios with limited resources and where context and speaker independence are needed.

Acknowledgements

Throughout my PhD years I have received great support and assistance from many wonderful persons and this dissertation would not have been possible without them.

I would first like to express my deep gratitude to my supervisor, Doctor Tillman Weyde, for putting your faith in me from the beginning, despite my unusual background. Your advice, both as a researcher and as a person, was invaluable to sharpen my thinking and brought my work and mindset to a higher level. In other words, you have made me a researcher.

I would like to thank my parents and my family for being always there, for their encouragement and reassurance. Your constant presence has been fundamental for me to face this PhD journey far from home. Thank you Paola, Giuseppe and Simone.

Thanks also to all my friends, lab mates, colleagues, and research team for the great times spent at City University and in London together. Your support and friendship made me feel at home and made the PhD challenge way easier. Thanks to Radha, Fatemeh, Francesco, Alessandro, Alex, Olga, Nadine, Charitos, Sarah, Benedikt, Mirela, Nico, Fosco, Banfi, Munin, Gino and Luigi.

I would also like to extend my warm regards to all the faculty in the Department of Computer Science at City for their help over the past three years. Thanks to Edoardo Alonso for reviewing my MPhil to PhD transfer report. Many thanks to the course officers especially Ann Marie De Here, Savita Afonso, David Mallo-Ferrer, Nathalie Chatelain, and staff members

Paula Green, Asif Nawaz, Gabrielle for all their valuable help on numerous occasions.

Finally, a very special thank goes my girlfriend Alice, whose love has always been the best encouragement and relief. Thanks for listening to and improving my ideas, for supporting me in the good and the bad times, for playing music with me and for making difficulties less of a challenge.

Contents

List of Figures

List of Tables

List of Abbreviations

ADAM	**Ada**ptive **M**oment Estimation
AI	**A**rtificial **I**ntelligence
ANN	**A**rtificial **N**eural **N**etwork
AT	**A**nti **T**ransfer
CCC	**C**concordance **C**orrelation **C**oefficient
CNN	**C**onvolutional **N**eural **N**etwork
DAT	**D**omain **A**dversarial **T**raining
EMODB	A **Data**base for german **Emo**tional Speech
FSDD	**F**ree **S**poken **D**igits **D**ataset
GRU	**G**ated **R**ecurrent **U**nit
GS	**G**ood **S**ounds Dataset
GSC	**G**oogle **S**peech **C**ommands
IEMOCAP	The **I**nteractive **Emo**tional Dyadic **Cap**ture Database
LSTM	**L**ong **S**hort **T**erm **M**emory
MHE	**M**inimum **H**yperspherical **E**nergy
MSE	**M**ean **S**quared **E**rror
MSSNSD	The **M**icrosoft **S**calable **N**oisy **S**peech of **D**ataset
MTS	**M**ulti **T**ime **S**cale
NLP	**N**atural **L**anguage **P**rocessing
PCA	**P**rincipal **C**omponent **A**nalysis
PP	**P**ercentage **P**oints
QCNN	**Q**uaternion **C**onvolutional **N**eural **N**etwork
R2He	**R**eal **2** **H**amilton space **E**ncoder
R2Hae	**R**eal **2** **H**amilton space **A**uto**e**ncoder
RAVDESS	The **R**yerson **A**udio **V**isual **D**atabase of **E**motional **S**peech and **S**ong
ReLU	**Re**ctified **L**inear **U**nit
ResNet	**Res**idual **Net**work
RH-emo	**R**eal to **E**motional **H**amilton space Autoencoder
SER	**S**peech **E**motion **R**ecognition
SGE	**S**ound **G**oodness **E**stimation
STFT	**S**hort **T**time **F**ourier **T**ransform
TESS	**T**oronto **E**motional **S**peech **S**et
VAD	**V**alence **A**rousal and **D**ominance space
VAE	**V**ariational **A**uto**e**ncoder
VGG	**V**isual **G**eometry **G**roup
WI	**W**eight **I**nitialization
WR	**W**ord **R**ecognition

Dedicated to my place: Vidor

Chapter 1

Introduction

1.1 Motivation

The sound of the human voice plays an important role in human communication, since it is able to carry information that goes beyond the meaning of the sequence of uttered words. The understanding of emotion is essential for a complete and correct interpretation of a speech signal, since it gives information about the feelings and potentially intentions of a speaker. In certain circumstances the emotional content of speech can even invert the literal meaning of a sentence. This happens for example when a sentence is pronounced with sarcasm. The study of Mehrabian and Wiener (1967) suggests that the amount of non-verbal information conveyed by speech is largely higher than the information carried by the mere words' sequence. Their specific approach quantifies the attributed information content of speech to the spoken words as 7%, while the non-verbal vocal expression conveys 38% and the facial expression 55%. These findings pertain to a specific setup in which the authors examined how recipients interpret incongruous emotional cues encompassing visual, lexical, and vocal aspects. While the experiment's scope is limited, and the aforementioned metrics shouldn't be regarded as definitive or dependable measurements, this research underscores the significance of acknowledging the influence of non-verbal communication. Nevertheless, emotion interpretation can be ambiguous even for humans, since

it strongly depends, among other factors, on the context, on the speech style of a person and on her/his cultural background (Scherer, 2003).

Human-machine interaction is becoming increasingly important in our everyday life and in this field a correct interpretation of human speech is fundamental. Research on automatic speech recognition reached near-human performance in recent years, making it possible to confidently identifying which words are uttered even in audio signals with non-optimal recording quality and for an extensive amount of different languages (Malik et al., 2021; Nassif et al., 2019). Nevertheless, this information without the understanding of speaker's non-verbal behavior is incomplete, and a correct interpretation of the emotional intention helps provide a more comprehensive account of the information conveyed by spoken communication (Mehrabian and Wiener, 1967). Speech Emotion Recognition (SER) is therefore acquiring a growing role in human-machine interaction, complementing research in automatic speech recognition and natural language processing.

1.2 Problem Overview and Research Objectives

Despite the impressive success that neural networks have achieved in this task, SER is still challenging due to the variability of emotional expression, especially in real-world scenarios where generalization to unseen speakers and contexts is required (Rybka and Janicki, 2013a; Hozjan and Kačič, 2003; Issa, Demirci, and Yazici, 2020; Wang et al., 2020; Akçay and Oğuz, 2020).

In context-independent scenarios, the generalization capabilities of SER models can be harmed by the presence of a different vocabulary in training, validation and test sets. This is particularly evident when only a limited set of words or sentences is recorded. In this context, the features learned by a network can be overly specialized to the specific words present in the training set. Therefore, even though the learned features can be effective for

emotion estimation, they can be inappropriate with different words or sentences. On the other hand, in speaker-independent contexts a SER estimator can overly specialize to the specific timbre or speech style of a person present in the training set. In this case, models can overly adapt to the way a specific person expresses emotions, for instance to intonation fluctuations, to global speech volume, and to the general character of his/her voice. This encourages models to fail recognizing equivalent emotions expressed by different individuals. Another obstacle to SER is the general scarcity of emotionally-labelled audio data, which is due to the high cost of the recording and labelling such data. Moreover, a further well-known difficulty is that emotional information in speech involves long-term temporal dependencies that are in the order of seconds (Rigoulot, Wassiliwizky, and Pell, 2013; Khorram et al., 2017; Lian et al., 2019). This forces models to analyze large temporal windows and, consequently, to use a large amount of resources.

In this thesis, we investigate solutions to the most influential and common issues related to SER: the difficulty of extracting emotion-related representations that are disentangled from other uncorrelated and potentially misleading features (as speech style and context), and the high resource demand of SER models. Although different types of data can be used to perform Emotion Recognition in general (among others, speech audio signals, electroencephalogram signals, face and body video recordings, text transcription), in this work we specifically focus on audio signals.

1.3 Research Questions

The primary focus of this research work addresses the following research questions:

- What are the major difficulties that neural networks exhibit when performing emotion recognition from audio speech signals?

We provide an answer to this question in Chapter 2 and 3. We identified the main SER challenges as the scarce generalization and high resource demand of models.

- Can a neural network learn emotion-related representations that are disentangled from the the context and from the speech style of a person? If yes, how?

  We respond to this question with the work presented in Chapter 4, 5 and 7. We show that it is possible to achieve such disentanglement in two different ways: enforcing time scale invariance and promoting independence from unwanted features with anti-transfer learning.

- Is it possible to inject invariance towards a specific feature in a neural network? If yes, could this improve the independence of learnt emotion-related representations from the context and from the speech style of a person?

  We address this question in Chapter 5, showing that that this behavior can be achieved through anti-transfer learning. We demonstrate that this technique can promote the independence of the learnt representations from the aforementioned confounding factors.

- How can we reduce the high amount of computational resources needed when performing Speech Emotion Recognition with neural networks?

  We deal with this question in Chapters 6 and 7. We show that the use of quaternion-valued neural networks (based on quaternion algebra, an extension of complex algebra) can reduce the resource demand of models while maintaining or improving their classification performance.

1.4 Main Contributions

In this research work, we present different approaches to overcome the aforementioned difficulties related to SER.

We propose specific solutions to achieve independence from context (i.e., which words are pronounced by a speaker) and from the speech style of a person when performing SER with neural networks. The first is achieved through a recurrent neural network design and the second relies on a custom convolution layer, called Multi-Time-Scale, that extracts features at multiple time scales, consequently improving invariance towards timing fluctuations of speech. As a more general solution to improve the invariance of the learnt representations from irrelevant features, we propose a novel learning paradigm, that we call anti-transfer learning, through which it is possible to diverge the features learnt by a network from the ones extracted by another network that is pretranied on a different task. This helps to learn emotion-related features that are independent from the context and speech style. Moreover, we propose a solution to overcome the high resource demand of models through the combination of quaternion information processing and a custom features extractor, called RH-emo, that maps real-valued spectrograms into the quaternion domain while optimizing the quaternion representation for SER tasks.

Lastly, we apply the anti-transfer learning paradigm to quaternion-valued neural networks fed with RH-emo embeddings, merging the disentanglement properties of the first and the resource saving capabilities of the latter.

The contributions of our work are specifically:

- We propose a convolution layer design for SER from audio signals that learns locally-scale-invariant features in the time dimension.

- For the first time, to the best of our knowledge, we introduce the concept of *anti-transfer* learning to achieve task-invariance between a pre-trained network and a new one.
- We define a novel method, RH-emo, that draws quaternion-valued embeddings from speech signals, where each quaternion component is tailored to a specific emotional characteristic.
- We apply anti-transfer learning to quaternion-valued neural networks, preserving the disentanglement properties of anti-transfer while optimizing its resource demand.
- We apply our approaches to multiple popular emotion-labelled speech datasets and with different neural network architectures, providing extensive analysis of the experimental results that confirm the effectiveness of the proposed methods.
- We provide open source code for all developed approaches[1] [2] [3]

1.5 Publications

1.5.1 Journal Papers

1. Eric Guizzo, Tillman Weyde, Simone Scardapane and Danilo Comminiello (2023). **'Learning Speech Emotion Representations in the Quaternion Domain'**, IEEE/ACM Transactions on Audio, Speech, and Language Processing 31: pp. 1200-1212.
2. Eric Guizzo, Tillman Weyde, Giacomo Tarroni (2021). **'Anti-Transfer Learning for Task Invariance in Convolutional Neural Networks for Speech Processing'**, Neural Networks 142: pp. 238-251.

[1] `https://github.com/ericguizzo/multi`
[2] `https://github.com/ericguizzo/anti_transfer`
[3] `https://github.com/ispamm/rhemo`

1.5.2 Conference Papers

1. Eric Guizzo, Tillman Weyde, Giacomo Tarroni, Danilo Comminiello (2023), **'Quaternion Anti-Transfer Learning for Speech Emotion Recognition'**, IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA). .

2. Eric Guizzo, Tillman Weyde, Jack Barnett Leveson (2020), **'Multi-Time-Scale Convolution for Emotion Recognition from Speech Audio Signals'**, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6489-6493.

3. Francesco Barbieri, Eric Guizzo, Federico Lucchesi, Giovanni Maffei, Fermin Moscoso del Prado Martin and Tillman Weyde (2019), **'Towards a Multimodal Time-Based Empathy Prediction System'**, IEEE International Conference of Automatic Face and Gesture Recognition. The work described in this paper won the OMG-Empathy Challenge 2018.

1.5.3 Code

The following repositories contain the code that reproduces the approaches presented in this thesis:

- `https://github.com/ericguizzo/multi_time_scale`
- `https://github.com/ericguizzo/anti_transfer`
- `https://github.com/ericguizzo/transferable_quaternion_embeddings`

1.6 Organization of the Thesis

Parts of this thesis are re-adapted from the papers listed in Section 1.5. This thesis is organized as follows:

- *Chapter 1: Introduction* provides an overview of the thesis, covering motivation behind the work, problems and research objectives, and the main contributions of the thesis. Publications and Code details are also presented.

- *Chapter 2: Literature Review* merges and extends the literature review sections of all papers listed in Chapter 1.5 and is divided in 4 sections. The first describes general strategies to perform SER and how datasets are configured. The second is dedicated to scale-invariance. The third regards general approaches and concepts that are influential for supervised task-specific invariance and the latter describes methods to improve SER models' efficiency, including quaternion audio representations and processing.

- *Chapter 3: Preliminary Study on a Context-Independent Scenario* is adapted from Conference Paper 2, with additional results and discussion that are not included in the original work. This Chapter contains preliminary collaborative work in which we identified the main research questions and motivations discussed in this dissertation. Here we introduce an approach to perform SER by predicting context-independent continuous time-varying emotional features. Specifically, we adopt an ensemble of different architectures, each specific to a different input domain (audio, video and face/body landmarks).

- *Chapter 4: Increasing Speaker-Independence: Multi-Time-Scale Modelling* is adapted from Conference Paper 3, with additional results and discussion that are not included in the original work. The Chapter describes a general method to achieve scale-invariance in CNNs, which in turn improves the independence of the learnt emotional features from the

speech style of a speaker. This approach is based on the Multi-Time-Scale convolution layer, that is a custom convolution layer that simultaneously extracts audio features at multiple time scales.

- *Chapter 5: A Supervised Method for Task-Specific Invariance: Anti-Transfer Learning* is adapted from Journal Paper 2, with additional discussion that is not included in the original work. This Chapter describes a method to supervisely disentangle the features learnt by a CNN from the features that a different pretrained CNN extracts. This approach permits to selectively improve the independence of SER models' predictions from any specific feature, including speech style and context.

- *Chapter 6: Improving efficiency: Speech Emotion Representations in the Quaternion Domain* is adapted from Journal Paper 1, with additional discussion that is not included in the original work. This chapter introduces a solution to overcome one of the most common problems related to SER: the high resource demand of SER models. This is achieved through the use quaternion-valued CNNs fed with custom quaternion embeddings extracted directly from the audio signals.

- *Chapter 7: Anti-transfer Learning in the Quaternion Domain* merges and extends the approaches presented in Chapter 5 and 6. We applied the concept of anti-transfer learning to quaternion-valued networks, merging the convenient properties of quaternion information processing with the task-specific disentanglement that anti-transfer learning provides. In this Chapter we present results and discussion that are not included in Conference Paper 1.

- *Chapter 8: Conclusions and Future Work* has the final remarks and summarizes about the problems addressed in the thesis and the proposed

approaches, followed by the future scope of work. All references are provided after the Conclusions.

- *Appendix A: Quaternion Convolutional Neural Networks* introduces the principles of quaternion algebra, quaternion Neural Networks and, more specifically, Quaternion CNNs.

- *Appendix B: Example Conversation from the OMG Empathy Dataset* includes a reference transcription of a story extracted from the dataset we adopted in Chapter 3.

Chapter 2

Literature Review

This Literature Review merges and extends all Literature Review sections of the Papers listed in Chapter 1.5 and is divided into four sections.

First, state-of-the-art strategies to perform SER are discussed and the general structure of emotion-labelled speech datasets is outlined. In the second part, we summarize existing approaches to achieve speaker and context invariance in SER models. After this, we focus on presenting methods and concepts related to supervised task-specific invariance. In the fourth part we present existing studies and techniques aimed at limiting the resource demand of SER models. This Literature Review responds to the first Research Question, which in turn is the foundation for the following studies presented in this thesis.

2.1 Strategies for Speech Emotion Recognition

A traditional approach to SER is based on two consecutive stages: hard-coded extraction of affect-related features followed by a learning-based classification or regression. Various combinations of features and machine learning models have been proposed. The most commonly used features are: base pitch, formant features, energy/spectral features, and prosody. A wide variety of learning models has been proposed: Artificial Neural Networks (Bhatti, Wang, and Guan, 2004; Cowie et al., 2001; Nicholson, Takahashi,

and Nakatsu, 2000), Bayesian Networks, (Ververidis and Kotropoulos, 2008), Hidden Markov Models (Mao, Chen, and Fu, 2009; Nwe, Foo, and De Silva, 2003), Support Vector Machines (Zhou et al., 2006; Hu, Xu, and Wu, 2007), and Gaussian Mixture Models (Neiberg, Elenius, and Laskowski, 2006). Generally, in state-of-the-art methods there was no default choice of features and classifier type (El Ayadi, Kamel, and Karray, 2011). With the advent of deep learning, end-to-end learning mostly replaced hard-coded feature extraction and selection, with models automatically extracting features from low-level representations of the input data (usually Fourier-based transforms, wavelet transforms or raw audio data). This enables a model to fine-tune the feature extraction for a specific task and, consequently, often obtain a higher accuracy compared to engineered feature extraction. A range of deep learning architectures have been adopted for SER. The most commonly used are variants of convolutional neural networks (Badshah et al., 2017; Sun, 2020; Issa, Demirci, and Yazici, 2020), recurrent neural networks (Lee and Tashev, 2015; Chernykh and Prikhodko, 2017) or combinations of the two (Trigeorgis et al., 2016; Lim, Jang, and Lee, 2016). Various studies directly compare the performance of approaches using end-to-end learning and hard-coded feature extraction, showing that the former generally provides a higher classification accuracy on the same data (Kim, Lee, and Provost, 2013; Mao et al., 2014; Huang et al., 2014; Han, Yu, and Tashev, 2014). Nevertheless, as a drawback, deep learning models generally requires a higher computational cost and longer training times than traditional machine learning techniques and the end-to-end learning usually requires a large amount of labelled data (Rossenbach et al., 2020; Laptev et al., 2020).

In the recent years, State-of-the-art (SOTA) approaches for SER tasks evolved from the use of plain deep networks, integrating complex solutions that can be radically different for each specific case and for each dataset as, among others, data augmentation (Pham, Dang, and Nguyen, 2021; Jothimani and

Premalatha, 2022; Etienne et al., 2018; Xu et al., 2021), attention (Pham, Dang, and Nguyen, 2021; Xu et al., 2021; Ho et al., 2020; Kakouros et al., 2022), adversarial attacks (Latif, Rana, and Qadir, 2018), multimodal processing (Ho et al., 2020; Bouali, Ahmed, and Mazouzi, 2022), speaker-aware processing (Kim and Vossen, 2021; Li et al., 2020b), transformer designs (Ho et al., 2020; Li et al., 2020b). The high degree of approaches variability, connected with the fact that many existing studies are based on different methods to compute the scores, different data splits and may use multiple data domains, make it difficult to clearly identify a unique SOTA approach for SER.

2.1.1 Emotion-Labelled Speech Datasets

As the aforementioned SER research work show, SER tasks are addressed with supervised learning strategies in the vast majority of cases, which rely on emotion-labelled databases of speech recordings. The design of such datasets has therefore a big impact on the quality of the SER models trained on them. One of the most influential aspects is the way the speakers express different emotional states (Ververidis and Kotropoulos, 2003; Akçay and Oğuz, 2020). To this respect, Akçay and Oğuz (2020) define 3 main categories of SER datasets:

- *Elicited emotional speech databases*: emotions are induced by creating situations that encourage certain emotional states in non-professional speakers.
- *Actor based speech databases*: emotions are acted by professional and trained artists.
- *Natural speech databases*: speech data is collected from real-world dialogues.

Each type of dataset has advantages and disadvantages. On the one hand, using professional actors ensures that all target emotions appear in the dataset, but emotions may sound forced and unnatural. This type of strongly-emphasised emotions can be easier to be detected by a SER model, however, a model trained on professionally-acted emotions is unlikely to generalize to real-world data, as this may present a completely different distribution. On the other hand, naturally-expressed emotions better reflect real world situations, but this setup does not allow to precisely control the amount of data recorded for each target emotion, and it is also possible that only a small subset of emotions is expressed (Cao, Verma, and Nenkova, 2015; Akçay and Oğuz, 2020).

SER datasets can be classified also according to the type of used vocabulary (usually referred as context):

- *Scripted scenario databases*: actors pronounce a selection of short sentences.
- *Semi-scripted scenario databases*: professional or non professional speakers talk about specific topics or interact with people that ask ask selected questions.
- *Non-scripted scenario databases*: free improvisation or real-world data.

Also in this case there are similar advantages and disadvantages. Scripted scenarios are less complex to classify with an algorithm due to the reduced variety of utterances, but it is very likely that the algorithm learns features that are overly specific to the pronounced words. Therefore, they hardly generalize to different spoken sentences (Jin et al., 2015). Natural scenarios better reflect real-world situations, but, again, it is possible that the amount of different emotions is highly unbalanced and only a small subset of target emotions is expressed. Semi-scripted datasets are a compromise between the other 2 and they are the most commonly used datasets for studies in which real-world application is intended.

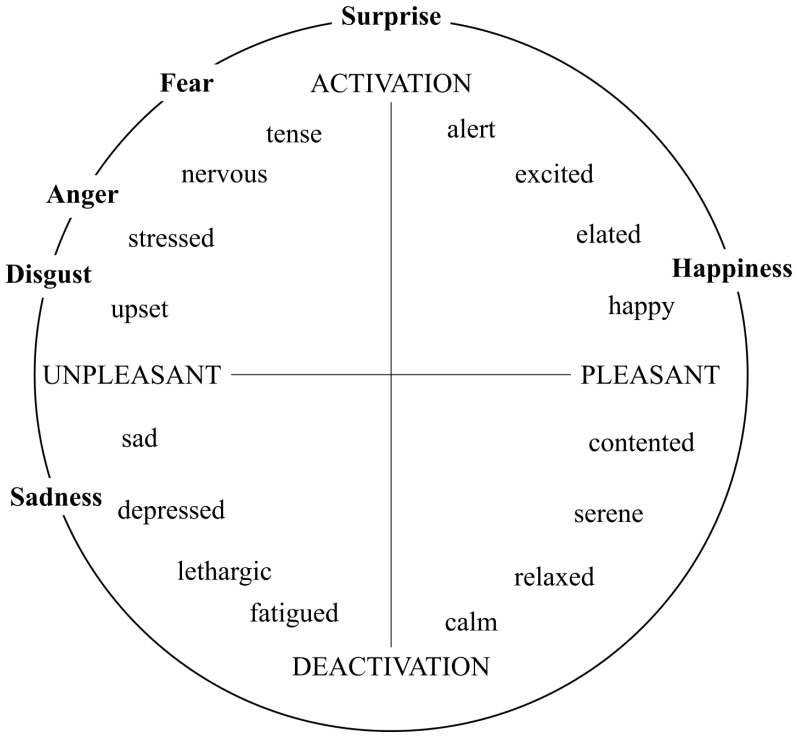


FIGURE 2.1: Example mapping from discrete to dimensional emotion space from *Stanford Encyclopedia of Philosophy* (Okasha, 2016).

2.1.2 Emotion labelling

As regards the emotion labelling strategies, there are mainly 2 options, which can coexist in the same dataset. *Discrete* models provide a set of fixed standard emotion categories. The most common are: happy, sad, angry, fearful, surprised, disgusted, neutral. On the other hand, *dimensional* models map emotions into a multidimensional continuous space (Ekman and Oster, 1979; Mehrabian, 1996). The most adopted solution is a 2D space in which the axes represent the *valence* and the *arousal*, where Valence, also known ad *pleasure*, describes the amount of emotional pleasantness, while arousal, also known as *activation*, measures the intensity of the emotion. The exact mapping is still subject of dispute and different studies can use slightly different mappings (Mihalache and Burileanu, 2021). Figure 2.1 shows an example mapping as exposed in the *Stanford Encyclopedia of Philosophy* (Okasha, 2016). A third axis called *dominance* can be added to this standard paradigm, forming

a tridimensional emotional space. This third metric measures the amount of self-confidence of an expressed emotional state.

Both strategies have pros and cons. On the one hand, discrete emotion labeling offers simplicity and result in easily interpretable outputs that are user-friendly and comprehensible to non-experts. Nevertheless, this comes at the cost of losing nuance and expressiveness, since fixed set of categories may not capture the full richness of emotional experiences, limiting therefore the models' capacity to reflect real-world complexity. On the other hand continuous emotion labelling allows for a more nuanced representation of emotions, embracing the intricacies of human emotions. However, annotating emotions with greater nuance requires a deeper understanding of emotional dimensions and their variations. Moreover, nuanced emotional labels can be more challenging to interpret, both for researchers and end-users. In addition, developing models that effectively leverage continuous emotional representations often necessitates more intricate architectures and advanced machine learning techniques compared to discrete representations. The choice between the two depends on the specific goals of the application, the available resources, and the desired balance between simplicity and accuracy in capturing emotional nuances. For these above reasons, discretely-labelled datasets are the most commonly used in the SER research community.

2.1.3 Adopted Datasets

Throughout the whole dissertation we have tested our approaches on 4 widely used emotion-labelled speech datasets:

1. *IEMOCAP*: The Interactive Emotional Dyadic Motion Capture Database (Busso et al., 2008). Acted/scripted/semi-scripted, 10 speakers, English language, 10000 utterances, 12 hours of audio, 6 discrete emotion labels + continuous emotion labels (valence-arousal-dominance).

2. RAVDESS, the Ryerson Audio Visual Database of Emotional Speech and Song (Livingstone and Russo, 2018a). Acted/scripted, 24 speakers, English language, 2542 utterances, 2:47 hours of audio, 8 emotion labels.

3. EmoDb, a Database of German Emotional Speech (Burkhardt et al., 2005). Acted/scripted, 10 speakers, German language, 535 utterances, 25 min of audio, 7 emotion labels.

4. TESS, the Toronto Emotional Speech Set (Dupuis and Pichora-Fuller, 2011). Acted/scripted, 2 speakers, English language, 2800 utterances, 1:36 hours of audio, 7 emotion labels.

We have chosen this specific combination of corpora for multiple reasons. In the first instance, IEMOCAP is among the most frequently used speech emotion datasets and contains both continuous and discrete classification, which is a strict requirement for the development of the RH-Emo approach, as we present in Chapter 6 of this dissertation. Moreover, it contains semi-improvised scenarios that enables to obtain a better insight of model's generalization, compared to the other scripted datasets that we adopted. On the other hand, in RAVDESS, EmoDb and TESS, being acted and scripted datasets, meticulously engineer emotional expressions by instructing professional actors to portray specific emotions under controlled conditions, thus enabling the isolation of emotional cues. This controlled variation provides a conducive environment to scrutinize the finer nuances of emotion recognition models. In contrast, spontaneous datasets encompass a broader spectrum of emotions that arise naturally in real-world interactions. This diversity often introduces inherent complexities like varying background noise, recording conditions and co-occurring emotions. While these traits mirror real-life scenarios, they do not allow for the isolation of specific features crucial for studying the effectiveness of anti-transfer learning, as discussed in

Chapter 5. The lack of systematic variation impedes the precise assessment of the efficacy of anti-transfer learning techniques in disentangling domain-specific variations from emotional cues, as these cues intermingle with the inherent intricacies of natural conversations.

2.2 Speaker and Context Independence

Several solutions have been developed to make SER models robust to unseen speakers and context. Among others, Kim et al. (2009) developed a hard-coded audio feature, the *ratio of a spectral flatness measure to a spectral center*, which helps models generalize to new speakers. Schuller et al. (2005) address this problem using an ensemble of acoustics and linguistic models merged through late fusion. Kotti and Paternò (2012) developed a cascade schema that initially aggregates similar emotions in macro-categories and then gradually goes towards the discrimination of specific emotions. Wang et al. (2015) developed a Fourier-based feature that describes the *voice quality*, which alongside its first and second-order differences helps generalization to unseen speakers. Savchenko and Savchenko (2021) propose to fine tune SER classifier for specific speakers and select the classifier to use with a speaker recognition system. Liu et al. (2019a) use a combination of convolutional neutral networks, attention-based bidirectional long short-term memory network and multiple linear support vector machines. Wang et al. (2020) developed a feature extraction strategy based on wavelet packet coefficient features. Lu et al. (2022) apply a multi-source unsupervised domain adaptation strategy to learn emotional features independent from their domain, in this case intended as the speaker identity. Instead Li et al. (2020a) achieve a similar disentangled emotion representation through an adversarial training network based on a gradient reversal technique with an entropy loss function that remove such speaker information.

2.2.1 Scale Invariance

Scale-invariance in convolutional neural networks has been addressed in a number of ways. The most common approach for audio by far is data augmentation (Salamon and Bello, 2017a; McFee, Humphrey, and Bello, 2015), which is frequently done by generating time-stretched variants of the training data. This procedure is usually part of a pipeline of different transformations, as in Schlüter and Grill (2015), which has proven effective in various tasks. However, in this approach the different scales in the data need to be learned by different filters in the network. Therefore, greater network capacity is required and there is no guarantee that scale-invariance is consistently achieved .

Another strategy for scale-invariance in neural networks is to design it into the training and inference methods, so that it is applied consistently and without the need for additional training examples. There are many existing approaches to achieve this and the majority of them use a pyramidal structure, in which the scale is progressively narrowed along the network. Dalal and Triggs (2005), Felzenszwalb et al. (2009), and Sermanet et al. (2013) use parallel models trained with images at descending resolutions and then combine the obtained predictions as an ensemble model. Szegedy et al. (2015) achieve scale invariance with multiple loss functions, separately computed in layers with different resolutions within the network. Inception networks (Szegedy et al., 2017; Szegedy et al., 2015; Szegedy et al., 2016) use parallel convolution layers with different filter sizes, matching features at different scales, but also increasing the number of variables in the network. While a variation of the latter approach relies on merging by summation or concatenation the features produced by the different re-scaled branches of a convolution layer (Long, Shelhamer, and Darrell, 2015; Hariharan et al., 2015; Kong et al., 2016; Liu, Rabinovich, and Berg, 2015; Bell et al., 2016). Wang

et al. (2019a) propose a convolutional architecture, in which a scaling factor is learned by the network for every layer.

The majority of studies of scale-invariance in neural networks is focused on computer vision tasks. In the acoustic domain, in addition to data augmentation techniques (Schlüter and Grill, 2015), scale-invariance can also be addressed through specific hard-coded transforms (Marchand and Peeters, 2016a) that are robust to some extent to scale variations. Nevertheless, since they are hard-coded, these methods need manual intervention and are usually highly task-specific, while embedding scale-invariance in the models provides a more generic solution that can be applied to multiple domains. The work of Zhu, Engel, and Hannun (2016) is an exception to this trend. They show that a network with n identically-sized filters performs worse than a network with the same number of filters, but split in 3 different sizes. Nevertheless, their models learn independent filters at different scales, increasing the number of free parameters. This achievement encourages the hypothesis that robustness to scale-invariance can enhance the performance of a CNN-based audio classifier.

Sequence-to-sequence alignment is another task that benefits from scale-invariant representation. In this domain, Time Warping is a leading technique that allows for stretching and compressing the time axis of sequences to find the optimal alignment that minimizes a defined distance metric between corresponding points (Putri and Lestari, 2015). This in turn permits to make predictions that are robust to temporal distortions and fluctuation, enhancing models' performance on disparate tasks as automatic speech recognition (Permanasari, Harahap, and Ali, 2019), time series analysis (Choi et al., 2020) and music information retrieval (Putri and Lestari, 2015).

Locally scale-invariant convolutional neural networks, as introduced for image recognition by Kanazawa, Sharma, and Jacobs (2014), are similar to our multi-time-scale convolution approach that we present in Chapter 4. This

method consists of performing feature-extraction through multiple parallel convolution layers, whose outputs are locally merged through max-pooling. This produces a self-contained structure that can substitute a canonical convolution layer. The key feature of their approach is the possibility of matching a feature at multiple scales without increasing the number of free variables in the network. It permits introducing several re-scaled parallel branches at different points in the network, providing higher flexibility then pyramidal architectures.

2.3 Task-Specific Invariance

2.3.1 Selective Representation Transfer

Transfer learning has been used with neural networks for a long time and in many different applications (Caruana, 1995; Bengio, 2012; Hamel et al., 2013; Shin et al., 2016; Tan et al., 2018a). Pretraining models has become standard practice in image classification and related tasks (Studer et al., 2019; Xie and Richmond, 2018; Han et al., 2018) and pretrained language models have become a common starting point in NLP (Qiu et al., 2020b). The transfer of knowledge from a trained network to a new task by re-using weights of a layer has been developed early on (Pratt, 1992; Gutstein, Fuentes, and Freudenthal, 2007).

The work of Gatys, Ecker, and Bethge (2016) on image style transfer is fundamental for the transfer of selective representations in neural networks. Based on the assumption that features become increasingly task-specific towards the last layer of a network (Yosinski et al., 2014), a strategy was developed by Gatys, Ecker, and Bethge (2016) to separate content and style of an image and to transfer the style alone to another image. The authors used a CNN that was pretrained on object recognition as a feature extractor

to estimate the style-related and the content-related information of an image in a CNN. The style of an image is represented by the Gram matrix computed on the initial layers, which contains information about texture, i.e. the co-occurrence of low-level features. The content is represented by the raw feature maps of the final layers. During the training of the style transfer network, the feature extractor separately extracts the style and the content from two different images and compares them to the corresponding features extracted from an image that is being generated, creating two deep feature loss values: style and content loss. The minimization of these losses promotes the generation of an image with the style of one image and the content of the other one. This idea received much attention in the computer vision community and has been further explored and improved (Ulyanov et al., 2016; Dumoulin, Shlens, and Kudlur, 2017). It has also been applied in the audio domain to audio style transfer with MelGan (Pasini, 2019), using both speech and music sources.

2.3.2 Deep Feature Losses

Deep feature losses involve comparing the feature representations of data at various layers of a neural network to measure their difference or similarity. The majority of studies regarding deep feature losses is based on the idea of encouraging a network to develop similar deep representations to a pretrained network in selected layers, e.g. in line with the work of Gatys, Ecker, and Bethge (2016). These losses play a pivotal role in various machine learning tasks: by assessing the agreement of feature representations at different layers of the network they enable models to capture semantic information and abstract features rather than focusing on low-level discrepancies. This, in turn, helps the models to improve the quality of the predictions or the generated content (Johnson, Alahi, and Fei-Fei, 2016; Zhang et al., 2018; Sahai,

Weber, and McWilliams, 2019).

Deep feature losses have been used in several computer vision tasks as texture synthesis (Gatys, Ecker, and Bethge, 2015), image super-resolution (Johnson, Alahi, and Fei-Fei, 2016) and conditional image synthesis (Chen and Koltun, 2017; Dosovitskiy and Brox, 2016). According to recent studies (Zhang et al., 2018; Doersch, Gupta, and Efros, 2015), deep feature losses are highly correlated to human perceptual judgements and are well suited to solve tasks related to semantic properties of data. Deep feature losses have several successful applications also in the audio domain. They have been used by Beckmann et al. (2019) to enhance the similarity between the deep representations of two networks and therefore transferring knowledge from one to the other, enhancing the networks' performance in several speech processing tasks. A deep feature loss was successfully used by Sahai, Weber, and McWilliams (2019) to perform audio source separation, obtaining a superior performance compared to spectrogram-based loss. Kegler, Beckmann, and Cernak (2019) applied the same conceptual idea to speech enhancement, language identification, speech, noise and music classification, and speaker identification.

2.3.3 Feature Diversity

Minimizing feature similarity has been shown earlier to improve robustness and generalization. In the context of ensemble models, Yao and Liu (2004) minimized mutual information between neural networks. More recently, the minimum hyperspherical energy (MHE) regularization was introduced by Liu et al. (2018) and applied to audio source separation by Perez-Lapillo, Galkin, and Weyde (2020). MHE encourages diverse weight vectors within a network to improve generalization, but it differs from our approach since we encourage dissimilarity of feature maps and with respect to another model.

2.3.4 Domain-Invariant Feature Learning

A common use case for transfer learning is domain adaptation, e.g. to different recording equipment or environments, and a common approach is to maximize the feature invariance to the domain of the data. *Mutual Information Minimization* is used in Wang, Liu, and Wang (2019) to extract features independent from the domain of the data points by maximizing the feature invariance to their domain indicator. In Domain Adversarial Training (DAT) (Ganin et al., 2016), a gradient reversal layer is introduced to maximize the loss on domain identification while minimizing the classification loss. A similar approach, but with a Siamese architecture, is introduced in Motiian et al. (2017). In Tzeng et al. (2017), a more general framework is presented, including generative adversarial approaches, that is also applied in domain adaptation for acoustic scene classification using unlabeled data for the target domain (Drossos, Magron, and Virtanen, 2019). Lu et al. (2022) apply a multi-source unsupervised domain adaptation strategy to a speaker-independent SER task. They achieve a multi-domainin variant representation of emotional speech through the use of a a hierarchical alignment layer in the network's feature extractor combined with the use of multiple discriminators that confuse the speaker information of emotion features.

Another approach to domain-invariant feature learning is contrastive learning. This self-supervised technique focuses on learning representations that are invariant to variations within the same domain while emphasizing differences between different data samples. By optimizing the model to distinguish between similar and dissimilar samples, contrastive learning can help the model capture meaningful features that are robust to intra-domain variability (Jaiswal et al., 2020; Le-Khac, Healy, and Smeaton, 2020). Among others, Zbontar et al. (2021) propose a unique loss function that encourages learned representations to be both highly informative and decorrelated. This

loss function enhances the quality of learned representations by minimizing redundancy while maximizing the captured information. Li et al. (2021) train models to predict future elements in a sequence while distinguishing them from negative samples (contrastive predictive coding), demonstrating the potential of contrastive learning techniques to enhance SER outcomes even in data-constrained scenarios. Contrastive learning for SER tasks has been also proposed by Alaparthi et al. (2022), applying this principle to transformer networks.

Temporal management can reinforce domain-invariant feature learning in various ways. Among others, Techniques like cycle-consistency (Wang, Liu, and Wang, 2019) and multi-domain prediction (Lei et al., 2021) encourage models to capture common temporal dynamics by incorporating domain-agnostic temporal representations and attention mechanisms, shifting network's focus towards features that transcend domain-specific temporal variations.

2.3.5 Causal Frameworks

A different way to obtain invariance towards the data domain is causal modelling that is aimed at understanding the mechanism by which changes in one variable influence changes in another. In practice, it involves considering and accounting for confounding factors that might lead to erroneous predictions due to spurious correlation between the cause (target) and the effect (observed data), where a feature is considered as spurious when it is not stable and does not hold on different data domains or distributions (Woodward, 2005).

While Neural Networks are excellent in finding statistical correlations between input features and target outcomes, these correlations do not necessarily imply a causal relationship (Kilbertus, Parascandolo, and Schölkopf,

2018). Causal frameworks focus on learning causal relationships and uncovering underlying mechanisms in data, rather than merely spurious surface correlations that might be present in data. This in turn involves identifying which properties of the analyzed data describe spurious correlations and which properties instead represent the phenomenon by investigating. In this context, an approach that is particularly relevant to our work is Invariant Risk Minimization (IRM) (Arjovsky et al., 2019). IRM focuses on learning representations that are domain invariant by minimizing the prediction risk across multiple related domains, referred to as *environmants*. This permits to exploit domain-specific causal knowledge about confounding factors that affect both the source and target domains to guide the selection of invariant features that account for these confounders and lead to better cross-domain generalization.

2.3.6 Disentanglement

The representation of independent properties of objects or processes has been recently explored in the literature and is usually referred to as *disentanglement* (Hung, Chen, and Yang, 2018; Chou et al., 2018; Nagrani et al., 2020; Lee et al., 2020). Methods for achieving disentanglement include adversarial training (Oldfield, Panagakis, and Nicolaou, 2019) or specific architectures, such as partitioned or factorized variational autoencoders (Li and Mandt, 2018; Hsu and Glass, 2018). Anti-transfer, the approach that we present in Chapter 5, can be considered a special case of disentanglement, aiming at the invariance to the internal representations of distinct orthogonal models.

2.4 Speech Emotion Recognition Models Optimization

2.4.1 Data Scarcity

A well-established solution to overcome the data scarcity in SER is transfer learning by weight initialization: network weights are initialized with values from a network that was pretrained with a different task, possibly on a different (usually large) dataset. Many variants of this method have been shown to improve the performance of SER models in limited-data scenarios and even when the task is rather distant from speech emotion (Macary et al., 2021; Pepino, Riera, and Ferrer, 2021; Guizzo, Weyde, and Tarroni, 2021). Also, various data augmentation strategies have been successfully adopted for the same purpose, e.g. Padi, Manocha, and Sriram (2020) and Shilandari, Marvi, and Khosravi (2022). On the other hand, the application of dimensionality reduction transformations to the model's input data is an established strategy for reducing resource demands while limiting the loss of useful information carried by the input data. Among others, autoencoders, PCA-based approaches, and transformer networks have been used in the field of SER (Fewzee and Karray, 2012; Patel, Patel, and Mankad, 2021; Pepino, Riera, and Ferrer, 2021), obtaining improvement both in the model's efficiency and classification accuracy.

2.4.2 Model Efficiency

Various techniques have emerged to enhance neural networks model efficiency without compromising performance. Among many others, Quantization reduces the precision of model parameters, compressing them to lower bit-widths. This minimizes memory and computation requirements, enhancing the model's suitability for edge devices. (Han, Mao, and Dally,

2015). Another relevant technique is pruning, involving removing unnecessary connections or neurons from a neural network. This process reduces model complexity, accelerates inference, and often improves generalization (Blalock et al., 2020). The pruning process can also be guided and optimized through knowledge-based localization of the relevant information, as in Balemans et al. (2020). However, aggressive quantization and pruning might adversely affect the model's performance (Han, Mao, and Dally, 2015; Blalock et al., 2020). Teacher-student methods transfer knowledge from a larger, more complex model (teacher) to a smaller, more efficient model (student). This technique optimizes the student's performance while significantly reducing model size (Meng et al., 2019). Another relevant branch of techniques is based on the lottery ticket hypothesis, which posits that within large neural networks, there exist "winning tickets", that are smaller subnetworks that, when trained in isolation, can achieve comparable performance. Identifying these subnetworks enables efficient model training, as Frankle and Carbin (2018) demonstrated. A further successful approach towards model's efficiency consists of decomposing layers of a neural network into low-rank matrices reduces the number of parameters, accelerating computations. This technique balances efficiency gains with accuracy preservation (Sainath et al., 2013). Dynamic computation a further relevant method which is based on adapting the model's computation complexity based on input data. In this context, MobileNets (Howard et al., 2017) introduce depth-wise convolutions, reducing computations for specific layers, dynamic-depth networks (Sun, Li, and Xu, 2022) adapt the depth (number of layers) of a neural network based on the complexity of the input data, while sparse activation strategies use only a subset of neurons based on the input data characteristics (Glorot, Bordes, and Bengio, 2011). Temporal management can be exploited to boost model efficiency by strategically processing time-dependent data. By identifying and capitalizing on key temporal patterns, models can reduce

redundant computations, focus on critical moments, and optimize memory usage. Techniques like multi-source features down-sampling (Tzinis, Wang, and Smaragdis, 2020) or temporal attention (Tan et al., 2023) can extract relevant information, streamlining computations. Leveraging temporal dependencies allows for more informed resource allocation, resulting in faster inference and reduced computational overhead, thus enhancing overall model efficiency.

2.4.3 Quaternion Information Processing

A recent and increasingly popular strategy to improve the efficiency and the performance of deep learning models is the use of quaternion information processing (Tay et al., 2019; Grassucci, Comminiello, and Uncini, 2021; Grassucci, Cicero, and Comminiello, 2022; Grassucci, Zhang, and Comminiello, 2021; Greenblatt and Agaian, 2018; Parcollet et al., 2018; Muppidi and Radfar, 2021).

Quaternion algebra is a branch of abstract algebra that deals with the mathematical properties and operations involving quaternions (please refer to Appendix 8.3 for the mathematical details. Quaternions are a specific type of mathematical object that extends the concept of complex numbers into four dimensions, as introduced by the Irish mathematician William Rowan Hamilton in the mid-19th century. Quaternion neural networks perform operations based on quaternion algebra. In quaternion layers input and weights matrices are treated as quaternion entities and Real-valued operations are substituted with quaternion-valued. These operations include quaternion multiplication (or Hamilton product), addition, and other specialized operations specific to quaternions. The most important difference with real-valued layers is that the dot product is substituted by Hamilton product, which has the same output shape of a real-valued equivalent one

but has less degrees of freedom, specifically using one quarter of the trainable parameters (see Appendix A for further details) (Tay et al., 2019). Performing operations in the quaternion domain permits to take advantage of intra-channel correlations in multi-channel signals (Bulow and Sommer, 2001; Mandic, Jahanchahi, and Took, 2010), i.e., among the color channels of RGB-encoded images. Quaternion-valued neural networks have also been successfully adopted in the audio domain (Comminiello et al., 2019b; Comminiello et al., 2019a; Qiu et al., 2020a; Muppidi and Radfar, 2021) and specifically for speech recognition (Parcollet et al., 2018) and speech emotion recognition (Muppidi and Radfar, 2021). Nevertheless, an intrinsic limitation of quaternion information processing is that it requires three or four-dimensional data as input, where intra-channel correlations exist (Greenblatt and Agaian, 2018; Grassucci, Comminiello, and Uncini, 2021; Grassucci, Zhang, and Comminiello, 2021; Grassucci, Cicero, and Comminiello, 2022). This is necessary to enable the benefits derived from the use of the Hamilton product instead of the regular dot product, as further discussed in Section A. In the audio domain, first-order Ambisonics (Furness, 1990) signals are naturally suited for a quaternion representation, being four-dimensional and presenting strong correlations among the spatial channels, and the application of quaternion networks to problems related to this audio format has already been extensively investigated (Comminiello et al., 2019a; Qiu et al., 2020a; Brignone et al., 2022; Grassucci et al., 2023). Nevertheless, in the vast majority of cases, audio-related machine-learning tasks deal with monaural signals, which are usually treated as vectors of scalars (time-domain signals), matrices of scalars (magnitude spectrograms), or 3D tensors (complex spectrograms). Hence they can not be naturally represented as a quaternion entity and additional processing is required to produce a suitable quaternion representation of these signals.

2.4.4 Quaternion Representations from Real-Valued Data

A number of different approaches have been proposed to overcome the necessity of having three or four-dimensional input data with intra-channel correlations. Among others, Parcollet et al. (2018) use Mel spectrograms, cepstral coefficients, and first and second-order derivatives as the four axes of the encoded quaternion. In contrast, Muppidi and Radfar (2021) convert Mel spectrograms to color-scaled images and use the RGB channels as axes of the encoded quaternion, following a computer vision-oriented approach. Parcollet et al. (2019) presented two learning-based approaches to map real-valued vectors into the quaternion domain, by producing through a network four-channel representations of the input data that present meaningful intra-channel correlations. On the one hand, the Real to H-space encoder Parcollet et al. (2019), applied to speech recognition tasks, consists of a simple real-valued dense layer applied at the beginning of a quaternion classifier network, which is trained jointly with the classifier. On the other hand, the Real to H-space Autoencoder, tested in the natural language processing field (conversation theme identification) (Parcollet et al., 2019) operates in an unsupervised way. Such a method contains a real-valued encoder and a quaternion-valued decoder, where the latter is expected to enable both the network's embeddings and output to present meaningful intra-channel correlations that can be exploited by a quaternion-valued classifier network.

Chapter 3

Preliminary Study on a Context-Independent Scenario

This Chapter includes an adapted version of Conference Paper 2 (see section 1.5) and extends it with additional results and discussions.

In the first stage of our work we participated to the above-mentioned *OMG-Empathy Prediction Challenge 2018* alongside with researchers from *AI* team of *Telefonica Alpha*, Barcelona.

This challenge represents the starting point of all other studies present in this dissertation. While developing our challenge solution, we found that speaker/context independence and high resource demand of models are fundamental problems to solve for successful real-world applications of SER. Based on this, we decided after the challenge to dedicate my entire PhD to explore ways to address these problems and we defined the research questions of this dissertation accordingly.

The OMG Challenge setup is configured as a context-independent SER scenario, as the models' performance is tested on an unseen vocabulary, but spoken by known actors. Here SER models are expected to develop a certain degree of invariance towards the vocabulary used by the speakers and generalize towards new words that are unseen in the training and validations set. Our solution is based on the processing of multiple different data streams, as

further explained below, but I have mainly worked on the part that concerns the audio information.

This Chapter, together with Chapter 4, addresses the second Research Question.

3.1 Introduction

We propose a system that separately processes multiple data streams (modalities) which are integrated at a late stage, so-called late fusion. The motivations for this modular approach are ease of development, enabling different parts of the team to separately optimize the processing of different modalities, and future extensibility, facilitating the integration of additional data streams in future iterations of our system.

The given dataset is composed of audio/visual recordings of dialog sessions with 2 actors each, discussing about 8 different topics:

- Talking about a childhood friend.
- How I started a band.
- My relation with my dog.
- I had a bad flight experience.
- I had an adventurous travelling experience.
- I cheated on an exam when I was younger.
- I won a martial arts challenge.
- I ate a very bad food item.

In Appendix A the reader can find an example transcription of a conversation extracted from this dataset. In every recording an actor leads on a

semi-scripted plot on one topic, while a listener freely interacts with the actor. 10 listeners and 4 actors in total participated to the recording of this dataset. A total of 80 recorded sessions is present in the dataset, that is 10 for every topic. In the set of 10 sessions of a topic, every session is recorded by a different listener. While each conductor recorded all sessions of 2 different topics, that is 20 sessions each. The average duration of each session is around 5'12", while the total duration of the dataset is around 512 minutes (approximately 7 hours). Every session is emotionally labelled with a continuous valence measure. This labelling was produced by self assessment of the same listeners that recorded the videos, who re-analyzed their sessions moving in real time a joystick to annotate how they felt in terms of valence using a continuous scale ranging from positive (1) to negative (-1) values. This continuous measure is quantized every video frame, so that 25 measures per second are given. The dataset is split topic-wise, providing for a context-independent scenario. In particular, 4 topics as training, 1 topic as validation and 3 topics as test set.

The main task of this challenge is to perform a continuous valence prediction over time, which in this context is referred as *empathy* of the listener with respect to the actor. The performance of a model is measured through the *Concordance Correlation Coefficient* (CCC) computed between the model's prediction and the actual valence created by the listeners. It is important to note that the test predictions are performed in a context (topic) unseen during the training. The main task is divided in 2 sub-tasks:

- *Personalized Empathy Track*: This modality takes into account the individual behavior of each subject in the training data, measuring the ability of proposed models to learn the empathetic behavior of each of the listeners over a newly perceived story.

- *General Empathy Track*: This modality takes into account the aggregated

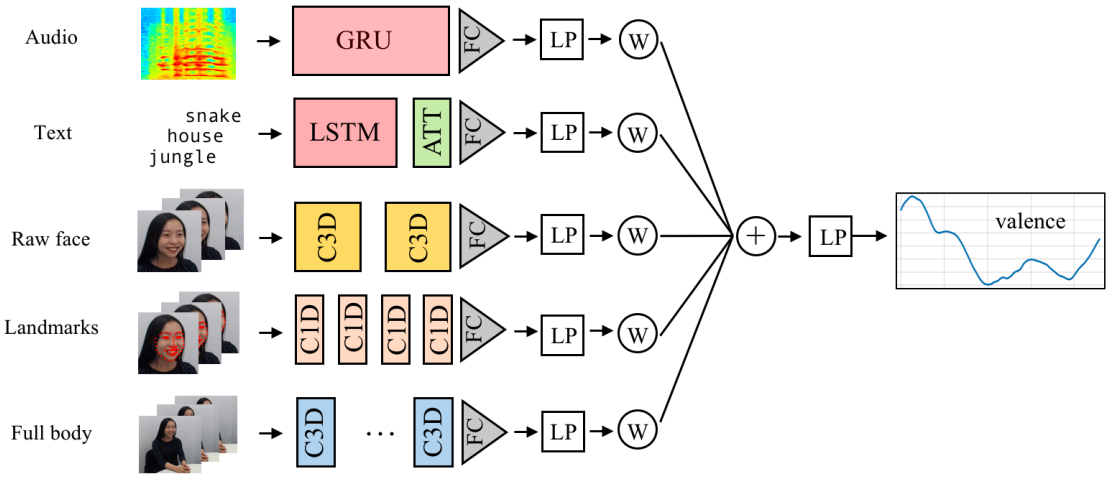


FIGURE 3.1: Schematic view of the whole system.

behavior of all the participants for each story, measuring the ability of proposed models to generalize the global empathetic behavior in a newly perceived story.

3.2 Methods

We integrate three different modalities (further broken down into five data streams) in the prediction of the valence ratings of the videos. These modalities are:

- *Image* information directly extracted from the videos.
- *Audio* information directly extracted from the videos.
- *Language* information obtained by automatic transcription of the audio data.

Figure 3.1 depicts a general schema of the model. Note that, in our model, the audio and language modalities each give rise to single data stream to be processed, whereas the image modality is further broken down into three input data streams: one corresponding to the full body of the subjects, another

one focusing on the face of the subject, and a final one that further synthesizes specific landmarks extracted from the subjects' faces. Note that we employ specific architectures across the five resulting systems, which are specifically optimized, of each modality. Nevertheless, the loss function and training and validation sets where held constant across the five sub-systems.

In all five systems, instead of using Story #1 as the validation set and all other ones as the training set (as was suggested by the instructions), we chose instead to use Story #2 as our validation set and the remaining ones as our training set. The reason for this is that we found the statistical properties of Story #1 to be rather unrepresentative of the stories in the original training set. In particular, the main frequency at which the ratings oscillated between positive and negative was found to differ significantly from the others.

All models where trained to minimize 1 – CCC as loss function, where CCC is the *Concordance Correlation Coefficient*. For a sequence of valence predictions x and and a sequence ground truth valences y, the CCC is defined as:

$$CCC = \frac{2\,\rho\,\sigma_x\,\sigma_y}{\sigma_x^2 + \sigma_y^2 + (\mu_x - \mu_y)^2} \tag{3.1}$$

where σ_i, μ_i refer to the means and standard deviations of the subscripted sequences, and ρ is the *Pearson's correlation coefficient* between x and y. The CCC measure is a correlation coefficient that additionally penalizes divergences in either mean of variance between the two data sequences. As we will see below, this motivates some additional post-processing of the output data.

We experimented with several architectures and hyperparameter values for each module. For brevity, we present only the configurations that provided the best performance.

3.2.1 Audio Model

This model is based on the audio information extracted from the video files. Every audio file is preprocessed in 4 consecutive stages: pre-emphasis, segmentation, Fourier transform, and normalization. In order to discard low-frequency-noise, we first pass the signal through an 8th order Butterworth high-pass filter with 100 Hz cut-off frequency. Then we apply an emphasis filter based on the following equation:

$$y(t) = \frac{2x(t) - x(t-1)}{3} \tag{3.2}$$

where $x(t)$ is an audio sample and $x(t-1)$ is the preceding sample. This acts as a gentle first order high pass filter that emphasizes the spectral range of speech, with the upper limit of 8kHz defined by the sampling frequency of 16kHz. Every file is segmented into 8-second slices with 20% overlap. Consequently, the STFT is computed for every slice using 16ms sliding windows with 10ms overlap. This results in exactly 4 STFT frames for each valence measure (since 1 valence every 40 milliseconds is provided). After this process, we discard the phase information and compute the power-law compression by exponentiating the spectrum magnitudes the power of 2/3 to approximate human perception (Weninger et al., 2014). This technique is adopted from the method of calculation of the Perceptual Linear Prediction Coefficients (Hermansky, 1990). Finally, we normalize the spectra to zero mean and unit standard deviation.

This neural network has a sequence-to-sequence design based on a Recurrent Neural Network. The model's task is to predict time sequences of 200 valence samples for 8 seconds of input. First, we apply a layer of bi-directional Gated Recurrent Units (GRU) with 250 neurons for the forward and 250 neurons for the backward representation of the input data. Batch

normalization and Dropout at 30% are applied to the GRUs' output to reduce overfitting. Then the signal is propagated into a fully-connected layer with 200 neurons using linear activation. We trained the model with a batch size of 50 samples, using the ADAM optimizer (Kingma and Ba, 2014) and applying Early Stopping. The mean CCC obtained by this model in the validation set is 0.32.

3.2.2 Language-based Model

The model for processing the linguistic input stream consists of a recurrent network that processes the dialogue transcript, which was obtained with the Amazon Transcribe service [1]. Beyond transcription, no further textual preprocessing is applied. The transcription results in a sequence of words, together with time-stamps indicating when each word starts and ends. Each word spans several frames, hence more than one valence value. We address this by associating each word to the average valence score of all the valence scores within its span. Each word is represented as a vector of 11 dimensions, consisting of the features extracted from two emotional lexicons (Warriner, Kuperman, and Brysbaert, 2013; Staiano and Guerini, 2014).

An LSTM network (Hochreiter and Schmidhuber, 1997) is used to predict a valence score after each word. The time window used (what the LSTM *sees* at each step) is a window of 100 words. The hidden vectors of the LSTM are merged with a weighted average implemented with the following attention

[1] `https://aws.amazon.com/transcribe/`

module as in Yang et al. (2016):

$$z_i = w_a h_i + b_a$$
$$\alpha_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}$$
$$s = \sum_{j=1}^{N} \alpha_j h_j$$

where $h_i \in \mathbb{R}^d$ is the hidden representation of the LSTM corresponding to the i^{th} word, with N the total number of words in the window. The weight vector $w_a \in \mathbb{R}^d$ and bias term $b_a \in \mathbb{R}$ map this hidden representation to a value that reflects the importance of this state for final valence. The values $z_1, ..., z_n$ are then normalized using a softmax function, yielding the attention weights α_i. The word sequence representation s (at each time frame) is defined as a weighted average of the vectors h_i.

For each input also the listener subject information is given in the input to the network, since the story transcript can be very similar across videos, but the labeling can be highly different depending on the subject. The subject feature is implemented as a trainable vector of size two (one vector for each of the 10 subjects).

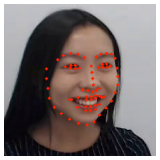
Finally, vector s, computed with the attention module, is concatenated with the subject vector, and a final analogous transformation is used to shrink the concatenated vector to one dimension (the final valence prediction of the model). This module achieves an average CCC of 0.32 across the validation set.

3.2.3 Vision Model

The visual model includes features extracted from the subjects who were listening to the story. Visual features are extracted from the face only, in order to capture facial expressions, but also from the whole body, in order to model

the subject body reactions to the story (i.e. posture and gestures). This leads to three modules for vision as follows.

(A) Raw face.



(B) Landmarks.

(C) Full body.

FIGURE 3.2: Vision data examples.

Raw Face

This module is dedicated to the prediction of valence from the subject's facial expression. It takes into account both the temporal evolution of the data and a subject feature vector. The cropped face images are obtained using the preprocessing script provided on the competition repository[2], using a temporal resolution of 10 frames for the face detection algorithm (Figure 3.2a). The obtained crops are subsequently turned to grey-scale, downsampled to a resolution of 48x48 pixels and normalized to have zero mean and unit standard deviation. The images are further organized with a 10 frame sliding window to obtain samples of shape 10x48x48. Each sample is matched with the valence label corresponding to the 10th frame. To predict valence from sequences of faces we use a neural network architecture composed of one 3 dimensional convolutional block of output shape 32 (two 3D convolutional

[2] `https://github.com/knowledgetechnologyuhh/OMGEmpathyChallenge`

layers with kernel size 3x3x3 and ReLU activation followed by a max-pooling layer and batch normalization) followed by a second 3D convolutional block of output shape 64. In addition we provide a subject feature vector encoding information about the subject and implemented as a trainable vector of size three (one vector for each of the 10 subjects), as in the text model. The concatenated layer is finally mapped to a fully connected ReLU layer of size 128, followed by a fully connected ReLU layer of size 32 and a single unit with linear activation. The network is trained with a batch size of 64 samples, using the ADAM optimizer. This module achieves a mean CCC of 0.14 on the validation set.

Face Landmarks

This module is dedicated to the prediction of valence from features extracted from image data, taking into account their temporal evolution. The data preprocessing consists of a facial landmark detection, performed frame-by-frame using the *dlib* library (King, 2009). We detect 68 landmarks points per frame on the subjects face, as shown in Figure 3.2b. Each point is defined by its (x, y) coordinates. This characterizes each video with 136 time series describing the temporal evolution of the landmarks points. Each time series is subsequently processed to have zero mean and unit standard deviation. The time series are further organized into 25 frames sliding sequences so to obtain samples of shape 25x136. Each sample is matched with the valence label corresponding to the 25th frame. For the first 25 frames, we perform constant-value padding. To predict valence, we use the 25-sample-long time series as inputs for a 1D Convolutional Neural Network architecture, composed of a first convolutional layer with 100 kernels, followed by a batch normalization layer and a convolutional block of three convolutional layers with 100, 160, and 160 kernels respectively. A 1D global average pooling is then applied, followed by a Fully Connected layer of size 32 that is mapped

to a single unit with linear activation. All the layers - except for the output layer - have ReLU activation functions, and a kernel size of 4. The network is trained with a batch size of 512 samples, using the ADAM optimizer. This module achieves a 0.12 CCC on the validation set.

Full-Body

This module is dedicated to predicting valence ratings out of full body subject images and it takes into account the sequential nature of the dataset. The full-body crop images are obtained using a preprocessing script that applies a cropping box manually selected to capture the position of the subject. An example is shown in Figure 3.2c. The obtained crops are then turned to grey-scale, downsampled to a resolution of 128x128 pixels and normalized by subtracting the mean and dividing by the standard deviation. Further the images are organized into 16 frame sliding windows so to obtain samples of shape 16x128x128 and each sample matched with the label corresponding to the 16th frame. Note that in the final setup the video sequences and respective labels are downsampled by a factor of 5 in order to expand the temporal window to approx. 3 seconds while maintaining a fixed sample shape. To predict valence ratings from full body video snippets we use a neural network architecture based on a version of the ResNet 16 (He et al., 2016) architecture adapted to 3-dimensional data [3]. This architecture was further modified for regression by replacing the last 3 fully connected ReLU layers with a 512 unit layer connected to a 32 unit layer followed by a single unit output. The network is trained with a batch size of 128 samples, using the ADAM optimizer. This module achieves a mean CCC of 0.31 on the validation set.

[3] `https://github.com/JihongJu/keras-resnet3d`

3.2.4 Postprocessing & Multimodal Integration

The predictions from each module are post-processed using a first order Butterworth low-pass filter and different cutoff frequencies adjusted for each individual module, ranging from 0.004 to 0.01 Hz. The filtered predictions $\hat{x}$ are then re-centered and re-scaled so that they match the training set in terms of per-subject means and standard deviations. This is a relatively ad-hoc procedure designed to optimize CCCs, under the assumption that the mean and standard deviations of the ground truth valences in the training set provide an approximation of those in the testing and validation sets.

Our best final predictions were obtained with a weighted average of the post-processed predictions of the single models. We set the weights approximately proportional to the CCC validation score of each modality. Audio, Text and Fullbody have similar performances, hence same weights (0.29), while the weights of Landmarks and Rawface are respectively 0.1 and 0.03. The average predictions are then filtered using a Butterworth low-pass filter of order 1 and cutoff frequency 0.01 Hz.

3.3 Results and Discussion

Figure 3.3 plots the predictions of the five different input streams and final integrated signal for Story #2 (validation) for Subjects 7 and 3 (the worst and best performing, respectively). In addition to this, we included a transcription of Story #2 and a link to its recorded video in Appendix A We observed that accurate predictions are associated with low disagreement across modules. Some models performed in average better than others, but the weighting scheme roughly proportional to the prediction accuracy of each input stream provided optimal results. This simple multimodal integration method performed less effectively in situations of perceptual ambiguity, where different modules predicted different, sometimes opposite, estimations.

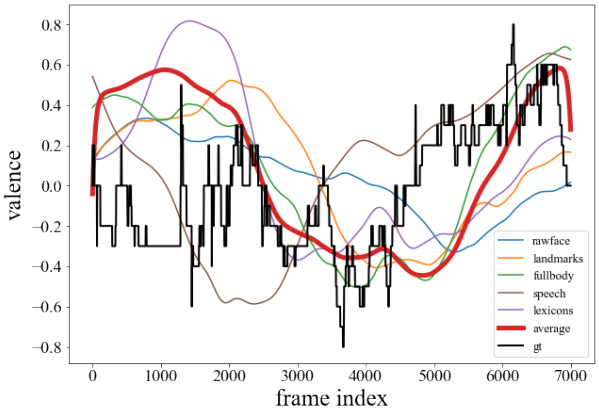
In the final evaluation results, our model's performance varied substantially across subjects and stories as is shown in Table 3.1. Performance was very good on Stories #3 and #6, but very poor on Story #7. Similarly, for two out of ten subjects the average CCC is negative, while for the rest the average CCC ranges from 0.09 up to 0.34.

As we indicated in the introduction, our system should be taken as a first approximation to a multimodal integration system. Although we have spent considerable time and effort in optimizing the valence predictions from the individual input streams, we believe there is yet much space for improving the method of integrating the predictions across modalities and modules. For instance, the weighted average we have used, could probably be improved upon by more sophisticated machine learning models (although our initial experiments failed to achieve this). Nevertheless, our work on this system has provided us with some valuable insights.

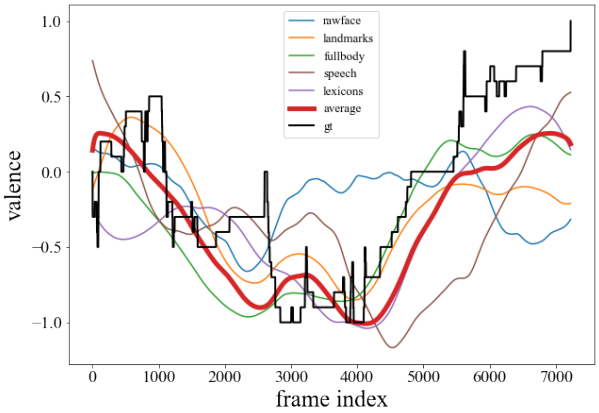
The first of this concerns the nature of the ground truth data themselves. As we have been training systems to try to match exactly the ground truth (and therefore optimize the CCC), we noticed that these data contain a large amount of high-frequency components The values oscillate between positive and negative several times within a few hundred milliseconds, which we suspect to be not solely reflective of emotional responses from the user, but caused by the input method. A more rigorous way of addressing this effect may help to estimate emotions better and may make the machine learning more effective.

Our second conclusion concerns the overall shape of the curves. We found that when the training data have similar overall shapes and the neural networks do worse when that shape changes. This suggest that time-warping procedures for training data enrichment might be useful to ensure that the systems generalize better.

Overall, the results show that the prediction of emphatic emotional reactions is still a challenging task that deserves further investigation.



(A) Story #2, subject 7 (lowest CCC results).



(B) Story #2, subject 3 (highest CCC result).

FIGURE 3.3: Example predictions: *speech* is for the audio module, *lexicons* for the language module, *gt* for the ground truth data and *average* for the overall prediction.

3.3.1 Additional Results

Our ensemble model provides a test CCC of 0.17 for both sub-tasks, surpassing the other participants by 0.03. We consequently won this challenge[4]. Table 3.2 shows the full challenge results with the average CCC obtained by all teams for the two challenge tracks, nevertheless not all teams submitted a

[4]Results are published at this link: `https://www2.informatik.uni-hamburg.de/wtm/omgchallenges/omg_empathy2018_results2018.html`

Subject	Story 3	6	7	Subj. Avg
1	0.43	0.62	-0.03	*0.34*
2	0.28	0.22	0.00	*0.17*
3	0.18	0.57	0.09	*0.28*
4	-0.11	0.62	0.04	*0.18*
5	0.09	0.86	-0.05	*0.30*
6	0.09	0.47	0.09	*0.22*
7	0.11	0.36	-0.21	*0.09*
8	0.16	-0.22	-0.01	*-0.02*
9	-0.11	0.11	-0.22	*-0.07*
10	0.08	0.66	0.09	*0.27*
Story Avg	*0.12*	*0.43*	*0.02*	*0.17*

TABLE 3.1: Experimental results on the test set. Final results of Personalized and Generalized track are both 0.17 as the model submitted to the two tasks was the same.

model for both. The challenge results provide valuable insights into the performance of different teams, showing the efficacy of their chosen approaches in relation to the utilized modalities. Notably, our team (Alpha-City), "USTC-AC," and "A*STAR AI" achieved competitive 0.17 scores in "Pers. Gen." by integrating audio, image, and text data, highlighting the potency of multi-modal fusion. Team "Rosie" secured a score of 0.08, showcasing the potential of audio, image, and semantic modalities. Both "Baseline" and "EIHW" teams garnered scores of 0.06 through audio and image approaches, emphasizing multi-modal benefits. The "Affective Bulls" team's scores of 0.02 and 0.03 underscore the complexities of emotion recognition in audio and image. Collectively, these results accentuate the role of modality integration in refining emotion recognition outcomes and the inner difficulty of predicting human emotional behavior in a continuous time-varying setup.

We also separately computed 19 test CCC given by the sole audio model, obtaining an average of 0.08. The CCC we obtained is overall noticeably low,

TABLE 3.2: OMG-Empathy 2018 Challenge results. The Personalized (Pers.) and Generalized (Gen.) tracks show the test CCC results obtained by the best model submitted by each team (listed in the first column). The last column indicates the modalities used by the models to produce the results

Team	Track		Modalities
	Pers.	Gen.	
Alpha-City	**0.17**	**0.17**	Audio+Images+Text
USTC-AC	0.14	0.14	Audio+Image+Time
A*STAR AI	0.14	0.14	Audio+Image+Text
Rosie	0.08	0.08	Audio+Image+Semantic
Baseline	0.06	0.06	Audio+Image
EIHW	-	0.06	Audio+Image
Affective Bulls	0.02	0.03	Audio+Image

confirming the difficulty of Deep Learning models to generalize to new contexts for datasets recorded in a non-scripted (or semi-scripted) fashion. Furthermore it points out the difference of human and a SOTA Deep Learning performance in a continuous and highly detailed emotion estimation task.

3.4 Further Developments with the Audio Model

After the *OMG-Empathy Prediction Challenge 2018* submission we continued experimenting on the same dataset and tasks, but taking in consideration only the audio information extracted from the videos. We tried several variants of our GRU-based approach. The most significant ones are:

- Simplifying the network's task, splitting it in 3 distinct models. One model identifies the macro trend of the valence (positive, neutral or negative), another model predicts the slope of the valence trend and a third model combines the outcomes of the other 2 to obtain the final prediction.

- Optimizing the input features. We tried PCA whitening and a LSTM based autoencoder trained on samples of the training set. These models reduced the features dimension of the input data from 129 to 50.

TABLE 3.3: Architecture of our best CNN model

Layer Type	Depth	Filter size	Activation function
Convolution	20	16x12	tanh
Pooling	n/a	2x2	n/a
Convolution	28	8x12	tanh
Pooling	n/a	2x2	n/a
Convolution	40	5x7	tanh
Pooling	n/a	2x2	n/a
Dropout at 30%	n/a	n/a	n/a
Fully Connected	200	n/a	tanh

- Concatenating handcrafted features to the input spectrograms spectrograms: pitch estimation, pitch salience, high frequency content, spectral centroid.

- Concatenating the predictions of the audio model and the above mentioned audio features and using this new feature set as input of another GRU-based model.

None of these variants improves the test CCC result.

After this stage, we moved to a different network design, which is a sequence-to-sequence Convolutional Neural Network. Table 3.3 depicts the design of this architecture. The Filter Size column is shown in the form *time x frequency*. The filter dimensions in the first convolution layer are 10 ms per unit in the time axis and 62 Hz per unit in the frequency axis. We trained and evaluated this model in the same conditions of the audio model described above: train / validation / test split of the dataset, pre/post-processing, and training parameters. We obtained a test CCC of 0.16, which is 0.01 inferior than the result we obtained with the multimodal ensemble model and 0.08 superior than the isolated GRU-based audio model.

Next, we pretrained the same CNN model on the *OMG-Emotion Behavior Dataset* (Barros et al., 2018). This dataset was developed for a previous OMG challenge and it is very similar to the one used in this work. For details, please refer to Barros et al. (2018). From this dataset we keep only the valence

annotations, discarding any other type of emotional information. Furthermore, we extract only the audio information from the videos and we apply the same preprocessing described of our submission. With this dataset we obtained a test CCC of 0.005. We expected this result to be very low, since we did not optimize the model for this specific dataset. After this stage, we retrained the CNN architecture on the OMG-Empathy dataset, initializing the model weights with the same weights of the pretrained model. In the end we obtained a test CCC of 0.17. Therefore, with this sequence-to-sequence CNN architecture pretrained on the OMG-Emotion Behavior dataset, we obtained a state of the art test CCC using only audio information, comparable with the CCC obtained with the multimodal model that our team submitted for the challenge.

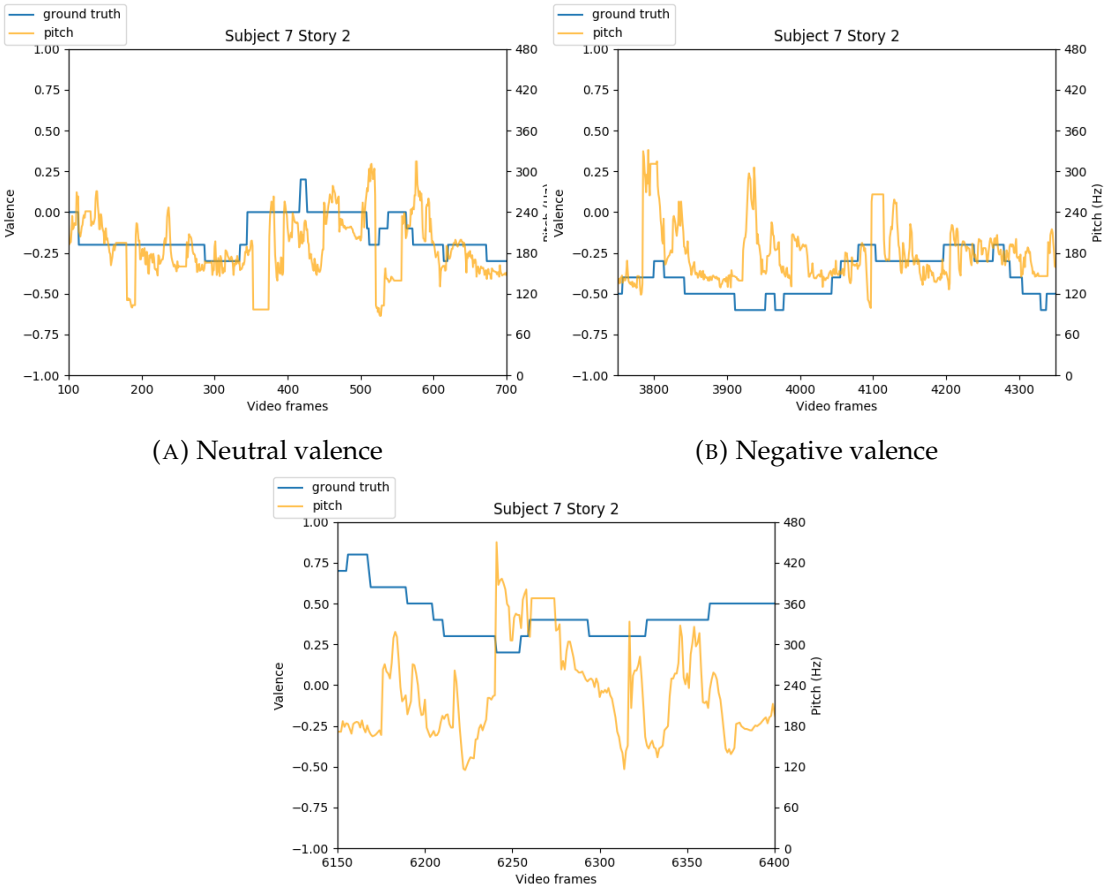
As described in the previous section, we tried to include the pitch information as input of the models, but this procedure did non improve the test CCC of our model. Nevertheless, we believe that a certain correlation between valence and pitch exists. Figure 3.4 shows a plot of the pitch estimation versus the valence for 3 audio segments taken from the same recorded session. We selected the fragments in order to show a possible correlation between pitch and valence in 3 windows with a different average valence measure. Figure 3.4a has a mean valence around 0, Figure 3.4b has a positive mean valence and Figure 3.4c has a negative mean valence.

In these figures a certain degree of correlation is visible. First, the global pitch trend moves towards the same direction on the valence measure. In fact, in the neutral sample the mean valence is -0.13 and the mean pitch is 186.3 Hz, in the positive sample the mean valence is 0.43 and the mean pitch is 230.9 Hz and in the negative sample the mean valence is -0.39 and the mean pitch is 177.9. In these examples a positive valence is associated to a higher pitch on average compared to a neutral state, while a negative valence is associated to a lower pitch on average. Furthermore, in certain regions of

valence transition a certain degree of analogy with the pitch trend is visible. In particular:

- *Neutral sample*: increasing between frames 300 and 400 and decreasing between frames 550 and 700.
- *Negative sample*: decreasing between frames 3900 and 400, increasing between frames 4050 and 4100 .
- *Positive sample*: decreasing between frames 6150 and 6230, increasing around frame 6250.

These examples suggest that the use of pitch information may have a potentially positive impact in the performance of SER classifiers.



(A) Neutral valence

(B) Negative valence

(C) Positive valence

FIGURE 3.4: Plot of pitch vs. valence in audio segments of the same recording session

3.5 Conclusions and remarks

In this Chapter we introduced a preliminary study on a context-independent SER scenario, presenting a solution to the *OMG-Empathy Prediction Challenge 2018*. We opted for a multimodal approach that processes independently audio, image and language information extracted from the original videos. Our solution ranked first, obtaining a CCC of 0.17.

As concerns audio-based SER, in this preliminary research stage we have not developed a novel technique. During the development of our solution, we experimented with standard deep learning architectures, focusing on trying different combinations of networks and fine tuning them to obtain the best possible results. However, this work has been fundamental to understand that context and speaker independence, as well as the generally high resource demand of models, are fundamental problems of SER. This encouraged us to further explore this field, aiming at finding novel ways to address the aforementioned problems and consequently defining the research questions of this dissertation.

Since the continuous valence prediction over time is not a common approach to SER, from here onward we decided to move to non-continuous SER classification problems. This permits to take advantage of different and more commonly used datasets, to compare our outcomes with existing works and results and to have more data to train and test our models.

Chapter 4

Improving Speaker-Independence: Multi-Time-Scale Modelling

This Chapter includes an adapted version of Conference Paper 3 (see section 1.5) and extends it with additional results and discussions.

In this stage, we focus on a speaker-independent SER scenario. Here models are expected to develop a certain degree of invariance to the speech style of speakers and generalize towards persons unseen in the training and validations set. We propose an approach that is based on a simple idea: the same words and sentences can be pronounced at different speeds, according to the speech style of a person, their cultural background and the context. Nevertheless, it is intuitive that also the emotion can depend to some extent on the speed of speech, for instance a *calm* utterance is likely to be pronounced slower, compared to the same utterance in an *angry* context. We hypothesize that resilience to a limited amount of speed fluctuations can help models generalize to unseen speakers, making the learnt features more robust to local dilation and compression on the time axis that can be due to the specific speech style of a person, rather than to the expressed emotion.

This Chapter, together with Chapter 3 addresses the second Research Question of this dissertation.

4.1 Introduction

To address the above-mentioned task and potentially other problems we introduce a Multiple-Time-Scale (MTS) Convolutional Neural Network architecture to create flexibility towards temporal variations when analyzing time-frequency representations of audio data, without augmenting the number of trainable parameters compared to standard convolution layers.

Convolutional Neural Networks (CNNs) have been extremely successful in recent years in a number of audio processing tasks, such as source separation, audio denoising, speech enhancement, speech and music transcription (Chandna et al., 2017; Jansson et al., 2017; Fu, Tsao, and Lu, 2016; Palaz, Collobert, et al., 2015; Zhao et al., 2015; Bittner et al., 2017). CNNs have also been extensively adopted for speech emotion recognition (SER) (Huang et al., 2014; Badshah et al., 2017; Mao et al., 2014; Trigeorgis et al., 2016; Lim, Jang, and Lee, 2016). Convolutional networks benefit from translation invariance of the processing on the time and frequency axis of a spectrogram or other time-frequency representations. However, in speech there are also variations in the speed of articulation between speakers and even of the same speaker in different situations. Therefore, allowing for matching the same kernel in multiple versions that are scaled differently on the time axis is the main idea in this work. We implement this in a self-contained layer architecture, the multi-time-scale (MTS) convolution layer, which increases the temporal flexibility in our networks compared to standard CNNs without increasing the number of model parameters. Separate treatment of dimensions is useful for speech processing with time-frequency representations, as opposed to image processing, where scaling is normally applied to both dimensions. MTS can be therefore considered as form of Dynamic Time Warping (DTW), even though the latter techniques are usually employed for sequence to sequence alignment tasks (Permanasari, Harahap, and Ali, 2019; Putri and Lestari, 2015;

Choi et al., 2020) and we test MTS only for sequence classification.

4.2 Method

Our approach is similar to Kanazawa, Sharma, and Jacobs (2014), as introduced in Section 2.2.1, but specifically adapted to the audio domain, where we analyse 2D magnitude spectrograms of speech audio. Since the time and frequency dimensions are of different nature in this representation, we treat them independently. Here, we focus on SER and address only time-scaling, while image processing techniques apply re-scaling to both dimensions with the same factor.

The core of our architecture is the multi-time-scale convolution layer (MTS), a custom 2D-convolution layer that can replace a standard convolution layer in a CNN design. The main feature of MTS is that it uses multiple versions of the learned kernel that are re-sampled on the time axis and performs parallel convolutions with them. This method enables the network to detect patterns at multiple time scales.

Figure 4.1 shows the architecture of one MTS layer with 3 parallel branches. In this example, the 2D spectrogram input, is convolved in parallel with the original kernel (in the center) and 2 time-stretched versions of the kernel (on both sides). The latter are generated by re-sampling the original kernel, applying linear interpolation. It is possible to independently apply different scaling factors for the 2 dimensions. These parallel convolutions produce 3 different feature maps, matching the feature of the original kernel at 3 different time scales. After this stage, the scaled feature maps are re-sampled again (applying linear interpolation) to match the shape of the original feature map. Then, a 3D max-pooling function is applied to merge the feature maps, selecting the scale with the maximal result in every time-frequency point. Therefore, the pooled feature map maintains the same dimension of

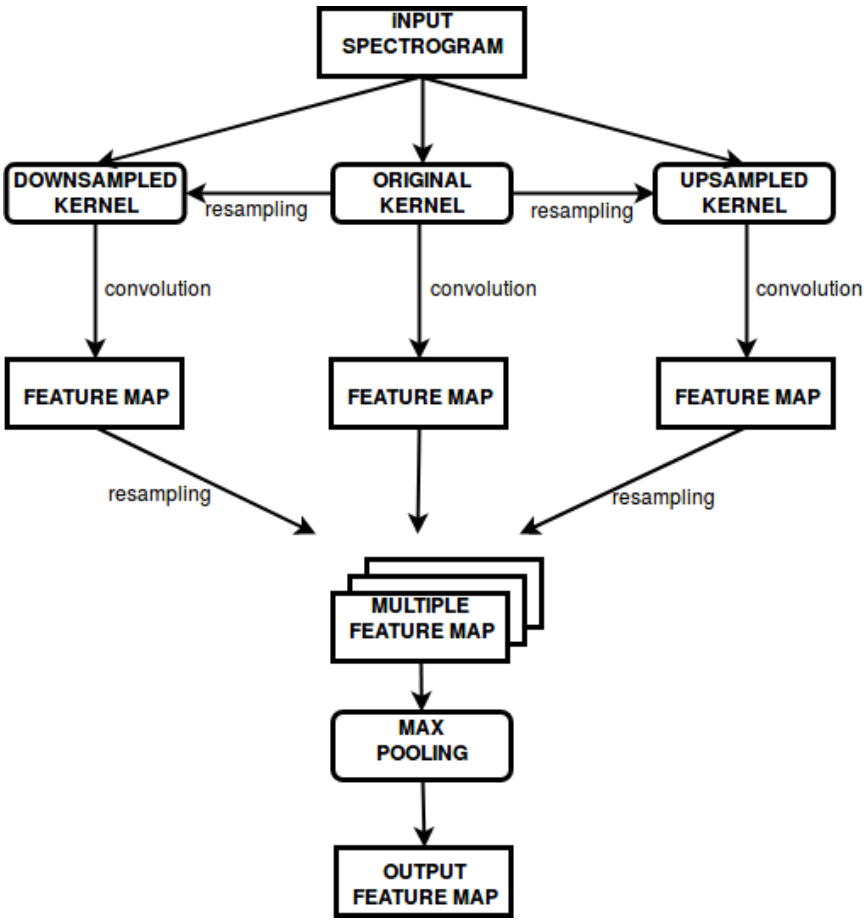


FIGURE 4.1: Example architecture of a Multi-Time-Scale convolution layer with 3 scale factors.

the feature map generated by the original kernel. During the training we average the weights of the original kernel and its scaled versions after each update. There is no constraint by design on the number of parallel branches that can be added to a MTS layer and MTS layers with different numbers of branches can be placed at various positions in the network. It is possible to fine-tune the scaling factors layer-by-layer. This approach provides a high degree of flexibility in the network design and enables scale invariance without increasing the number of free parameters. We have implemented this method in PyTorch and made it available as open source[1].

Our method is different from Kanazawa, Sharma, and Jacobs (2014) in

[1] `https://github.com/ericguizzo/multi_time_scale`

that it re-scales only one dimension and that we re-sample the kernels. Although re-sampling the data or kernel is equivalent in terms of results up to numerical variations, our method is somewhat more efficient. Moreover, Kanazawa, Sharma, and Jacobs (2014) augment test data by re-scaling. At least for SER tasks, we believe that this practice would not give a good estimate of the generalization capabilities of the models and thus we test without augmentation.

4.3 Evaluation

We have evaluated the performance of MTS on 4 benchmark datasets for speech emotion recognition:

1. EmoDb, a database of German emotional speech (Burkhardt et al., 2005). 10 speakers, German language, 535 utterances, 25 min of audio, 7 emotion labels: angry, bored, disgusted, anxious/fearful, happy, sad. Actors pronounce 10 different sentences which could be used in everyday communication.

2. RAVDESS, the Ryerson Audio Visual Database of Emotional Speech and Song (Livingstone and Russo, 2018b). 24 speakers, English language, 2542 utterances, 2:47 hours of audio, 8 emotion labels: happy, sad, angry, fearful, surprised, disgusted, calm, neutral. Actors pronounce 2 sentences: “Kids are talking by the door” and “Dogs are sitting by the door”.

3. TESS, the Toronto Emotional Speech Set (Dupuis and Pichora-Fuller, 2010). 2 speakers, English language, 2800 utterances, 1:36 hours of audio, 7 emotion labels: happy, sad, angry, disgusted, neutral, pleasant surprise, fearful. Actors say “Say the word ...” followed by 200 different words.

4. IEMOCAP, the Interactive Emotional Dyadic Motion Capture Database (Busso et al., 2008). 5 speakers, English language, 7529 utterances, 9:32 hours of audio, 10 emotion labels: neutral, angry, happy, excited, sad, frustrated, fearful, surprised, disgusted, other. Actors perform improvisations or scripted scenarios on defined topics.

For each dataset we keep only the audio information and the emotion labels, discarding any other types of data. We also discard the "song" data from RAVDESS. IEMOCAP is the only highly inbalanced dataset, therefore we removed the rarest labels from it, keeping only neutral, angry, happy and sad samples. Every sound file is pre-processed in 3 consecutive stages: resampling to 16 kHz, Short-Time Fourier Transform and normalization. For EmoDb, RAVDESS and TESS datasets every file is zero-padded to obtain equally-sized data. Since the IEMOCAP dataset contains longer recordings we segmented them into 4-second frames with 2-second overlap. The STFT is computed using 20 ms sliding windows with 10 ms overlap. Then, we normalize the magnitude spectra to zero mean and unit standard deviation.

TABLE 4.1: Accuracy results for all datasets. N ist the number of audio recordings per dataset. A1-4 are the network architectures.The best results per dataset are highlighted in bold font.

Dataset	N	Type	A1	A2	A3	A4
EmoDb	0.5k	Stand.	64.3	66.26	66.91	62.75
	0.5k	MTS	66.5	**70.97**	70.68	66.28
RAVDESS	1.4k	Stand.	42.09	39,84	42.56	47.41
	1.4k	MTS	47.85	44.95	51.32	**55.85**
TESS	2.8k	Stand.	47.45	49.6	50.61	40.78
	2.8k	MTS	51.76	48.75	**53.05**	51.71
IEMOCAP	5.5k	Stand.	48.93	50.48	49.0	54.96
	5.5k	MTS	49.0	50.84	49.86	**55.01**

We divide every dataset using approximately 70% of the data as training, 20% for validation and 10% as test set. Furthermore, we perform every experiment with 4-fold cross-validation. We make sure that samples from the same speaker appear only in the same set, in order to get a meaningful measure of

TABLE 4.2: Best scale factors used per each dataset. Usage factors of the parallel branches (last column) relate to scaling factors in the same row.

Dataset	Best scale factors	Use of parallel branches
EmoDb	n/a 0.7, 1, 1.428	n/a 0.47, 0.05, 0.48
RAVDESS	n/a 0.5, 1, 2	n/a 0.45, 0.06, 0.49
TESS	n/a 0.5, 0.7, 1, 1.428, 2.	n/a 0.41, 0.04, 0.05, 0.07, 0.43
IEMOCAP	n/a 0.5, 0.7, 1, 1.428, 2	n/a 0.39, 0.04, 0.04, 0.05, 0.48

the models' capability to generalize to new speakers, because new speakers are likely to produce patterns at different speeds. For this reason, our results are not directly comparable to most published results on the same datasets computed with randomly-split training, validation and test sets as, for example Zeng et al. (2019) and Verma and Mukhopadhyay (2016). Furthermore, many published results rely on different preprocessing types (Shegokar and Sircar, 2016; Schuller et al., 2005; Wang et al., 2015), other use multi-modal features rather than only audio (Schuller et al., 2005; Cho et al., 2018; Tripathi and Beigi, 2018) and other use different architectures (Shegokar and Sircar, 2016; Tripathi and Beigi, 2018; Cho et al., 2018; Verma and Mukhopadhyay, 2016; Schuller et al., 2005; Wang et al., 2015). For the same reason, we do not apply any kind of augmentation to the input data.

Rather than aiming at a state-of-art classification accuracy for these datasets, we focus on evaluating the performance of MTS layers compared to standard convolution with the same number of channels, i.e. without increasing the number of trainable variables. Therefore, we arranged our experiments in order to obtain consistent results within our set-up, with the same conditions for all datasets. We perform this comparison for 4 different CNN architectures with different capacity:

1. Convolution (1 channel, [10,5] kernel) - fully connected (200 neurons) - fully connected output layer.

2. Convolution (10 channels, [10,5] kernel) - fully connected (200 neurons) - fully connected output layer.

3. Convolution (10 channels, [10,5] kernel) - max pooling ([2,2] kernel) - convolution (10 channels, [10,5] kernel - fully connected (200 neurons) - fully connected output layer.

4. AlexNet: 5 convolutions and max pooling, 2 fully connected layers. See Krizhevsky, Sutskever, and Hinton (2012) for a detailed description.

The kernel dimensions above are in the form [time,frequency]. The activation function is ReLU for hidden and softmax for output units. In all experiments we use the ADAM optimizer with L2 regularization and Cross Entropy loss. We perform a grid search to find the best regularization parameter. We train for a maximum of 500 epochs, applying early stopping with 10 epochs patience for validation loss improvement. In architectures A1, A2 and A3, MTS is applied to all convolutional layers, while in A4 only the first 2 layers are augmented with MTS. We tested MTS with 3, 5 and 7 parallel branches, using logarithmically spaced scale factors in these combinations: (0.25, 1, 4), (0.5, 1, 2), (0.7, 1, 1.428), (0.8, 1, 1.25), (0.9, 1, 1.111), (0.95, 1, 1.053), (0.25, 0.5, 1, 2, 4), (0.5, 0.7, 1, 1.428, 2), (0.8, 0.9, 1, 1.111, 1.25), (0.25, 0.5, 0.7, 1, 1.428, 2, 4), (0.7, 0.8, 0.9, 1, 1.111, 1.25, 1.428). In each experiment, we apply the same combination of stretch factors to all MTS-enabled layers.

Table 4.1 shows the results we obtained for all datasets and all architectures. The first 3 columns show the dataset, total number of data points and the type of convolution layer(s). Columns A1-A4 show the mean test accuracy across folds obtained with each architecture (as listed above). Table 4.2 instead shows the best scale factors used for each dataset (second column)

and their average percentage of usage (last column). These values refer to the MTS model with the best accuracy in Table 4.1. The latter values are obtained computing how many pixels of each parallel feature map pass the Max Pooling layer, and thus go into the output feature map.

The results clearly show that MTS consistently improves the generalization for all datasets. We reach a maximum improvement of 8.04 percentage points (RAVDESS) and with an average of 3.78 with a standard deviation of 3.45 across all datasets and architectures. For all model/architecture combinations except one (A2 with TESS), MTS outperforms standard convolution. We performed a two-sided Wilcoxon signed-rank test comparing the standard and MTS results, which shows statistical significance with $p < 0.001$. The mean improvement is higher for the smaller datasets, which confirms that enabling pattern recognition at different time scales with MTS improves generalisation. Considering the general scarcity of emotion-labelled speech data, this is a desirable feature for SER applications.

The best-performing models on different datasets used different combinations of scaling factors. In particular, for the smaller datasets applying only 3 factors gives the best results. Architectures with 5 parallel branches perform better for the larger datasets. MTS models tend to use mostly 2 scale factors (see last column of table 4.2). In every case, at least 2 parallel branches give a high contribution, confirming that MTS is actually matching patterns at multiple time-scales.

We found that MTS is more effective at larger kernel sizes. In an experiment with an MTS version of ResNet18, where most kernels are very small (3x3), we achieved no improvement with MTS.

Training a MTS-enabled network generally takes longer than a standard CNN. In a test with architecture A2, it took on average 1.3 times longer per epoch to train MTS models with 3 branches and 1.52 times longer for MTS models 5 branches. Moreover, MTS networks need on average more epochs

to converge (27.85 vs 32.26 epochs for CNN vs MTS average overall).

We also tested modified variants of MTS:

- Applying a penalty to the re-sampled feature maps, to give the model a preference for the unscaled kernel.
- Performing the training using standard convolution layers and substitute them with MTS layers with shared weights only at inference time.
- Concatenating the used scaling factor for each time-frequency point to the output feature map of an MTS layer.

Each of these modifications reduced the performance of MTS models. Therefore, we kept the simplest variant described above.

4.4 Additional Results

We performed further experiments on MTS. In particular:

- We tested the performance of MTS augmenting a ResNet18 architecture (He et al., 2016)
- We tested MTS for a Speech Classification task, with 2 datasets of different size.
- We tested MTS for an image recognition task, with 2 popular datasets.

Testing MTS on ResNet18

In addition to the 4 architectures included in Table 4.1 as A1, A2, A3 and A4, we built a MTS version of a standard ResNet18 architecture, as described in He et al. (2016). In order to make our data compatible with the input dimension of ResNet18 we applied 2 additional steps to our preprocessing pipeline. First we re-sampled all spectra to a dimension of 256x256, applying linear interpolation. Then, we concatenated in the channels dimension 3 identical

versions of the reshaped spectra. We applied MTS to all convolution layers of ResNet18. For the rest, the experiments are conducted under the same conditions. Table 4.3 shows the results we obtained comparing standard convolution and MTS on ResNet18 for all datasets.

TABLE 4.3: Accuracy results with ResNet, for all datasets. N ist the number of audio recordings per dataset. Test accuracy refers to the best model in our grid search. The usage factors relate to scaling factors in the same row. The best results per dataset are highlighted in bold font.

D.set	N	Type	Test acc.	Scale factors	Use of parall.
EmoDb	0.5k	Stand.	**69.26**	n/a	n/a
	0.5k	MTS	52.5	0.7, 1, 1.428	0.47, 0.04, 0.49
RAVDESS	1.4k	Stand.	**46.64**	n/a	n/a
	1.4k	MTS	41.78	0.5, 1, 2	0.47, 0.05, 0.48
TESS	2.8k	Stand.	**46.45**	n/a	n/a
	2.8k	MTS	36.98	0.5, 1, 2.	0.46, 0.07, 0.47
IEMOCAP	5.5k	Stand.	**52.01**	n/a	n/a
	5.5k	MTS	50.6	0.5, 1, 2	0.47, 0.05, 0.48

In this case, the standard convolution layer always outperforms MTS, with a maximum improvement of 16.76 percentage points for the EmoDb dataset and 7.37 percentage points on average. Furthermore, it is evident a generally inverse trend between the dataset size and the improvement of standard convolution over MTS. In particular, the improvement is the highest for the smallest dataset (16.76 for EmoDb), is lower and similar for the 2 middle-sized datasets (respectively 4.86 and 6.67 for RAVDESS and TESS) and is the lowest for the biggest dataset (1.41 for IEMOCAP). However, the best accuracy with ResNet for all dataset is always lower than the best accuracy we reported with the A1, A2, A3 and A4 architectures. As discussed above, with these 4 architectures, MTS has shown an inverse trend compared to ResNet18, providing an improvement over standard convolution that is bigger for the smaller datasets.

We believe that this drop of performance of MTS with ResNet18 is due to the small size of the kernels in the convolution layers of ResNet18 (3x3

and 5x5). Since the implementation of MTS is based on the resampling of the kernels, it is possible that such small dimensions cause strong approximation when re-sampling, consequently lowering the resolution of the learned features. This suggests that a different implementation based on resampling of the input data of an MTS layer instead of resampling the convolution kernels could provide more high-resolution features. This approach has been used by Kanazawa, Sharma, and Jacobs (2014) with positive results on image recognition tasks.

Testing MTS on a Speech Recognition Task

Besides Speech Emotion Recognition, we tested the performance of MTS also on a Speech Recognition task. We used 2 datasets of different size:

- *Google Speech Commands* (GSC) (Warden, 2018). 2618 speakers. English language. 105829 utterances. Approximately 29 hours of audio. Each datapoint contains only one spoken word and is exactly 1-second long. There are 35 different words in the dataset.

- *Free Spoken Digits Dataset* (FSDD) (Jackson et al., 2018). 4 speakers. English language. 2000 utterances. Each datapoint contains only one spoken word (digits from 0 to 9) and is exactly 1-second long.

We performed this experiment in the same above-described conditions, but using only architectures A1, A2 and A3. The only difference lies in the kernel dimensions we adopted for the convolution layers. While for Emotion Recognition we used [10,5] kernels (in the form *time x frequency*) in all layers and architectures, for Speech Recognition we downsized them to a dimension of [5,5]. This choice was made because for Speech Recognition the smaller kernels provided higher accuracy on average. Table 4.4 shows the results we obtained in this experiment.

TABLE 4.4: Accuracy results for Speech Recognition tasks, for GSC and FSDD datasets. N is the number of audio recordings per dataset. A1-A3 are the architectures we tested. The usage factors of parallel branches (last column) relate to scaling factors in the same row. The best results per dataset are highlighted in bold font.

D.set	N	Type	A1	A2	A3	Scale factors	Use of parall.
FSDD	2k	Stand.	**62.24**	59.33	59.23	n/a	n/a
	2k	MTS	58.33	60.09	57.42	0.7, 1, 1.428	0.62, 0.02, 0.36
GSC	105k	Stand.	71.73	67.76	**75.94**	n/a	n/a
	105k	MTS	60.9	57.8	68.8	0.7, 1, 1.428	0.45, 0.11, 0.44

In all cases but one (FSDD dataset with architecture A2) the standard convolution provides a higher accuracy compared to MTS. The overall accuracy is higher with normal convolution, for both datasets. The highest improvement of standard convolution over MTS is for architecture A1 with the GSC dataset: 10.83 percentage points, while the average improvement is 5.48 among all dataset/architecture combinations. On average, taking into account all architectures for a dataset, the improvement is higher for the biggest dataset GSC, being 7.14 percentage points, while for the smallest dataset (FSDD) the improvement is 2.15 percentage points.

Also in this case, MTS was applied to relatively small kernels ([5,5]), therefore the same considerations exposed in the previous section do apply. Furthermore, the amount of scale invariance in the GSC and FSDD dataset is smaller with respect to the emotion-labelled datasets. In fact, these 2 datasets contain only 1-second samples with only one spoken word, while the other datasets contain more varied scenarios and generally longer recordings, in which there was not the restriction of fitting their utterances in a predefined length. In this circumstance the actors were more free to express their own speech style, consequently augmenting possible differences between recordings of different subjects.

Testing MTS on a Image Recognition Task

In order to asses the effectiveness of MTS also in a domain different from audio, we tested our approach on an image recognition task. We used popular datasets for this purpose:

- MNIST: 70 000 28x28 images in total, containing handwritten digits.
- CIFAR10: 60 000 32x32 colour images in total, containing 10 different classes of objects.

In this case, the preprocessing is different from what we applied for audio data. We first converted all images to gray-scale (1 single channel), then we normalized to 0 mean and unit standard deviation. For the rest, this experiments are performed in the same conditions, with the same architectures A1, A2 and A3. Table 4.5 shows the results we obtained.

TABLE 4.5: Accuracy results for Image Recognition tasks, for CIFAR10 and MNIST datasets. N is the number of images per dataset. A1-A3 are the architectures we tested. The usage factors of parallel branches (last column) relate to scaling factors in the same row. The best results per dataset are highlighted in bold font.

Dataset	N	Type	A1	A2	A3	Scale factors	Use of parall.
CIFAR10	60k	Stand.	20.37	34.01	23.91	n/a	n/a
	60k	MTS	22.26	**34.28**	26.24	0.7, 1, 1.428	0.47, 0.06, 0.47
MNIST	70k	Stand.	68.87	89.15	78.26	n/a	n/a
	70k	MTS	66.55	**90.74**	75.15	0.7, 1, 1.428	0.45, 0.13, 0.42

In this experiment the overall accuracy per dataset is higher when MTS is applied to the networks. In particular, the improvement of MST over standard convolution is 0.27 percentage points for CIFAR10 and 1.59 for MNIST, 0.93 on average for both. The average improvement in all dataset/architecture combinations is 0.1. On all cases but 2 (MNIST with A1 architecture and MNIST with A3 architecture) MST provides superior performance with respect to standard convolution.

This experiment demonstrated that MTS has a positive impact also on tasks related to the visual domain. Nevertheless, on image recognition the average improvement is inferior than the improvement given for Emotion Recognition from speech. This encourages further developments in this direction, in particular using architectures fine tuned for the visual domain and applying our method to more challenging datasets.

4.5 Conclusions and remarks

In this chapter, we propose multi-time-scale convolution layer (MTS) for CNNs applied to audio analysis, specifically emotion recognition from speech. The MTS performs parallel 2D-convolutions using a standard kernel and its re-sampled versions to match patterns at different time scales. This method enables the network to learn to some extent time-invariant features without increasing its number of trainable parameters or the number of training examples. We evaluated our approach on speech emotion recognition with unknown speakers, using 4 different datasets and applying it to networks of different size and structure. We found a consistent and statistically significant improvement in test accuracy across all datasets and models, up to 8.04 percentage points for RAVDESS and on average 3.78 across all datasets and architectures. MTS is particularly effective on smaller datasets, which makes MTS well suited for Speech Emotion Recognition where labelled data is scarce.

Despite the positive results that MTS provides on this scenario, it has also important drawbacks. The first and most critical is that its effectiveness is limited to networks that employ large convolution kernels, as shown in Chapter 4.4. This discourages the use of MTS with most popular and powerful architectures that are based on small convolution kernels, as the

ResNet and VGG network families. Another limitation is that the independence to speed fluctuations, and consequently to a speaker's speech style, that MTS achieves comes at the cost of increasing the resource demand. In fact, as exposed in Chapter 4.3, training with MTS layers take on average 1.3 times longer compared to the correspondent standard CNN, due to the additional convolutions operations (without increasing the number of trainable parameters). For these reasons, in the following Chapters of this dissertation we discontinued the use of MTS. Instead, we based our following research approaches on larger architectures based on small kernels, which generally provide better results and are more conveniently comparable with existing research and results.

Chapter 5

A Supervised Method for Task-Specific Invariance: Anti-Transfer Learning

This Chapter includes an adapted version of Journal Paper 2 (see Section 1.5).

In this stage of our research work we focus on a supervised approach to achieve invariance to a desired feature, when the latter can be extracted by a pretrained network. This is achieved through a novel training paradigm that we call **anti-transfer learning**. We apply this strategy to SER tasks, demonstrating that it can successfully improve speaker and context independence of SER models, consequently enhancing the models' performance. Moreover, we test anti-transfer learning also on word recognition and sound goodness estimation, demonstrating that this approach can successfully generalize to tasks and domains different from SER.

This Chapter addresses the third Research Question of this dissertation.

5.1 Introduction

In recent years, transfer learning has become a popular method in speech and audio processing to make use of existing deep learning models that have been trained on large datasets. The assumption underlying transfer learning

is that the internal representations learned to solve one task will be relevant for another task. This can improve the performance of a model in terms of training time and overall accuracy even across tasks and domains, and has been proven to be particularly useful in cases when data availability for the target task is limited (Oord, Dieleman, and Schrauwen, 2014; Bansal et al., 2019; Dieleman, Brakel, and Schrauwen, 2011; Wang et al., 2019c).

We introduce here the concept of anti-transfer learning, which is based on the idea that if a neural network can be used to teach another network what to do, it may also be used to teach what *not* to do. Based on the observation that some tasks may be irrelevant and confounding or undesirable to influence the target task, we try to avoid representations learned for one task when learning to solve another. We call the task which should not influence the predictions an *orthogonal* task, as our intention is that the predictions of our target should be independent of it. What constitutes an orthogonal task depends on the nature of the tasks and the intention of the user. We see two main application scenarios: first, improving generalization by discouraging reliance on spurious associations, e.g., word recognition and speaker identity, and second, discouraging undesirable bias, e.g. that gender or ethnicity should not influence financial decisions.

In this chapter we focus on the first scenario, and particularly on audio applications. Spurious correlations occur frequently in real-world data and are sometimes unavoidable. E.g., we expect the word 'joy' to be associated with a happy expression in natural speech. This association may be useful to resolve ambiguities, but a model overly reliant on this may not generalize in cases where the association does not hold, e.g. the word 'joy' pronounced with a sad expression. Similarly, the frequency of word use is not equally distributed between different speakers, genders or cultures, but we would prefer our models not to depend on these features when they recognize words, both in the interest of generalization and in avoidance of bias

or stereotyping. This problem could be addressed by creating or collecting more data, that contains all variants of emotional expressions for all words, or all words uttered by all speakers but this not practical in general. However, with anti-transfer we can discourage the use of emotional features for word recognition, or speaker identity for emotion recognition, respectively, and thus avoid that dependency and improve generalization from limited datasets.

Anti-transfer can be used to address open research problems in speech and audio processing, such as speaker or context invariance in word or emotion recognition (Kitza, Schlüter, and Ney, 2018; Milde and Biemann, 2018; Liu et al., 2019b; Jalal, Moore, and Hain, 2019; Rybka and Janicki, 2013b; Bhaykar, Yadav, and Rao, 2013). In our experiments, we compare anti-transfer learning to regular transfer learning and learning from scratch on speech and music audio tasks. A common approach for transfer learning with deep learning models is to use a pretrained network as starting point through weight initialization, i.e. re-training a pretrained network or part of it (Tan et al., 2018b). Support for this approach is built into popular machine learning libraries, such as *Tensorflow*[1] and *PyTorch*[2], along with models pretrained on disparate tasks. In anti-transfer learning, we penalize instead the use of features that have been learned for the orthogonal task when training for the target task. Our results show that this leads to greater invariance to the target predictions from the orthogonal task and improves the generalization of the models.

5.1.1 Differences from Similar Techniques

The approach presented in this chapter is inspired by the work of Gatys, Ecker, and Bethge (2016) on style transfer on images and re-uses elements of

[1] `https://www.tensorflow.org/`
[2] `https://pytorch.org/`

that work, in particular, the use of deep feature losses. However, our main idea to perform the opposite. That is: discouraging specific deep representations that have been particularly useful for a task that is irrelevant for a target task and should thus be avoided when training for the target task, in order to not develop spurious correlations.

We actually address similar problems with a similar approach as Beckmann et al. (2019) and Kegler, Beckmann, and Cernak (2019) but while they maximize the similarity with representation of a pretrained network, our aim is to minimize it. There are also substantial differences in the implementation as we use Gram aggregation and a different similarity measure, as explained in Section 5.2. When viewing orthogonal tasks as domains, we can compare domain adaptation (Wang, Liu, and Wang, 2019) to anti-transfer. Although, this is different from our approach in terms of applications, as we are training to transfer between tasks, within or between domains.

Moreover, there is a parallelism between anti-transfer and Domain Adversarial Training (Ganin et al., 2016; Motiian et al., 2017; Tzeng et al., 2017; Drossos, Magron, and Virtanen, 2019). However, there are two notable differences between these approaches and ours: first, we directly compare the feature activations in our loss function as opposed to propagating gradients derived from domain labels, and second, most of these approaches require labeled data from the source domain (analog to our orthogonal task), while anti-transfer only requires a pretrained model, which does not have to be trained on the same dataset.

Finally, a connection can be observed between anti-transfer learning and Invariant Risk Minimization, being both aimed at obtaining a certain degree of independence of predictions from the data domain (referred as environment for IRL). As DAT, IRL benefits from a unified training stage that directly solves the final task obtaining domain invariance predictions. Nevertheless, this is only possible under the assumption that multiple domains are present

in the training data (even though domain labelling is not strictly required for IRL). On the other hand, as specified specified above, since anti-transfer requires only a pretrained model, it enables to obtain domain invariant predictions even using a dataset collected on a single domain.

5.2 Method

The main idea of anti-transfer learning is to encourage dissimilarity of a model's deep representations with respect to another model with the same architecture but pretrained on an orthogonal task. We focus here on CNNs which have been immensely popular in recent years and achieve state of the art results on many audio tasks, e.g. Jansson et al. (2017), Neumann and Vu (2017), and Salamon and Bello (2017b).

5.2.1 Approach

We achieve anti-transfer learning through the introduction of an anti-transfer loss term during training, that is a deep feature loss (Dosovitskiy and Brox, 2016). The anti-transfer loss measures the similarity between the deep representations that the network is learning and a pretrained network with the same architecture. By adding this term as a penalty to the loss function we encourage the trained network to develop deep representations that are different from the pretrained network. In other words, we encourage the network being trained to develop feature representations that are good for its target task but different from those developed to solve the orthogonal task in the pretrained network. This reduces the trained network's dependency on the orthogonal task's classes, e.g. the dependency of word recognition on speaker identity.

Figure 5.1 depicts a block diagram of a generic CNN with *anti-transfer* learning applied. As the diagram shows, this architecture has two parallel

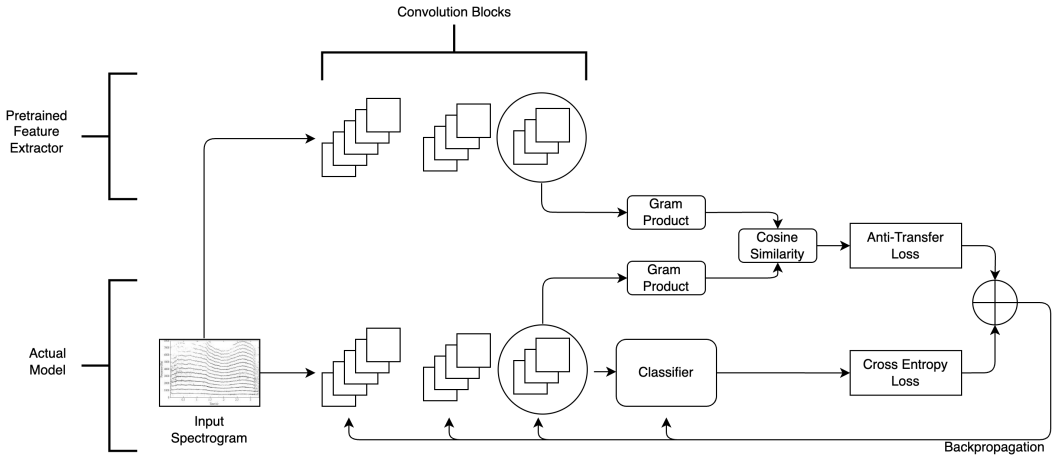


FIGURE 5.1: Block diagram of a CNN network with **anti-transfer learning** applied to a classification task. We use spectrograms of audio signals as the input, but anti-transfer is not specific to the audio domain or spectrogram representations.

networks: a *pretrained feature extractor* (in the upper part), which is the convolutional part of the pretrained network, with non-trainable weights and the CNN classifier that is currently being trained (in the lower part).

Our implementation is based on the VGG16 Architecture (Simonyan and Zisserman, 2015), a deep CNN, with details shown in Table 5.1. We selected this architecture since it has been proven to be effective in computing a deep feature loss in the audio domain (Beckmann et al., 2019). Nevertheless, the same concept and implementation can be translated to any other CNN design.

5.2.2 Anti-Transfer Loss

The anti-transfer loss is computed in the forward pass. The input data, a spectrogram in our experiments, is forward propagated in parallel through both networks. The feature maps of the n^{th} convolution layer in both networks are extracted and aggregated in the channel-wise Gram matrix G, which is computed for each network, similarly to the approach used by Gatys, Ecker, and Bethge (2016) to compute the style matrix of an image. The Gram

TABLE 5.1: The **VGG16** architecture. In this example configuration the network has an input dimension of 244x244x1 and 1000 output classes.

Layer	Channels	Size	Kernel	Stride	Activation
Input	1	244x244	-	-	-
2x Convolution	64	224x224	3x3	1	relu
Max Pooling	64	128x128	3x3	2	relu
2x Convolution	128	224x224	3x3	1	relu
Max Pooling	128	56x56	3x3	2	relu
2x Convolution	256	56x56	3x3	1	relu
Max Pooling	256	28x28	3x3	2	relu
3x Convolution	512	28x28	3x3	1	relu
Max Pooling	512	14x14	3x3	2	relu
3x Convolution	512	14x14	3x3	1	relu
Max Pooling	512	7x7	3x3	2	relu
Fully Connected	-	25088	-	-	relu
Fully Connected	-	4096	-	-	relu
Fully Connected	-	4096	-	-	relu
Output	-	1000	-	-	softmax

matrix is computed as the inner product between the vectorized feature maps F for each pair of channels:

$$G_{ij} = F_i \cdot F_j. \tag{5.1}$$

where i, j are the channel numbers. The Gram matrix correlates the information of each channel pair over all points x, y, consequently reducing the dimensionality of a feature map from 3 dimensions, (c, x, y), to 2, (c, c), where c, x, y are the number of channels, rows and columns, respectively. We then calculate the anti-transfer (AT) loss L_{AT} as a scalar coefficient β multiplied by the squared cosine similarity of the vectorized Gram matrices G_p (for the pretrained net) and G_t (for the net being trained):

$$L_{AT} = \beta \left(\frac{G_p \cdot G_t}{||G_p||\,||G_t||} \right)^2. \tag{5.2}$$

The aggregation with the Gram matrix serves to compare all possible channel combinations at once, using a limited amount of memory. This is essential

for consistently measuring the similarity of the feature maps, where permutations can occur along the channel dimension. We choose the squared cosine similarity since it is naturally limited in the interval [0,1] and therefore it can have only a limited impact in the overall loss function. Moreover, we square it to apply a stronger penalty when the similarity is high and we re-scale by the coefficient β as an hyperparameter to fine-tune the performance of AT learning.

The diagram in Figure 5.1 shows the AT loss calculated on the last convolution layer, but it is possible to apply the the AT loss to any of the convolution layers. Furthermore, it is possible to combine the AT loss of multiple layers in the same training, summing their AT loss values. The total AT loss is added to the standard loss function during the training of the network (cross entropy in our case, but AT can be used with any loss function).

The complete objective function we minimize per datapoint is therefore:

$$L_{TOT} = -\sum_{i=1}^{n} t_i \log(p_i) + \sum_{s \in S_{AT}} L_{ATs} \tag{5.3}$$

where n is the number of classes, t_i is 1 if i is the true class and 0 otherwise, p_x is the predicted probability of class i, S_{AT} is the set of convolution layers where anti-transfer is computed, and L_{ATs} is the anti-transfer loss computed for convolution layer s.

5.2.3 Variations

As we present in Section 5.4, we test several aggregation strategies and similarity measures. The best combination is Gram matrix aggregation and squared cosine similarity, which is detailed above. Different aggregation and similarity functions can be used by adapting equations (5.1) and (5.2).

Moreover, we combine two orthogonal tasks in *dual AT* loss. To achieve this, we first train a model with anti-transfer for one orthogonal task. We use

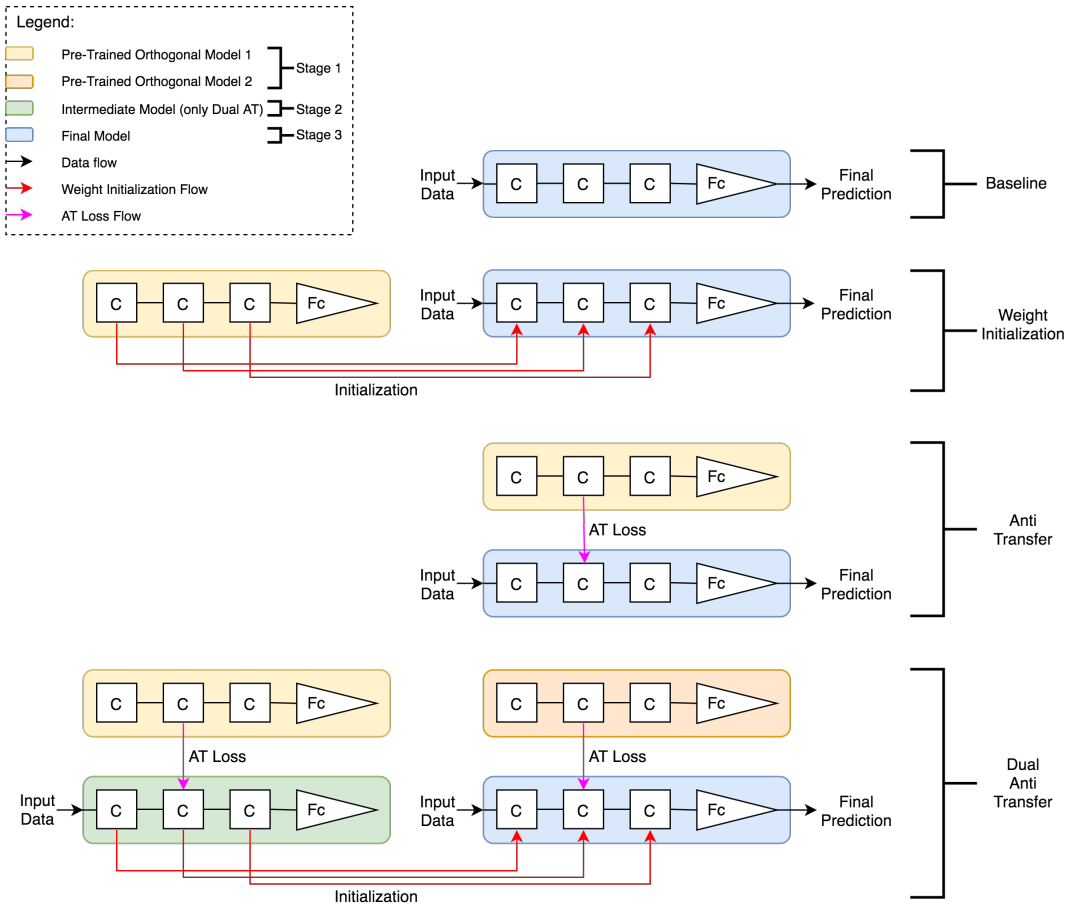


FIGURE 5.2: Block diagram of our **training strategies**. The color coding reflects 3 consecutive temporal stages. Stage 1: pretraining of the orthogonal models (yellow and, only for dual AT, orange). Stage 2: only for dual AT, training of the intermediate model applying AT (green). Stage 3: training of the final models (blue) applying different transfer learning strategies: no transfer (baseline), weigh-initialization of the convolution layers, anti-transfer and dual anti-transfer. Different information flows are represented with differently colored arrows: the data flow is shown in black, the weight-initialization flow in red and the AT loss flow in magenta.

the result of that training to initialize the weights of a new model, which is then trained with anti-transfer on the second orthogonal task. It is worth noting that we apply the weight initialization to all convolution layers at once, while we apply anti-transfer to only one convolution layer per experiment.

5.3 Experimental Set-up and Results

We test anti-transfer learning on several audio classification tasks with 20 different combinations of training and pretraining tasks in order to evaluate the behavior of anti-transfer learning in a variety of set-ups. We have three main classification tasks: word recognition (WR), speech emotion recognition (SER) and sound goodness estimation (SGE) (i.e. how well musical notes are played by musicians (Romani Picas et al., 2017)).

SER and SGE tasks are evaluated with two types of splitting the dataset into training, validation and test set: random split and class split by speaker or instrument. The class split types provide a more challenging task than the random split. This is because these (orthogonal) classes reflect different data distributions in the random split the training, validation and test set distributions are the same. On the other hand, splitting by speaker or instrument presents a more realistic task for many applications. The class-split is based on labels used in the orthogonal tasks (see Section 5.3.1). This enables us to assess more directly the AT trained networks' invariance to the orthogonal task classes as discussed in Section 5.4.8. For WR, we use only random split, but we added different types of background noise to the audio samples to create more challenging classification tasks. We test 3 scenarios: noise-free, low noise and high noise (see below for details).

Our experiments are set up to test the effectiveness of anti-transfer learning, comparing it to the most common transfer learning method of weight initialization (WI) and to a baseline method without any transfer learning. In this way we can compare anti-transfer to regular transfer learning in the specific case of pretraining on orthogonal tasks. In addition, we perform two further experiments (presented in Section 5.4.1). In the first one we freeze the convolution layers in the WI modality up to the same layer where we apply

the AT loss. This avoid possible dissipation of prior knowledge when training. In the second experiment we invert β in the AT loss, so that similarity of feature activations is encouraged instead of dissimilarity, i.e. performing the opposite of regular AT. Figure 5.2 shows a diagram of the different training strategies we compared. We perform 3 consecutive training stages. First, we pretrain the models on the orthogonal tasks. Then, only for dual AT, we apply AT to train an intermediate model on the final task. The weights of the intermediate model are then used to initialize the final model. Finally, we train our final models, applying the different transfer learning strategies.

5.3.1 Datasets

We use six different datasets overall. For our experiments, we extract subsets from larger datasets to reduce training times and adjust class imbalances. While this limits comparability to published results, it enabled us to perform a much broader range of experiments as reported in this and the following section.

1. *Google Speech Commands*: A Dataset for Limited-Vocabulary Speech Recognition V2 (GSC) (Warden, 2018). Task: Single-word speech recognition. 2.7 hours of audio. 10 different words (digits) recorded by more that 2000 non-professional speakers in various acoustic environments.

2. *MS-SNSD*: The Microsoft Scalable Noisy Speech Dataset. A dataset and online subjective test framework (Reddy et al., 2019). Approximately 20 hours of audio. Task: background noise type recognition. 11 different types of noise mixed with speech audio signals at volumes between -20 and -40 dBfs.

3. *Librispeech*: An ASR corpus based on public domain audio books (Panayotov et al., 2015). Task: Single-word speech recognition. 100 hours of

audio, 40 speakers, 1000 single-word labels. One-word excerpts from audio book recordings.

4. *IEMOCAP*: The Interactive Emotional Dyadic Motion Capture Database (Busso et al., 2008). Tasks: speech emotion recognition, speaker recognition. 7:30 hours of audio, 5 speakers, 4 emotion labels: neutral, angry, happy, sad. Actors perform semi-improvised or scripted scenarios on defined topics.

5. *Nsynth*: A large-scale, high-quality dataset of annotated musical sounds. (Engel et al., 2017). Task: instrument recognition. 66 hours of audio. 11 different instrument macro-categories. One-note recordings of musical instruments.

6. *Good-Sounds*: A dataset to explore the quality of instrumental sounds (GS) (Romani Picas et al., 2017). Tasks: sound goodness estimation, instrument recognition. 14 hours of audio. 12 different instruments, 5 different goodness rates. One-note recordings of acoustic musical instruments, played by professional musicians.

The above descriptions refer to the subsets we extracted (or generated, for MS-SNSD), not to the original size and arrangement of these datasets. Please refer to the references above for the original specifications.

For each target task, we pretrain on two different tasks for transfer and anti-transfer learning. For word recognition we train on GSC and we pretrain on speech emotion recognition (IEMOCAP) and on background noise type recognition (MS-SNSD). For speech emotion recognition we train on IEMOCAP and we pretrain on speaker recognition with the same training dataset (IEMOCAP) and on word recognition with a larger dataset (Librispeech). For sound goodness estimation we train on Good-Sounds, we pretrain on instrument recognition with the same training dataset (Good-Sounds) and with a larger dataset (Nsynth).

5.3.2 Processing Stages, Training Parameters and Training Strategies

We paid particular attention to performing all experiments (trainings and pretrainings) in the same conditions, in order to isolate the influence of anti-transfer and weight initialization in the results. All experiments are performed in a Python and PyTorch environment, using the VGG16 network architecture (Simonyan and Zisserman, 2015) (in the implementation from the *torchvision* library[3]).

We apply two architectural modifications to the standard implementation: we reduce the channel number of the very first layer to 1 (since we use single-channel magnitude spectrograms) and we vary the number of output neurons to match the classes to the task.

We apply the same preprocessing to all datasets:

1. We first down-sample all audio data to 16kHz sampling rate.

2. Then we zero-pad/segment all sounds in order to have data-vectors of the same length for each task. We segment the audio as follows:

    - In the word recognition target task, we use 1-second sound samples as provided in the GSC. For the orthogonal noise classification task, we first generate approximately 20 hours of noisy speech from MS-SNSD and then we extract 1-second fragments with no overlap. For emotion recognition, we extract 1-seconds fragments from IEMOCAP.
    - In the speech emotion recognition target task, we use 4-seconds sound samples from IEMOCAP. For the orthogonal task of word recognition, we extract segments containing only one word[4] from Librispeech and then zero-pad them to 4-seconds.

[3] `https://github.com/pytorch/vision/blob/main/torchvision/models/vgg.py`
[4] We use `https://github.com/bepierre/SpeechVGG` for this.

- In the sound goodness recognition target and the orthogonal instrument recognition task, we use 6-second sounds, applying zero-padding to both Nsynth and Good-Sounds sounds.

3. Only for GSC, we add noise to the segmented speech sounds at 3 different levels: no noise, low noise (-40 to -20 dBfs) and high noise (-10 to 0 dBfs). The noise sounds are from the MS-SNSD datasets. Like for MS-SNSD we use the MS-SNSD code[5] to perform this operation.

4. Next we compute the Short-Time-Fourier-Transform (STFT) using 16 ms sliding windows with 50% overlap, applying a Hamming window and discarding the phase information.

5. Finally, we normalize the magnitude spectra of each dataset to zero mean and unit standard deviation, based on the training set's mean and standard deviation.

We perform all neural network trainings and pretrainings with the same hyperparameters. We use a learning rate of 0.0005, a batch size of 13 and the ADAM optimizer (Kingma and Ba, 2014). We apply dropout at 50% but neither L_1 nor L_2 regularization. We randomly initialize the weights of all networks, except in the case of weight initialization from a pretrained network (for WI and dual AT). We train for a maximum of 50 epochs and apply early stopping by testing at the validation loss improvement with a patience of 5 epochs. We divide every dataset using subsets of approximately 70% of the data for training, 20% for validation and 10% for the test set. All of the above settings are kept constant for all datasets in all configurations: non-transfer, transfer, anti-transfer/dual anti-transfer and also for all pretrainings.

These experiments are not designed to produce state of the art results on these datasets, because we want to focus on the impact of anti-transfer learning. Therefore we used specific subsets and we did not optimise network

[5] `https://github.com/microsoft/MS-SNSD`

TABLE 5.2: Results of the **pretraining** in terms of classification accuracy. *Classes* is the number of different class labels. *Hours* describes the amount of recorded material in the subset that we used. The *Train* and *Test* columns contain the accuracy on the train and test sets.

Dataset	**Task (Recognition)**	**Classes**	**Hours**	**Accuracy**	
				Train	**Test**
Librispeech	Speech	1000	100	97.6	91.8
IEMOCAP (1 sec)	Speech Emotion	4	7.3	85.6	51.9
IEMOCAP (4 sec)	Speaker	5	7.3	99.8	96.5
Good-Sounds	Instrument	12	14	100.0	100.0
Nsynth	Instrument	11	66	98.1	69.9
MS-SNSD	Noise Type	11	20	100.0	99.8

TABLE 5.3: Accuracy results for the **word recognition (WR)** target task on the Google Speech Commands (GSC) dataset with 3 levels of background noise added: None, Low and High. We pretrain on noise type recognition (Nse) with MS-SNDS dataset (MSS) and speech emotion recognition (Emo) with IEMOCAP dataset (IEC). We compare between no transfer learning (None), regular transfer learning by weight initialization (WI), anti-transfer (AT) and dual anti-transfer (Dual AT, using two pretraining tasks). The order of the pretraining tasks is shown in the second column. The best results per column are highlighted in bold font.

Transfer	**Pretraining**		**Train accuracy Noise level**			**Test accuracy Noise level**		
Type	**Task**	**Data**	**None**	**Low**	**High**	**None**	**Low**	**High**
None	n/a	n/a	98.45	97.94	97.23	95.32	93.67	90.44
WI	Noise	MSS	98.33	97.85	96.34	94.83	93.97	90.51
WI	Emo	IEC	98.67	97.69	97.36	95.40	93.51	90.35
AT	Noise	MSS	99.57	99.11	98.42	95.70	94.81	90.99
AT	Emo	IEC	99.02	99.09	98.36	95.57	94.91	**91.38**
Dual **AT**	Emo + Nse	IEC + MSS	**99.84**	**99.49**	98.29	**96.60**	94.91	90.98
Dual **AT**	Nse + Emo	MSS + IEC	99.31	99.17	**98.89**	95.64	**95.20**	90.67

architectures and hyperparameters to the individual datasets in order to exclude any other sources of performance variation.

TABLE 5.4: Accuracy results for the **speech emotion recognition (SER)** target task on the IEMOCAP dataset. Comparison between no transfer learning (None), weight initialization (WI) and anti-transfer (AT) with pretraining on different datasets. In particular, we compared anti-transfer with pretraining on the same dataset (IEMOCAP) but on an orthogonal task (speaker recognition) and on a bigger dataset (Librispeech) on a different orthogonal task (word recognition). We test 2 different train/ validation/ test split: random (Rand) and speaker-wise (Speaker). The best results per column are highlighted in bold font.

Transfer	**pretraining**		**Train accuracy Split Type**		**Test accuracy Split Type**	
Type	**Task**	**Dataset**	**Rand**	**Speaker**	**Rand**	**Speaker**
None	n/a	n/a	69.0	67.8	63.7	57.2
WI	Word	Librispeech	66.9	66.9	63.4	59.2
WI	Speaker	IEMOCAP	70.7	66.9	64.8	58.5
AT	Word	Librispeech	72.0	68.6	**66.9**	61.1
AT	Speaker	IEMOCAP	**75.5**	**74.5**	66.5	**61.3**

5.3.3 Classification Results

Table 5.2 shows the results of the pretraining in terms of classification accuracy. There is wide variation in performance on the different tasks, with the Good-Sounds and MS-SNSD saturating or almost saturating on the train and test set for instrument and background noise type recognition.

Tables 5.3, 5.4 and 5.5 show the results obtained on the target tasks of word recognition, speech emotion recognition and sound goodness estimation, respectively. These tables contain the baseline results without transfer learning (None), with standard transfer learning using weight initialization (WI) and with anti-transfer learning (AT) on 20 pre-task/actual-task combinations in total. While for SER and SGE we test only anti-transfer with one orthogonal task at a time, for WR we additionally test dual anti-transfer (Dual AT), applying two orthogonal tasks as described in Section 5.2.

We applied anti-transfer to one layer of the VGG16 network with each of the 13 convolution layers for each task. The reported anti-transfer test accuracy results reflect the choice of layer that reached the best validation

TABLE 5.5: Accuracy results for **sound goodness estimation (SGE)**. For the target task we use the Good-Sounds dataset. We compare no transfer learning (None), weight initialization (WI) and anti-transfer (AT). In particular, we compare anti-transfer with pretraining on the same dataset (Good-Sounds) and on a bigger dataset (Nsynth). We test 2 different train/validation/test splits: random (Rand) and instrument-wise (Instr). The best results per column are highlighted in bold font.

Transfer	**pretraining**		**Train accuracy Split Type**		**Test accuracy Split Type**	
Type	**Task**	**Dataset**	**Rand**	**Instr**	**Rand**	**Instr**
None	n/a	n/a	91.8	42.2	83.8	22.8
WI	Instrument	Nsynth	93.4	40.5	84.7	29.6
WI	Instrument	Good-Sounds	93.3	**42.3**	84.9	23.9
AT	Instrument	Nsynth	**96.8**	41.0	**86.3**	30.0
AT	Instrument	Good-Sounds	93.9	36.4	85.7	**34.3**

accuracy. In all experiments, the coefficient β is fixed to 1 since, as further analyzed in Section 5.4.6, this provides the best accuracy results.

The results (Tables 5.3, 5.4 and 5.5) show that anti-transfer improves the test accuracy in all cases and interestingly improves also the training accuracy in all cases but one (sound goodness estimation with instrument-wise split dataset, Table 5.5), compared to both the baseline and weight initialization. We have a maximum improvement in the test accuracy of 11.5 percentage points (pp) (for sound goodness estimation with instrument-wise split dataset, Table 5.5) and a maximum improvement in the training accuracy of 6.7 pp (for speech emotion recognition with speaker-wise split dataset, Table 5.4). The overall average improvement is of 4.11 pp for the test accuracy and of 2.35 pp for the training accuracy.

We performed a two-sided Wilcoxon signed-rank test comparing the baseline results obtained without AT and transfer learning by weight initialization and the results obtained with AT applied, which shows statistical significance with $p < 0.001$.

Figure 5.3 shows the average gain achieved by anti-transfer learning in

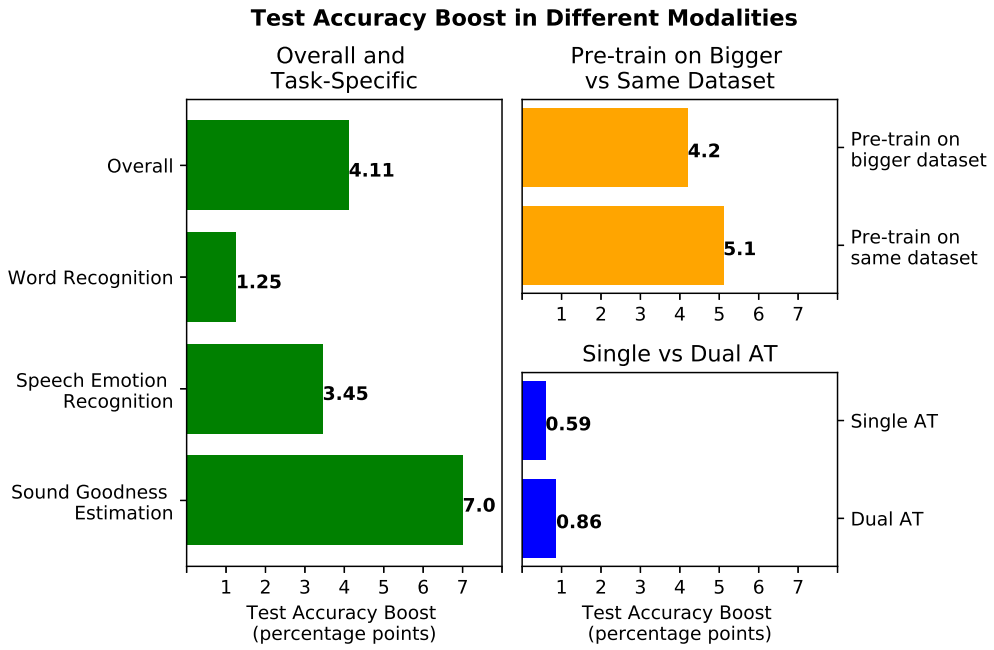


FIGURE 5.3: Average **improvement** by applying anti-transfer learning on different applications and different settings compared to the baseline (no transfer learning). The overall and the task-specific measures (word recognition, speech emotion recognition, sound goodness estimation, in green) show the average over the best improvements on each task/split or task/noise level configuration. The other measures show the average improvement over all experiments of a modality. The pretraining on bigger/same dataset modality (orange lines) is computed for Good-Sounds pretrained on NSynth and itself and for IEMOCAP pretrained on Librispeech and itself. The single/dual AT modality (blue lines) is computed for the Google Speech Commands dataset pretrained on MS-SNSD and IEMOCAP.

the test accuracy for different tasks and settings. It has practical relevance that the improvement in the networks' generalization is higher when anti-transfer is applied with a feature extractor trained on an orthogonal task with the same dataset as opposed to a different but larger dataset (we tested this property only on SER and SGE: IEMOCAP pretrained on speaker recognition vs IEMOCAP trained on speech emotion recognition and Good-Sounds pretrained on instrument recognition vs Good-Sounds trained on sound goodness estimation).

Another interesting aspect is that using dual anti-transfer provides a higher

accuracy boost compared to anti-transfer on a single orthogonal task (we tested this only on WR: GSC pretrained on speech emotion recognition and background noise type recognition). This suggests that the task invariance effect of anti-transfer learning can be cumulative, opening the possibility of pretraining on multiple orthogonal tasks.

5.4 Analysis and Discussion

The results in the previous section show a robust improvement resulting from the use of anti-transfer learning. Here we investigate various aspects of the method for understanding and optimizing its performance.

5.4.1 Ablation Study: Encouraging Similarity vs. Dissimilarity

As an ablation study, we performed additional experiments where we encourage the models to develop representations that are similar instead of dissimilar to the models pretrained on orthogonal tasks. The results are shown in Figure 5.4. We tested two methods for encouraging feature similarity. The first consists of inverting the sign of the β hyperparameter to encourage similarity instead of dissimilarity through the AT loss. This operation can be considered as the opposite of the regular AT (in line with Beckmann et al. (2019)). The second consists of weight initialization and freezing (i.e. setting as not trainable) all convolution layers from the input layer of the network up to the layer where we apply AT. This test is complementary to the comparison between AT and WI, since in regular WI the knowledge transferred from the pretrained model may completely disappear during the training because of the *catastrophic forgetting* phenomenon (Parisi et al., 2019). This experiment shows the model's performance when we avoid this phenomenon

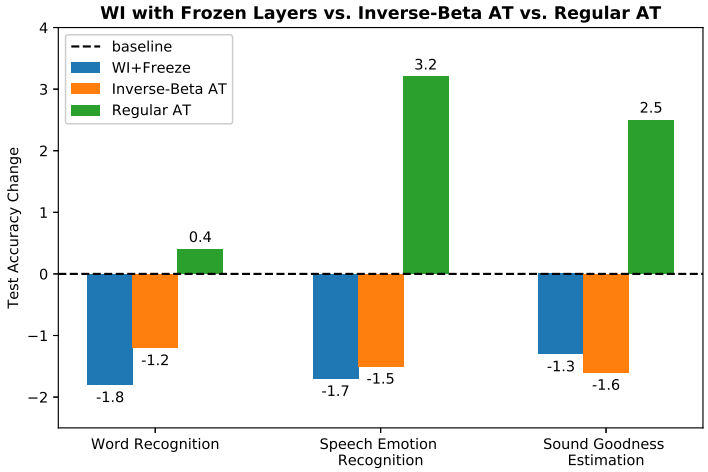


FIGURE 5.4: Comparison of **regular AT** (encouraging feature dissimilarity with the orthogonal task, green columns), **inverse-beta AT** (encouraging feature similarity, orange columns) and **weight initialization** with **frozen** convolution layers (until the same layer where we apply AT, blue columns) on all target tasks: word recognition, speech emotion recognition and sound goodness estimation. The improvement in the test accuracy (percentage points) is shown, comparing to the baseline results (no AT nor WI applied, black segmented line: 95.3% for word recognition, 63.7% for speech emotion recognition, 83.8% for sound goodness estimation).

by freezing the initial layers. We performed these two experiments using the task/orthogonal-task/AT-layer combination that yielded the best performance in each case, which are:

- Word recognition: GSC with no further noise added, background noise type recognition pretraining (MS-SNSD), layer 5.
- Speech emotion recognition: IEMOCAP random split, word recognition pretraining (Librispeech), layer 5.
- Sound goodness estimation: GS random split, instrument recognition pretraining (NSynth), layer 6.

The results show that both inverse-Beta-AT and freeze-WI configurations

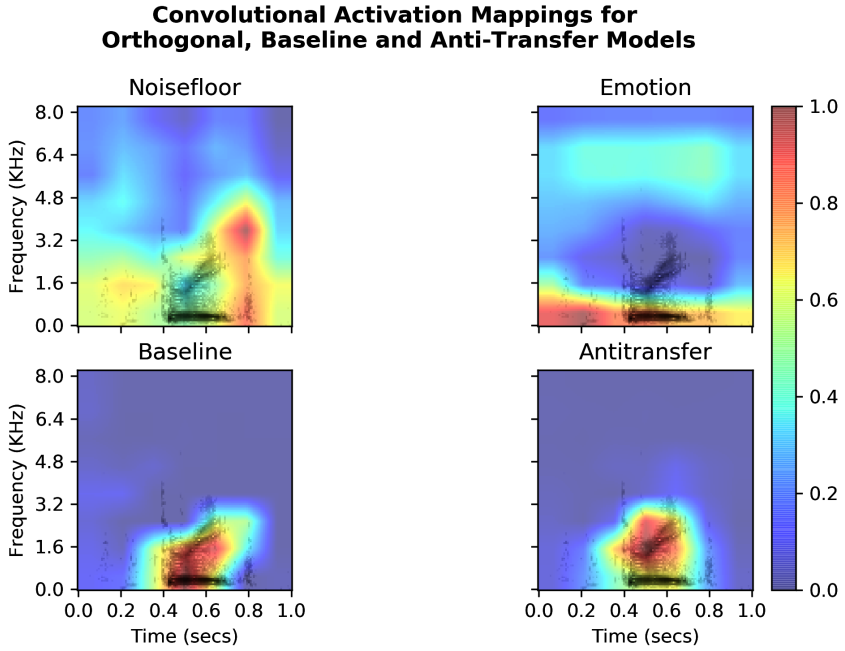


FIGURE 5.5: Grad-CAM convolutional **feature activations** of different models for the same input datapoint from the test set of the GSC with low noise added (word recognition target task). The activations have been computed in the last (13th) convolution layer, where we applied anti-transfer for this experiment. In all plots, the magnitude spectra are shown in black. The top row shows: the activation of the model trained for the first orthogonal task (noise type recognition, left), the activation of the model trained for the second orthogonal task (emotion recognition, right). The bottom row shows: the activation of the baseline model (no transfer or anti-transfer, left), the activation of the dual anti-transfer model (pretrained on noise type and emotion recognition, right)

lead to a decreased performance compared to regular AT and to the baseline (no transfer learning). These results support the motivating idea of anti-transfer learning: given a suitable choice of orthogonal tasks, avoiding similar representations can improve learning and generalization on the target task. Conversely, while transfer learning has proven efficient and effective in many settings, for orthogonal tasks like in our experiments it can actually be detrimental.

5.4.2 Convolutional Feature Activations

In order to support a visual interpretation of the deep representations generated with anti-transfer learning, we applied the Grad-CAM technique (Selvaraju et al., 2020) to our trained models.[6] In a CNN, Grad-CAM produces class-discriminative localization maps of a convolution layer using the gradient of the classification score with respect to the convolutional features present in that layer. This produces a heatmap of the same dimension as the input data, showing which parts of the input matrix are most important for classification. An in-depth description of this technique can be found in Selvaraju et al. (2020).

For this visualization we used the GSC dataset with low noise added, where we apply dual AT. We selected this specific case to better assess the effectiveness of our approach in moving away from unwanted features, showing the behavior of AT with 2 simultaneous orthogonal tasks. Figure 5.5 shows the Grad-CAM activations obtained for a datapoint of the test set, containing a male voice saying the world “eight" with added “office-like" background noises at low volume. The voice appears as a column in the center of the lower half of the spectrogram (approximately from 0.4 until 0.7 secs), while the background noise appears mainly as vertical spikes outside of the center (approximately at 0.12, 0.22, 0.38, 0.8 secs). The activations shown are obtained for the two models trained on the orthogonal tasks (background noise recognition and emotion recognition), the baseline model (no transfer learning) and the dual AT model with AT applied on the last convolution layer (pretrained first on background noise recognition and then on emotion recognition). As expected, the background noise type recognition model focuses mostly on pixels outside the center, in particular on the spike at 0.8 secs. The emotion recognition model focuses instead mostly on the lower

[6] We used a modified version of `https://github.com/jacobgil/pytorch-grad-cam` .

frequencies in the spectrum (approximately below 800 Hz), which is the normal range for the fundamental frequency of the human voice. The baseline model successfully focused on the speech signal in the center, although it slightly expands also towards the noise spike at 0.8 secs and it has a high activation in the low-frequency region where emotion information is more present (according both to our orthogonal model and our research experience). Similarly, the dual AT model is focused on the speech signal center, but it adjusted its attention towards the mid frequencies, where most format and consonant information is present, decreasing its activation on both the low-frequency area (emotion) and the spike at 0.8 secs (background noise). This example confirms that the dual AT model developed a certain degree of invariance to both orthogonal tasks (noise type and emotion recognition) when predicting the between the target task (word recognition), which underpins the observed effectiveness of anti-transfer learning in our experiments.

5.4.3 Layer Selection

We tested all layers in all task/orthogonal task combinations and Figure 5.6 shows the average per-layer improvement in both train and test accuracy that we obtained in the speech emotion recognition task. In this case, computing the anti-transfer loss with layer 5 provides the best performance, although layers 7 and 13 yield comparable results. Moreover, in both training and test, layer 9 yields the lowest performance and it is the only one that leads to a slight training accuracy decrease. However, most other layers also lead to improvements and the situation may vary when using different architectures or datasets. Also for word recognition layer 5 yields the best results, but for sound goodness estimation we obtained the best performance with layer 6.

In summary, there is no overall unequivocal best choice for the layer to use for the anti-transfer loss. Our intuitive expectation was the last layers

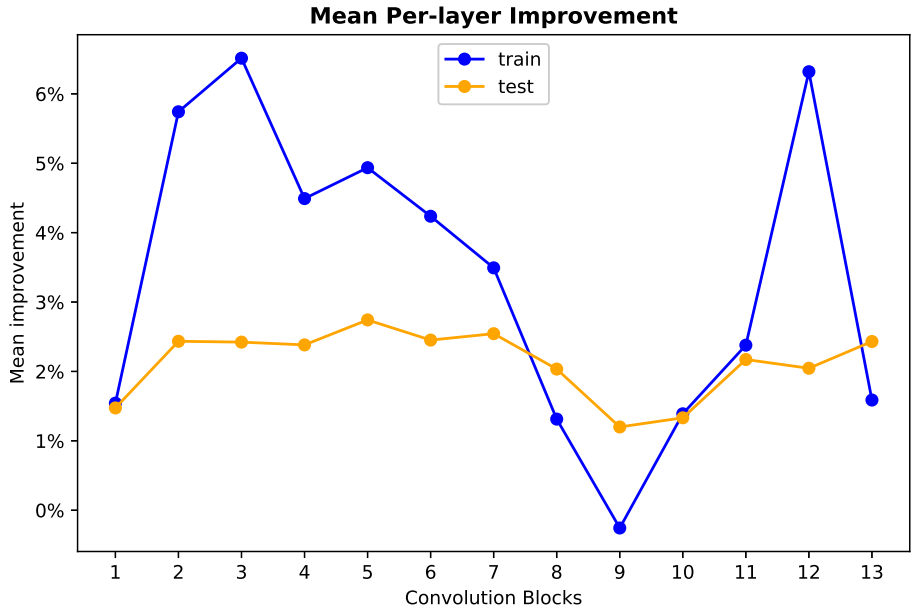


FIGURE 5.6: Mean **per-layer improvement** on speech emotion recognition (IEMOCAP random split) with pretraining on word recognition (Librispeech). The improvement refers to the baseline with no weight initialization.

would be most effective, as they should be most task-specific according to Yosinski et al. (2014). It is interesting to observe that these results are not reflected in our layer-wise evaluation, but we do not currently have an explanation for this.

Based on these results, we experimented with training using the anti-transfer loss on multiple layers at the same time. We tried to use three layers at once in three configurations: the first convolution layers, the last ones, the best ones according to the results above. These configurations yielded worse results than the baseline setting (non-transfer learning). However, this may be because we did not perform hyperparameter optimization on this approach, therefore further exploration could potentially lead to positive results.

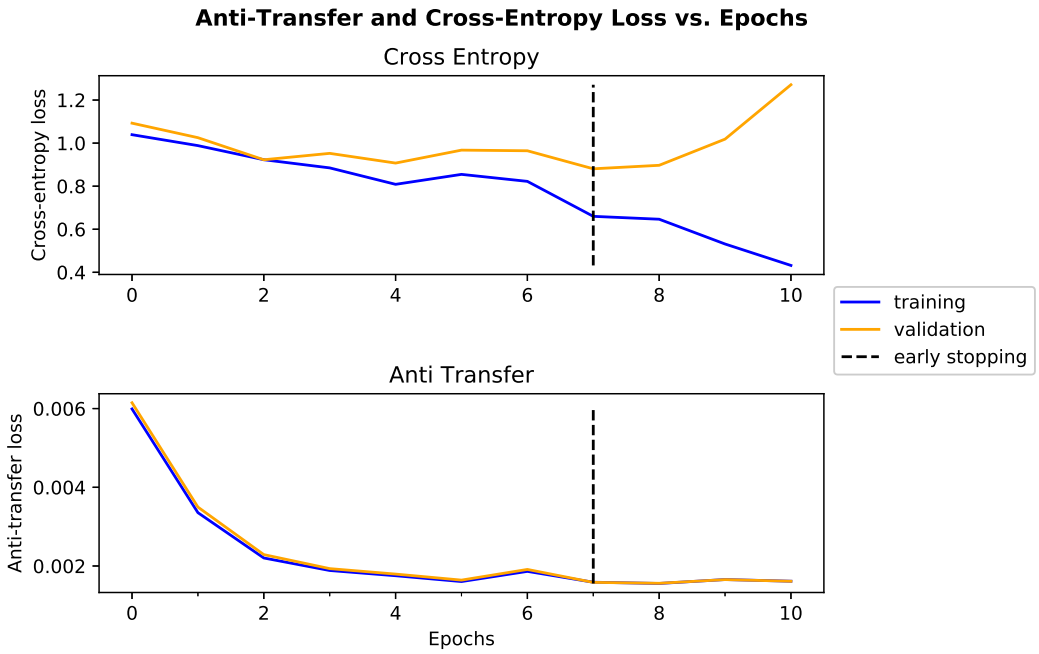


FIGURE 5.7: Evolution of the train and validation cross-entropy loss and train and validation anti-transfer **loss** during the training. This example refers to training on speech emotion recognition as target task (IEMOCAP random-split) and pretraining on word recognition (Librispeech) and anti-transfer applied to the 5th convolution layer.

5.4.4 Learning Dynamics

Figure 5.7 shows the development of the classification loss (cross-entropy) and the anti-transfer loss during training for speech emotion recognition (IEMOCAP random-split), with pretraining on word recognition (Librispeech) and anti-transfer applied to the 5th convolution layer. Here it is evident that the network is actually learning to differentiate its deep representations from the pretrained ones, as the anti-transfer loss is substantially reduced. Moreover, as we expected, the anti-transfer loss is already low from the first epoch because the randomly initialized feature maps start mostly uncorrelated to the ones of the pretrained network. The relatively low magnitude of the anti-transfer loss with respect to the cross-entropy loss indicates that anti-transfer plays a "preventive" role during training, keeping the deep representations from becoming correlated.

TABLE 5.6: Accuracy results for different **channel aggregation** methods and different **similarity functions**. All results are computed for speech emotion recognition as target task (IEMOCAP random-split) with pretraining on word recognition (Librispeech) and anti-transfer on the 5th convolution layer. The best training, validation and test accuracy results overall are highlighted in bold font.

	Sigmoid MSE Similarity			**Squared Cos Similarity**		
Aggregation	**Train**	**Val**	**Test**	**Train**	**Val**	**Test**
Mean	68.2	68.3	63.9	68.7	66.1	60.0
Sum	68.4	68.2	63.8	69.5	66.0	60.0
Comp Mul	71.0	67.5	63.9	70.4	67.0	63.1
Max	68.3	66.3	65.0	**76.7**	66.2	66.7
Gram Matrix	76.3	65.9	65.8	72.5	**68.7**	**66.9**

5.4.5 Aggregation and Distance Functions

Table 5.6 shows the results of experiments performed to select the best channel aggregation and similarity function to compute the anti-transfer loss. All aggregation types refer to a function applied pixel-by-pixel along the channel dimension. *Comp Mul* stands for compressed multiplication (feature activation values raised to the power of 0.001 and then multiplied along the channel dimension). The compression is necessary when multiplying pixel-by-pixel to avoid rounding to 0 during the multiplication of many small numbers.

As an alternative to Squared Cosine Similarity we used *Sigmoid MSE Similarity*, which we define as the negative standard Mean Squared Error with a sigmoid function applied to avoid excessive loss values. Without the sigmoid, the training led to very high absolute values in the feature maps, which minimizes the AT loss, but also drastically decreased the accuracy. We also tried several approaches to compute the similarity for all possible channel combinations without using any aggregation method, but all of them were too expensive in terms of computation or memory. We find that to aggregate the channel information using the Gram matrix and to compute the matrix similarity with squared cosine similarity gives the best results, which is why

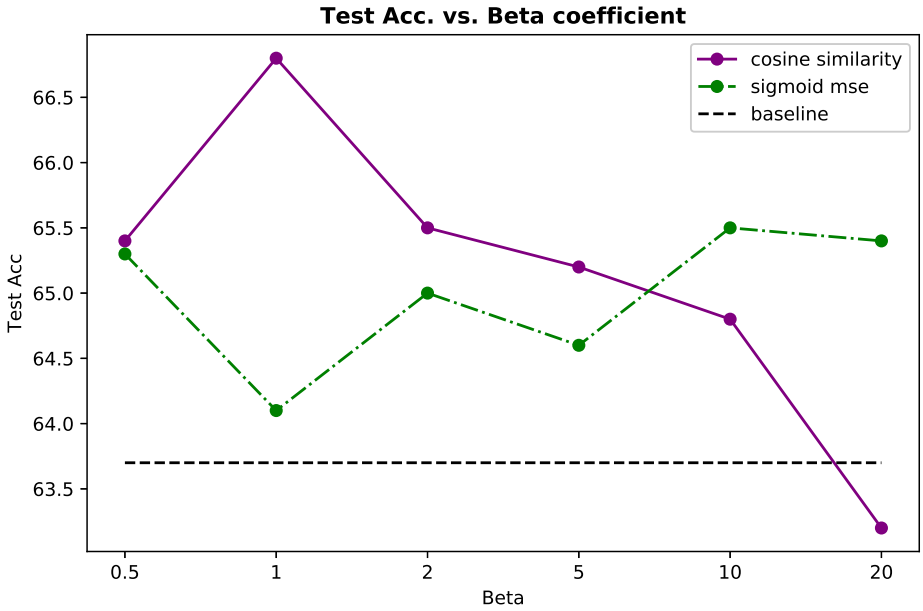


FIGURE 5.8: Variation of the test accuracy for **different β parameters** (weight of the AT loss) using Gram aggregation with the squared cosine similarity (solid line) and the sigmoid MSE (dash-dotted line). This example refers to training on speech emotion recognition as target task (IEMOCAP random-split), with pretraining on word recognition (Librispeech) and anti-transfer applied to the 5th convolution layer (the one yielding the best result).

we used this combination in the experiments in the previous Section 5.3.

5.4.6 AT Loss Weight

The β hyperparameter that determines the weight of the anti-transfer loss has a clear impact on the performance. As shown in Figure 5.8 for emotion recognition (IEMOCAP with random splitting) with pretraining on Librispeech and anti-transfer applied to the 5th convolution layer, we get the best result using the squared cosine similarity with a β value of 1, obtaining a performance gain of approximately 3 percentage points. The performance gain is smaller for all other values and there is no improvement with a very high (20) β. For sigmoid MSE the performance gain is smaller, but less dependent

on the β value. For practical purposes, $\beta = 1$ with cosine similarity seems to be a good default choice.

5.4.7 Computation and Memory Costs

The improved accuracy comes at a cost of increased computational and memory demands at training time. The following considerations refer to our specific implementation and different strategies may have different trade-offs. For instance, it is possible to precompute all needed Gram matrices in advance and avoid loading the pretrained feature extractor into the GPU RAM. This would lead to lower memory demand and computation time during training, but it would be incompatible with in-place data augmentation and other online approaches.

With reference to our implementation, regarding computational time, training a network with anti-transfer learning takes on average approximately 2.8 times longer compared to the same network with standard training. This refers only to the training with anti-transfer applied, not including any time needed to pre-train the feature extractor on the orthogonal task. Moreover, learning with AT loss requires more memory than standard training, since it requires fitting into memory the trained network, the pretrained feature extractor, the feature maps and the Gram matrices to be compared. The size of these depends on the chosen architecture, the input data dimension and the the number of channels of the convolution layer(s) used to compute the anti-transfer loss. The additional memory M_t required to compute the anti-transfer loss using a network pretrained on an orthogonal task t can be calculated as:

$$M_t = E_t + \sum_{l \in L} 2(\#G_l + \#F_l) \times bytes\_per\_number \tag{5.4}$$

where E_t is the size of convolutional part of the pretrained network, L is the set of layers used for anti-transfer, $\#G_l$ is the size of the Gram matrix

computed on the layer l , $\#F_l$ is the size of the feature map of layer l. The term inside the summation is multiplied by 2 because we compute the above-described matrices both for the currently-trained and the pretrained network. The term $\#G_l$ is determined by $batchSize \times numChannels^2$, while $\#F_l$ depends on all dimensions of the input data, on the network's architecture and on the layer parameters. The $bytes\_per\_number$ is 4 in our case. In our specific test case with the VGG16 network, the whole network occupies ~1620 MB, while the feature extractor E_t requires additional ~1150 MB, The dimension of one batch with one single GSC data point preprocessed as described above is $[1, 1, 126, 129]$. With this configuration the term $G_l + F_l$ is ~62 MB when anti-transfer is computed only on the first layer (shape [1,64,126,129]) and is ~714 MB when computed on the last layer (shape [1,512,7,8]).

5.4.8 Discussion

Results Anti-transfer leads to a robust improvement in test results in all our experiments. The learning dynamics, data splits and the visualization show that the similarity between the pretrained and the new network's representations is effectively reduced. It seems that avoiding features from orthogonal tasks is generally helpful. The improvement with anti-transfer is generally greater when the baseline accuracy is lower.

Training results are also improved in most cases. This is unexpected, as we assumed that learning from scratch would find a good fit for the training set and that anti-transfer would act as a regularizer that only benefits generalization. However, it seems that for suitably chosen orthogonal tasks avoiding the representation of the pretrained network not only avoids fitting to confounding aspects of the data, but even leads to a better fit to the target task during training. This contravenes the common assumption that end-to-end learning with deep learning leads to a near-optimal fit to the training

data. Instead it shows that the use of prior knowledge, here in the form of an orthogonal task, can help not just to improve generalization.

In some cases, the train/test split was separating classes that the network was aiming to recognise in the orthogonal pretraining task (speaker-wise split for emotion recognition vs speaker recognition, Table 5.4, and instrument-wise split for sound goodness estimation vs instrument recognition, Table 5.5). When the orthogonal task was speaker or instrument recognition, we observed a significantly improved generalization to unseen speakers in the speaker-wise split and to new musical instruments in the instrument-wise split, respectively. This indicates that the models are actually developing a degree of invariance to the orthogonal tasks, which is also illustrated in the visualization example (Figure 5.5).

The results show that pretraining on the same dataset provides higher improvement on average, compared to pretraining on a different dataset, even a much bigger one (Figure 5.3). It is surprising that the larger dataset does not have a more positive effect. We hypothesize that a more specific separation of representations can be developed for the specific dataset with orthogonal task labels on the same dataset by more directly modelling the interactions between different tasks. Thus, anti-transfer is a well-suited approach to exploit datasets provided with multiple labels, but the use of models pretrained on different data is still effective and both can also be combined.

Related Work As mentioned in Section 5.1, when pretraining with an orthogonal label of the same dataset, AT is similar to Domain Adversarial Training (DAT) (Ganin et al., 2016) if we consider an orthogonal task class as a domain. As mentioned, AT has the practical advantage of only needing a pretrained model rather than requiring labeled data from the source domain. This makes it possible to use models pretrained by third parties,

which can be beneficial in the case of models pretrained on very large or private datasets. Even though in our test cases AT with models pretrained on the same dataset provided the best improvement, models pretrained on bigger and different datasets (which is not possible with DAT) still provided a good improvement over the baseline.

As mentioned in Section 5.1, the idea of anti-transfer is related to *Speech-VGG* (Beckmann et al., 2019), which applies a deep feature loss to encourage similarity of deep representations, instead of dissimilarity as in anti-transfer. The experiments by Beckmann et al. (2019) are comparable with our inverse-Beta experiment in Section 5.4.1, where we show that encouraging similarity causes a drop in performance for orthogonal tasks. However, Beckmann et al. (2019) obtain a performance improvement with their approach applied to related target/pretrained task combinations: word recognition vs. speech inpainting, language identification and speech/music classification. This confirms that the selection of orthogonal tasks for anti-transfer is important. The cosine similarity of the labels for target and orthogonal task on the same dataset could be a test for the degree of task orthogonality. In our examples, the labels of the orthogonal task on the same dataset were completely orthogonal because of the nature of the dataset, but it may be wort testing this more systematically in future work.

As we introduce in Section 5.1, anti-transfer falls into the broad category of disentanglement. Our method does not directly enable pinpointing specific disentangled components in the data, e.g. as in source separation, but in effect it leads to separate deep representations for different tasks, as visualized in Figure 5.5. An advantage of anti-transfer is that it is a supervised training approach, which tends to be more efficient than adversarial or VAE methods.

Limitations Limitations of anti-transfer apply to: resources, orthogonal tasks, models and data availability, pretrained model accuracy.

Anti-transfer needs additional memory and computation resources at training time. Invariance to simple transformations can sometimes be achieved with simpler models, e.g. Guizzo, Weyde, and Leveson (2020), Wang et al. (2019b), and Marchand and Peeters (2016b), but complex tasks, like speaker recognition justifies in our view the increased resources used. Memory demands can have an impact in practice as GPU memory is often a bottleneck. Since earlier layers are similarly effective as later layers, but use less memory, using them can offer a better ratio of cost to performance gain. To make anti-transfer more practical on GPUs with limited memory, other ways of reducing memory demand can be investigated.

A practical limitation when using pretrained networks, is that the target task network needs to have the same structure (up to the AT layer) as the orthogonal task network. This can be a limitation if the network structure is not well suited for the target task.

Anti-transfer training is sensitive to the weight on the AT loss in our experiments, especially using the cosine similarity, although a value of $\beta = 1$ worked well in all our experiments. Still, some effort should be made to tune this hyperparameter when using anti-transfer learning.

A more conceptual limitation is the need for an orthogonal task. However, identifying the orthogonal task is often straightforward, as the elements that cause model performance to decrease are known, e.g. speaker identity, text, emotion, recording equipment, acoustical conditions. Finding or creating orthogonal task labels on the same or a similar dataset, or a model pretrained on an orthogonal task, can be a limitation, depending on the application.

In addition to this, benefits of AT can only be expected if the pretrained model is effective and even then there may be relevant representations that

the pretrained model has not learned. However, perfect avoidance of the representation learnt for the orthogonal task or perfect invariance to the orthogonal task is not required to improve performance and generalization, as our experiments have shown. The situation would be different for undesirable labels, where invariance to the orthogonal task in itself is an important target. Our measurement of this invariance has mainly been indirect through performance. Our visualization example was encouraging but to guarantee algorithmic fairness, more stringent measurements would be required.

Applications Anti-transfer is in principle applicable in all situations where suitable datasets are available, in particular when invariance to a specific task is desired. Even though we implemented AT for VGG16, it can be applied to other CNN designs. Also, it is directly applicable to feed-forward and to recurrent networks and it can be adapted to attention-based models. We have only tested classification tasks, but there is nothing in general to prevent the application of this method to regression, or more complex tasks (e.g. automatic speech recognition) or other domains (e.g. computer vision).

As mentioned, AT can have applications in areas such as algorithmic fairness, where model outputs should be independent of sensitive variables, e.g. financial decisions should not depend on gender or ethnicity. The variable is not necessarily explicit in the input data, e.g. the gender of a person could be not mentioned in their financial data, but models could estimate it and use that estimate as the basis for a decision. With more direct measurements of the degree of invariance to the sensitive variable, AT could be suitable to improve algorithmic fairness.

There are many pretrained models available for many tasks, particularly in computer vision and natural language processing. These models can be used for anti-transfer in many tasks, with the limitation that the orthogonal task must be known to be independent of the target and the the network

architecture must have sufficient overlap, i.e. the structure of the networks must be the same from the input up to the the layer(s) used in anti-transfer.

5.5 Conclusions and Remarks

In this study, we introduced anti-transfer learning for speech processing with neural networks, a novel method improving generalization by instilling invariance to an orthogonal task when training a network on a target task. When applying anti-transfer, we use a pretrained network with the same structure as the target network. In training the target network we apply a deep feature loss that discourages similarity between convolutional layers in the pretrained and target network to encourage the development of an internal representation that is independent of the orthogonal task. Our experiments with several classification tasks on speech and music audio in different configurations show improved results for all tasks. We observe a robust improvement over the learning form scratch and over transfer learning by weight initialization.

Our analysis provides evidence that anti-transfer achieves a degree of invariance to the orthogonal tasks, e.g. speaker identity, when the network is applied to the target task, e.g. speech emotion recognition. While there is a cost of pretraining and of the anti-transfer learning itself, the improved generalization may often be worth it. Readily available trained models remove the cost of pretraining and there may be further optimizations possible to address memory and computation costs.

With the increasing availability of public datasets and pretrained models chances grow that a suitable dataset or model can be found, but the selection of the orthogonal task needs careful consideration. Transfer learning is generally seen as a straightforward way to improve the performance of deep neural networks by using additional data. Our results show that taking into

account the nature of the pretraining tasks is important and that treating related and orthogonal tasks differently can boost generalization significantly.

Applications can benefit from improved generalization in many domains where there are natural changes to a signal that are independent of the target task, such as room acoustics, ambient noise, degradation through transmission, etc., as in the tasks we addressed in our experiments. A potential application of anti-transfer is to avoid the use of specific signal properties in areas such as algorithmic fairness, where being invariant to gender or ethnicity is a socially important goal. This will need further work on measuring and controlling the level of invariance as well as a discussion of the specific goals.

Despite the useful properties of anti-transfer, one of its major bottlenecks is the increase of computational resource demand that it introduces, as discussed in Chapter 5.4.7. The next Chapter of this dissertation presents a general solution to optimize the resource demand of SER models that is based on quaternion information processing. Subsequently, in Chapter 7, and as a final stage of this dissertation, we propose a solution to exploit the properties of anti-transfer while limiting the resource demand. Here we merge the two approaches presented in this Chapter and in Chapter 6 applying anti-transfer to quaternion-valued network. We also extend the evaluation of anti-transfer to new datasets and network architectures.

Chapter 6

Improving efficiency: Speech Emotion Representations in the Quaternion Domain

This Chapter includes an adapted version of Journal Paper 1 (see Section 1.5). This research was in part developed when I was Research Fellow at La Sapienza University, during my PhD suspension in 2021. That project was refined, improved and submitted to "IEEE Transactions on Speech Audio and Language Processing" after my resumption to studies at City in 2022.

In this study we propose a solution for a common issue in SER: high resource-demanding models. Broadly speaking, we propose to map speech signals into a compact multi-channel latent representation that permits having different "emotional viewpoints" of the signal, which are signal representations individually related to different components of human emotion, namely: valence arousal and dominance. This enables the use of quaternion-valued neural networks with real-valued signals as input. We show that our approach improves the performance of SER models, while drastically reducing their resources demand. Moreover, the improvement is consistent also in cases where data is very scarce, which is a common scenario in the SER field.

This Chapter addresses the final research question of this work.

6.1 Introduction

Our approach is based on quaternion information processing, which is a well-established strategy to minimize models' resource demand without reducing their performance, as we discuss in detail in Section 2.4.3 and Appendix A.

The proposed model, named Real to Emotional H-Space (RH-emo), is a hybrid real-quaternion autoencoder-classifier architecture that is trained in a semi-supervised fashion in order to optimize each axis of the embedding dimension to different emotional characteristics: the first channel is optimized for discrete emotion recognition and the 3 other channels are individually optimized for the classification of the valence, arousal and dominance (as shown in Figure 6.1). As will be further explored from Section 6.3 onward, when used as a feature extractor that feeds into quaternion neural networks (QNNs), RH-emo improves the performance in SER tasks while consideably reducing the number of trainable parameters and computing resources, compared to equivalent real-valued models processing plain spectrograms.

This approach has two advantages: it improves the performance of SER models even in situations where data is scarce and it drastically reduces the number of network parameters, consequently reducing the resource demand. We extend the approach of the quaternion autoencoder in Parcollet et al. (2019) by specializing the learned quaternion representation for our specific task (SER), where the different axes are optimized for the detection of different emotional characteristics that are coherent with the most used criteria of emotion classification. Moreover, we implement it with a more complex architecture (deep convolutional autoencoder) and we apply it to a different domain: emotion recognition from speech audio.

6.2 The Proposed RH-emo Model

6.2.1 Approach

The main aim of RH-emo is to map real-valued spectrograms to the quaternion domain, building compact emotion-related quaternion embeddings where each axis is optimized for a different emotional characteristic. In the embedded dimension, the real axis of the quaternion is optimized for the discrete classification of 4 emotions: *neutrality, anger, happiness, sadness* and the 3 complex axes are optimized for the prediction of emotion in a *valence, arousal* and *dominance* 3D space. This representation exploits the natural predisposition of quaternion algebra to process data where a 4 or 3-channels representation is meaningful. Nevertheless, in most machine learning applications of quaternion algebra, the input data is naturally organized with a meaningful shape, as happens for instance with RGB/RGBA images (where the color/alpha channels are treated as different quaternion axes) and first-order Ambisonics audio signals (where the 4 spatial channels are considered as the quaternion axes). In our case, instead, such quaternion representation is created through a semi-supervised learning procedure, where the different axes are forced to contain information related to different complementary emotion characteristics. Therefore, in a certain sense, the axes of this embedded dimension can be thought of as different “emotional points of view" of an audio signal.

RH-emo is intended to be used as a pretrained feature extractor to enable the use of quaternion-valued neural networks for SER tasks applied to monoaural audio signals. On the one hand, the emotion-related disentanglement among channels helps to enhance the performance of SER models. Whereas, on the other hand, the reduced dimensionality together with the enabled possibility to classify the data with quaternion-valued networks permits to spare a large number of network parameters, consequently lowering

the resource demand and speeding up the training.

6.2.2 RH-emo Architecture

RH-emo is a hybrid real/quaternion autoencoder network. Its structure is similar to R2Hae Parcollet et al. (2019), nevertheless, RH-emo is based on a convolutional design and it embraces multiple classification branches, as opposed to R2Hae. We used a public PyTorch implementation of quaternion convolution layers and operators[1]. As Figure 6.1 shows, our RH-emo is composed of three components: an encoder $E(X)$ acting on the (real-valued) input spectrogram, producing an embedded vector. The output of the encoder is then fed separately to a (quaternion-valued) decoder $D(Z)$ to reconstruct the original spectrogram, and to a classification head $C(Z)$ for performing emotion recognition. The classifier outputs four separate predictions y_D, y_v, y_a and y_d which are, respectively, a discrete and a continuous (in the valence, arousal, dominance space) categorization of the emotional content of the spectrogram. The specific architecture for each of these blocks, as well as the loss function we optimize and the training strategy we adopt is described more in detail in the following paragraphs.

Encoder

The input data, a magnitudes-only real-valued spectrogram in our case, is forward propagated through a real-valued autoencoder made up of 3 convolution blocks. Each block contains a 2D convolution layer (ReLU activations, 3x3 kernels, single-pixel stride, increasing channels number: 1, 2, 4), followed by max-pooling layers of dimension [2x2], [2x1], [2x1]. Moreover, only between the first and the second block, a batch normalization layer is

[1]https://github.com/Orkis-Research/Pytorch-Quaternion-Neural-Networks

present. The encoder produces an embedded vector that presents a dimensionality reduced by a factor of 0.25 compared to the input. In our experiments, we use input spectrograms with a shape of 1x512x128 (channels, time-steps, frequency-bins) and the embedded dimension created by the encoder has a shape of 4x64x64. The embedded vector is then forward propagated in parallel into four distinct real-valued classifiers and also into a quaternion-valued decoder. It is therefore important that the embedded vector contains a number of elements that is multiple of four, in order to be properly treated as a quaternion by the decoder section of the network.

Classifiers

Each classifier consists of a sequence of 3 real-valued fully connected layers, where the first 2 contain 4096 neurons and are followed by a dropout layer. In the first classifier, the output layer contains 4 output neurons (the number of emotional classes to be classified) and softmax activation. Instead, the other 3 classifiers are identical and have one single output neuron with sigmoid activation, as they are individually aimed at a binary classification task: the prediction of "high" or "low" valence, arousal, and dominance, respectively.

Decoder

The decoder mirrors the encoder's structure, but uses quaternion-valued 2D transposed convolutions with a stride that mirrors the pooling dimensions of the encoder, instead of the sequence of 2D real-valued convolutions and 2x2 max-pooling and a quaternion-valued batch normalization layer instead of its real-valued counterpart. The output of the decoder is therefore a matrix with the same dimensions as the input, but with 4 channels instead of a single one.

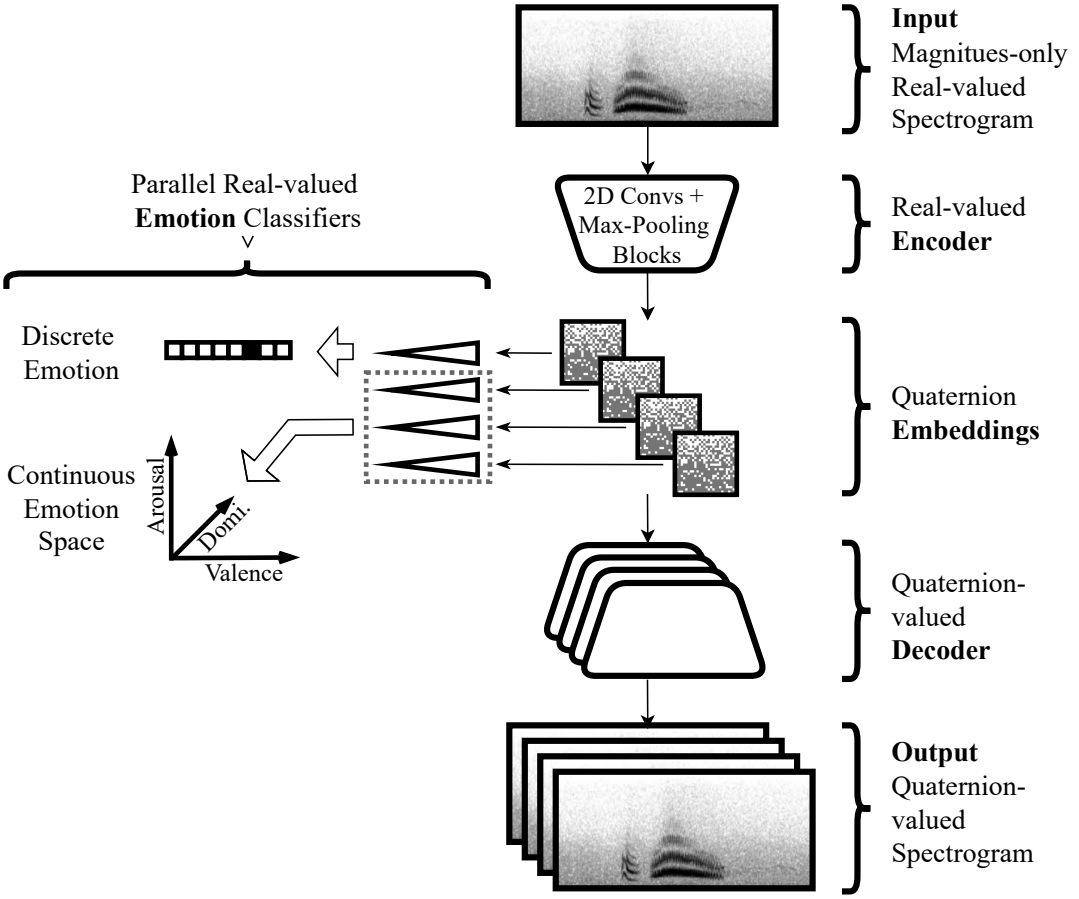


FIGURE 6.1: RH-emo Block Diagram. An input magnitudes-only spectrogram is first propagated into a real-valued convolutional encoder that generates embeddings with a [4x64x64] shape. The network is then split into two branches: a completely unsupervised quaternion-valued decoder that reconstructs the input spectrogram projecting it in a four-channel quaternion space and a set of 4 parallel real-valued supervised classifiers, each connected to one of the four channels of the embeddings and separately classifying different emotion characteristics: discrete emotion, valence, arousal, and dominance.

6.2.3 Loss Function

The loss function we minimize during the training of RH-emo is a weighted sum of the binary crossentropy reconstruction loss between the input spectrogram and the decoder's output, the categorical crossentropy classification loss of the emotion labels predicted by the supervised classifier in the middle of the network (discrete, valence and arousal).

The objective function we minimize is therefore:

$$\begin{aligned}\mathcal{L} = \mathrm{BCE}_(X, Y_r) + \beta \cdot \{\mathrm{CE}_(p, t) \\ + \alpha \cdot [\mathrm{BCE}_(v_p, v_t) + \mathrm{BCE}_(a_p, a_t) + \mathrm{BCE}_(d_p, d_t)]\}\end{aligned} \tag{6.1}$$

where BCE is the binary crossentropy loss, CE is the categorical crossentropy loss, β and α are scalar weight factors, X is the input spectrogram, Y_r is the decoder's output re-mapped to the real domain through the split activation function (as discussed below), p and t are respectively the discrete emotion prediction and truth label, v_p/v_t, a_p/a_t and d_p/d_t are respectively the valence, arousal and dominance prediction and truth labels.

For the reconstruction loss computation it is necessary to map the quaternion-valued decoder output back to the real domain, in order to have the same shape as the input vector. For this purpose we use a stratagem similar to the "split activation" described in Parcollet et al. (2019) and Ujang, Took, and Mandic (2011): we perform an element-wise mean across the channel dimension of the quaternion output, bringing back the 4-channels vector to a single-channel shape. During the training, this forces the model to not weigh the intra-channel correlations among the quaternion axes in the reconstruction term of the loss (the leftmost term of eq. (6.1)). Our expectation is that this leaves room for the emotion recognition term of the loss (the rightmost term of eq. (6.1)) for tuning these correlations, making them related to the emotional information.

6.2.4 Training Strategy

For the RH-emo training, we use the Interactive Emotional Dyadic Motion Capture Database (IEMOCAP) dataset (Busso et al., 2008), which includes: 5 speakers, 7529 utterances, 9:32 hours of audio, 10 emotion labels and it is in

the English language. We selected this specific dataset for the following reasons: it is one of the most popular SER datasets, it contains a large number of datapoints, it is not limited to a restricted set of sentences, emotions are expressed by actors with a natural feeling rather than being over-emphasized (Busso et al., 2008) and it is labelled both in the discrete and continuous (valence, arousal, dominance) emotional domains.

We apply 4 preprocessing stages to the raw data: we first extract 4-seconds non-overlapped fragments (or zero-pad if a datapoint is shorter that this duration). Then, we compute the short-time Fourier transform (STFT) using 16 ms sliding windows with 50% overlap, applying a Hamming window and discarding the phase information. After this point, we normalize the whole dataset between 0 and 1 and, in the end, we zero-pad the spectrograms to match a shape of 512 (time-steps) x 128 (frequency-bins).

To permit proper convergence, we perform the training in 2 consecutive stages: we first train the network until convergence with the β weight set to 0. This removes the rightmost term from eq. (6.1), consequently eliminating the emotion classification part of the loss. Doing so, we train the network in a completely unsupervised way only to perform a quaternion projection of the real input spectrogram, without taking into account any emotion-related information. After this stage, we re-train the network adding also the classification term in the loss in order to specialize the learnt representations to the emotion recognition task, but also maintaining the embedded vector in a quaternion-compatible shape that is meaningful for the decoder part of the network. For this stage we performed a grid search to find the best combination of the emotion classification weights β and α and we ended up using $\beta = 0.01$ and $\alpha = 100$. This means that overall we weight more the reconstruction error in the loss function (thanks to the low β), and we weight more the dimensional emotion classification compared to the discrete classification (thanks to the high α).

TABLE 6.1: CNNs pretraining results for IEMOCAP

Arch.	Method	Params	Train acc.	Test acc.
RH-emo	/	1.3×10^8	80.34	60.7
VGG16	Real	1.6×10^8	74.88	62.87
	RH-emo+Quat	1×10^7	72.25	71.10
AlexNet	Real	5.7×10^7	71.02	63.33
	RH-emo+Quat	1×10^7	71.81	70.31
ResNet	Real	2.3×10^7	61.05	57.20
	RH-emo+Quat	4.9×10^6	73.03	**71.20**

While for the first, completely unsupervised, training stage we use all data available with IEMOCAP, in the second supervised stage we use only a subset of the dataset, including only the datapoints related to 4 emotions (*angry, happy, neutral, sad)* and we merge the classes *happy* and *excited* as one single emotion class *happy*. This is a standard procedure with IEMOCAP, as the other labels contained in the dataset are highly imbalanced. For both training stages, we use subsets of approximately 70% of the data for training, 20% for validation, and 10% for the test set. We use a learning rate of 0.001 in the first stage and of 0.000001 in the second one, a batch size of 20 and the Adam optimizer (Kingma and Ba, 2014). We use dropout at 50% in the classification branches for the second training stage. We apply early stopping by testing at the validation loss improvement with a patience of 100 epochs in the first stage and of 30 epochs for the second one.

After these 2 training stages, we obtain a test reconstruction loss (the isolated leftmost term of eq. (6.1)) of 0.00413 and competitive test classification accuracy: 60.7% for the discrete classification and respectively 65.4%, 75.3% and 70.2% for the valence, arousal, and dominance dimensions.

6.3 Evaluation

In order to test the capabilities and properties of RH-emo, we compare the classification accuracy for SER tasks obtained with real-valued CNN networks and equivalent quaternion-valued versions of them (QCNNs). For the quaternion versions we keep the same architecture of the real CNNs, but we use quaternion-valued convolution and quaternion-valued fully connected layers instead of the canonical real-valued ones, with the exception of the final layer of the networks, which are real-valued also in the QCNNs. For the real networks, we use the magnitudes-only spectra as input, while for the quaternion networks we use the embeddings generated with RH-emo pretrained on IEMOCAP. Moreover, we compare and combine our approach with a standard transfer learning method performed on the same dataset (IEMOCAP): pretraining with weight initialization. Therefore we have two distinct types of pretraining: the pretraining of the RH-emo network, which we use to compute the emotional embeddings, and the pretraining of the CNNs that we use to perform the actual SER task. Both pretrainings are performed on the IEMOCAP dataset. To avoid confusion, from here on we will refer to the first as RH-emo pretraining and to the latter as CNN's pretraining.

Figure 6.2 depicts all cases we include in our experimental setup. The color coding of Figure 6.2 shows the 3 consecutive stages of our experiments: first we pretrain RH-emo (yellow), then we pretrain the CNNs (orange) on IEMOCAP and finally we train or retrain the CNNs on other datasets. We have two types of baseline: the first one, shown in the upper row of Figure 6.2, is a standard real-valued CNN with randomly-initialized weights. As a further baseline, as depicted in the second row of Figure 6.2, we test a standard transfer learning approach applied to the real-valued CNNs: we pretrain on IEMOCAP (the same dataset used to train RH-emo) and we then

initialize all weights of the SER CNNs but the ones of the final classification layer. The last two rows of Figure 6.2, instead, show our approach, where we use RH-emo as a feature extractor to feed quaternion-valued CNNs. In the third row, only RH-emo pretraining happens, while in the last row both RH-emo and CNNs pretraining are performed. In the latter case, we first pretrain RH-emo, then we pretrain the CNN on IEMOCAP, and finally, we re-train the same CNN on different datasets.

6.3.1 Experimental Setup

We evaluate RH-emo with 3 benchmark SER datasets:

1. RAVDESS, the Ryerson Audio Visual Database of Emotional Speech and Song (Livingstone and Russo, 2018a). 24 speakers, English language, 2542 utterances, 2:47 hours of audio, 8 emotion labels.

2. EmoDb, a Database of German Emotional Speech (Burkhardt et al., 2005). 10 speakers, German language, 535 utterances, 25 min of audio, 7 emotion labels.

3. TESS, the Toronto Emotional Speech Set (Dupuis and Pichora-Fuller, 2011). 2 speakers, English language, 2800 utterances, 1:36 hours of audio, 7 emotion labels.

The preprocessing pipeline for these datasets is identical to the one we applied to IEMOCAP, as described in Section 6.2, except for the final normalization step. For the quaternion-valued networks we normalize data between 0 and 1 (as required by RH-emo) and for the real-valued network we normalize to 0 mean and unity standard deviation to permit proper convergence.

We apply this approach to 3 popular CNN architectures with increasing capacity: VGG16 (Simonyan and Zisserman, 2015), AlexNet (Krizhevsky, Sutskever, and Hinton, 2012) and ResNet-50 (He et al., 2016), based on the

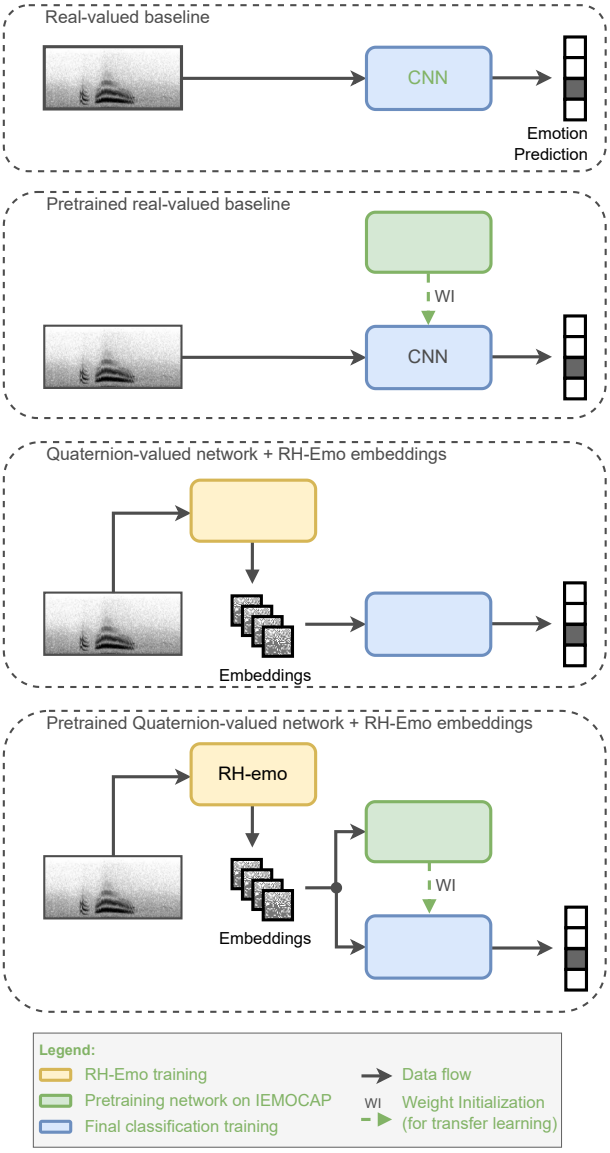


FIGURE 6.2: Block diagram of our experimental setup. The yellow-to-blue color coding reflects 3 consecutive training stages. There are 2 separate pretraining stages: RH-emo pretraining (yellow) and CNNs pretraining (green). The straight arrows indicate the data flow, while the dotted arrows, accompanied by the word WI, show where the weights of a pretrained network are used to initialize the initial weights of an identical network (transfer learning). The real-valued baseline is a regular CNN with random weight initialization, upper row. The pretrained real-valued baseline is the same network, but its weights are initialized with the ones of an identical network pretrained on IEMOCAP (the same dataset used to train RH-emo), second row. The quaternion-valued network is a quaternion-valued version of the real-valued baselines, in which (4 channel) input is generated by forward propagating the input spectrogram in RH-emo's encoder, third row. The pretrained quaternion-valued network is identical to the latter, but the weights of the CNN are initialized with the ones of an identical network pretrained on IEMOCAP, last row.

TABLE 6.2: Results for RAVDESS

Arch.	Method	Params	Train acc.	Test acc.
VGG16	Real	1.6×10^8	47.10	41.06
	RH-emo+Quat	1×10^7	55.50	49.85
	Real-Pre	1.6×10^8	67.86	45.30
	RH-emo+Quat-Pre	1×10^7	67.08	53.79
AlexNet	Real	5.7×10^7	54.55	46.36
	RH-emo+Quat	1×10^7	50.62	43.94
	Real-Pre	5.7×10^7	83.54	51.06
	RH-emo+Quat-Pre	1.4×10^7	63.16	47.58
ResNet	Real	2.3×10^7	72.84	43.48
	RH-emo+Quat	4.9×10^6	91.29	**55.15**
	Real-Pre	2.3×10^7	22.16	18.79
	RH-emo+Quat-Pre	4.9×10^6	89.54	52.42

TABLE 6.3: Results for EmoDb

Arch.	Method	Params	Train acc.	Test acc.
VGG16	Real	1.6×10^8	72.74	70.00
	RH-emo+Quat	1×10^7	79.54	50.00
	Real-Pre	1.6×10^8	78.16	52.00
	RH-emo+Quat-Pre	1×10^7	75.00	47.00
AlexNet	Real	5.7×10^7	63.1	47.00
	RH-emo+Quat	1×10^7	82.3	49.00
	Real-Pre	5.7×10^7	71.45	67.00
	RH-emo+Quat-Pre	1.4×10^7	77.63	71.00
ResNet	Real	2.3×10^7	99.47	48.00
	RH-emo+Quat	4.9×10^6	99.73	**73.00**
	Real-Pre	2.3×10^7	100.00	72.00
	RH-emo+Quat-Pre	4.9×10^6	99.73	46.00

Torchvision implementations[2]. These implementations present an adaptive average pooling layer between the convolution-based feature extractor and the fully-connected classifier. This permits to obtain an identical output shape from the feature extractor for any input dimension. We removed this layer from the only VGG16, in order to test the behavior of our approach also in this situation. Doing this, in fact, the feature extractor presents a reduced output dimensionality when the networks are fed with the quaternion embeddings (75% smaller than using the real spectrograms), enabling to spare a

[2] `https://pytorch.org/vision/stable/_modules/torchvision.html`

TABLE 6.4: Results for TESS

Arch.	Method	Params	Train acc.	Test acc.
VGG16	Real	1.6×10^8	99.54	97.62
	RH-emo+Quat	1×10^7	98.87	97.62
	Real-Pre	1.6×10^8	99.95	99.52
	RH-emo+Quat-Pre	1×10^7	98.72	97.85
AlexNet	Real	5.7×10^7	99.18	98.01
	RH-emo+Quat	1×10^7	99.54	98.56
	Real-Pre	5.7×10^7	100.00	98.01
	RH-emo+Quat-Pre	1.4×10^7	99.75	98.81
ResNet	Real	2.3×10^7	100.00	97.38
	RH-emo+Quat	4.9×10^6	100.00	**99.76**
	Real-Pre	2.3×10^7	59.88	57.53
	RH-emo+Quat-Pre	4.9×10^6	100.00	99.28

TABLE 6.5: Test Accuracy Results

Dataset	Average improvement			Best improvement
	No pret.	Pret.	Overall	
IEMOCAP	9.74	/	/	7.87
RAVDESS	6.01	12.88	9.45	4.09
EmoDb	2.34	-9.00	-3.34	1.00
TESS	0.97	13.63	7.30	0.24

major number of network parameters.

For all experiments we used a learning rate of 0.00001, ADAM optimizer, and a batch size of 20 samples, we apply early stopping with the patience of 20 epochs on the validation loss and we split the training, validation, and test sub-sets with approximately 70%, 20% and 10% of the data, respectively.

The main aim of this research is to provide a valid comparison between the proposed approach (quaternion-valued CNNs fed with RH-emo embeddings) and standard equivalent real-valued architectures, isolating as much as possible the pure difference between them. We configured our experimental setup in order to show the performance difference between real and corresponding quaternion CNNs fed with the emotional quaternion embeddings.

Therefore, we paid attention to performing each experiment in as-close-as-possible conditions, rather than optimizing each architecture for each different dataset, in order to highlight the properties of our approach. State-of-the-art results for SER tasks usually involve more complex solutions, as explained in Section 2. Moreover, the state-of-the-art approach can be radically different for each dataset, and therefore using the best method for each dataset would not permit having the same configuration for all possible aspects in both RH-Emo experiments and the baselines. This would add much more complexity to the setup, consequently making it less straightforward to isolate and understand the properties of our approach. Because of these reasons and the fact that many existing studies are based on different methods to compute the scores, different data splits and may use multiple data domains, our results can not be directly compared to the current state-of-the-art accuracy for these datasets, which, to the best of our knowledge are 75.60% for IEMOCAP (Kakouros et al., 2022), 87.5% for RAVDESS (Bouali, Ahmed, and Mazouzi, 2022), 88.47% for EmoDb (Pham, Dang, and Nguyen, 2021) and 99.6% for TESS (Jothimani and Premalatha, 2022).

6.3.2 Experimental Results

Table 6.1 shows the CNNs pretraining results we obtained on IEMOCAP, while Table 6.3, 6.2, and 6.4 expose the results on RAVDESS, EmoDb and TESS, respectively. Table 6.5, shows the average and best test accuracy improvement provided by our approach, among all CNN architectures for each dataset. Here, average improvement refers to the difference between the average test accuracy among all real-valued and the equivalent quaternion-valued outcomes, whereas the best improvement is the difference between the best real-valued and the best quaternion-valued accuracy we obtained.

The results clearly show that our approach enhances or maintains the

model's performance while improving its efficiency. For all datasets, the quaternion CNNs fed with RH-emo embeddings provide the best test accuracy overall, with an average accuracy improvement of 6.01 percentage points (pp) for RAVDESS, 2.34 pp for EmoDb, and 0.97 for TESS in the case we do not apply CNNs pretraining. The only case where our approach does not improve the test accuracy is with the EmoDb dataset, applying CNNs pretraining, where we have a performance drop of 9 pp. In the other cases where we applied CNNs pretraining, our approach provides a strong average improvement of 12.88 and 13.63 pp, respectively for RAVDESS and TESS. The results computed on IEMOCAP (Table 6.1 and first row of Table 6.5) depict a limit case, where knowledge is not transferred to different data because the same dataset is used for the RH-emo pretraining and for SER. Therefore here we did not apply any CNNs pretraining. Also in this special case is evident that models benefit from the use of quaternion-valued SER CNNs fed with emotional embeddings, with an average improvement of 9.74 pp among all CNN designs we tested.

We performed a two-sided Wilcoxon signed-rank test comparing the best real-valued results and the best quaternion-valued results for all dataset and networks, which results in a p value of 0.15. Although the p value is higher than the typical significance level of 0.05, it is worth noting that the sample size is very limited in this setup and there is still some indication of a potential difference between the groups. Moreover, it is essential to consider that, despite the significance results, the use of RH-emo plus quaternion-valued networks substantially reduces the network parameters, thus saving a considerable amount of resources.

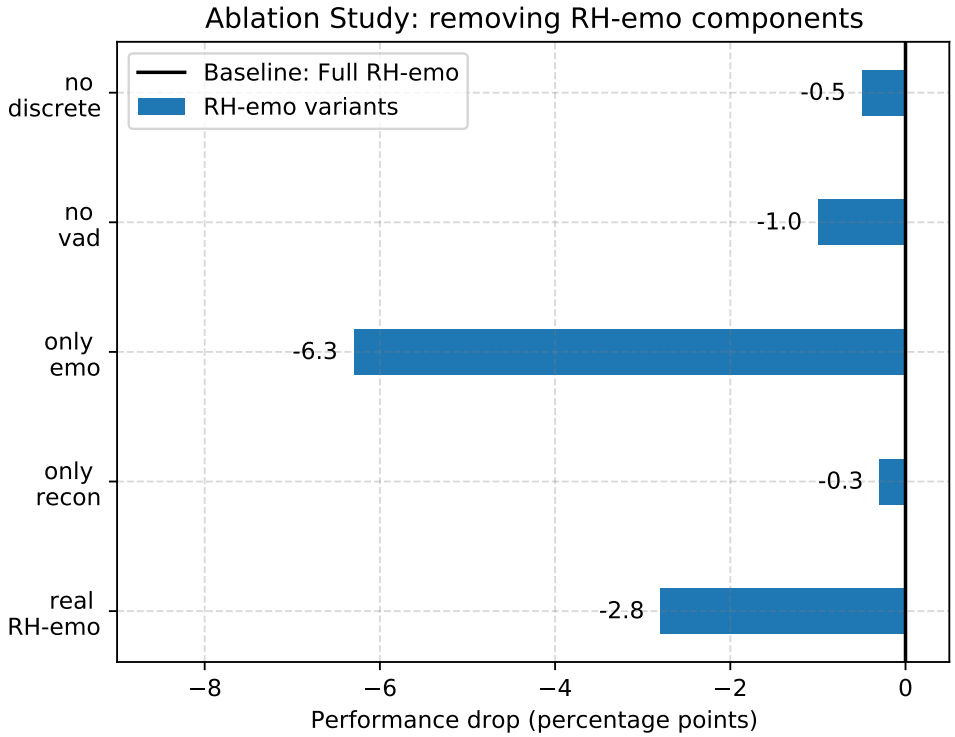


FIGURE 6.3: Ablation study results. The x axis shows the average drop in test accuracy (among the quaternion-valued VGG16, AlexNet and ResNet-50 for all corpora) obtained with different variants of RH-emo. Each row refers to a variant of RH-emo where we removed a specific component, namely: a completely real-valued network, only reconstruction, only emotion recognition, no valence-arousal-dominance (vad) estimation, and no discrete emotion classification.

6.4 Ablation Studies

In order to further explore the properties of our approach and to support its foundations, we performed additional experiments and ablation studies. For these studies we applied the same experimental setup presented in Section 6.3, altering only specific details, as described below.

6.4.1 Removing RH-emo Components

In this study, we alter the RH-emo structure and test the emotion recognition accuracy using the embeddings generated from the modified RH-emo

networks. We compared the full RH-emo, as described in Section 6.2, to the following altered versions:

- Real: identical to the regular network, but the decoder part is real-valued and no split activation is applied to the reconstructed output in the loss function.
- Reconstruction only: we removed the supervised classification branch, resulting in a completely unsupervised real-quaternion hybrid autoencoder.
- Emotion only: we removed the unsupervised reconstruction branch from the network, obtaining a completely supervised and real-valued emotion classification CNN. In this configuration, there are still 4 target outputs, each with a dedicated classifier (discrete emotion, valence, arousal, dominance).
- Discrete emotion only: we removed the valence, arousal and dominance classifiers, keeping only the discrete emotion classification branch. The rest of the network is unaltered.
- Valence-arousal-dominance only: we removed the discrete emotion recognition branch, keeping only the branches for valence, arousal and dominance. The rest of the network is unaltered.

Figure 6.3 exposes the results of this ablation study. In the figure, we show the mean test accuracy improvement obtained for all corpora with the quaternion-valued VGG16, AlexNet, and ResNet-50 over the real-valued baselines. Each row shows the results obtained feeding the quaternion-valued networks with the embeddings created with the above-described variants of RH-emo. These results consistently confirm the foundation of our approach. The performance of all variants is inferior to the full RH-emo. The results point out that the unsupervised branch of RH-emo is fundamental

to obtain useful embeddings, in fact, the emotion-only version, where the decoder part of RH-emo is removed, provides the most severe drop in performance compared to all variants and also the baseline. As we expected, the quaternion-valued decoder of the actual RH-emo outperforms the completely real-valued version (by 2.8pp). This supports our hypothesis that a quaternion-value decoder is able to create embeddings that present more suitable intra-channel correlations for the quaternion-valued CNNs. Moreover, also here, the quaternion approach leads to faster (pre)training and less memory demand due to the lower amount of parameters. The completely unsupervised variant (recognition-only) is conceptually similar to R2Hae (Parcollet et al., 2019), but it relies on a convolutional design and it is applied to a different domain. This ablation study shows that the addition of a classification branch to R2Hae provides an improvement in performance (by 0.3 pp in our case) and therefore the semi-supervision can be considered a valuable extension to R2Hae. This ablation study also shows that the classification of emotion in the valence-arousal-dominance space is more influential in the creation of stronger embeddings. In fact, the RH-emo variant without discrete classification provides superior accuracy compared to the discrete-only version (by 0.5 pp) This is further supported by the fact that, as a result of an extensive grid search, we apply a stronger weight to the valence-arousal-dominance term of the loss function (the α term in eq. (6.1)).

6.4.2 Removing RH-emo Pretraining and Backpropagation

We performed an additional ablation study where we alter how the RH-emo weights are initialized and backpropagated during the SER training. Figure 6.4 depicts the results of this study, showing the average difference in test accuracy per-dataset among all CNN designs. On the one hand, we initialized

the weights of RH-emo with random values while we regularly backpropagate the gradients of the RH-emo's encoder layers (blue rows). By doing this, we completely ignore the RH-emo pretraining and we force the QCNN network to perform an end-to-end training, directly learning how to map the real-valued input spectrograms into quaternion-compatible representations to feed the QCNNs. This approach is conceptually similar to (R2He) (Parcollet et al., 2019). On the other hand, we regularly initialize the weights of RH-emo with the pretrained RH-emo network, but we don't backpropagate the RH-emo layers (orange rows). The results of this experiment strongly support the foundation of our approach. The removal of RH-emo pretraining causes a consistent and substantial decrease in the QCNNs test performance, of 29.4, 3.25, and 6.97 pp for RAVDESS, EmoDb, and TESS, respectively. This confirms the importance of the prior training of the RH-emo encoder, as presented in Section 6.2, for the development of adequate quaternion emotional embeddings. On the contrary, the lack of backpropagation of the RH-emo layers does not provide a consistent performance drop. While the performance decreases for EmoDb (25 pp) and for TESS (0.22 pp), a narrow accuracy boost is evident for RAVDESS (0.91 pp). Moreover, the performance difference is averagely inferior compared to the no-pretraining case.

6.4.3 Reducing Training Data

As a further study, we re-trained all CNNs and QCNNs, progressively decreasing the amount of training and validation data. The size of the test set, instead, is kept unaltered, in order to have a consistent performance measure that can be compared with the other results presented in this paper. Figure 6.5 shows the outcomes of this experiment. Each line shows the trend of the average test accuracy among all CNN architectures, at different reduction rates of the data. Specifically, we trained on 100%, 75%, 50%, 25%, 10%,

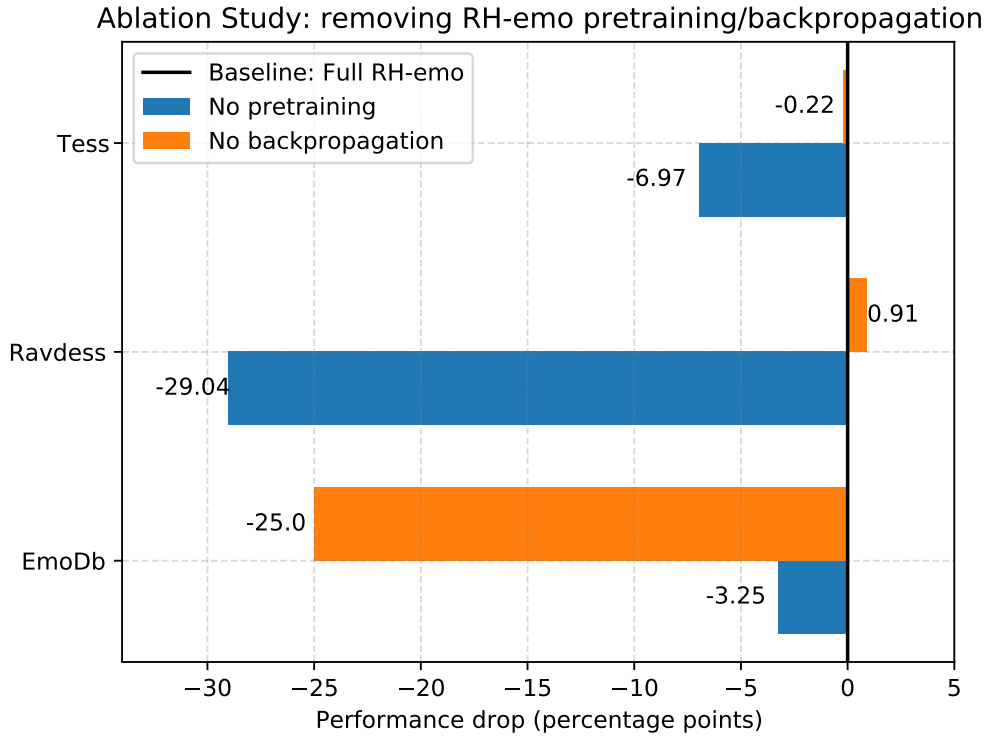


FIGURE 6.4: Ablation study results. The x axis shows the average difference in test accuracy (among the quaternion-valued VGG16, AlexNet and ResNet-50) obtained by removing the RH-emo pretraining (blue lines) and backpropagation (orange lines).

5% and 1% of the available data. The yellow and red lines are the baselines, respectively with and without CNNs pretraining on IEMOCAP. Instead, the green and blue lines show the trend for the QCNNs + RH-emo, respectively with and without CNNs pretraining.

The results of this ablation study clearly point out that our method can provide consistent performance improvement even in conditions with less data. In all cases but one (5% of training data) our pretrained approach surpasses both real-valued baselines. This is a convenient property for SER tasks, considering the general scarcity of emotion-labelled speech audio data.

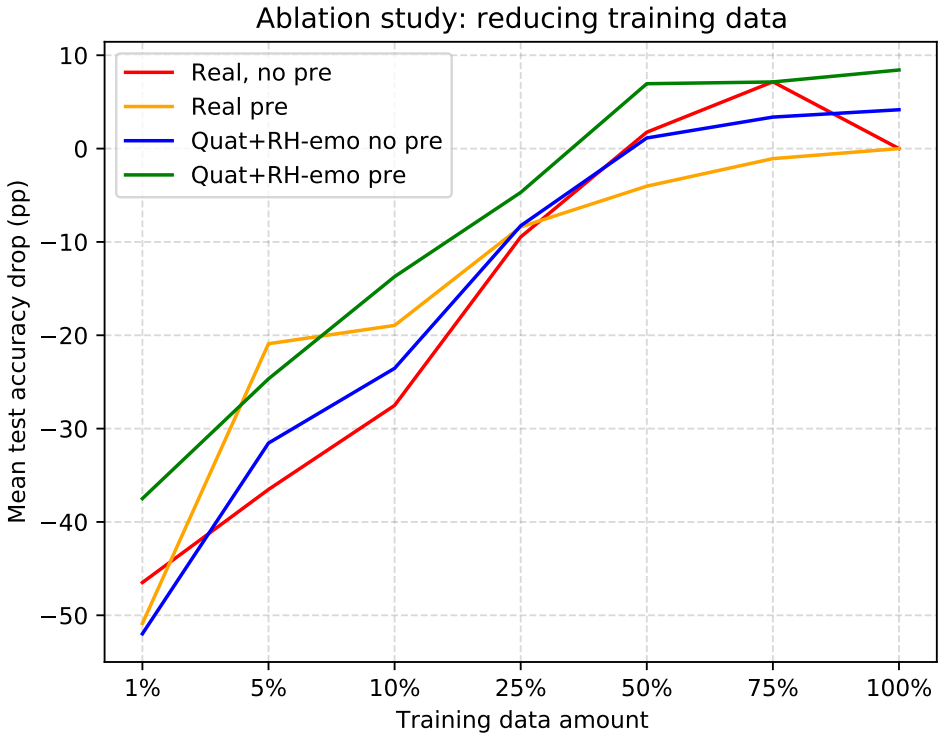


FIGURE 6.5: Ablation study results. The y axis shows the test accuracy drop of each model, compared to the baselines that use 100% of the training data. Each point in the line shows the average performance among the real-valued (red, yellow) and quaternion-valued (blue, green) VGG16, AlexNet, and ResNet-50 architectures for all corpora. The x axis shows the percentage of available training and validation data used. The data reduction rates shown in the x axis are a discrete set: we trained only on the data percentage values that are shown and not on intermediate values. We use the full test set in all cases, in order to have a consistent performance measure.

6.5 Discussion

6.5.1 Resource Savings

RH-emo permits to spare a considerable amount of parameters. Compared to the real counterparts, the quaternion VGG16 uses the ~6% of the parameters, while the quaternion AlexNet and ResNet-50 use the ~25%. The difference between the VGG16 and the others is due to the lack of adaptive average pooling (as described above). Therefore, on the one hand, the use

of quaternion-valued layers instead of real-valued ones permits to drop the number of parameters by a factor of 0.25, while, on the other hand, the smaller feature dimensionality obtained with the embeddings further cuts down the number of parameters by a factor of 0.25. This in turn permits the reduction of the model's memory requirements and training time. In our implementation, the embeddings computation happens during the training for every batch and, therefore, both the main network and the RH-emo feature extractor are loaded into the memory. This simulates a plausible application scenario of RH-emo, where the embeddings need to be computed in real-time. Although it is possible to pre-compute the embeddings as part of the preprocessing pipeline, further reducing the memory demand and computation time. As regards the memory demand, in our setup the quaternion networks require on average the 84.2% of memory, compared to their real-valued equivalents. For the VGG16 (where we don't apply average pooling) the memory demand is approximately the 70%, for AlexNet the 89% and for ResNet-50 the 93%. Regarding the training time, the epoch duration of our quaternion networks compared to the real networks is approximately the 15.9% for VGG16, 88.1% for AlexNet and 162.6% for ResNet-50. These outcomes show that the maximum efficiency in terms of both memory demand and computation time is obtained for VGG16, where we take advantage of the reduced dimensionality of the embeddings. On the other hand, the accuracy improvement for ResNet-50 comes at the cost of an increased computation time with respect to the real networks, but still reducing the model's memory demand.

6.5.2 Reconstruction Properties

Figure 6.6 shows an example of the decoder's output of the pretrained RH-emo model. The *Input* subplot is the input magnitudes-only spectrogram

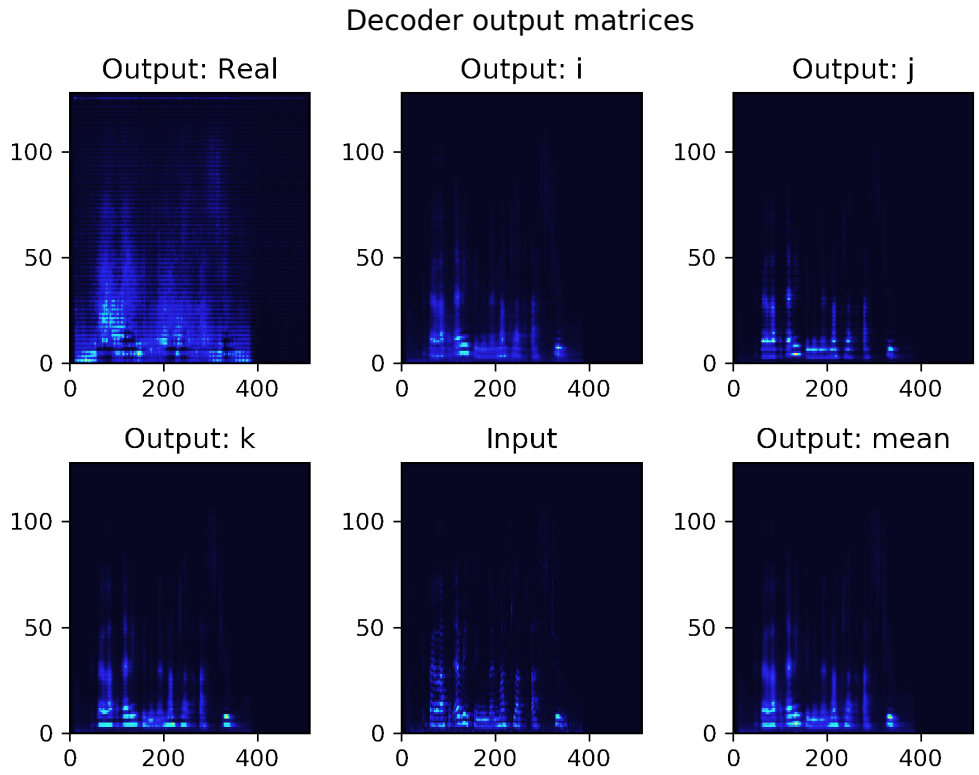


FIGURE 6.6: Example of RH-emo quaternion reconstruction of a speech spectrogram. *Input* is the magnitudes-only input spectrogram, *Output: real*, $\hat{\imath}, \hat{\jmath}, \hat{\kappa}$ are the four output matrices of RH-emo, respectively reconstructed from the discrete emotion, valence, arousal and dominance axes of the embeddings, *Output: mean* is the pixel-wise average of *Output: real*, $\hat{\imath}, \hat{\jmath}, \hat{\kappa}$ and is the matrix that is compared to the input in the loss function.

and the *Output: mean* is the element-wise mean of the quaternion separate axes and, therefore, the actual matrix that is compared to the input in the loss function. The sub-plots labelled as *Output: real*, $\hat{\imath}, \hat{\jmath}, \hat{\kappa}$ depict the separate quaternion axes, which are generated from the emotional embeddings: *real* from the discrete emotion classification matrix, and $\hat{\imath}, \hat{\jmath}, \hat{\kappa}$ from the valence, arousal and dominance channels, respectively.

By comparing the *Input* and the *Output: mean* plots, it is evident that the reconstruction is not perfect. While the time-wise articulation of the speech seems to be accurately reproduced, the model is not able to reconstruct in detail the most feeble harmonics of the signal. Although it is interesting the way the different quaternion axes are differentiated. In the real axis the model

seems to perform an operation similar to amplitude compression (obtainable, for instance, computing the square root of the matrix), bringing up the signal's quietest portions around the speech region. Instead, in the 3 complex axes ($\hat{\imath}, \hat{\jmath}, \hat{\kappa}$) different aspects of the signal are highlighted, focusing on different harmonics and/or temporal areas. Our intuition is that these representations may represent different "emotional points of view" of the input speech signal.

6.5.3 Limitations

Besides the numerous advantages that our approach provides, there are also some intrinsic limitations. The main constraint of our approach is that a pretrained RH-emo network can be used for only a fixed time scale. In this work we considered a temporal window of 4 seconds, which is well suited for most SER tasks and datasets. If a different time scale is needed, then a specific RH-emo has to be trained on purpose. Another limitation is that training with an end-to-end fashion is not possible, as a pretrained RH-emo is needed and the omission of the RH-emo pretraining stage leads to a drastic decrease of model's performance, as shown in Section 6.4.2.

6.5.4 Applications

The advantages provided by the combination of RH-emo and quaternion-valued networks suggest several application scenarios. Due to the substantial saving of trainable parameters, memory and training time, our approach is particularly suited for situations where limited resources are available and performance can not be sacrificed. Another useful property of RH-emo is that while the embeddings carry the necessary information to perform SER tasks (as proven by our experimental results), they also provide speaker anonymity,

as it is not possible to reconstruct the input spectrogram without the RH-emo pretrained weights. This could be exploited in situations where sensible speech data must me used for SER tasks.

6.6 Conclusions and Remarks

In this chapter we presented RH-emo, a semi-supervised approach to obtain quaternion emotional embeddings from real speech spectrograms. This method enables to perform speech emotion recognition tasks with quaternion-valued convolutional neural networks, using real-valued magnitudes spectrograms as input. We use RH-emo pretrained on IEMOCAM to extract quaternion embeddings from speech spectrograms, where the individual axes are optimized for the classification of different emotional characteristics: valence, arousal, dominance, and overall discrete emotion.

We compare the performance on SER tasks of real-valued CNNs fed with regular spectrograms and quaternion-valued CNNs fed with RH-emo embeddings. We evaluate our approach on a variety of cases, using 4 popular SER datasets (IEMOCAP, RAVDESS, EmoDb, TESS) and with 3 widely-used CNN designs of increasing capacity (ResNet-50, AlexNet and VGG16). Our approach provides a consistent improvement in the test accuracy for all datasets, while using a considerably lower amount of resources. We obtained an average improvement of 6.01 pp for RAVDESS, 2.34 pp for EmoDb, and 0.97 pp for TESS and we spared up to 94% of the trainable parameters, up to the 30% of GPU memory and up to 84.1% of training time. Moreover, we performed additional experiments and ablations studies that confirm the properties and foundations of our approach. The results show that the combination of RH-emo and QCNNs is a suitable strategy to circumvent the high resource demand of SER models and that our approach provides consistent

performance improvement also in scenarios where the available training data is scarce.

In the next Chapter of this dissertation we merge the approach presented on this Chapter and anti-transfer learning. This enables to exploit the selectable disentanglement properties of the latter, while reducing the resource demand of models and maintaining their performance.

Chapter 7

Anti-Transfer Learning in the Quaternion Domain

In this final study of this dissertation we aimed at improving our anti-transfer learning approach by solving one of its major issues: its high resources demand. This chapter contains an extension of the work presented in Conference Paper 1, merging and further expanding the approaches presented in Chapter 5 and 6.

7.1 Introduction

As introduced in Chapter 5, anti-transfer Learning is a suitable approach to selectively promote task invariance through the introduction of a deep feature loss term, the *anti-transfer loss*. When properly minimized, this enhances the divergence of the features developed in a specific layer of a CNN with the features developed by an identical network trained on an orthogonal task. We demonstrate that anti-transfer learning can improve the performance of models in different domains and tasks. In particular, we show that it can improve the independence of SER models to the context (specific uttered words that can bias the result) and characteristics of the speaker's voice.

Nevertheless, the improved accuracy comes at a cost of increased computational and memory demands at training time. On the one hand, the

increased memory demand of anti-transfer is mainly due to the fact that the gram matrices and the pre-trained classifier network need to be kept in memory on top of the input data and the main network. On the other hand, the increased computational resources are mainly related to the additional forward propagation of the input data to both the main and the parallel networks and the computation of the gram matrices and the distance function. As discussed in Chapter 5, this leads to an increase in computation time, with an average epoch requiring 2.8 times longer to train than without anti-transfer.

In order to reduce the impact of anti-transfer on the resource demand, we propose to exploit the properties of quaternion information processing. In particular, we apply anti-transfer learning to quaternion-valued CNNs fed with embeddings generated with RH-emo (as described in Chapter 6). We have shown that this technique is able to improve or maintain the performance of SER models, while substantially reducing the number of trainable parameters, and thus resources. Therefore, we expect that combining anti-transfer and Rh-Emo QCNNs will maintain or enhance the properties of anti-transfer learning, but reducing the model's demand for memory and computation time.

In this study we expand the work of Chapter 5 by using more networks (adding AlexNet and ResNet50 to the already present VGG16), by applying anti-transfer also to quaternion versions of these networks and by applying dual anti-transfer for SER tasks. Moreover, we expand Chapter 6 by applying anti-transfer to Rh-Emo QCNNs and evaluating model performance on a new setup where we test on speakers and/or contexts unseen during the training.

7.2 Method

The anti-transfer loss is defined as a scalar coefficient β multiplied by the squared cosine similarity of the vectorized Gram matrices computed on the feature maps of 2 identical layers, where one is pretrained on an orthogonal task and one is being trained, as explained in Chapter 5. Equation 5.2 formally shows this principle.

While we originally tested this approach only on real-valued CNNs, the objective of this additional research is to apply the same anti-transfer learning principle to quaternion-valued CNNs.

We left the original anti-transfer loss definition unchanged, as it is already compatible with feature maps that are generated by quaternion-valued convolution layers for the following reasons. In the first instance, the output of quaternion convolution layers has the same output shape as corresponding real-valued convolution layers. However, in quaternion convolution layers, consecutive kernels are aggregated in groups of 4, where each group uses $1/4$ of the sub-matrices (channels) to build its kernels. For instance, a real-valued convolution layer that has 4 channels as input and 4 channels as output contains 4 kernels shaped as $4\,channels \times time\ dim \times features\ dim$, i.e., it contains a total of 16 sub-matrices shaped as $time\ dim \times features\ dim$. An equivalent quaternion-valued convolution layer contains only 4 different sub-matrices in total, because the same 4 are permuted and reused to build all different kernels. Therefore, all kernels are made with the same sub-matrices, but in each kernel, they have a different ordering along the channel dimension Grassucci, Zhang, and Comminiello, 2022. This behavior causes the output feature maps to encode in quadral structures, since consecutive groups of 4 output channels are generated with kernels that re-use the same shifted sub-matrices. Despite their quadral organization, the output matrices of a quaternion convolution layer can be treated as real-valued matrices, because

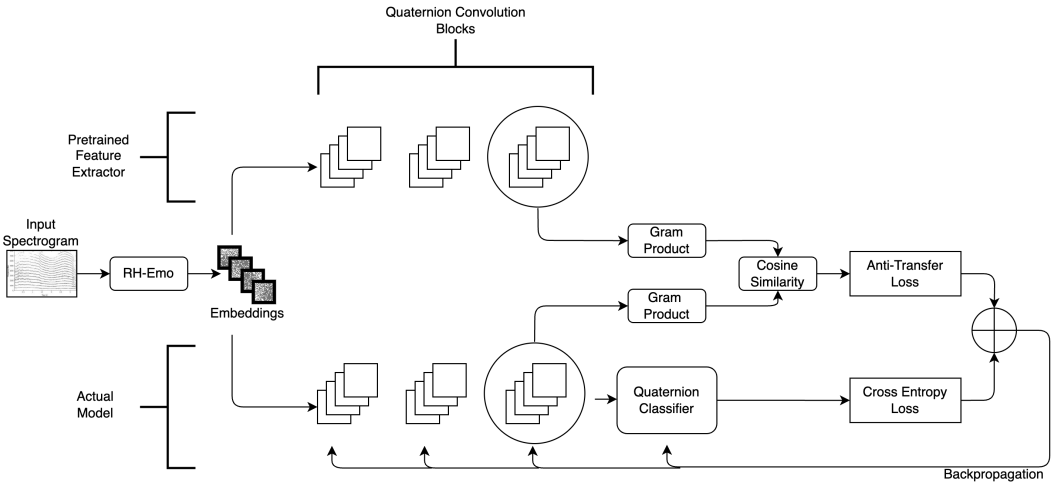


FIGURE 7.1: Block diagram of a QCNN network with **anti-transfer learning** applied to a SER classification task. The input spectrogram is first propagated to RH-emo, which generates quaternion emotional embeddings that feed the QCNNs. Anti-transfer is applied to the feature maps generated by selected quaternion convolution blocks.

each channel contains unique information.

We use the Gram matrix as a feature aggregation strategy in order to compare all possible channel combinations at once in the ATL, using a limited amount of memory and limiting the impact of possible channel permutations. For this, we calculate the Gram matrix on the quaternion components as real numbers. By doing this, we treat correlations within the quaternion axes in the same way as correlations between components of different quaternion which is justified by the unique information contained on each axis.

Figure 7.1 depicts a block diagram of a QCNN with anti-transfer learning applied. An input spectrogram is first forward propagated through a pretrained RH-emo network. The latter generates quaternion emotional embeddings that feed both the pretrained feature extractor and the network being currently trained. The anti-transfer loss is then computed exactly as for real-valued CNNs, as described above.

TABLE 7.1: Results of the **pretraining** in terms of classification accuracy. The *Train* and *Test* columns contain the accuracy on the train and test sets.

Network	Type	Task	Accuracy	
			Train	Test
VGG16	Real	Speaker	99.8	96.5
	Real	Word	99.8	99.5
	Quat	Speaker	99.24	89.47
	Quat	Word	99.85	99.64
AlexNet	Real	Speaker	99.88	99.88
	Real	Word	99.68	99.65
	Quat	Speaker	93.28	75.81
	Quat	Word	98.25	96.44
ResNet	Real	Speaker	91.93	91.78
	Real	Word	99.88	99.87
	Quat	Speaker	90.67	78.77
	Quat	Word	94.02	93.25

7.3 Evaluation

We evaluate the potentialities and properties of anti-transfer learning applied to QCNNs for SER tasks, with an experimental setup similar to the one described in Chapter 6. In the following Section we report the most important details.

7.3.1 Experimental Setup

Our objective is to compare AT performed in the real and quaternion domain to each other and to the same real and quaternion networks without any AT strategy applied. In this study we used the same datasets as in Chapter 6:

1. IEMOCAP, Interactive Emotional Dyadic Motion Capture Database (Busso et al., 2008),

2. RAVDESS, the Ryerson Audio Visual Database of Emotional Speech and Song (Livingstone and Russo, 2018a).

3. EmoDb, a Database of German Emotional Speech (Burkhardt et al., 2005).

4. TESS, the Toronto Emotional Speech Set (Dupuis and Pichora-Fuller, 2011).

We apply the same preprocessing stages: we cut signals into 4-seconds non-overlapping frames, we compute the STFT and discard the phase information and we zero-pad all matrices to a shape of 512 x 128. In addition to this, we normalize the magnitudes to a 0 to 1 range for the quaternion valued networks (as required by RH-emo) and to 0 mean and unity standard deviation for real-valued networks, which helps convergence.

For the quaternion networks we keep the same architecture as for the real CNNs, but we use quaternion-valued convolution and quaternion-valued fully connected layers instead of the canonical real-valued ones. There is an exception in the final layer of the networks, which are real-valued also in the QCNNs. For the real-valued networks, we use the magnitudes-only spectra as input, while for the quaternion networks we use the embeddings generated with RH-emo pretrained on IEMOCAP.

We use the same implementation of VGG16 (Simonyan and Zisserman, 2015), AlexNet (Krizhevsky, Sutskever, and Hinton, 2012) and ResNet-50 (He et al., 2016) mentioned in Chapter 6. Note that we removed the adaptive average pooling layer between the convolutional feature extractor and the dense classifier section on the VGG16 networks. This layer serves to reduce the feature extractor output shape to a fixed input shape of the classifier. When the RH-emo embeddings are used as input, however, the feature map is smaller than the classifier input. Removing the adaptive pooling layer and reducing the classifier input thus prevents the generation of redundant output in the adaptive layer and saves a significant number of network parameters.

As performed in Chapter 6, for all experiments we used a learning rate of 0.00001, the ADAM optimizer, and a batch size of 20 samples, we apply early stopping with a patience of 20 epochs on the validation loss and we split the training, validation, and test sub-sets with approximately 70%, 20% and 10% of the data, respectively.

We configured the experimental setup in order to show the difference of using AT on real and quaternion valued CNNs, as well as the difference of using or removing AT for the same networks. To this extent, we have paid attention to perform all experiments with the exact same configuration in order to isolate at best the features we want to highlight. This makes our results incompatible with published state-of-the-art results for the same datasets, as the latter are obtained with more complex setups that are different for each dataset and recreating them would make it more difficult to isolate and understand the properties of our approach, as further explained in Section 6.3.

We perform AT with 2 orthogonal tasks: Word Recognition and Speaker Recognition, as in the SER experiments of Chapter 5. For the first orthogonal task we pretrained our networks on the Librispeech (Panayotov et al., 2015) dataset, where we extracted single word segments of 1000 different classes. For the latter, instead, we used IEMOCAP. Table 7.1 shows the pretraining accuracy results for all network architectures.

7.3.2 Experimental Results

Tables 7.2, 7.3, 7.4 and 7.5 show the experimental results we obtained for IEMOCAP, RAVDESS, EmoDb and TESS, respectively. Here we compare the performance achieved by real and quaternion-valued baselines without AT applied to the same networks with AT applied in different ways:

- Single AT on speaker recognition (Single S)
- Single AT on word recognition (Single W)

TABLE 7.2: Results for IEMOCAP

Arch.	Method	AT Type	Params	AT Layers	Rnd-s acc.	Act-s acc.
VGG16	Real	/	1.6×10^8	/	62.87	58.29
	Real-AT	Single S	1.6×10^8	11,10	72.86	59.27
	Real-AT	Single W	1.6×10^8	1,11	70.09	59.35
	Real-AT	Dual SW	1.6×10^8	11,9	73.04	**60.41**
	Real-AT	Dual WS	1.6×10^8	8,8	71.07	59.67
	Quat	/	1×10^7	/	71.1	57.8
	Quat-AT	Single S	1×10^7	11,12	72.23	58.37
	Quat-AT	Single W	1×10^7	2,13	**73.57**	57.97
	Quat-AT	Dual SW	1×10^7	5,12	71.25	58.46
	Quat-AT	Dual WS	1×10^7	4,9	72.14	58.62
AlexNet	Real	/	5.7×10^7	/	63.33	55.85
	Real-AT	Single S	5.7×10^7	2,2	73.3	**59.11**
	Real-AT	Single W	5.7×10^7	2,2	73.57	57.07
	Real-AT	Dual SW	5.7×10^7	2,2	**74.11**	58.7
	Real-AT	Dual WS	5.7×10^7	2,2	73.57	58.13
	Quat	/	1.4×10^7	/	70.31	54.98
	Quat-AT	Single S	1.4×10^7	2,2	70.45	58.62
	Quat-AT	Single W	1×10^7	2,2	70.45	56.38
	Quat-AT	Dual SW	1.4×10^7	1,2	68.66	58.29
	Quat-AT	Dual WS	1.4×10^7	2,2	70.8	56.24
ResNet	Real	/	2.3×10^7	/	57.2	42.6
	Real-AT	Single S	2.3×10^7	5,3	70.89	48.46
	Real-AT	Single W	2.3×10^7	5,5	71.79	54.07
	Real-AT	Dual SW	2.3×10^7	2,5	70.8	53.58
	Real-AT	Dual WS	2.3×10^7	1,2	70.45	57.15
	Quat	/	4.9×10^6	/	71.2	56.65
	Quat-AT	Single S	4.9×10^6	5,5	71.43	57.325
	Quat-AT	Single W	1×10^7	4,1	71.43	56.59
	Quat-AT	Dual SW	4.9×10^6	5,1	72.14	57.48
	Quat-AT	Dual WS	4.9×10^6	1,4	**73.04**	**58.04**

- Dual AT, first on speaker and then on word recognition (Dual SW)
- Dual AT, first on word and then on speaker recognition (Dual WS)

We recall that dual AT is achieved by training a model with anti-transfer for one orthogonal task and using the result of that training to initialize the weights of a new model, which is then trained with anti-transfer on the second orthogonal task, as introduced in Section 5.2. In this study we use the

same AT layer on both orthogonal tasks when we perform dual AT. The tables show the test accuracy obtained by splitting the training, validation and test set randomly (Rnd-s acc. column) and actor-wise (Act-s acc. column). While in the first split setting the samples recorded from all actors appear in all sets, in the latter setting the models are tested on speakers totally unseen during the training stage, and therefore better show the model's generalization capabilities. On each case, we select the AT layer that gives the best validation accuracy (even though we report the test accuracy). Column "AT Layers" lists the layers that respectively provide the exposed Rnd-s acc. and Act-s acc. results.

7.3.3 Discussion

In general, the experimental results we obtained demonstrate that AT can be successfully applied to quaternion-valued neural networks, even though the average accuracy performance improvement is higher for real-valued ones. Moreover, the outcomes show that AT is compatible with multiple CNNs designs, even though its impact is different for different networks. This further supports the foundation of AT, showing that it is not restricted to the real-valued VGG16 network, as by our first study on AT presented in Chapter 5.

Table 7.6 shows the performance difference that we have obtained isolating AT applied in different setups AT provides a consistent improvement both for real and quaternion-valued networks. For all datasets, the best test accuracy result is provided by a network where AT is applied for the random split configuration. Whereas, in the actor-wise split arrangement there is one only case where the real network without AT gives the best result: TESS with ResNet architecture. AT provides an average improvement of 3.2 percentage points (pp) for real-valued networks and of 1.51 pp for quaternion-valued networks. Nevertheless, for the cases considered in this study, there is not

TABLE 7.3: Results for RAVDESS

Arch.	Method	AT Type	Params	AT Layers	Rnd-s acc.	Act-s acc.
VGG16	Real	/	1.6×10^8	/	41.06	45.0
	Real-AT	Single S	1.6×10^8	2,9	49.56	46.67
	Real-AT	Single W	1.6×10^8	9,1	42.89	41.67
	Real-AT	Dual SW	1.6×10^8	2,2	45.56	40.62
	Real-AT	Dual WS	1.6×10^8	1,9	46.22	43.75
	Quat	/	1×10^7	/	49.85	36.67
	Quat-AT	Single S	1×10^7	2,2	57.11	51.67
	Quat-AT	Single W	1×10^7	12,13	55.11	48.33
	Quat-AT	Dual SW	1×10^7	3,2	57.78	**55.0**
	Quat-AT	Dual WS	1×10^7	11,2	**60.44**	46.67
AlexNet	Real	/	5.7×10^7	/	46.36	45.0
	Real-AT	Single S	5.7×10^7	1,2	47.78	**53.33**
	Real-AT	Single W	5.7×10^7	2,1	44.22	50.0
	Real-AT	Dual SW	5.7×10^7	2,2	47.56	**53.33**
	Real-AT	Dual WS	5.7×10^7	2,2	46.89	48.33
	Quat	/	1.4×10^7	/	43.94	38.33
	Quat-AT	Single S	1.4×10^7	2,3	50.22	35.0
	Quat-AT	Single W	1×10^7	2,4	49.56	36.67
	Quat-AT	Dual SW	1.4×10^7	2,1	**53.78**	43.33
	Quat-AT	Dual WS	1.4×10^7	2,4	50.89	40.0
ResNet	Real	/	2.3×10^7	/	43.48	48.33
	Real-AT	Single S	2.3×10^7	3,3	50.67	53.33
	Real-AT	Single W	2.3×10^7	4,4	44.44	55.0
	Real-AT	Dual SW	2.3×10^7	5,1	51.11	50.0
	Real-AT	Dual WS	2.3×10^7	1,5	47.11	53.33
	Quat	/	4.9×10^6	/	55.15	48.33
	Quat-AT	Single S	4.9×10^6	2,2	57.78	56.67
	Quat-AT	Single W	1×10^7	3,5	**59.78**	**58.33**
	Quat-AT	Dual SW	4.9×10^6	3,2	56.44	55.
	Quat-AT	Dual WS	4.9×10^6	3,3	56.89	**58.33**

a significantly best approach overall. In the random split configuration, for RAVDESS and EmoDb the quaternion-valued approach gives the best result, real and quaternion networks equally saturate the test accuracy for TESS, and the real-valued approach provides the best performance for IEMOCAP. Instead, in the actor-wise split arrangement, for IEMOCAP and TESS the real-valued method gives the best test accuracy, for EmoDb real and quaternion networks achieve the same absolute best result and for RAVDESS the quaternion approach provides the highest accuracy. Single AT achieves an

TABLE 7.4: Results for EmoDb

Arch.	Method	AT Type	Params	AT Layers	Rnd-s acc.	Act-s acc.
VGG16	Real	/	5.7×10^7	/	**70.0**	50.0
	Real-AT	Single S	5.7×10^7	3,2	59.17	48.75
	Real-AT	Single W	5.7×10^7	11,12	58.33	**52.5**
	Real-AT	Dual SW	5.7×10^7	12,4	60.0	50.0
	Real-AT	Dual WS	5.7×10^7	11,3	55.0	48.75
	Quat	/	1.4×10^7	/	50.0	43.75
	Quat-AT	Single S	1.4×10^7	2,2	43.33	50.0
	Quat-AT	Single W	1×10^7	2,10	55.0	47.5
	Quat-AT	Dual SW	1.4×10^7	3,2	50.83	50.0
	Quat-AT	Dual WS	1.4×10^7	2,10	50.83	46.25
AlexNet	Real	/	2.3×10^7	/	47.0	43.75
	Real-AT	Single S	2.3×10^7	2,2	59.17	45.0
	Real-AT	Single W	2.3×10^7	2,1	59.17	43.75
	Real-AT	Dual SW	2.3×10^7	2,2	57.5	45.0
	Real-AT	Dual WS	2.3×10^7	2,2	**60.83**	45.0
	Quat	/	4.9×10^6	/	49.0	**52.5**
	Quat-AT	Single S	4.9×10^6	1,1	51.67	50.0
	Quat-AT	Single W	1×10^7	1,1	52.5	50.0
	Quat-AT	Dual SW	4.9×10^6	1,1	50.0	50.0
	Quat-AT	Dual WS	4.9×10^6	5,1	51.67	50.0
ResNet	Real	/	1.6×10^8	/	48.0	51.25
	Real-AT	Single S	1.6×10^8	3,5	57.5	52.5
	Real-AT	Single W	1.6×10^8	4,4	58.33	52.5
	Real-AT	Dual SW	1.6×10^8	5,5	54.17	**53.75**
	Real-AT	Dual WS	1.6×10^8	4,3	56.67	50.0
	Quat	/	1×10^7	/	73.0	47.5
	Quat-AT	Single S	1×10^7	3,2	**73.33**	55.0
	Quat-AT	Single W	1×10^7	4,4	63.33	51.25
	Quat-AT	Dual SW	1×10^7	4,3	66.67	**53.75**
	Quat-AT	Dual WS	1×10^7	4,4	66.67	51.25

average improvement of 1.97 pp, whereas the improvement for dual AT is of 2.75 pp. In the random split configuration the average improvement is of 2.71 pp, instead on the actor-wise split setup it is limited to 2 pp. AT has a different impact on each network architecture, with an average improvement of 1.89 pp for VGG16, 2.65 pp for AlexNet and 2.52 for ResNet50. It has a different effect also on each individual dataset, providing an average performance boost of 4.37 pp, 3.64 pp, 0.7 pp and 0.71 pp for IEMOCAP, RAVDESS, EmoDb and TESS, respectively. Furthermore, also the 2 different orthogonal

TABLE 7.5: Results for TESS

Arch.	Method	AT Type	Params	AT Layers	Rnd-s acc.	Act-s acc.
VGG16	Real	/	1.6×10^8	/	97.62	14.29
	Real-AT	Single S	1.6×10^8	3,4	99.64	13.29
	Real-AT	Single W	1.6×10^8	3,4	99.64	**26.43**
	Real-AT	Dual SW	1.6×10^8	3,2	**100.0**	21.0
	Real-AT	Dual WS	1.6×10^8	3,6	**100.0**	25.57
	Quat	/	1×10^7	/	97.62	13.86
	Quat-AT	Single S	1×10^7	1,11	97.86	19.43
	Quat-AT	Single W	1×10^7	11,4	98.57	18.29
	Quat-AT	Dual SW	1×10^7	2,11	98.93	19.0
	Quat-AT	Dual WS	1×10^7	9,3	97.86	18.29
AlexNet	Real	/	5.7×10^7	/	98.01	14.43
	Real-AT	Single S	5.7×10^7	3,3	**99.29**	14.43
	Real-AT	Single W	5.7×10^7	3,3	**99.29**	14.29
	Real-AT	Dual SW	5.7×10^7	3,3	**99.29**	16.14
	Real-AT	Dual WS	5.7×10^7	3,3	**99.29**	14.86
	Quat	/	1.4×10^7	/	98.56	21.57
	Quat-AT	Single S	1.4×10^7	4,1	98.57	23.0
	Quat-AT	Single W	1×10^7	2,4	98.57	**23.29**
	Quat-AT	Dual SW	1.4×10^7	2,1	98.57	21.86
	Quat-AT	Dual WS	1.4×10^7	3,1	98.93	20.0
ResNet	Real	/	2.3×10^7	/	97.38	**40.0**
	Real-AT	Single S	2.3×10^7	4,3	99.29	17.0
	Real-AT	Single W	2.3×10^7	3,1	98.57	16.14
	Real-AT	Dual SW	2.3×10^7	4,2	98.93	18.57
	Real-AT	Dual WS	2.3×10^7	4,4	99.29	18.0
	Quat	/	4.9×10^6	/	99.76	19.71
	Quat-AT	Single S	4.9×10^6	5,5	99.64	20.0
	Quat-AT	Single W	1×10^7	4,2	99.29	20.0
	Quat-AT	Dual SW	4.9×10^6	4,3	99.64	20.57
	Quat-AT	Dual WS	4.9×10^6	1,2	**100.0**	19.71

tasks used to pretrain the AT classifiers have a distinct impact on the model's performance: speaker recognition boosts the average test accuracy by 2.5 pp and word recognition by 2.21 pp.

To substantiate the most pertinent claims inferred from the results we performed a two-sided Wilcoxon signed-rank test comparing the best baseline results without AT applied and the best results with AT applied, revealing statistical significance with $p < 0.001$. This confirms the significance results we obtained in Chapter 5 in a broader setup and strongly supports the claim

TABLE 7.6: Test accuracy performance gain by applying anti-transfer learning in multiple setups

Comparison setup	Test acc. improvement
real / quat	**3.2** / 1.51
singleAT / **dualAT**	1.97 / **2.75**
randSplit / actorSplit	**2.71** / 2
speaker / word	**2.5** / 2.21
VGG16 / **AlexNet** / ResNet-50	1.89 / **2.65** / 2.52

that AT can be successfully applied in the quaternion domain and with multiple CNN network designs.

7.3.4 Per-layer AT performance

Figures 7.2, 7.3, 7.4 and 7.5 show the average test accuracy obtained for each convolution block of our networks, with different modalities: single AT quaternion-valued, dual AT quaternion-valued, single AT real-valued and dual AT real-valued, respectively. While for Tables 7.2, 7.3, 7.4 and 7.5 we have selected the best layers according to the validation accuracy score, in order to simulate a real use case, in this Section we show the per-layer test accuracy performance, in order to show how each layer performs on data completely unseen during the training. On each Figure, 5 colored lines are present, indicating the combination of orthogonal task and train/test split types adopted. Each point in these lines shows the average test accuracy obtained for a specific convolution layer among all corpora. Due to resources restrictions, for the ResNet50, we tested AT only on blocks of 10 convolution layers and not for each individual one. On all graphs, the green dotted line draws the mean value among all visible lines, indicating therefore the average behavior of AT among all setups, per AT layer.

The Figures show that the impact of AT is widely different on the architectures we adopted. Moreover, AT does not provide a performance improvement for all cases, confirming the importance of the AT layer selection, as

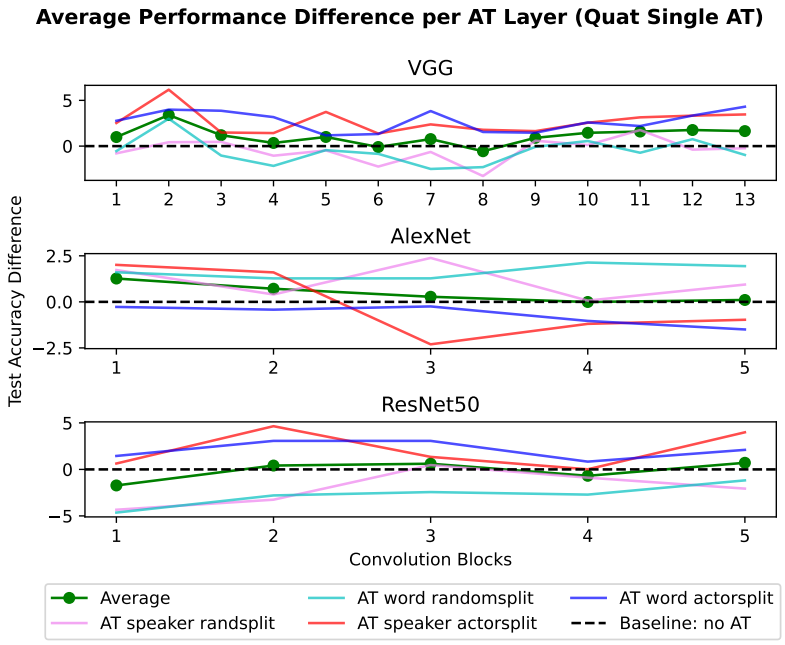


FIGURE 7.2: Average test accuracy for quaternion-valued single AT for each AT layer in different modalities.

discussed in Chapter 5. For the VGG16 network, there is a consistent performance drop for the middle layers. AlexNet shows an overall decreasing trend, with the first layers providing an average higher accuracy compared to the latter. For ResNet50, instead, the trend is flatter, with an average peak on the middle layers and lower performance on the extremes. In general, the graphs show that the performance of single AT is generally closer to the performance of dual AT for quaternion-valued networks, whereas dual AT provides a higher performance gain for real-valued ones, especially for AlexNet. For VGG16 and ResNet50 the highest improvement is given on the actor-wise split configuration for quaternion networks and on the random split configuration this applies for the real networks. Instead, the improvement on AlexNet is consistently higher for the random split configuration for both quaternion and real networks.

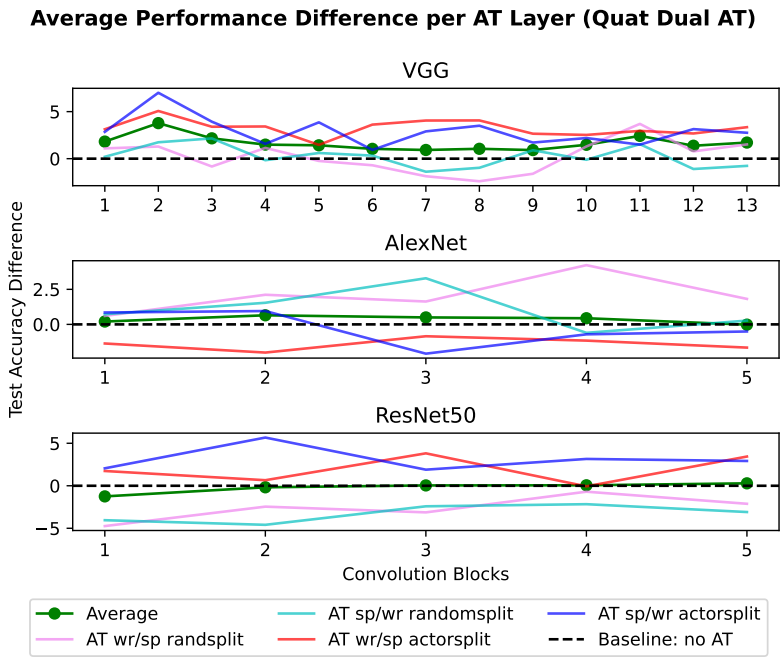


FIGURE 7.3: Average test accuracy for quaternion-valued dual AT for each AT layer in different modalities.

7.3.5 Resurce demand

Figure 7.6 and 7.7 respectively show the memory demand and training time difference for real and quaternion-valued networks and for each AT layer. We recall that on the AlexNet and ResNet50 architectures an adaptive average pooling layer is present between the convolutional feature extractor and the fully-connected classifier, while we removed this layer from the only VGG16, enabling to spare a major number of network parameters (as explained in Section 6.3.1). In addition to this, we recall that RH-emo can be used both as a pure pretrained feature extractor with no trainable parameters, and as a re-trainable part of a quaternion-valued network. As exposed in Section 6.4.2, the first case comes at the cost of a reduced average performance of models, however, on the other hand, it enables the computation of the quaternion embeddings before the training stage. This significantly reduces the memory demand at training time, as it eliminates the need to keep

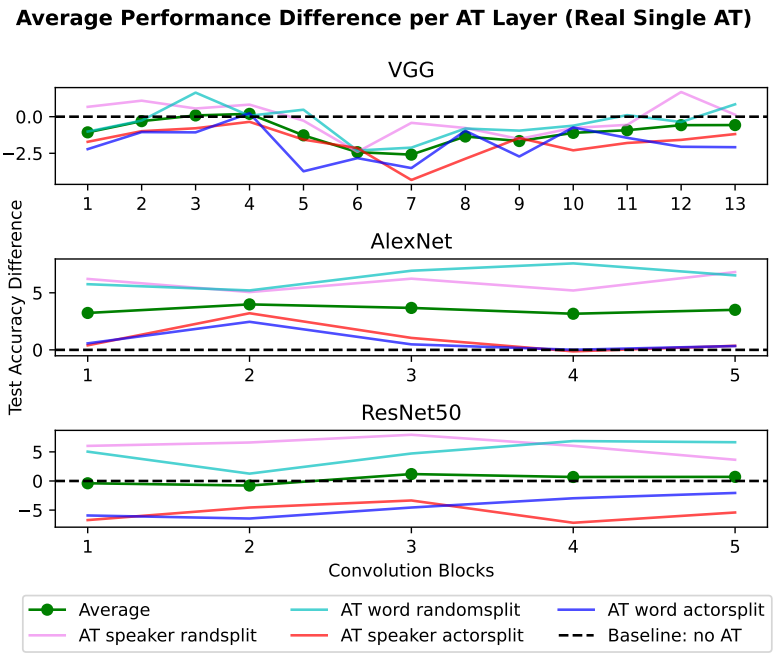


FIGURE 7.4: Average test accuracy for real-valued single AT for each AT layer in different modalities.

the RH-emo network in RAM.

In general, the best resource saving are provided by the VGG16 network with pre-computed embeddings, which saves 79.77% of the training time and 77.6% of the memory, on average among all AT layers.

It is evident that the deeper is the AT layer, the higher are the average training time and memory demand of models, both for real and quaternion-valued networks. This is mainly related to the fact that in the specific architectures we adopted, the number of convolution channels is increasingly larger from the first to the last feature extractor layer, resulting in larger Gram matrices to compare. Moreover, adding layers to a network naturally increases the number of operations required for the forward propagation.

As regards the training times, the VGG16 is the only architecture where the use of quaternion processing enhances the performance, with an average reduction of 29.98 seconds per epoch. This points out that the removal of the adaptive average pooling layer is fundamental to take advantage of the

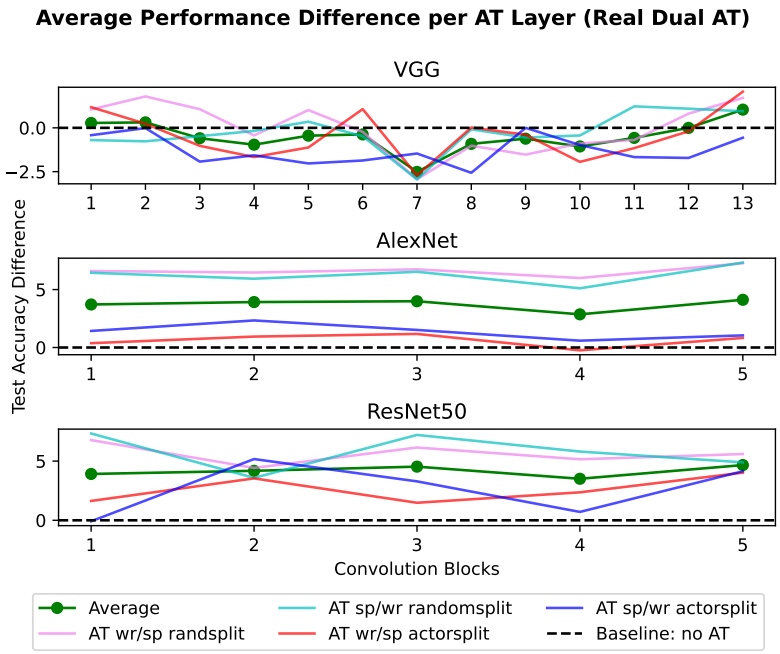


FIGURE 7.5: Average test accuracy for real-valued dual AT for each AT layer in different modalities.

resource optimization capabilities of quaternion information processing with the network architectures we tested. On the other hand, for the AlexNet and ResNet50 the quaternion domain processing comes at a cost of an average training time increasing of 0.19 and 22.57 seconds respectively.

Figure 7.7 shows the total amount of VRAM used during the training of our models, using a batch size of 10. VGG16 is the only architecture that provides a lower memory requirement for all AT layers when RH-emo embeddings are computed for each batch during training, with an average improvements of 4620 Mb. Again, this shows that the removal of the adaptive average pooling layer is fundamental in scenarios with limited resources. For AlexNet, the memory occupied by the QCNN + RH-emo is consistently higher for all convolution blocks (1388 Mb), while for ResNet50 real-valued networks occupy a lower amount of memory in the first 25 layers, whereas the quaternion-valued networks perform better on the last 25 layers, with an average difference of 83 Mb. On the other hand, if RH-emo is used as a

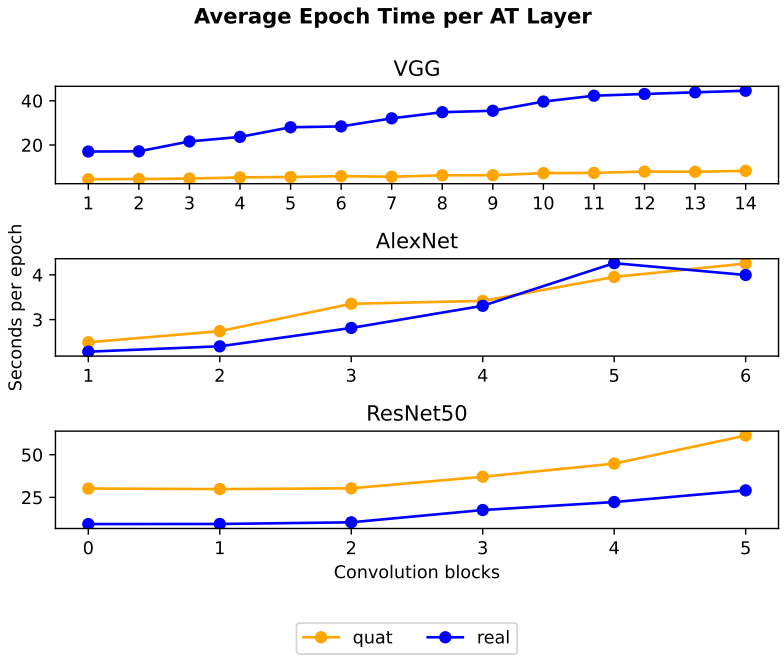


FIGURE 7.6: Average training time required for an epoch per AT layer.

pure pretrained features extractor, and therefore it is not fit in RAM at training time, quaternion-valued networks always require less memory resources compared to real ones, with an average decrease of 6677, 669 and 1973 Mb for VGG16, AlexNet and ResNet50, respectively.

7.3.6 Limitations

Despite the benefits provided by performing AT on quaternion-valued networks, there are important limitations. First, while it is possible to reduce the training time with AT applied on quaternion-valued networks, this requires to remove the adaptive average pooling layer from the standard implementations. This procedure is fundamental also to save memory during training. The best memory performance is given when the RH-emo embeddings are pre-computed and therefore RH-emo is not fit in the RAM alongside the other models and data at training time. Nevertheless, we have shown in

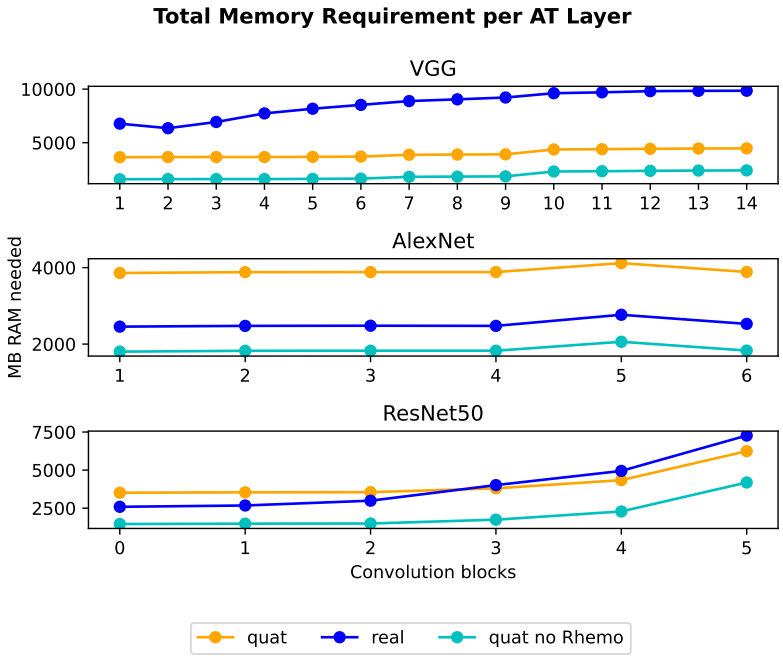


FIGURE 7.7: Total memory occupied by a model during training per AT layer.

Chapter 6 that this procedure can harm models' accuracy performance. Furthermore, the effectiveness of AT is strictly related to the choice of AT layer. According to our experiments there is no universal best choice for different architectures and datasets and choosing the best layer on each occasion may require to run computationally expensive and time consuming grid search experiments.

7.4 Conclusions and Remarks

In this Chapter we have presented a novel approach to exploit the benefits of QCNNs with RH-emo embeddings to reduce the impact of anti-transfer learning on resource demand, merging the approaches presented in Chapter 5 and 6.

This method retains most of anti-transfer benefits, while requiring a fraction of memory and computation time. We have not modified the original

AT implementation, as the output feature maps of quaternion convolution layers can be treated as matrices generated from equivalent real-valued convolution layers, and are therefore compatible with the computation of the AT loss.

The experimental results demonstrate that AT can be successfully applied to RH-emo QCNNs and to different convolutional designs (VGG16, AlexNet and ResNet50), obtaining a significant improvement in the test accuracy performance for SER tasks: 3.2 pp and 1.5 pp on average for real and quaternion-valued networks, respectively. Moreover, the average improvement is consistent in all different configurations we have tested (random split vs. actor split, real vs. quaternion-valued network, with pretraining on speaker vs. word recognition, single vs. dual AT) and with four different datasets: IEMOCAP, RAVDESS, EmoDb and TESS. The combination of AT and QCNNs fed with RH-emo embeddings permits to save a significant amount of resources: up to 79.77% of the training time (VGG16) and up to 77.6% of memory (VGG16 with pre-computed embeddings), making this approach a relevant option to exploit the disentanglement properties of AT in scenarios with restricted resources.

Chapter 8

Conclusions and Future Work

Human emotional expression is manifested with different nuances, which are related to personal characteristics and experience. This makes speech emotion recognition a challenging and ambiguous task even for humans.

This project started from the empirical observation that neural network models aimed at emotion recognition from speech audio signals consistently exhibit difficulties to generalize to unseen context and speakers. The main reason is that the features learned by a network can be overly specialized to the specific words or to the speech style of a person present in the training set. Moreover, since emotional information in speech involves long-term temporal dependencies, models need to analyze large temporal windows. This aspect, connected with the complexity of learning representation from audio signals, leads to SER models that need large amounts of memory and computation. In this thesis, we propose solutions to these issues in SER with neural networks and demonstrate their effectiveness.

As a preliminary study, we have applied standard neural network designs to address context dependency on an SER task, determining the need for specific solutions to disentangle the learnt representations from misleading features, in order to enhance the generalization of SER models.

We have initially focused on multi-time-scale modelling to enhance the independence of features learnt by CNNs to possible speed fluctuations of uttered words, addressing a speaker independent scenario. To this end, we

have developed a Multi-Time Scale convolution layer that learns features at different time scales at the same time, without increasing the number of trainable parameters. We have shown that invariance to a limited amount of speed fluctuations can help models generalize to unseen speakers. It makes the learnt features more robust to local dilation and compression on the time axis that can be due to the specific speech style of a person, rather than to the expressed emotion. Moreover, the improvement we obtain by applying Multi-Time Scale to CNN architectures is higher for smaller datasets, which confirms that enabling pattern recognition at different time scales with Multi-Time-Scale improves generalisation. Considering the general scarcity of emotion-labelled speech data, this is a desirable feature for SER applications. Nevertheless, despite the useful properties of Multi-Time Scale, its most critical limitation is that its effectiveness is limited to networks that employ large convolution kernels. This makes it incompatible with popular and powerful CNN architectures that are based on small convolution kernels, such as the ResNet and VGG network families.

In a second stage of our work, we have developed a general and supervised approach to achieve invariance towards a specific signal property, when that property can be extracted by a pretrained network. This approach goes beyond the specificity of Multi-Time-Scale to speed fluctuation, and therefore speaker invariance, and avoids its restriction to large-kernel CNNs. We achieve this disentanglement through the novel paradigm of anti-transfer learning, where we use a deep feature loss to maximize the dissimilarity of the features that are being learnt by a network to the features of another identical network that has been pretrained on an orthogonal task (the detection of the irrelevant property). Among other audio-related tasks, we have shown that anti-transfer learning can enhance the generalization capabilities of SER models, when performing anti-transfer with orthogonal networks pretrained

on word and speaker recognition. With the increasing availability of public datasets and pretrained models, chances grow that a suitable dataset or model can be found, but the selection of the orthogonal task needs careful consideration. Transfer learning is generally seen as a straightforward way to improve the performance of deep neural networks by indirectly using additional data. Our results show that taking into account the nature of the pretraining tasks is important. Treating related and orthogonal tasks differently can boost generalization significantly. On the other hand, despite the supervised disentanglement properties of anti-transfer, its main limitation is that it increases the demand for memory and training time. Nevertheless, the improved generalization may often be worth the effort, especially when pretrained models are available for orthogonal tasks.

In the third stage of this PhD, we have explored the application of quaternion information processing to CNN models for SER, aiming to reduce the high resource demand. In general, quaternion-valued networks can achieve comparable or better performance than real-valued networks, while reducing the number of network parameters and consequently lowering their resource demand. A drawback of quaternion-valued networks is that they need to be fed with data organized into quadral representation, where an intra-channel correlations must exist. In order to enable the use of QCNNs with monophonic audio signals we have developed RH-emo, a pretrained feature extractor that creates emotion-optimized embeddings that can be treated as a quaternion entity, and therefore can be used as input to a quaternion-valued network. We show that the combination of QCNNs and RH-emo embeddings enhances the performance of SER models, while cutting 3/4 of the network parameters, and consequently lowering the memory demand and training time of models. Due to the substantial saving of trainable parameters, memory and training time, our approach is particularly suited for situations where limited resources are available and performance can not be

sacrificed.

As a final study of this thesis, we have merged anti-transfer learning and quaternion information processing in order to take advantage of the disentanglement that anti-transfer provides, but limiting its high demand of resources. We show that anti-transfer can be effectively applied to QCNNs fed with RH-emo embeddings without modifying the AT loss algorithm and obtaining comparable results while saving a substantial amount of parameters and thus memory and training time. The generalization improvement we obtained with AT is consistent in all setups, making the combination of anti-transfer and RH-emo QCNNs suitable for SER scenarios with limited resources and where context and speaker independence are needed.

While our primary focus has been on understanding and enhancing the recognition of emotional cues in audio speech signals, the approaches we presented in this dissertation show potential implications and applications in diverse areas of speech processing and also to domains different from audio. On the one hand, anti-transfer can be considered as a versatile strategy for selectively introducing invariance to specific features. Although we have tested it across various audio-related tasks, its implementation is not specifc to audio or emotion recognition and offers potential utility in a broader spectrum of deep neural network-based tasks Moreover, its reliance on latent representation comparisons makes it adaptable to network architectures beyond CNNs, including recurrent and transformer networks. On the other hand, the implementation of RH-Emo can be potentially generalized to any problem where direct inference of a quadral representation from input data poses a challenge. While in our experiments we have fine-tuned RH-Emo quaternion axes for emotion-related features, this fundamental concept readily extends its applicability to various problem domains by integrating distinct classifiers or regressors into the RH-Emo training pipeline. Furthermore, it's worth highlighting that this approach is not constrained to the utilization of

CNNs, underscoring its flexibility.

8.1 Ethical, Legal, and Social Implications

The field of Speech Emotion Recognition is intertwined with pressing Ethical, Legal, and Social Implications (ELSI) due to the sensitive nature of speech data and human emotional behavior. ELSI considerations are paramount to ensure that SER technology aligns with ethical principles, legal frameworks, and societal values. Moreover, the ethical implications of resource-saving techniques must be considered as these techniques, while optimizing efficiency, raise concerns about potential compromises in fairness and equity. The work presented in this dissertation aligns with ELSI principles, encompassing means to ensure fairness and equitable outcomes from SER models. In the first instance, we have shown that anti-transfer learning can be adopted to introduce invariance towards specific features and when the latter are selected to be sensible characteristics as, for instance, gender and ethnicity, our approach can be used to obtain algorithmic fairness and to avoiding bias amplification. Therefore, by disentangling undesirable associations, anti-transfer learning can serve to promote equity, aligning with ethical principles of non-discrimination and inclusivity. Moreover, we have demonstrated that this technique can be successfully combined with resource saving techniques as quaternion-valued neural network. This aspect has important ethical implications, ensuring that resource conservation and efficiency, which are central components of sustainable practices aimed at reducing the environmental impact of human activities, can be obtained without amplifying existing biases or lead to exclusion.

8.2 Future work

The positive results obtained in this PhD thesis project encourage further investigation.

An immediate research objective is identifying more application areas for anti-transfer learning and studying it on larger datasets in different domains, which will enable a better understanding of the performance and comparison to standard benchmarks. Potential quantification of orthogonality of labels and its relationship to anti-transfer learning should be also addressed in future work. Further general goals for longer term research are a deeper understanding of how to measure invariance or achieve it across multiple tasks beyond dual anti-transfer and it will be interesting to apply anti-transfer learning to different neural network architectures, including non-convolutional ones. Moreover, we intend to further investigate the application of anti-transfer in the quaternion domain by devising similarity measures in the AT loss that directly exploit the quaternion structure of the feature maps, for example using quaternion distance metrics instead of the real-valued cosine distance. This could further reduce the resource demand when applying AT to quaternion-valued networks.

Another future research objective is to test RH-emo with different datasets and architectures, with multiple time scales and to different tasks. Moreover, we intend to apply the basis principle of RH-emo to different tasks where a quadral representation of input data can not be directly inferred from data, as for speech emotion. This means adapting the supervised part RH-emo to the nature of different tasks, for instance music genre recognition, where the embedded dimensions of the autoencoder are optimized for tempo, harmonic key, spoken words and instrument type recognition.

8.3 Concluding remarks

Despite the positive results we have obtained in this PhD thesis, many challenges remain open in the field of emotion recognition from speech audio signals. To date, the performance of neural networks for SER is far from matching the state-of-the-art for more popular tasks, such as medical image classification, behavioral data analysis or weather forecasting, in terms of evaluation metrics that are currently achieved. This situation calls for research into obtaining more accurate models and the creation of more extensive emotion labelled data collections. Moreover, due to the high degree of subjectivity of how emotion is perceived and expressed, novel and less biased methods for labelling data with emotional characteristics could help the design of improved SER models.

This line of work has raised an fundamental question: is it possible to learn a universal representation of human emotion that is completely disentangled from any personal or cultural bias? And if such representation exists, what could it tell us about the very nature of emotions? At the moment, the available emotion-labelled data is clearly not sufficient to get even close to answer these questions, but the methods presented in this PhD thesis can be a starting point to research towards this direction.

Appendix A

Quaternion Convolutional Neural Networks

Operations between quaternion numbers are defined in the quaternions algebra $\mathbb{H}$. A quaternion Q is a four-dimensional extension of a complex number, defined as $\mathbf{q} = q_0 + q_1\hat{\imath} + q_2\hat{\jmath} + q_3\hat{\kappa} = q_0 + q$, where, q_0, q_1, q_2, q_3 are real numbers, and $\hat{\imath}$, $\hat{\jmath}$ and $\hat{\kappa}$ are the quaternion unit basis. In this representation q_0 is the real part and $q_1\hat{\imath} + q_2\hat{\jmath} + q_3\hat{\kappa}$ is the imaginary part, where $\hat{\imath}^2 = \hat{\jmath}^2 = \hat{\kappa}^2 = -1$ and $\hat{\imath}\hat{\jmath} = -\hat{\jmath}\hat{\imath}$. From the latter assumption follows that the quaternion vector multiplication is not commutative. A quaternion can also be represented as a matrix of real numbers as:

$$\mathbf{q} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \tag{A.1}$$

Analogously to real and complex numbers, a set of operations can be defined in the quaternion space:

- **Addition**: $\mathbf{q} + \mathbf{p} = (q_0 + p_0) + (q_1 + p_1)\hat{\imath} + (q_2 + p_2)\hat{\jmath} + (q_3 + p_3)\hat{\kappa}$
- **Conjugation**: $\mathbf{q}^* = q_0 - q_1\hat{\imath} - q_2\hat{\jmath} - q_3\hat{\kappa}$
- **Scalar multiplication**: $\lambda\mathbf{q} = \lambda q_0 + \lambda q_1\hat{\imath} + \lambda q_2\hat{\jmath} + \lambda q_3\hat{\kappa}$

- **Element multiplication** (or **Hamilton product**):

$$\begin{aligned}\mathbf{q} \otimes \mathbf{p} &= (q_0 + q_1\hat{\imath} + q_2\hat{\jmath} + q_3\hat{\kappa})\,(p_0 + p_1\hat{\imath} + p_2\hat{\jmath} + p_3\hat{\kappa}) \\ &= (q_0p_0 - q_1p_1 - q_2p_2 - q_3p_3) \\ &+ (q_0p_1 + q_1p_0 + q_2p_3 - q_3p_2)\,\hat{\imath} \\ &+ (q_0p_2 - q_1p_3 + q_2p_0 + q_3p_1)\,\hat{\jmath} \\ &+ (q_0p_3 + q_1p_2 - q_2p_1 + q_3p_0)\,\hat{\kappa}.\end{aligned} \tag{A.2}$$

The quaternion convolutional neural network (QCNN) is an extension of the real-valued convolutional neural network to the quaternion domain. For each input vector of a quaternion layer the dimensions are split into four parts to compose a quaternion representation. In a quaternion-valued fully-connected layer, the parameters matrices are treated as a single quaternion entity with four components, although they are manipulated as matrices of real numbers (Gaudet and Maida, 2018).

In a quaternion layer, the dot product operations used in real layers are replaced with the Hamilton product (Equation A.2) between the input vector and a quaternion-represented weight matrix. This allows to process all input channels together as a single entity maintaining original intra-channels dependencies, because the weights submatrices are shared among the input channels. Consequently, quaternion layers permit to spare the 75% of free parameters compared to their real-valued equivalents because, as shown in Equation A.2, the same components are re-used to build the output matrix.

In a QCNN, the convolution of a quaternion filter matrix with a quaternion vector is performed as the Hamilton product between the real-valued matrices representation of the input vector and filters. A quaternion convolution between a quaternion input vector $\mathbf{x} = x_0 + x_1\hat{\imath} + x_2\hat{\jmath} + x_3\hat{\kappa}$ and a quaternion filter $W = W_0 + W_1\hat{\imath} + W_2\hat{\jmath} + W_3\hat{\kappa}$ can be defined as:

$$W * x = \begin{bmatrix} W_0 & -W_1 & -W_2 & -W_3 \\ W_1 & W_0 & -W_3 & W_2 \\ W_2 & W_3 & W_0 & -W_1 \\ W_3 & -W_2 & W_1 & W_0 \end{bmatrix} * \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y'_0 \\ y'_1\hat{\imath} \\ y'_2\hat{\jmath} \\ y'_3\hat{\kappa}' \end{bmatrix} \tag{A.3}$$

The optimization of quaternion-valued networks is identical to the one of a real network and can be achieved through regular backpropagation. This is possible because of the use of split activation and loss functions, as introduced in (Parcollet et al., 2019; Ujang, Took, and Mandic, 2011). These functions map a quaternion-like entity back to the real domain, consequently enabling the use of standard loss functions for the network training.

Appendix B

Example Conversation from the OMG Empathy Dataset

The official OMG Empathy Challenge page [1] offers a web player where the reader can interactively audition the recorded video of Story #2 "How I started a band". The following text is the transcription of this conversation obtained with the Amazon Transcribe service [2].

ACTOR: So we meet again.

LISTENER: Yeah, I meet you again.

ACTOR: How are you?

LISTENER: Yeah, I'm great.

ACTOR: Had a good week so far?

LISTENER: Yes.

ACTOR: Fantastic. Yeah, so today I'm going to tell you a bit more about myself. You got to hear a bit about more, or you got to hear a bit about me before, but now you're going to get to know a bit more. Do you know that I'm in the band?

LISTENER: Yes, I know.

ACTOR: You did know that I'm in the band?

[1] https://www2.informatik.uni-hamburg.de/wtm/omgchallenges/omg_empathy_description_19.html

[2] https://aws.amazon.com/transcribe/

LISTENER: Yeah.

ACTOR: Impressive. So I'm actually a big fan of music. Do you like music?

LISTENER: Yeah, I like music.

ACTOR: Because I actually do like music quite a lot. And actually, after I moved to Sweden, I really got into it, kind of. I really got into this whole. Aesthetic line of stuff, kind of. And I did actually nag my parents, like, I want to get a guitar, kind of. They said it was just a phase and it would go over, but I never really gave up, kind of. I kept trying to convince them to get me things. The best I could do, kind of is that eventually when we were going to choose kind of upper high school, then wanted to go to this music school, kind of. So I nagged them and eventually they said, that okay, fine, sure, you can go to this music school as long as you can read all these other subjects, kind of. Eventually they essentially did kind of give up on this point. I got what I wanted, so I was really excited to go to the school, kind of. And the school was amazing. I learned a lot while I was there. The only problem I had really was with the classical theory, classical, like music theory, which is like learning different kind of how to play different things and how the history of music also was a big part of that. So I had a lot of trouble with that mostly because of the teacher. And the teacher was really like I was pretty sure she was a demon in the shape of a person, kind of. She was just an awful human being. She was really bad. I did manage to make friends because of that, though, because what happened was that I met four other guys who kind of also hated we kind of developed a friendship over hating her together, kind of, which was great.

LISTENER: Yeah.

ACTOR: So that went on and I went on really enjoying learning how to play music and learning a lot about musical theory, as well as doing the other courses while still hating this one teacher because she was really an awful human being, kind of. And yeah, so that kept going on. And eventually, at some point in time, what happened was that we had to give this presentation in classical musical theory. And I was like and it was supposed to be a presentation front of the entire class. And I was really stressing out about this. I studied a lot. I was very nervous because partially because I wanted to do good and do well, right? But then also because I was so scared of this teacher and what she would say, because she would obviously judge me. So I went up and I gave my presentation. And for a moment I thought I'd done a really good job. People applauded. But then she really tore into my work. Everything presented. She said that this was wrong. Why did you say this? You need to speak up more. You didn't do well. And criticism was really harsh and mean. It was really, like, bitter criticism. So I really couldn't handle I really started to cry a bit when she started to be so cruel to me. And then I got home, I cried a lot. I talked to my parents and I said, look, this isn't going to work. I can't go to a school where I have to be with this teacher. And she's really just an awful person because we talked to the school about this teacher before and really they said that basically she's there to stay, kind of, so we couldn't do anything about it. So eventually I had to just swap schools, kind of, because I couldn't stay there. I needed that class to pass. What happened was, because the school shared some courses with the main school, I could swap over to the main school, kind of, and continue my education there. And that kind of worked. It was okay. I would have preferred to stay

at the main where I was the musical, but with that teacher, it really just didn't work. So time goes on, kind of. I go to college and then I meet these kind of four people I was friends with during the school, music school, because I didn't keep in contact with that much with them after that, which is cooler. I got to meet them again, but it was good to meet them again. And we just talked a bit and they were still really into music. And I was still, like I was still into music. I hadn't played or done anything serious with music for a while. But we eventually decided, like, yeah, we should start a band. Well, they suggested to me that I should join the band, kind of, because they were really playing with each other a bit, right? So I joined as the vocalist and they kind of all were they were on drums, bass, guitar, guitar, stuff like that. Yeah. So actually, we played a bit and then eventually we had our first show and it was really amazing. Like, when we were performing, I felt so happy, kind of. It was really a great experience, and it was really nice I could meet these guys and do that stuff with them. So I do have a bad recluse in a bad. That's how it happened.

Bibliography

Akçay, Mehmet Berkehan and Kaya Oğuz (2020). "Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers". In: *Speech Communication* 116, pp. 56–76.

Alaparthi, Varun Sai, Tejeswara Reddy Pasam, Deepak Abhiram Inagandla, Jay Prakash, and Pramod Kumar Singh (2022). "ScSer: Supervised contrastive learning for speech emotion recognition using transformers". In: *2022 15th international conference on human system interaction (HSI)*. IEEE, pp. 1–7.

Arjovsky, Martin, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz (2019). "Invariant risk minimization". In: *arXiv preprint arXiv:1907.02893*.

Badshah, Abdul Malik, Jamil Ahmad, Nasir Rahim, and Sung Wook Baik (2017). "Speech emotion recognition from spectrograms with deep convolutional neural network". In: *Int. Conf. on Platform Techn. and Service (PlatCon)*. IEEE, pp. 1–5.

Balemans, Dieter, Wim Casteels, Simon Vanneste, Jens de Hoog, Siegfried Mercelis, and Peter Hellinckx (2020). "Resource efficient sensor fusion by knowledge-based network pruning". In: *Internet of Things* 11, p. 100231.

Bansal, Sameer, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater (2019). "Pre-training on high-resource speech recognition improves low-resource speech-to-text translation". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis,*

MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Association for Computational Linguistics, pp. 58–68. DOI: 10.18653/v1/n19-1006. URL: https://doi.org/10.18653/v1/n19-1006.

Barros, Pablo, Nikhil Churamani, Egor Lakomkin, Henrique Siqueira, Alexander Sutherland, and Stefan Wermter (July 2018). "The OMG-Emotion Behavior Dataset". In: pp. 1–7. DOI: 10.1109/IJCNN.2018.8489099.

Beckmann, Pierre, Mikolaj Kegler, Hugues Saltini, and Milos Cernak (2019). "Speech-VGG: A deep feature extractor for speech processing". In: *CoRR* abs/1910.09909. arXiv: 1910.09909. URL: http://arxiv.org/abs/1910.09909.

Bell, Sean, C Lawrence Zitnick, Kavita Bala, and Ross Girshick (2016). "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks". In: *CVPR*, pp. 2874–2883.

Bengio, Yoshua (2012). "Deep Learning of Representations for Unsupervised and Transfer Learning". In: *Unsupervised and Transfer Learning - Workshop held at ICML 2011, Bellevue, Washington, USA, July 2, 2011*. Ed. by Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham W. Taylor, and Daniel L. Silver. Vol. 27. JMLR Proceedings. JMLR.org, pp. 17–36. URL: http://proceedings.mlr.press/v27/bengio12a.html.

Bhatti, Muhammad Waqas, Yongjin Wang, and Ling Guan (2004). "A neural network approach for human emotion recognition in speech". In: *IEEE Int. Symp. on Circuits and Syst. (ISCAS)*. Vol. 2. IEEE, pp. II–181.

Bhaykar, Manav, Jainath Yadav, and K. Rao (Feb. 2013). "Speaker dependent, speaker independent and cross language emotion recognition from speech using GMM and HMM". In: pp. 1–5. ISBN: 978-1-4673-5950-4. DOI: 10.1109/NCC.2013.6487998.

Bittner, Rachel M, Brian McFee, Justin Salamon, Peter Li, and Juan Pablo Bello (2017). "Deep Salience Representations for F0 Estimation in Polyphonic Music". In: *ISMIR*, pp. 63–70.

Blalock, Davis, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag (2020). "What is the state of neural network pruning?" In: *Proceedings of machine learning and systems* 2, pp. 129–146.

Bouali, Yassamine Lala, Olfa Ben Ahmed, and Smaine Mazouzi (2022). "Cross-Modal Learning for Audio-Visual Emotion Recognition in Acted Speech". In: *2022 6th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. IEEE, pp. 1–6.

Brignone, Christian, Gioia Mancini, Eleonora Grassucci, Aurelio Uncini, and Danilo Comminiello (2022). "Efficient Sound Event Localization and Detection in the Quaternion Domain". In: *IEEE Trans. Circuits and Systems II: Express Briefs* 69.5, pp. 2453–2457.

Bulow, Thomas and Gerald Sommer (2001). "Hypercomplex signals-a novel extension of the analytic signal to the multidimensional case". In: *IEEE Trans. Signal Process.* 49.11, pp. 2844–2852.

Burkhardt, Felix, Astrid Paeschke, Miriam Rolfes, Walter F Sendlmeier, and Benjamin Weiss (2005). "A database of German emotional speech". In: *Europ. Conf. on Speech Commun. and Techn.*

Busso, Carlos, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan (2008). "IEMOCAP: Interactive emotional dyadic motion capture database". In: *Language Resources and Evaluation* 42.4, pp. 335–359.

Cao, Houwei, Ragini Verma, and Ani Nenkova (2015). "Speaker-sensitive emotion recognition via ranking: Studies on acted and spontaneous speech". In: *Computer speech & language* 29.1, pp. 186–202.

Caruana, Rich (1995). "Learning many related tasks at the same time with backpropagation". In: *Advances in neural information processing systems*, pp. 657–664.

Chandna, Pritish, Marius Miron, Jordi Janer, and Emilia Gómez (2017). "Monoaural audio source separation using deep convolutional neural networks". In: *International conference on latent variable analysis and signal separation*. Springer, pp. 258–266.

Chen, Qifeng and Vladlen Koltun (2017). "Photographic Image Synthesis with Cascaded Refinement Networks". In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, pp. 1520–1529. DOI: 10.1109/ICCV.2017.168. URL: https://doi.org/10.1109/ICCV.2017.168.

Chernykh, Vladimir and Pavel Prikhodko (2017). "Emotion recognition from speech with recurrent neural networks". In: *arXiv preprint arXiv:1701.08071*.

Cho, Jaejin, Raghavendra Pappagari, Purva Kulkarni, Jesús Villalba, Yishay Carmiel, and Najim Dehak (2018). "Deep Neural Networks for Emotion Recognition Combining Audio and Transcripts". In: *Interspeech*, pp. 247–251.

Choi, Wonyoung, Jaechan Cho, Seongjoo Lee, and Yunho Jung (2020). "Fast constrained dynamic time warping for similarity measure of time series data". In: *IEEE Access* 8, pp. 222841–222858.

Chou, Ju-Chieh, Cheng-chieh Yeh, Hung-yi Lee, and Lin-Shan Lee (2018). "Multi-target Voice Conversion without Parallel Data by Adversarially Learning Disentangled Audio Representations". In: *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*. Ed. by B. Yegnanarayana. ISCA, pp. 501–505. DOI: 10.21437/Interspeech.2018-1830. URL: https://doi.org/10.21437/Interspeech.2018-1830.

Comminiello, Danilo, Marco Lella, Simone Scardapane, and Aurelio Uncini (2019a). "Quaternion convolutional neural networks for detection and localization of 3D sound events". In: *IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*. IEEE, pp. 8533–8537.

Comminiello, Danilo, Michele Scarpiniti, Raffaele Parisi, and Aurelio Uncini (2019b). "Frequency-domain adaptive filtering: From real to hypercomplex signal processing". In: *IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*. IEEE, pp. 7745–7749.

Cowie, Roddy, Ellen Douglas-Cowie, Nicolas Tsapatsoulis, George Votsis, Stefanos Kollias, Winfried Fellenz, and John G Taylor (2001). "Emotion recognition in human-computer interaction". In: *IEEE Signal Process. Mag.* 18.1, pp. 32–80.

Dalal, Navneet and Bill Triggs (2005). "Histograms of oriented gradients for human detection". In.

Dieleman, Sander, Philemon Brakel, and Benjamin Schrauwen (2011). "Audio-based Music Classification with a Pretrained Convolutional Network". In: *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA, October 24-28, 2011*. Ed. by Anssi Klapuri and Colby Leider. University of Miami, pp. 669–674. URL: http://ismir2011.ismir.net/papers/PS6-3.pdf.

Doersch, Carl, Abhinav Gupta, and Alexei A. Efros (2015). "Unsupervised Visual Representation Learning by Context Prediction". In: *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, pp. 1422–1430. DOI: 10.1109/ICCV.2015.167. URL: https://doi.org/10.1109/ICCV.2015.167.

Dosovitskiy, Alexey and Thomas Brox (2016). "Generating Images with Perceptual Similarity Metrics based on Deep Networks". In: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. Ed. by

Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, pp. 658–666. URL: http://papers.nips.cc/paper/6158-generating-images-with-perceptual-similarity-metrics-based-on-deep-networks.

Drossos, Konstantinos, Paul Magron, and Tuomas Virtanen (2019). "Unsupervised Adversarial Domain Adaptation Based on The Wasserstein Distance For Acoustic Scene Classification". In: *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA 2019, New Paltz, NY, USA, October 20-23, 2019*. IEEE, pp. 259–263. DOI: 10.1109/WASPAA.2019.8937231. URL: https://doi.org/10.1109/WASPAA.2019.8937231.

Dumoulin, Vincent, Jonathon Shlens, and Manjunath Kudlur (2017). "A Learned Representation For Artistic Style". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=BJO-BuT1g.

Dupuis, Kate and M Kathleen Pichora-Fuller (2010). *Toronto Emotional Speech Set (TESS)*. University of Toronto, Psychology Department.

— (2011). "Recognition of emotional speech for younger and older talkers: Behavioural findings from the Toronto emotional speech set". In: *Canadian Acoust.* 39.3, pp. 182–183.

Ekman, Paul and Harriet Oster (1979). "Facial expressions of emotion". In: *Annual review of psychology* 30.1, pp. 527–554.

El Ayadi, Moataz, Mohamed S Kamel, and Fakhri Karray (2011). "Survey on speech emotion recognition: Features, classification schemes, and databases". In: *Pattern Recogn.* 44.3, pp. 572–587.

Engel, Jesse H., Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan (2017). "Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders". In: *Proceedings*

of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 1068–1077. URL: http://proceedings.mlr.press/v70/engel17a.html.

Etienne, Caroline, Guillaume Fidanza, Andrei Petrovskii, Laurence Devillers, and Benoit Schmauch (2018). "Speech emotion recognition with data augmentation and layer-wise learning rate adjustment". In: *arXiv preprint arXiv:1802.05630* 68.

Felzenszwalb, Pedro F, Ross B Girshick, David McAllester, and Deva Ramanan (2009). "Object detection with discriminatively trained part-based models". In: *IEEE PAMI* 32.9, pp. 1627–1645.

Fewzee, Pouria and Fakhri Karray (2012). "Dimensionality reduction for emotional speech recognition". In: *Int. Conf. on Privacy, Security, Risk and Trust and Int. Conf. on Social Comput.* IEEE, pp. 532–537.

Frankle, Jonathan and Michael Carbin (2018). "The lottery ticket hypothesis: Finding sparse, trainable neural networks". In: *arXiv preprint arXiv:1803.03635*.

Fu, Szu-Wei, Yu Tsao, and Xugang Lu (2016). "SNR-Aware Convolutional Neural Network Modeling for Speech Enhancement". In: *Interspeech*, pp. 3768–3772.

Furness, Roger K (1990). "Ambisonics–An overview". In: *AES 8th Int. Conf. The Sound of Audio*. Audio Engineering Society.

Ganin, Yaroslav, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky (2016). "Domain-Adversarial Training of Neural Networks". In: *J. Mach. Learn. Res.* 17, 59:1–59:35. URL: http://jmlr.org/papers/v17/15-239.html.

Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge (2015). "Texture Synthesis Using Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information*

Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada. Ed. by Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, pp. 262–270. URL: http://papers.nips.cc/paper/5633-texture-synthesis-using-convolutional-neural-networks.

— (2016). "Image Style Transfer Using Convolutional Neural Networks". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, pp. 2414–2423. DOI: 10.1109/CVPR.2016.265. URL: https://doi.org/10.1109/CVPR.2016.265.

Gaudet, Chase J and Anthony S Maida (2018). "Deep quaternion networks". In: *IEEE Int. Joint Conf. on Neural Netw. (IJCNN)*. IEEE, pp. 1–8.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio (2011). "Deep sparse rectifier neural networks". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, pp. 315–323.

Grassucci, E., E. Cicero, and D. Comminiello (2022). "Quaternion Generative Adversarial Networks". In: *Generative Adversarial Learning: Architectures and Applications*. Ed. by R. Razavi-Far, A. Ruiz-Garcia, V. Palade, and J. Schmidhuber. Cham: Springer International Publishing, pp. 57–86. ISBN: 978-3-030-91390-8.

Grassucci, Eleonora, Danilo Comminiello, and Aurelio Uncini (2021). "A Quaternion-Valued Variational Autoencoder". In: *IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*. IEEE, pp. 3310–3314.

Grassucci, Eleonora, Gioia Mancini, Christian Brignone, Aurelio Uncini, and Danilo Comminiello (2023). "Dual Quaternion Ambisonics Array for Six-Degree-of-Freedom Acoustic Representation". In: *Pattern Recognition Lett.* 166, pp. 24–30. ISSN: 0167-8655.

Grassucci, Eleonora, Aston Zhang, and Danilo Comminiello (2021). "PHNNs: Lightweight neural networks via parameterized hypercomplex convolutions". In: *arXiv preprint arXiv:2110.04176*.

— (2022). "PHNNs: Lightweight Neural Networks via Parameterized Hypercomplex Convolutions". In: *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13. DOI: 10.1109/TNNLS.2022.3226772.

Greenblatt, Aaron B and Sos S Agaian (2018). "Introducing quaternion multi-valued neural networks with numerical examples". In: *Information Sciences* 423, pp. 326–342.

Guizzo, Eric, Tillman Weyde, and Jack Barnett Leveson (2020). "Multi-Time-Scale Convolution for Emotion Recognition from Speech Audio Signals". In: *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. IEEE, pp. 6489–6493. DOI: 10.1109/ICASSP40776.2020.9053727. URL: https://doi.org/10.1109/ICASSP40776.2020.9053727.

Guizzo, Eric, Tillman Weyde, and Giacomo Tarroni (2021). "Anti-transfer learning for task invariance in convolutional neural networks for speech processing". In: *Neural Netw.* 142, pp. 238–251.

Gutstein, Steven, Olac Fuentes, and Eric Freudenthal (2007). "Knowledge Transfer in Deep Convolutional Neural Nets". In: *Proceedings of the Twentieth International Florida Artificial Intelligence Research Society Conference, May 7-9, 2007, Key West, Florida, USA*. Ed. by David Wilson and Geoff Sutcliffe. AAAI Press, pp. 104–109. URL: http://www.aaai.org/Library/FLAIRS/2007/flairs07-020.php.

Hamel, Philippe, Matthew E. P. Davies, Kazuyoshi Yoshii, and Masataka Goto (2013). "Transfer Learning In MIR: Sharing Learned Latent Representations For Music Audio Classification And Similarity". In: *Proceedings of the 14th International Society for Music Information Retrieval Conference, ISMIR 2013, Curitiba, Brazil, November 4-8, 2013*. Ed. by Alceu de Souza

Britto Jr., Fabien Gouyon, and Simon Dixon, pp. 9–14. URL: http://www.ppgia.pucpr.br/ismir2013/wp-content/uploads/2013/09/76_Paper.pdf.

Han, Kun, Dong Yu, and Ivan Tashev (2014). "Speech emotion recognition using deep neural network and extreme learning machine". In: *INTERSPEECH*, pp. 223–227.

Han, Song, Huizi Mao, and William J Dally (2015). "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding". In: *arXiv preprint arXiv:1510.00149*.

Han, Yunfei, Tonghai Jiang, Yupeng Ma, and Chunxiang Xu (2018). "Pretraining Convolutional Neural Networks for Image-Based Vehicle Classification". In: *Adv. Multim.* 2018, 3138278:1–3138278:10. DOI: 10.1155/2018/3138278. URL: https://doi.org/10.1155/2018/3138278.

Hariharan, Bharath, Pablo Arbeláez, Ross Girshick, and Jitendra Malik (2015). "Hypercolumns for object segmentation and fine-grained localization". In: *CVPR*, pp. 447–456.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition". In: *IEEE Conf. on Comp. Vision and Pattern Recog.* Pp. 770–778.

Hermansky, Hynek (1990). "Perceptual linear predictive (PLP) analysis of speech". In: *the Journal of the Acoustical Society of America* 87.4, pp. 1738–1752.

Ho, Ngoc-Huynh, Hyung-Jeong Yang, Soo-Hyung Kim, and Gueesang Lee (2020). "Multimodal approach of speech emotion recognition using multi-level multi-head fusion attention-based recurrent neural network". In: *IEEE Access* 8, pp. 61672–61686.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Howard, Andrew G, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam (2017). "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861*.

Hozjan, Vladimir and Zdravko Kačič (2003). "Context-independent multilingual emotion recognition from speech signals". In: *Int. J. of Speech Techn.* 6.3, pp. 311–320.

Hsu, Wei-Ning and James R. Glass (2018). "Disentangling by Partitioning: A Representation Learning Framework for Multimodal Sensory Data". In: *CoRR* abs/1805.11264. arXiv: 1805.11264. URL: http://arxiv.org/abs/1805.11264.

Hu, Hao, Ming-Xing Xu, and Wei Wu (2007). "GMM supervector based SVM with spectral features for speech emotion recognition". In: *IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*. Vol. 4. IEEE, pp. IV–413.

Huang, Zhengwei, Ming Dong, Qirong Mao, and Yongzhao Zhan (2014). "Speech emotion recognition using CNN". In: *ACM Int. Conf. on Multimedia*. ACM, pp. 801–804.

Hung, Yun-Ning, Yi-An Chen, and Yi-Hsuan Yang (2018). "Learning Disentangled Representations for Timber and Pitch in Music Audio". In: *CoRR* abs/1811.03271. arXiv: 1811.03271. URL: http://arxiv.org/abs/1811.03271.

Issa, Dias, M Fatih Demirci, and Adnan Yazici (2020). "Speech emotion recognition with deep convolutional neural networks". In: *Biomed. Signal Process. and Control* 59, pp. 1–11.

Jackson, Zohar, Csar Souza, Jason Flaks, and H Nicolas (2018). *Jakobovski/free-spoken-digit-dataset v1. 0.7*.

Jaiswal, Ashish, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon (2020). "A survey on contrastive self-supervised learning". In: *Technologies* 9.1, p. 2.

Jalal, Md Asif, Roger K. Moore, and Thomas Hain (2019). "Spatio-Temporal Context Modelling for Speech Emotion Classification". In: *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019, Singapore, December 14-18, 2019*. IEEE, pp. 853–859. DOI: 10.1109/ASRU46091.2019.9004037. URL: https://doi.org/10.1109/ASRU46091.2019.9004037.

Jansson, Andreas, Eric J. Humphrey, Nicola Montecchio, Rachel M. Bittner, Aparna Kumar, and Tillman Weyde (2017). "Singing Voice Separation with Deep U-Net Convolutional Networks". In: *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*. Ed. by Sally Jo Cunningham, Zhiyao Duan, Xiao Hu, and Douglas Turnbull, pp. 745–751. URL: https://ismir2017.smcnus.org/wp-content/uploads/2017/10/171_Paper.pdf.

Jin, Qin, Chengxin Li, Shizhe Chen, and Huimin Wu (2015). "Speech emotion recognition with acoustic and lexical features". In: *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, pp. 4749–4753.

Johnson, Justin, Alexandre Alahi, and Li Fei-Fei (2016). "Perceptual Losses for Real-Time Style Transfer and Super-Resolution". In: *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Vol. 9906. Lecture Notes in Computer Science. Springer, pp. 694–711. DOI: 10.1007/978-3-319-46475-6_43. URL: https://doi.org/10.1007/978-3-319-46475-6_43.

Jothimani, S and K Premalatha (2022). "MFF-SAug: Multi feature fusion with spectrogram augmentation of speech emotion recognition using convolution neural network". In: *Chaos, Solitons & Fractals* 162, p. 112512.

Kakouros, Sofoklis, Themos Stafylakis, Ladislav Mosner, and Lukas Burget (2022). "Speech-based emotion recognition with self-supervised models

using attentive channel-wise correlations and label smoothing". In: *arXiv preprint arXiv:2211.01756*.

Kanazawa, Angjoo, Abhishek Sharma, and David Jacobs (Dec. 2014). "Locally Scale-Invariant Convolutional Neural Networks". In: *NIPS Deep Learning and Representation Learning Workshop*.

Kegler, Mikolaj, Pierre Beckmann, and Milos Cernak (2019). "Deep speech inpainting of time-frequency masks". In: *CoRR* abs/1910.09058. arXiv: 1910.09058. URL: http://arxiv.org/abs/1910.09058.

Khorram, Soheil, Zakaria Aldeneh, Dimitrios Dimitriadis, Melvin McInnis, and Emily Mower Provost (2017). "Capturing long-term temporal dependencies with convolutional networks for continuous emotion recognition". In: *INTERSPEECH*, pp. 1253–1257.

Kilbertus, Niki, Giambattista Parascandolo, and Bernhard Schölkopf (2018). "Generalization in anti-causal learning". In: *arXiv preprint arXiv:1812.00524*.

Kim, Eun Ho, Kyung Hak Hyun, Soo Hyun Kim, and Yoon Keun Kwak (2009). "Improved emotion recognition with a novel speaker-independent feature". In: *IEEE/ASME transactions on mechatronics* 14.3, pp. 317–325.

Kim, Taewoon and Piek Vossen (2021). "Emoberta: Speaker-aware emotion recognition in conversation with roberta". In: *arXiv preprint arXiv:2108.12009*.

Kim, Yelin, Honglak Lee, and Emily Mower Provost (2013). "Deep learning for robust feature generation in audiovisual emotion recognition". In: *IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*. IEEE, pp. 3687–3691.

King, Davis E (2009). "Dlib-ml: A machine learning toolkit". In: *The Journal of Machine Learning Research* 10, pp. 1755–1758.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Kitza, Markus, Ralf Schlüter, and Hermann Ney (2018). "Comparison of BLSTM-Layer-Specific Affine Transformations for Speaker Adaptation". In: *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*. Ed. by B. Yegnanarayana. ISCA, pp. 877–881. DOI: 10.21437/Interspeech.2018-2022. URL: https://doi.org/10.21437/Interspeech.2018-2022.

Kong, Tao, Anbang Yao, Yurong Chen, and Fuchun Sun (2016). "Hypernet: Towards accurate region proposal generation and joint object detection". In: *CVPR*, pp. 845–853.

Kotti, Margarita and Fabio Paternò (2012). "Speaker-independent emotion recognition exploiting a psychologically-inspired binary cascade classification schema". In: *International journal of speech technology* 15.2, pp. 131–150.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "ImageNet classification with deep convolutional neural networks". In: *Advances in Neural Inform. Process. Syst. (NIPS)* 25, pp. 1097–1105.

Laptev, Aleksandr, Roman Korostik, Aleksey Svischev, Andrei Andrusenko, Ivan Medennikov, and Sergey Rybin (2020). "You do not need more data: Improving end-to-end speech recognition by text-to-speech data augmentation". In: *Int. Congress on Image and Signal Process., BioMed. Engin. and Informatics (CISP-BMEI)*. IEEE, pp. 439–444.

Latif, Siddique, Rajib Rana, and Junaid Qadir (2018). "Adversarial machine learning and speech emotion recognition: Utilizing generative adversarial networks for robustness". In: *arXiv preprint arXiv:1811.11402*.

Le-Khac, Phuc H, Graham Healy, and Alan F Smeaton (2020). "Contrastive representation learning: A framework and review". In: *Ieee Access* 8, pp. 193907–193934.

Lee, Jinkyu and Ivan Tashev (2015). "High-level feature representation using recurrent neural network for speech emotion recognition". In: *INTERSPEECH*, pp. 1537–1540.

Lee, Jongpil, Nicholas J. Bryan, Justin Salamon, Zeyu Jin, and Juhan Nam (2020). "Disentangled Multidimensional Metric Learning for Music Similarity". In: *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. IEEE, pp. 6–10. DOI: 10.1109/ICASSP40776.2020.9053442. URL: https://doi.org/10.1109/ICASSP40776.2020.9053442.

Lei, Jianjun, Yanan Shi, Zhaoqing Pan, Dong Liu, Dengchao Jin, Ying Chen, and Nam Ling (2021). "Deep multi-domain prediction for 3D video coding". In: *IEEE Transactions on Broadcasting* 67.4, pp. 813–823.

Li, Haoqi, Ming Tu, Jing Huang, Shrikanth Narayanan, and Panayiotis Georgiou (2020a). "Speaker-invariant affective representation learning via adversarial training". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 7144–7148.

Li, Jingye, Donghong Ji, Fei Li, Meishan Zhang, and Yijiang Liu (2020b). "Hitrans: A transformer-based context-and speaker-sensitive model for emotion detection in conversations". In: *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 4190–4200.

Li, Mao, Bo Yang, Joshua Levy, Andreas Stolcke, Viktor Rozgic, Spyros Matsoukas, Constantinos Papayiannis, Daniel Bone, and Chao Wang (2021). "Contrastive unsupervised learning for speech emotion recognition". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 6329–6333.

Li, Yingzhen and Stephan Mandt (2018). "Disentangled Sequential Autoencoder". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine

Learning Research. PMLR, pp. 5656–5665. URL: http://proceedings.mlr.press/v80/yingzhen18a.html.

Lian, Zheng, Jianhua Tao, Bin Liu, and Jian Huang (2019). "Unsupervised representation learning with future observation prediction for speech emotion recognition". In: *INTERSPEECH*, pp. 3840–3844.

Lim, Wootaek, Daeyoung Jang, and Taejin Lee (2016). "Speech emotion recognition using convolutional and recurrent neural networks". In: *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, pp. 1–4.

Liu, Wei, Andrew Rabinovich, and Alexander C Berg (2015). "Parsenet: Looking wider to see better". In: *arXiv preprint arXiv:1506.04579*.

Liu, Weiyang, Rongmei Lin, Zhen Liu, Lixin Liu, Zhiding Yu, Bo Dai, and Le Song (2018). "Learning towards Minimum Hyperspherical Energy". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*. Ed. by Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, pp. 6225–6236. URL: http://papers.nips.cc/paper/7860-learning-towards-minimum-hyperspherical-energy.

Liu, Zhen-Tao, Peng Xiao, Dan-Yun Li, and Man Hao (2019a). "Speaker-independent speech emotion recognition based on CNN-BLSTM and multiple SVMs". In: *Intelligent Robotics and Applications: 12th International Conference, ICIRA 2019, Shenyang, China, August 8–11, 2019, Proceedings, Part III 12*. Springer, pp. 481–491.

Liu, Zhentao, Peng Xiao, Dan-Yun Li, and Man Hao (2019b). "Speaker-Independent Speech Emotion Recognition Based on CNN-BLSTM and Multiple SVMs". In: *Intelligent Robotics and Applications - 12th International Conference, ICIRA 2019, Shenyang, China, August 8-11, 2019, Proceedings, Part III*. Ed. by Haibin Yu, Jinguo Liu, Lianqing Liu, Zhaojie Ju, Yuwang Liu, and Dalin Zhou.

Vol. 11742. Lecture Notes in Computer Science. Springer, pp. 481–491. DOI: 10.1007/978-3-030-27535-8_43. URL: https://doi.org/10.1007/978-3-030-27535-8_43.

Livingstone, Steven R and Frank A Russo (2018a). "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English". In: *PloS One* 13.5.

— (2018b). "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English". In: *PloS one* 13.5, e0196391.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). "Fully convolutional networks for semantic segmentation". In: *CVPR*, pp. 3431–3440.

Lu, Cheng, Yuan Zong, Wenming Zheng, Yang Li, Chuangao Tang, and Björn W Schuller (2022). "Domain invariant feature learning for speaker-independent speech emotion recognition". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30, pp. 2217–2230.

Macary, Manon, Marie Tahon, Yannick Estève, and Anthony Rousseau (2021). "On the use of Self-supervised Pre-trained Acoustic and Linguistic Features for Continuous Speech Emotion Recognition". In: *IEEE Spoken Language Techn. Workshop (SLT)*. IEEE, pp. 373–380.

Malik, Mishaim, Muhammad Kamran Malik, Khawar Mehmood, and Imran Makhdoom (2021). "Automatic speech recognition: a survey". In: *Multimedia Tools and Applications* 80, pp. 9411–9457.

Mandic, Danilo P, Cyrus Jahanchahi, and C Cheong Took (2010). "A quaternion gradient operator and its applications". In: *IEEE Signal Process. Lett.* 18.1, pp. 47–50.

Mao, Qirong, Ming Dong, Zhengwei Huang, and Yongzhao Zhan (2014). "Learning salient features for speech emotion recognition using convolutional neural networks". In: *IEEE Trans. Multimedia* 16.8, pp. 2203–2213.

Mao, Xia, Lijiang Chen, and Liqin Fu (2009). "Multi-level speech emotion recognition based on HMM and ANN". In: *WRI World Congress on Comput. Science and Informat. Engin.* Vol. 7. IEEE, pp. 225–229.

Marchand, Ugo and Geoffroy Peeters (2016a). "Scale and shift invariant time/frequency representation using auditory statistics: Application to rhythm description". In: *IEEE MLSP*, pp. 1–6.

— (2016b). "Scale and shift invariant time/frequency representation using auditory statistics: Application to rhythm description". In: *26th IEEE International Workshop on Machine Learning for Signal Processing, MLSP 2016, Vietri sul Mare, Salerno, Italy, September 13-16, 2016*. Ed. by Francesco A. N. Palmieri, Aurelio Uncini, Kostas I. Diamantaras, and Jan Larsen. IEEE, pp. 1–6. DOI: 10.1109/MLSP.2016.7738904. URL: https://doi.org/10.1109/MLSP.2016.7738904.

McFee, Brian, Eric J Humphrey, and Juan Pablo Bello (2015). "A software framework for musical data augmentation". In: *ISMIR*, pp. 248–254.

Mehrabian, Albert (1996). "Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament". In: *Current Psychology* 14, pp. 261–292.

Mehrabian, Albert and Morton Wiener (1967). "Decoding of inconsistent communications." In: *Journal of personality and social psychology* 6.1, p. 109.

Meng, Zhong, Jinyu Li, Yong Zhao, and Yifan Gong (2019). "Conditional teacher-student learning". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 6445–6449.

Mihalache, Serban and Dragos Burileanu (2021). "Dimensional models for continuous-to-discrete affect mapping in speech emotion recognition". In: *University Politehnica of Bucharest Scientific Bulletin, Series C* 83.4, pp. 137–148.

Milde, Benjamin and Chris Biemann (2018). "Unspeech: Unsupervised Speech Context Embeddings". In: *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*. Ed. by B. Yegnanarayana. ISCA, pp. 2693–2697. DOI: 10.21437/Interspeech.2018-2194. URL: https://doi.org/10.21437/Interspeech.2018-2194.

Motiian, Saeid, Marco Piccirilli, Donald A. Adjeroh, and Gianfranco Doretto (2017). "Unified Deep Supervised Domain Adaptation and Generalization". In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, pp. 5716–5726. DOI: 10.1109/ICCV.2017.609. URL: https://doi.org/10.1109/ICCV.2017.609.

Muppidi, Aneesh and Martin Radfar (2021). "Speech Emotion Recognition Using Quaternion Convolutional Neural Networks". In: *IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*. IEEE, pp. 6309–6313.

Nagrani, Arsha, Joon Son Chung, Samuel Albanie, and Andrew Zisserman (2020). "Disentangled Speech Embeddings Using Cross-Modal Self-Supervision". In: *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. IEEE, pp. 6829–6833. DOI: 10.1109/ICASSP40776.2020.9054057. URL: https://doi.org/10.1109/ICASSP40776.2020.9054057.

Nassif, Ali Bou, Ismail Shahin, Imtinan Attili, Mohammad Azzeh, and Khaled Shaalan (2019). "Speech recognition using deep neural networks: A systematic review". In: *IEEE access* 7, pp. 19143–19165.

Neiberg, Daniel, Kjell Elenius, and Kornel Laskowski (2006). "Emotion recognition in spontaneous speech using GMMs". In: *Int. Conf. on Spoken Language Process.*

Neumann, Michael and Ngoc Thang Vu (2017). "Attentive Convolutional Neural Network Based Speech Emotion Recognition: A Study on the Impact of Input Features, Signal Length, and Acted Speech". In: *Interspeech*

2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017. Ed. by Francisco Lacerda. ISCA, pp. 1263–1267. URL: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/0917.html.

Nicholson, Joy, Kazuhiko Takahashi, and Ryohei Nakatsu (2000). "Emotion recognition in speech using neural networks". In: *Neural Comput. & Applicat.* 9.4, pp. 290–296.

Nwe, Tin Lay, Say Wei Foo, and Liyanage C De Silva (2003). "Speech emotion recognition using hidden Markov models". In: *Speech Communic.* 41.4, pp. 603–623.

Okasha, Samir (2016). *Philosophy of Science: Very Short Introduction*. Oxford University Press.

Oldfield, James, Yannis Panagakis, and Mihalis A. Nicolaou (2019). "Adversarial Learning of Disentangled and Generalizable Representations for Visual Attributes". In: *CoRR* abs/1904.04772. arXiv: 1904.04772. URL: http://arxiv.org/abs/1904.04772.

Oord, Aäron van den, Sander Dieleman, and Benjamin Schrauwen (2014). "Transfer Learning by Supervised Pre-training for Audio-based Music Classification". In: *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR 2014, Taipei, Taiwan, October 27-31, 2014*. Ed. by Hsin-Min Wang, Yi-Hsuan Yang, and Jin Ha Lee, pp. 29–34. URL: http://www.terasoft.com.tw/conf/ismir2014/proceedings/T007_118_Paper.pdf.

Padi, Sarala, Dinesh Manocha, and Ram D Sriram (2020). "Multi-Window Data Augmentation Approach for Speech Emotion Recognition". In: *arXiv preprint arXiv:2010.09895*.

Palaz, Dimitri, Ronan Collobert, et al. (2015). *Analysis of cnn-based speech recognition system using raw speech as input*. Tech. rep. Idiap.

Panayotov, Vassil, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur (2015). "Librispeech: An ASR corpus based on public domain audio books". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*. IEEE, pp. 5206–5210. DOI: 10.1109/ICASSP.2015.7178964. URL: https://doi.org/10.1109/ICASSP.2015.7178964.

Parcollet, Titouan, Mohamed Morchid, Xavier Bost, Georges Linarès, and Renato De Mori (2019). "Real to H-space autoencoders for theme identification in telephone conversations". In: *IEEE/ACM Trans. Audio, Speech, Language Process.* 28, pp. 198–210.

Parcollet, Titouan, Ying Zhang, Mohamed Morchid, Chiheb Trabelsi, Georges Linarès, Renato De Mori, and Yoshua Bengio (2018). "Quaternion convolutional neural networks for end-to-end automatic speech recognition". In: *INTERSPEECH*, pp. 22–26.

Parisi, German Ignacio, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter (2019). "Continual lifelong learning with neural networks: A review". In: *Neural Networks* 113, pp. 54–71. DOI: 10.1016/j.neunet.2019.01.012. URL: https://doi.org/10.1016/j.neunet.2019.01.012.

Pasini, Marco (2019). "MelGAN-VC: Voice Conversion and Audio Style Transfer on arbitrarily long samples using Spectrograms". In: *CoRR* abs/1910.03713. arXiv: 1910.03713. URL: http://arxiv.org/abs/1910.03713.

Patel, Nivedita, Shireen Patel, and Sapan H Mankad (2021). "Impact of autoencoder based compact representation on emotion detection from audio". In: *J. of Ambient Intelligence and Humanized Comput.*, pp. 1–19.

Pepino, Leonardo, Pablo Riera, and Luciana Ferrer (2021). "Emotion Recognition from Speech Using Wav2vec 2.0 Embeddings". In: *INTERSPEECH*, pp. 3400–3404.

Perez-Lapillo, Joaquin, Oleksandr Galkin, and Tillman Weyde (2020). "Improving Singing Voice Separation with the Wave-U-Net Using Minimum Hyperspherical Energy". In: *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. IEEE, pp. 3272–3276. DOI: 10.1109/ICASSP40776.2020.9053424. URL: https://doi.org/10.1109/ICASSP40776.2020.9053424.

Permanasari, Yurika, Erwin H Harahap, and Erwin Prayoga Ali (2019). "Speech recognition using dynamic time warping (DTW)". In: *Journal of physics: Conference series*. Vol. 1366. 1. IOP Publishing, p. 012091.

Pham, Nhat Truong, Duc Ngoc Minh Dang, and Sy Dzung Nguyen (2021). "Hybrid Data Augmentation and Deep Attention-based Dilated Convolutional-Recurrent Neural Networks for Speech Emotion Recognition". In: *arXiv preprint arXiv:2109.09026*.

Pratt, Lorien Y. (1992). "Discriminability-Based Transfer between Neural Networks". In: *Advances in Neural Information Processing Systems 5, [NIPS Conference, Denver, Colorado, USA, November 30 - December 3, 1992]*. Ed. by Stephen Jose Hanson, Jack D. Cowan, and C. Lee Giles. Morgan Kaufmann, pp. 204–211. URL: http://papers.nips.cc/paper\/641-discriminability-based-transfer-between-neural-networks.

Putri, Rifki Afina and Dessi Puji Lestari (2015). "Music information retrieval using Query-by-humming based on the dynamic time warping". In: *2015 International Conference on Electrical Engineering and Informatics (ICEEI)*. IEEE, pp. 65–70.

Qiu, Xinchi, Titouan Parcollet, Mirco Ravanelli, Nicholas Lane, and Mohamed Morchid (2020a). "Quaternion neural networks for multi-channel distant speech recognition". In: *INTERSPEECH*, pp. 329–333.

Qiu, Xipeng, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang (2020b). "Pre-trained Models for Natural Language Processing:

A Survey". In: *CoRR* abs/2003.08271. arXiv: 2003.08271. URL: https://arxiv.org/abs/2003.08271.

Reddy, Chandan K. A., Ebrahim Beyrami, Jamie Pool, Ross Cutler, Sriram Srinivasan, and Johannes Gehrke (2019). "A Scalable Noisy Speech Dataset and Online Subjective Test Framework". In: *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*. Ed. by Gernot Kubin and Zdravko Kacic. ISCA, pp. 1816–1820. DOI: 10.21437/Interspeech.2019-3087. URL: https://doi.org/10.21437/Interspeech.2019-3087.

Rigoulot, Simon, Eugen Wassiliwizky, and Marc D Pell (2013). "Feeling backwards? How temporal order in speech affects the time course of vocal emotion recognition". In: *Frontiers in Psychology* 4, p. 367.

Romani Picas, Oriol, Hector Parra Rodriguez, Dara Dabiri, and Xavier Serra (June 2017). *Good-sounds dataset*. DOI: 10.5281/zenodo.820937. URL: https://doi.org/10.5281/zenodo.820937.

Rossenbach, Nick, Albert Zeyer, Ralf Schlüter, and Hermann Ney (2020). "Generating synthetic audio data for attention-based speech recognition systems". In: *IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*. IEEE, pp. 7069–7073.

Rybka, Jan and Artur Janicki (2013a). "Comparison of speaker dependent and speaker independent emotion recognition". In: *Int. J. of Applied Math. and Comput. Science* 23.4.

— (2013b). "Comparison of speaker dependent and speaker independent emotion recognition". In: *Int. J. Appl. Math. Comput. Sci.* 23.4, pp. 797–808. DOI: 10.2478/amcs-2013-0060. URL: https://doi.org/10.2478/amcs-2013-0060.

Sahai, Abhimanyu, Romann Weber, and Brian McWilliams (2019). "Spectrogram Feature Losses for Music Source Separation". In: *27th European Signal Processing Conference, EUSIPCO 2019, A Coruña, Spain, September 2-6,*

2019. IEEE, pp. 1–5. DOI: 10.23919/EUSIPCO.2019.8903019. URL: https://doi.org/10.23919/EUSIPCO.2019.8903019.

Sainath, Tara N, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran (2013). "Low-rank matrix factorization for deep neural network training with high-dimensional output targets". In: *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, pp. 6655–6659.

Salamon, Justin and Juan Pablo Bello (2017a). "Deep convolutional neural networks and data augmentation for environmental sound classification". In: *IEEE Signal Processing Letters* 24.3, pp. 279–283.

— (2017b). "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification". In: *IEEE Signal Process. Lett.* 24.3, pp. 279–283. DOI: 10.1109/LSP.2017.2657381. URL: https://doi.org/10.1109/LSP.2017.2657381.

Savchenko, Lyudmila and Andrey V Savchenko (2021). "Speaker-aware training of speech emotion classifier with speaker recognition". In: *Speech and Computer: 23rd International Conference, SPECOM 2021, St. Petersburg, Russia, September 27–30, 2021, Proceedings 23*. Springer, pp. 614–625.

Scherer, Klaus R (2003). "Vocal communication of emotion: A review of research paradigms". In: *Speech communication* 40.1-2, pp. 227–256.

Schlüter, Jan and Thomas Grill (2015). "Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks". In: *ISMIR*, pp. 121–126.

Schuller, Björn, Ronald Müller, Manfred Lang, and Gerhard Rigoll (2005). "Speaker independent emotion recognition by early fusion of acoustic and linguistic features within ensembles". In: *Eurospeech*.

Selvaraju, Ramprasaath R., Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra (2020). "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *Int.*

J. Comput. Vis. 128.2, pp. 336–359. DOI: 10.1007/s11263-019-01228-7. URL: https://doi.org/10.1007/s11263-019-01228-7.

Sermanet, Pierre, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun (2013). "Overfeat: Integrated recognition, localization and detection using convolutional networks". In: *arXiv preprint arXiv:1312.6229*.

Shegokar, Pankaj and Pradip Sircar (2016). "Continuous wavelet transform based speech emotion recognition". In: *IEEE ICSPCS*, pp. 1–8.

Shilandari, Arash, Hossein Marvi, and Hossein Khosravi (2022). "Speech Emotion Recognition using Data Augmentation Method by Cycle-Generative Adversarial Networks". In: *Signal, Image and Video Process.*

Shin, Hoo-Chang, Holger R. Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel J. Mollura, and Ronald M. Summers (2016). "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning". In: *IEEE Trans. Medical Imaging* 35.5, pp. 1285–1298. DOI: 10.1109/TMI.2016.2528162. URL: https://doi.org/10.1109/TMI.2016.2528162.

Simonyan, Karen and Andrew Zisserman (2015). "Very deep convolutional networks for large-scale image recognition". In: *Int. Conf. on Learning Representations (ICLR)*.

Staiano, Jacopo and Marco Guerini (2014). "Depechemood: a lexicon for emotion analysis from crowd-annotated news". In: *arXiv preprint arXiv:1405.1605*.

Studer, Linda, Michele Alberti, Vinaychandran Pondenkandath, Pinar Goktepe, Thomas Kolonko, Andreas Fischer, Marcus Liwicki, and Rolf Ingold (2019). "A Comprehensive Study of ImageNet Pre-Training for Historical Document Image Analysis". In: *2019 International Conference on Document Analysis and Recognition, ICDAR 2019, Sydney, Australia, September 20-25, 2019*. IEEE, pp. 720–725. DOI: 10.1109/ICDAR.2019.00120. URL: https://doi.org/10.1109/ICDAR.2019.00120.

Sun, Ting-Wei (2020). "End-to-end speech emotion recognition with gender information". In: *IEEE Access* 8, pp. 152423–152438.

Sun, Yi, Jian Li, and Xin Xu (2022). "Meta-GF: Training Dynamic-Depth Neural Networks Harmoniously". In: *European Conference on Computer Vision*. Springer, pp. 691–708.

Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi (2017). "Inception-v4, inception-resnet and the impact of residual connections on learning". In: *Thirty-First AAAI Conference on Artificial Intelligence*.

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich (2015). "Going deeper with convolutions". In: *CVPR*, pp. 1–9.

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna (2016). "Rethinking the inception architecture for computer vision". In: *CVPR*, pp. 2818–2826.

Tan, Cheng, Zhangyang Gao, Lirong Wu, Yongjie Xu, Jun Xia, Siyuan Li, and Stan Z Li (2023). "Temporal attention unit: Towards efficient spatiotemporal predictive learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18770–18782.

Tan, Chuanqi, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu (2018a). "A Survey on Deep Transfer Learning". In: *Artificial Neural Networks and Machine Learning - ICANN 2018 - 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III*. Ed. by Vera Kurková, Yannis Manolopoulos, Barbara Hammer, Lazaros S. Iliadis, and Ilias Maglogiannis. Vol. 11141. Lecture Notes in Computer Science. Springer, pp. 270–279. DOI: 10.1007/978-3-030-01424-7_27. URL: https://doi.org/10.1007/978-3-030-01424-7_27.

— (2018b). "A Survey on Deep Transfer Learning". In: *Artificial Neural Networks and Machine Learning - ICANN 2018 - 27th International Conference*

on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III. Ed. by Vera Kurková, Yannis Manolopoulos, Barbara Hammer, Lazaros S. Iliadis, and Ilias Maglogiannis. Vol. 11141. Lecture Notes in Computer Science. Springer, pp. 270–279. DOI: 10.1007/978-3-030-01424-7_27. URL: https://doi.org/10.1007/978-3-030-01424-7_27.

Tay, Yi, Aston Zhang, Luu Anh Tuan, Jinfeng Rao, Shuai Zhang, Shuohang Wang, Jie Fu, and Siu Cheung Hui (2019). "Lightweight and efficient neural natural language processing with quaternion networks". In: *Proc. of the 57th Ann. Meeting of the Assoc. for Computat. Linguistics*, pp. 1494–1503.

Trigeorgis, George, Fabien Ringeval, Raymond Brueckner, Erik Marchi, Mihalis A Nicolaou, Björn Schuller, and Stefanos Zafeiriou (2016). "Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network". In: *IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*. IEEE, pp. 5200–5204.

Tripathi, Samarth and Homayoon Beigi (2018). "Multi-modal emotion recognition on iemocap dataset using deep learning". In: *arXiv preprint arXiv:1804.05788*.

Tzeng, Eric, Judy Hoffman, Kate Saenko, and Trevor Darrell (2017). "Adversarial Discriminative Domain Adaptation (workshop extended abstract)". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=B1Vjl1Stl.

Tzinis, Efthymios, Zhepei Wang, and Paris Smaragdis (2020). "Sudo rm-rf: Efficient networks for universal audio source separation". In: *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, pp. 1–6.

Ujang, Bukhari Che, Clive Cheong Took, and Danilo P Mandic (2011). "Quaternion-valued nonlinear adaptive filtering". In: *IEEE Trans. Neural Netw.* 22.8, pp. 1193–1206.

Ulyanov, Dmitry, Vadim Lebedev, Andrea Vedaldi, and Victor S. Lempitsky (2016). "Texture Networks: Feed-forward Synthesis of Textures and Stylized Images". In: *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*. Ed. by Maria-Florina Balcan and Kilian Q. Weinberger. Vol. 48. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 1349–1357. URL: http://proceedings.mlr.press/v48/ulyanov16.html.

Verma, Devika and Debajyoti Mukhopadhyay (2016). "Age driven automatic speech emotion recognition system". In: *2016 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, pp. 1005–1010.

Ververidis, Dimitrios and Constantine Kotropoulos (2003). "A review of emotional speech databases". In: *Proc. Panhellenic Conference on Informatics (PCI)*. Vol. 2003, pp. 560–574.

— (2008). "Fast and accurate sequential floating forward feature selection with the Bayes classifier applied to speech emotion recognition". In: *Signal Process.* 88.12, pp. 2956–2970.

Wang, Huiyu, Aniruddha Kembhavi, Ali Farhadi, Alan L Yuille, and Mohammad Rastegari (2019a). "ELASTIC: Improving CNNs With Dynamic Scaling Policies". In: *CVPR*, pp. 2258–2267.

Wang, Huiyu, Aniruddha Kembhavi, Ali Farhadi, Alan L. Yuille, and Mohammad Rastegari (2019b). "ELASTIC: Improving CNNs With Dynamic Scaling Policies". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, pp. 2258–2267. DOI: 10.1109/CVPR.2019.00236. URL: http://openaccess.thecvf.com/content_CVPR_2019/html/Wang_ELASTIC_Improving_CNNs_With_Dynamic_Scaling_Policies_CVPR_2019_paper.html.

Wang, Ke, Jiayong Liu, and Jingyang Wang (2019). "Learning Domain-Independent Deep Representations by Mutual Information Minimization". In: *Comput.*

Intell. Neurosci. 2019, 9414539:1–9414539:14. DOI: 10.1155/2019/9414539. URL: https://doi.org/10.1155/2019/9414539.

Wang, Kunxia, Ning An, Bing Nan Li, Yanyong Zhang, and Lian Li (2015). "Speech emotion recognition using Fourier parameters". In: *IEEE Affective Computing* 6.1, pp. 69–75.

Wang, Kunxia, Guoxin Su, Li Liu, and Shu Wang (2020). "Wavelet packet analysis for speaker-independent emotion recognition". In: *Neurocomputing* 398, pp. 257–264.

Wang, Pengwei, Liangchen Wei, Yong Cao, Jinghui Xie, Yuji Cao, and Zaiqing Nie (2019c). "Understanding Semantics from Speech Through Pre-training". In: *CoRR* abs/1909.10924. arXiv: 1909.10924. URL: http://arxiv.org/abs/1909.10924.

Warden, Pete (2018). "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition". In: *CoRR* abs/1804.03209. arXiv: 1804.03209. URL: http://arxiv.org/abs/1804.03209.

Warriner, Amy Beth, Victor Kuperman, and Marc Brysbaert (2013). "Norms of valence, arousal, and dominance for 13,915 English lemmas". In: *Behavior research methods* 45.4, pp. 1191–1207.

Weninger, Felix, John R Hershey, Jonathan Le Roux, and Björn Schuller (2014). "Discriminatively trained recurrent neural networks for single-channel speech separation". In: *2014 IEEE global conference on signal and information processing (GlobalSIP)*. IEEE, pp. 577–581.

Woodward, James (2005). *Making things happen: A theory of causal explanation*. Oxford university press.

Xie, Yiting and David Richmond (2018). "Pre-training on Grayscale ImageNet Improves Medical Image Classification". In: *Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part VI*. Ed. by Laura Leal-Taixé and Stefan Roth. Vol. 11134. Lecture Notes in

Computer Science. Springer, pp. 476–484. DOI: 10.1007/978-3-030-11024-6_37. URL: https://doi.org/10.1007/978-3-030-11024-6_37.

Xu, Mingke, Fan Zhang, Xiaodong Cui, and Wei Zhang (2021). "Speech emotion recognition with multiscale area attention and data augmentation". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 6319–6323.

Yang, Zichao, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy (2016). "Hierarchical attention networks for document classification". In: *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 1480–1489.

Yao, Xin and Yong Liu (2004). "Evolving Neural Network Ensembles by Minimization of Mutual Information". In: *Int. J. Hybrid Intell. Syst.* 1.1, pp. 12–21. URL: http://content.iospress.com/articles/international-journal-of-hybrid-intelligent-systems/his004.

Yosinski, Jason, Jeff Clune, Yoshua Bengio, and Hod Lipson (2014). "How transferable are features in deep neural networks?" In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. Ed. by Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, pp. 3320–3328. URL: http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.

Zbontar, Jure, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny (2021). "Barlow twins: Self-supervised learning via redundancy reduction". In: *International Conference on Machine Learning*. PMLR, pp. 12310–12320.

Zeng, Yuni, Hua Mao, Dezhong Peng, and Zhang Yi (2019). "Spectrogram based multi-task audio classification". In: *Multimedia Tools and Applications* 78.3, pp. 3705–3722.

Zhang, Richard, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang (2018). "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, pp. 586–595. DOI: 10.1109/CVPR.2018.00068. URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html.

Zhao, Mengyuan, Dong Wang, Zhiyong Zhang, and Xuewei Zhang (2015). "Music removal by convolutional denoising autoencoder in speech recognition". In: *APSIPA ASC*, pp. 338–341.

Zhou, Jian, Guoyin Wang, Yong Yang, and Peijun Chen (2006). "Speech emotion recognition based on rough set and SVM". In: *IEEE Int. Conf. on Cognitive Informat.* Vol. 1. IEEE, pp. 53–61.

Zhu, Zhenyao, Jesse H Engel, and Awni Hannun (2016). "Learning multiscale features directly from waveforms". In: *arXiv:1603.09509*.