DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Robotics, Cognition and Intelligence

Transfer Learning for Emotion Recognition on Multimodal Data from Children with Autistic Spectrum Condition (ASC)

Elisabeth Wittmann

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Robotics, Cognition and Intelligence

Transfer Learning for Emotion Recognition on Multimodal Data from Children with Autistic Spectrum Condition (ASC)

Transfer-Lernen für Emotionserkennung mit Multimodalen Daten von Kindern mit Autismus-Spektrum-Störung (ASC)

Author:	Elisabeth Wittmann
Supervisor:	P.D. Georg Groh
Advisor:	M. Sc. Gerhard Hagerer, Dr. Nicholas Cummins
Submission Date:	16.08.2018

I confirm that this master's thesis in robotics, cognition and intelligence is my own work and I have documented all sources and material used.

Munich, 16.08.2018 Elisabeth Wittmann

Acknowledgments

Ich möchte mich bei allen bedanken, die mir während der Arbeit mit Rat und Tat zur Seite standen. Besonders bei Gerry, für die fachliche Unterstützung zu jeder erdenklichen Uhrzeit und die vielen ausgiebigen Skype Gespräche. Natürlich auch bei Georg und Nick die mich immer unterstützt haben. Außerdem bei allen, die mir anderweitig ausgeholfen haben. Ob nun durch Probelesen oder durch ein leckeres Frühstück zum Entspannen. Danke ihr Lieben!

Abstract

Recognizing emotions is a vital part of human communication, whereas children suffering from Autistic Spectrum Condition (ASC) show severe problems in that regard. Thus, recent research aims at supporting ASC children in their psychological development by giving them intelligent toys at hand to playfully learn and reflect on the emotions they express and recognize. This thesis contributes to this research by automatically recognizing emotions in acoustic speech signals from ASC children. Since according data tends to be available in small amounts due to reduced communication abilities among ASC affected personalities, inferring knowledge from related tasks to emotion recognition on ASC children is of crucial importance to the problem. To that end, this paper evaluates a novel and promising transfer learning technique called Progressive Neural Networks on two ASC children data sets: ASC-Inclusion and DE-ENIGMA. This was done for paralinguistic cross–task learning using ASC-Inclusion data. There they proved as a promising alternative to the traditional pre–training/fine–tuning approach. Secondly, Recurrent Progressive Neural Networks are introduced as an recurrent extension of Progressive Neural Networks to be used for time continuous prediction. Those were tested for cross–task learning on valence and arousal on the DE-ENIGMA data set and for cross-corpus learning with both data sets. They slightly improve the results achieved and thus are regarded as promising alternative to multi–task learning.

Contents

Glossary

NN Neural Network

ASC Autistic Spectrum Disorder

TD Typically Developing

eGeMAPS extended Geneva Minimalistic Acoustic Parameter Set

ComParE Computational Paralinguistics Challenge

UAR Unweighted Average Recall

ROC Receiver Operator Curve

AUC Area Under Curve

RELU Rectified Linear Unit

CNN Convolutional Neural Network

LOSO Leave One Speaker Out

CCC Concordance Correlation Coefficient

SVM Support Vector Machine

ILSVRC ImageNet Large Scale Visual Recognition Challenge

LSTM Long Short Term Memory

GRU Gated Recurrent Unit

eGEMAPS extended Geneva Minimalistic Acoustic Parameter Set

GEMAPS Geneva Minimalistic Acoustic Parameter Set

LLD Low Level Descriptor

HNR Harmonic to Noise Ratio

MFCC Mel Frequency Cepstral Coefficients

RNN Recurrent Neural Network

CLDNN Convolutional, Long-Short-Term Memory Deep Neural Network

COMPARE Computational Paralinguistics Challenge

RELU Rectified Linear Unit

R-PNN Recurrent Progressive Neural Network

AVEC Audio/ Visual Emotion Recognition Challenge

PCC Pearson Correlation Coefficient

MTL Multi Task Learning

RMSE Root Mean Squared Error

1. Introduction

In this chapter one can find a basic introduction to this thesis. The aim of this thesis is introduced in section 1.1 and the structure of the thesis explained in section 1.2.

1.1. Aim

This thesis forms a part of the DE-ENIGMA project which will be presented in chapter 2. For one part it aims to provide a very first emotion recognition baseline for the DE-ENIMGA data set. Predicting arousal and valence scores on the acoustic data of the children. This is on one hand performed on acoustic functionals for only one score for the full utterance, on the other hand time continuous predictions are computed for the utterances. The final scores are presented and evaluated in chapter 7. As a second part of this thesis the performance of Progressive Neural Networks on emotion recognition and on other para-linguistic tasks is investigated. Progressive neural networks are evaluated for cross–task learning and compared to other transfer learning techniques in chapter 5. Recurrent Progressive Neural Networks are introduced which are an extension of the Progressive Neural Networks to become applicable for time continuous predictions as well. Those are tested and evaluated for cross–corpus learning in chapters 6 and 7.

1.2. Structure

Following this introduction, the experiments and research of this thesis are presented and evaluated. First in chapter 2 additional basic information on the DE-ENIGMA project is given. The machine learning concepts used are described in chapter 3 in more detail. Following this in the chapter 4 different acoustic features are explained. Additionally, the two feature sets used during this thesis extended Geneva Minimalistic Acoustic Parameter Set (eGEMAPS) and Computational Paralinguistics Challenge (COMPARE) are described in more detail and feature extraction is explained. In chapter 5 preliminary experiments on the ASC-Inclusion data set are presented. Those compare simple Neural Networks (NNs) with the common pre–training/fine–tuning and the progressive neural network approach for cross–task learning. They are

evaluated using cross–validation on the tasks of typicality, gender and binary valence and arousal recognition. Additionally, the feature sets eGEMAPS and COMPARE are evaluated. Cross–culture–evaluation is tested on the presented problems as well. In chapter 6 simple NNs, Recurrent Neural Networks (RNNs) and cross–corpus learning with Recurrent Progressive Neural Networks (R-PNNs) are used to perform emotion recognition on the ASC-Inclusion data set. The results are presented and evaluated. The same is done for the DE-ENIGMA data set in chapter 7. Additionally, multi–task–learning and cross–task learning with R-PNNs is considered. Finally, in chapter 8 the results of the previous experiments are summarized and compared to previous work. Additionally, concepts for future work are given. In the attachment additional, extensive results of the experiments are given and the code and pretrained networks are provided on the attached DVD.

2. Related Work

In this chapter related work to this thesis is presented. One can find a description of the DE-ENIGMA project in the context of which this thesis was written in section 2.1. Next the second data set, ASC-Inclusion is described. The approach to include robotics into Autistic Spectrum Disorder (ASC) therapy is described in section 2.3 and a general introduction to emotion recognition is given in section 2.4.

2.1. The DE-ENIGMA project

The *DE-ENIGMA - Playfully Empowering Autistic Children* project is an international, interdisciplinary project. A collaboration of organizations and universities in Serbia, Germany, Romania and Great Britain work on it together. It aims to use robotics and artificial intelligence to help autistic children with their socio-emotional skills.

To achieve this a training program including a social humanoid robot is being developed. A small humanoid robot is used to interact with children in a supervised manner. The robot is called *Zeno* by *RoboKind* and resembles a cartoonist astronaut as can be seen in figure 2.1. The data collection therapy sessions with *Zeno* were subject to a *Wizard of Oz* strategy. The therapist controlled the robot from a hidden panel to engage with the child. *Zeno* was able to express a set of emotions during the session which the child was asked to identify. *Zeno* could react to the children's expressions if the therapist prompted this. Data collection sessions were held in Serbia and Great Britain.

Figure 2.1.: The Robot *Zeno* Used During Therapy Sessions [35]

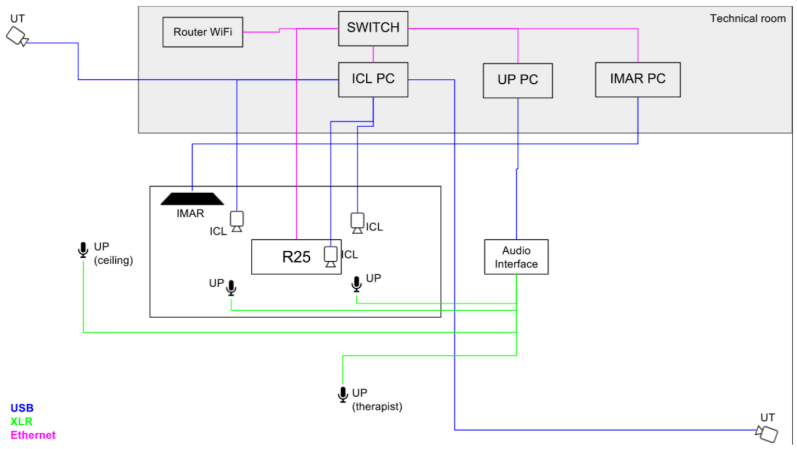


Figure 2.2.: The Data Collection Setup Used During Therapy Sessions [35]

2.1.1. Data Collection

The therapy sessions were recorded to allow for analysis and the training of algorithms later on therefore the setup in figure 2.2 was used. It consists of three webcams. One is placed behind the robot, the other two on each side of the robot. The robot itself has a built in camera. Additional spatial data was gathered using a Kinect device placed behind the table. In addition to the low sound resolution cameras, high definition microphones were placed in the room on both sides of the child (indicated in the figure 2.2 as 'UP'). Additionally, one was mounted on the ceiling. The therapist wore a close talk microphone. [35]

In total 50 children participated in the experiments whose recordings were used as the data set. Of those 11 were female while the remaining 39 were male children. Their age ranged between five and twelve years with an average a little under nine years.

2.1.2. DE-ENIGMA Data set

The DE-ENIGMA data set was used as one data set for emotion recognition. The prior experiments were performed with the ASC-Inclusion data set which will be described later, due to temporal issues with the DE-ENIGMA annotation. The data collection process has already been described in section 2.1.1. The annotation was distributed into different tiers. The first one indicating speech activity, the second one using annotated verbal cues like laughing and crying, the third annotated vocal behavior as echolalia, the

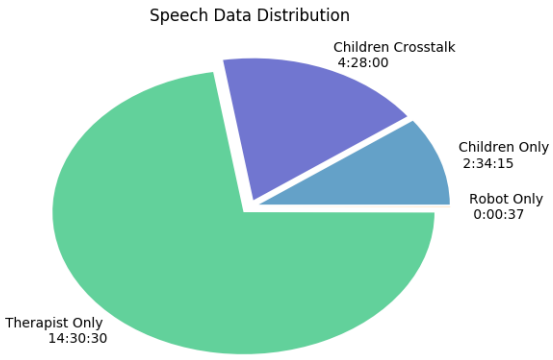


Figure 2.3.: Distribution of Speech Data in the DE-ENIGMA Data set

fourth had traits like gender, age, and language skills, whilst the fifth had continuous annotations for valence and arousal. The available speech data distribution can be found in figure 2.3.

Emotion annotation was performed by native speakers. For annotation purposes they were given a joystick which they were asked to move up and down according to their perception of the current valence/arousal level articulated by the child. There were two sessions, one for the valence annotation and one for arousal. Annotations were made including video information. Each therapy session was annotated by three to five annotators. To combine those annotations a gold standard per session was computed according to the AVEC EWE Gold Standard [42]. Examples from the data set can be found in figure 2.4 and figure 2.5. Up to now there is no baseline for Emotion Recognition on DE-ENIGMA data.

2.2. ASC-Inclusion

The ASC-Inclusion data set served as the second data set for this thesis. It is used for the preliminary experiments with Progressive Neural Networks for para-linguistic tasks. It was chosen because it is a comparatively large data set focusing on emotion recognition on ASC children.

The ASC-Inclusion dataset was created as part of the *ASC-Inclusion Project* [56]. Vocalizing autistic children and a control group of typically developing children were asked to pronounce parts of an emotion-evoking story. This lead to a categoric emotion

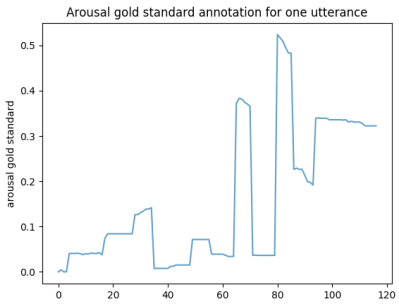


Figure 2.4.: Arousal Gold Standard Example

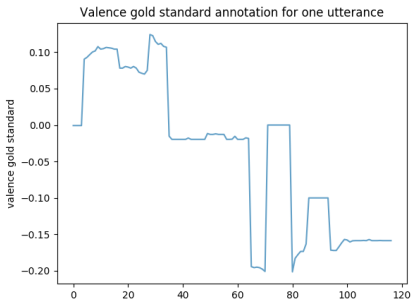


Figure 2.5.: Valence Gold Standard Example

annotation. An example for one of the stories can be found in [10]. Those utterances were recorded and later exploited for different classification tasks. For example, on typicality, categorical emotion recognition, and binary valence and arousal scores [10], [11]. The data set has constantly been evolving and growing. For this project English, Hebrew and Swedish data has been used. The distribution of utterances per language can be found in the following tables 2.1, 2.2 and 2.2. Information about the language, typicality, age and gender was annotated as well. Additional English utterances from extended emotional categories were neglected.

Language	Total # of utterances	# male utterances	# ASC utterances
English	636	357	352
Hebrew	529	308	308
Swedish	729	522	332

Table 2.1.: Numbers of utterances per language and category

Language	# of speakers	# male speaker	# ASC speaker
English	18	10	8
Hebrew	17	11	7
Swedish	20	16	9

Table 2.2.: Numbers of speakers per language and category

Emotion	English	Hebrew	Swedish
Happy	88	79	99
Sad	71	59	97
Angry	70	58	80
Surprised	71	59	97
Afraid	71	56	80
Proud	89	68	98
Ashamed	71	54	58
Calm	52	40	60
Neutral	53	56	78

Table 2.3.: Numbers of utterances per language and emotion

Many classification models for different tasks were already developed and published on those datasets. One example for emotion recognition on the Hebrew part of the ASC-Inclusion dataset can be found in [10]. Those results will be compared to the ones achieved during this work in the final chapter 8.

2.3. Robotics and Autism

ASC manifests itself in many different ways and affects more than five million people in the European Union. It has two main indicators, one being the social interaction capabilities of those affected, the second being rigid patterns in thinking and behaivour. Both of those must show in a patient [2]. People affected by ASC vary on a grand scale, from people with learning disorders to patients with normal to high intellectual abilities. Some never achieve basic language skills whilst others just have unexpected ways of language use. Those implications and their effects are described in [19]. The underlying reason for the features of ASC are still under discussion. Theories such as the *theory of mind hypothesis* [5] and the *weak central coherence theory* [16] try to explain them in a structured manner. Another quite recent approach suggests that patients suffer from attenuated Bayesian priors. This implicates that they do not perceive the world through a filter of their prior believes. They systematically undervalue their prior knowledge about the world and therefore perceive the world differently [40].

To enhance their socio-emotional skills explicit training sessions are widely used and accepted. During those sessions patients practice recognizing and displaying emotions with their caregivers or therapists. The DE-ENIGMA project aims to incorporate a robot companion in this kind of therapy. The first training sessions followed an approach outlined in [25].

Robots have already been used in autism therapy before. They ranged in appearance

from simple toy structures like balls to humanoids, able to display complex facial behaviors. Robots were able to elicit social behavior with the children and seem to be a promising tool when teaching of socio-emotional skills. They are applied for different uses within therapy. Those include the use for diagnosis of ASC in early childhood, to enhance self–initiated interactions between the child and the robot or in a triadic manner with another human. Robots have proven helpful in the settings of turn taking activities and imitation games with children. They were also explored as emotion recognition training tools just as in the DE-ENIGMA project or as points of joint attention [41]. This is thought to happen, because robots are more predictable and follow structures in their behavior. This fulfills the special need of these children for rigidity and gives them a feeling of safety. They might, depending on their style and functional capabilities, provide a bridge between inanimate toys and human interaction allowing the children to generalize their experiences during therapy to their daily live. Up to now, most of the trials were mainly qualitative and did not provide quantitative insight [50].

2.4. Emotion recognition

To enhance the robot's capabilities during therapy, participants of the DE-ENIGMA project aim to construct an interaction software for the robot. This shall enable him to react to emotions displayed by the child and display emotions himself. *Zeno* needs a way to understand and interpret the emotions displayed by the children. This data can then be interpreted by a decision module and evoke a reaction of the robot. This would allow therapists to move forward from the *Wizard of Oz* setting that they are currently using.

There exist two main strategies to measure emotion. One is to describe emotions as categorical distinct states [65]. There are six basic states 'Happy, Sad, Neutral, Surprised, Angry and Disgusted' or even more precise states like 'Calm, Quiet, Bored, Frustrated...'. As one can see, this state–based description can be extended to different sets of emotions of different sizes. It is hard to compare data sets which use different sets of emotional states. Additionally, during normal conversations or in a therapy setting the emotional state can't always be rigidly defined. For example, the transition from neutral to slightly happy could be more precisely described along a continuous axis. In this project the widely accepted concept of arousal and valence measures is used [54]. Arousal is a continuous value which specifies the current state of excitedness or activatedness. Valence on the other hand gives a measure of the positivity or pleasantness. Those values can be used to specify emotion on a continuous circular scale [45]. Categorical emotions can be placed on this circular model as can be seen in

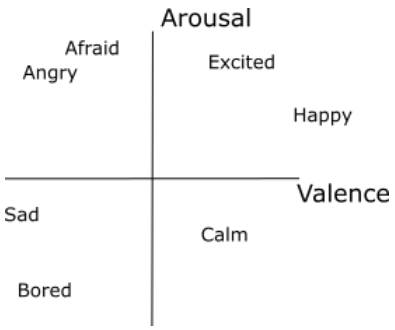


Figure 2.6.: Categorical Emotions on a Two–Dimensional Circular Model. Based on [45]

figure 2.6 but the continuous scale allows for a more fine–grained description of the emotional state and has been found to apply to young children as well [46].

This thesis focuses on emotion recognition using acoustic data. The utterances and sounds made by the child are converted to two numerical values which give an estimate of arousal and valence.

3. Machine Learning Concepts

In the following chapter an overview of the machine learning concepts used during this thesis will be given. Neural networks will be described in detail, as well as the related concepts of transfer learning and recurrent neural networks.

3.1. Neural Networks

Neural Networks (NNs) have been derived from neuroscience. They are a very crude approximation of the far more complicated human neuron. They have an activation function and adaptive parameters called weights on the connections to other neurons [37], [67], [43].

They are based on the concept that any function can be represented as a linear combination of nonlinear parameterized functions. The parameters or in other words the connecting weights are learned from training data via gradient descent and error back–propagation. To enhance the capabilities of those networks more than one hidden layer can be used. The network is then called a deep neural network. It allows for more levels of abstraction from the original data in comparison to just a one–layered network and can achieve the same accuracy or level of abstraction with less neurons and parameters than just a one layered network. In theory any function can be approximated with just a one layered network if it has enough neurons. In this thesis we will follow naming conventions used in [6]. This includes that the superscript $^{(1)}$ indicates that currently layer one of the network is considered. Weights and Biases are described as $w_{ji}^{(1)}$ and $w_{j0}^{(1)}$ respectively. a_j is known as the activation and $h()$ denotes the nonlinear activation function. $z_j()$ is considered the output of the layer j.

$$a_j = \sum_{i=1}^{D}(w_{ji}^{(1)}x_i + w_{j0}^{(1)}) = \sum_{i=0}^{D}(w_{ji}^{(1)}x_i) \tag{3.1a}$$

$$z_j = h(a_j) \tag{3.1b}$$

Standard Equations for the Activation and Output of a Neural Network

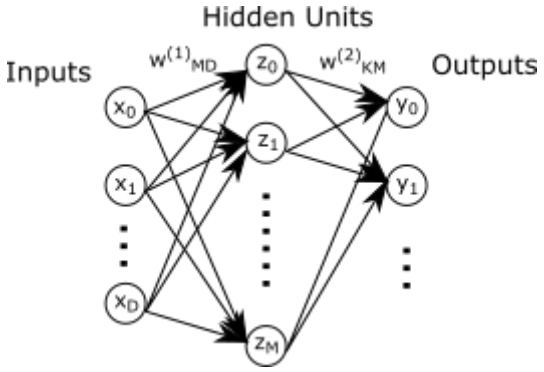


Figure 3.1.: Structure of a Neural Network based on [6]

$$y_k = \sum_{i=0}^{M} (w_{kj}^{(2)} z_j) \tag{3.2}$$

Output of a Neural Network with One Hidden Layer

Following those conventions, a NN with one hidden layer can be shortened and denoted as can be seen in equation 3.2.

The weights are optimized through error back–propagation. Typically, an error function like the sum of squared errors 3.3 is minimized with respect to the weights. Labeled data is needed and used.

A more detailed explanation can be found in [6].

Neural networks played an important role in advancing machine learning and data science over the last decade. Neural Networks and their more specialized offspring's like convolutional neural networks have enabled a rise in artificial intelligence. It is now possible to for example recognize items in pictures with very high accuracy, use neural networks in reinforcement learning to teach a program to play arcade games, or even beat a human in the highly complicated game of Go [60], [38], [59].

$$E(w) = \frac{1}{2} \sum_{n=1}^{N} ||y(x_n, w) - t_n||^2 \tag{3.3}$$

Sum of Squared Error Function to be Minimized

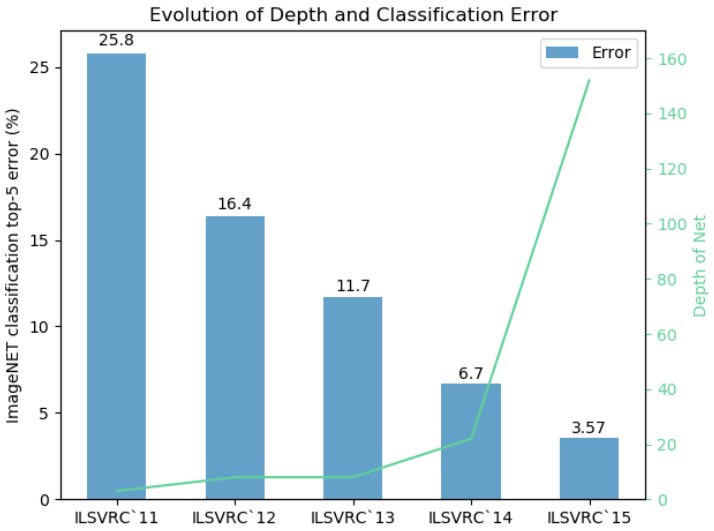


Figure 3.2.: Development of Depth and Classification Error in the ILSVRC adapted from [22]

3.2. Transfer Learning

Due to the tremendous increase in computational power during recent years models with an increasing size and depth became feasible. One example of this trend can be seen in the number of parameters of the winning models in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [44]. Those models recognize more than 1000 different objects in images. This is illustrated in figure 3.2. This amount of computation takes quite some time, even with the currently available hardware architectures.

In addition to the time–consuming training procedure models with many parameters also tend to overfit when used with too little data. This means that the model learns to exactly copy the training data but fails to create an abstraction. It therefore did not learn the real underlying features and fails when applied to unseen data. This can be compared to a student who has to study for a math exam. The student learns the example calculations by heart but does not grasp the concept behind it. They will be perfectly able to reproduce the example calculations. When confronted with a test which obviously will use similar but not the same questions he or she will fail. In machine learning this is called overfitting and can be avoided by using separate training and evaluation data. The model is trained only on the training data and evaluated on the evaluation data. Overfitting takes place, when the model score increases on the

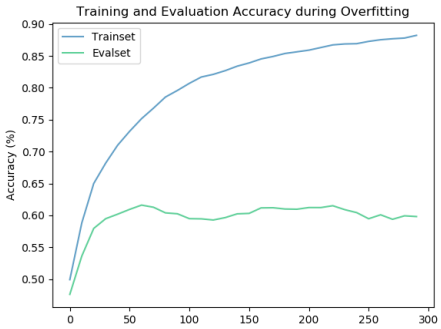


Figure 3.3.: An Example of Training and Evaluation Accuracy During Overfitting

training data set but decreases on the evaluation data as can be seen in figure 3.3. There are many different techniques to prevent overfitting such as early stopping or dropout which were also applied in this thesis.

One of the main challenges in machine learning is the scarcity of data especially in the field of supervised learning where data needs to be annotated. Annotations are costly since they require either human labor or sophisticated algorithms. Data augmentation techniques such as the introduction of noise or cropping and flipping were introduced to cope with this difficulty. Another method to cope with scarce data is the use of transfer learning. This is especially useful in settings with highly specialized data or unusual tasks. This could for example be emotion recognition on pictures from faces. To enhance the performance of models on those data sets additional information can be drawn from similar, larger data sets. One can use bigger normal image data sets, which are available, as a pre–training for specialized models like those described above.

There are many possibilities to transfer knowledge from one task to another. The most widely used approach is called fine–tuning. This signifies that an already trained model from one task, like an image recognition model trained on the whole Google Image net data set, is copied as it is. This means not only copying the architecture, but also the weights learned by the model. The last layer of the model is than changed to the required output format and the model is retrained using the goal tasks data. Here it is possible to back propagate only the last layer or to adapt the weights within the whole network. This is also shown in figure 3.4.

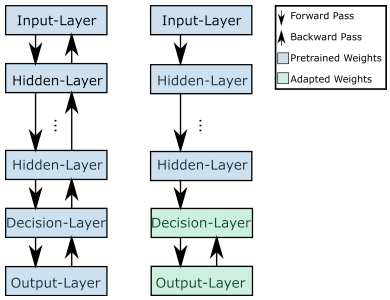


Figure 3.4.: Fine–tuning Approach in Transfer Learning.

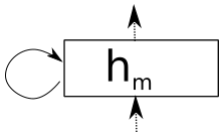


Figure 3.5.: A Recurrent Cell

3.3. Recurrent Neural Networks

An even greater amount of knowledge and information can be gained from treating speech data as time series data since for emotion recognition time continuity plays an important role. This is not only true for human perception of emotions but can as well be exploited in the machine learning concept. Speech data is temporal data. Often the current set of sounds can't provide all the information needed to determine its implication. Just as a single word can have different meanings in a story. To capture long time dependencies that would get lost otherwise, time dependent models are often used with speech data or written data.

Different approaches exist to apply neural networks to temporal data. The most widely known approach are recurrent neural networks. Those exploit temporal dependencies via an internal state which gets updated by the current input. This is illustrated in figure 3.5.

There are different approaches for the actual setup of a recurrent network cell. The most popular architecture is that of Long Short Term Memory (LSTM) cell as described in [24] and [17]. This cell has an input gate, an output gate and a forget gate. The internal status is dependent on the current input and all previous inputs. This architecture is superior to some prior architectures by offering a forget gate. Due to this invention the internal state does not constantly grow bigger over time. This also helps with vanishing or exploding gradients which were quite a big problem when tackling

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{3.4a}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{3.4b}$$

$$\hat{C}_t = tanh(W_c[h_{t-1}, x_t] + b_c) \tag{3.4c}$$

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \tag{3.4d}$$

$$h_t = \sigma(W_o[h_{t-1}, x_t] + b_o) * tanh(C_t) \tag{3.4e}$$

Equations of the LSTM cell [17]

$$f_t = 1 - i_t \tag{3.5}$$

Forget Gate of a GRU

long term dependencies. An overview of the equations and architecture is provided in equation 3.4 and figure 3.6.

LSTM cells are not the only possible architecture. In *LSTM. A search space Odyssey* [20] different cell architectures were investigated. It was found that the Gated Recurrent Unit (GRU) cell seems to be an alternative to the LSTM cell while not having as many parameters therefore speeding up computations. GRUs have been described in [9]. The main difference is that the output of the forget gate is coupled to the input gate as described in equation 3.5.

3.4. Progressive Neural Networks

Another rather recent approach towards transfer learning are so called Progressive Neural Networks. In transfer learning the effect of forgetting the first task is commonly known and an accepted drawback, but with Progressive Neural Networks a new task can be learned while the old task will not be forgotten. In addition, the new task can make use of features learned by the first one. This approach can scale to more than just one task or data set, leveraging knowledge across more tasks or data sets. The architecture of a progressive neural network is illustrated in figure 3.7.

This architecture is achieved by first training a complete normal neural network on the first task or the first data set. This model is then frozen and copied just like it is

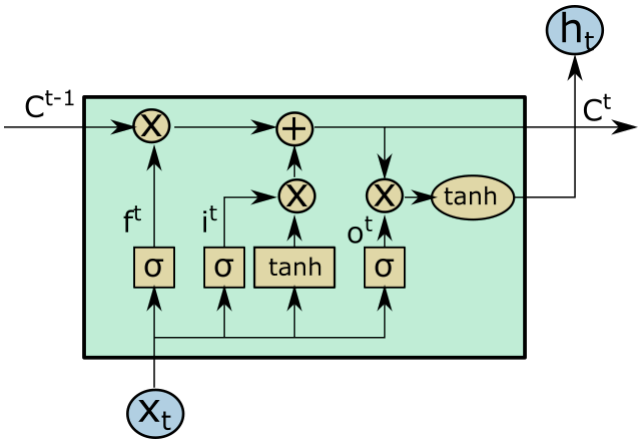


Figure 3.6.: A LSTM Cell based on [17]

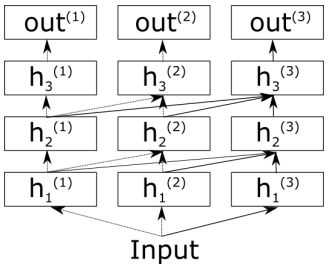


Figure 3.7.: Architecture of a Progressive Neural Net

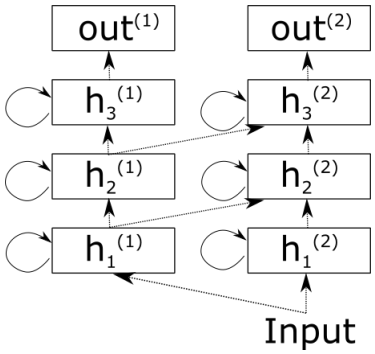


Figure 3.8.: An Overview of a Recurrent Progressive Network

done in fine–tuning. Instead of replacing the last layer, another network is added to it. The old model has frozen parameters and therefore does not change. The new network has normal interconnections between its own layers which are backpropagated and trained normally. Additionally, the outputs of each layer of the previous model are used as inputs for the current model. The new model can then choose to use those different level features by adjusting the corresponding weights thereby it is possible for the network to ignore some features from the previous model but also to learn some completely new features without losing previous ones in the process.

Progressive neural networks have already proven useful in different contexts such as Reinforcement Learning [47] and have been applied successfully to emotion recognition [18].

3.5. Recurrent Progressive Networks

One of the drawbacks of recurrent neural networks is their difficulty to apply them to transfer learning. There are approaches to combine different training datasets called Multi Task Learning (MTL) as for example done in [68] or to use different levels in the processing. This could mean to evaluate if the incoming data is human speech and applying a laughter detection only to the corresponding data [21]. A few architectures use partially shared layers as reported in [34].

One possibility to extend the use of Progressive Neural Networks is to apply them to recurrent architectures. To the authors current knowledge this has not yet been tested before. This extension would then lead to an architecture as can be referenced in figure 3.8.

The freedom of the progressive network architecture to ignore unwanted feature

inputs by setting their weights to a small number could prevent them from unwanted dependencies. Recurrent progressive networks are tested and evaluated on the task of emotion recognition in chapter 6 and 7 of this thesis.

4. Audio Features

First an overview and an anatomical explanation of different audio features will be provided 4.1.2, as well as a discussion for their use in emotion recognition 4.1.3. A differentiation between Low Level Descriptor (LLD)-Features and Feature-Functionals will be made. To describe the selection process of the used features set, different possibilities will be considered. The Computational Paralinguistics Challenge (COMPARE) and extended Geneva Minimalistic Acoustic Parameter Set (eGEMAPS) feature sets will be looked at in more detail. In addition, learned and custom features sets will also be evaluated.

4.1. Audio Features

Since sound and therefore speech is an audible pressure wave and most of the information lies within the frequency domain it is not normally used directly for recognition tasks. Its direct pressure measurements do not convey directly understandable information. Many machine learning concepts are not focused on time-domain data and rely on extracted features that capture the conveyed information. This preprocessing step is designed to discard a lot of information that is considered unnecessary or redundant in the waveform. [12]

There exist few algorithms that perform on the level of the waveform as it is, those rely on convolutions to extract information in an automated and learned manner [66]. Most other algorithms depend on extracted features from this sound wave. There are a multitude of available features and transformations that can be performed on the original sound wave. Those features, their extraction process and their significance for emotion recognition will be explained in this section.

4.1.1. Low Level Descriptors and Functionals

Audio features can be separated into LLD-features and their functionals. LLDs are very closely related to the signal itself. They are computed on a short part of the signal, typically a few milliseconds. They can compute frequency, spectral or energy parameters. Since they are computed on just a short interval of the signal, one can find time dependencies in changes of those descriptors over time. Some descriptors also

have an anatomical correspondence. This will be further explored in the next section. They provide an understandable basis for the information encoded in the original waveform and can be processed easily by time-dependent systems like Recurrent Neural Networks (RNNs). Functionals on the other hand, provide just one set of features for a complete utterance. They do so by combining the extracted LLDs. They for example include the arithmetic mean of one LLD and their coefficient of variation or standard deviation. More statistical parameters like different percentiles or slopes can be included as well. On the other hand functional features that are computed for the whole utterance at once can be included. This includes features like the rate of loudness peaks or mean length of voiced regions. Those functional descriptors result in one single parameter set describing the full utterance. This allows the data to be used on non-time-dependent machine learning algorithms.

4.1.2. Anatomical and Mathematical Feature Interpretation

Human voice is produced in the vocal tract through vibration of the vocal folds. The airflow through the glottis can be seen as the sound source whereas the vocal tract can be considered as a filter for the speech signal. This filter transforms the source waveform of the final speech signal [62]. This has long been subject to a lot of research where scientists try to model the human voice production to find and understand its components and parameters. Some of those models include the three-mass model which describes the vibration of the vocal folds and its tissues and the linear or nonlinear source-filter theory that describes the interaction of the voice production and the vocal tract [29], [63], [64], [15]. From those models arise the first parameters to describe speech which can be used as features. Those include the formants F1-F5, and the fundamental frequency F0, and the second harmonic (2F0). The fundamental frequency and the second harmonic arise directly from the glottal flow and the vibration of the vocal folds whereas the formants depend on vocal tract resonances that transform the air flow.

Other typical parameters like pitch, shimmer, loudness and Mel Frequency Cepstral Coefficientss (MFCCs) take the listener into account as well. They encode the nonlinear human perception. A detailed description of all the parameters used within this thesis and their implementation in the opensmile toolkit can be found in [12].

4.1.3. Relevance for Emotion Recognition

In everyday life we commonly recognize each other's emotions. This human ability is key to our social understanding and greatly helps with communication. Studies show that humans are able to perceive emotions solely from the sound of the voice

with up to sixty percent accuracy [51], [3]. Those experiments were either performed with nonsense utterances or sentences that could have different emotional contexts. This accuracy by itself is not very high. It also varies among the emotional categories where anger can be detected most accurately. On the other hand, untrained subjects can recognize the main acoustic features from voice with comparatively high reliability [4]. In daily life we have a lot of other cues to combine for emotion recognition. This includes facial expression, current context or the content of the utterance. It has been found that acoustic features provide additional information to only facial emotion expression [53]. There exist different theories on the physical effects of emotions and their reasons [51]. They state, that emotion influences our physis and therefore the muscles used for voice production. This in combination with the models of the vocal tract lead to theories of how certain features should change with emotion. Those features included mainly fundamental frequency, energy, pitch and shimmer. Those theories were tested and only partially found to be correct [51], [30]. But, nevertheless a strong correspondence can be found between the acoustic parameters of speech and emotion. Features for emotion recognition were researched for a long time. Some features are well known to have great correspondence for certain emotional states [69], [8]. Those features and correspondences are as well visible in trials performed on the singing voice [52]. Even more importantly those features have as well been found to apply to Autistic Spectrum Disorder (ASC) patients as well [36].

4.2. Overview of available features sets

A lot of feature sets are available. Some of them have gained more reputation or have been used more often in development. Two of those which originated at the INTERPEECH conference will be presented in more detail. The COMPARE feature set, and the eGEMAPS feature set. Additionally, custom or learned feature sets will be described as well.

4.2.1. Custom Feature Sets

Out of all those descriptors, functionals and features described above one can choose and evaluate them at will. To do so effectively extensive research or experience is needed but by this procedure the most accurate feature set for the task at hand can be designed. Features can be extracted using standard procedures like the OpenSmile toolkit [13] or Praat [7]. A disadvantage of this approach is that it is hard to compare. Recognition models trained on different custom feature sets are not comparable. For those two reasons custom feature sets were not taken into consideration for the task at hand.

4.2.2. eGEMAPS Feature Set

The eGEMAPS feature set also has its origins in the INTERSPEECH conference. It is an extension of the Geneva Minimalistic Acoustic Parameter Set (GEMAPS) feature set. Both intend to provide an agreeable minimalistic specialized expert feature set for acoustic emotion recognition task. The LLD-features and functionals were curated carefully by an interdisciplinary team of professionals. This feature set should provide a baseline comparison for different acoustic recognition systems. Those who previously mostly used custom feature sets should now as well evaluate their systems using the standardized feature set to enable comparisons between the different approaches and systems. The selection of features to be used in the GEMAPS or eGEMAPS feature set was based on three main principles. Features needed to show a high potential of being correlated affective psychological changes and in addition they should have already proven useful in prior studies. One must be able to extract them reliably and they should as well have a theoretical foundation [14]. Using those criteria 18 LLD were chosen which can be grouped as follows:

- Frequency Related Parameters
  - Pitch
  - Jitter
  - Formant 1, 2 and 3 Frequency
  - Formant 1 bandwidth
- Energy Related Parameters
  - Shimmer
  - Loudness
  - Harmonic to Noise Ratio (HNR)
- Spectral Parameters
  - Alpha Ratio
  - Hammarberg Index
  - Spectral Slope (0-500 Hz, 500-1500 kHz)
  - Formant 1, 2 and 3 Relative Energy
  - Harmonic difference H1-H2
  - Harmonic difference H1-A3

Those are converted to functionals by taking their arithmetic mean and their coefficient of variation. For loudness and pitch additional functionals are computed and slope information is considered. In addition, six temporal parameters were selected:

- Rate of Loudness Peaks
- Mean Length and Standard Deviation of Voiced Regions
- Mean Length and Standard Deviation of Unvoiced Regions
- Number of Continuously Voiced Regions per Second

This leads to 18 LLD features and 56 corresponding functionals in the GEMAPS feature set. For the eGEMAPS feature set seven additional cepstral and dynamic descriptors were considered.

- MFCC 1-4
- Spectral Flux
- Formant 2-3 bandwidth

This results in 25 LLD features and 88 functional features for the eGEMAPS feature set. Information on the justification, meaning or significance of those parameters can either be found in the section above 4.1.2 or the original eGEMAPS paper [14]. All the features available in the eGEMAPS feature set are also included in the COMPARE feature set. Those two feature sets were compared in the experiments in chapter 5.

4.2.3. COMPARE Feature Set

The COMPARE feature set has been developed and used during the INTERSPEECH challenges of the past years. It is developed as a brute force data set which tries to include all possibly useful features. Therefore, it has 141 LLDs and 6373 functional features. The functional features include energy, spectral, cepstral and voicing related LLD functionals as well as some HNR, psycho acoustic spectral sharpness and spectral harmonicity descriptors. In this work the version of INTERSPEECH 2016 [58] has been used which can be reproduced using the OpenSmile toolkit. This feature set combines a lot of features which provide many clues about the emotional state of the speaker [57]. On the other hand, this great number of features might slow down computation. To compare the performance of this data set to the more specialized eGEMAPS data set preliminary trials were conducted.

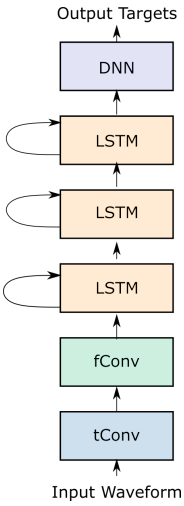


Figure 4.1.: The architecture of the CLDNN. tConv stands for Time-Convolution and fConv for Frequency Convolution. Adapted from [49]

4.2.4. Learned Feature Sets

Another approach to feature sets is to learn them during training. This process can be divided into different approaches. Firstly, learning low level features directly from the waveform. And secondly learning high level features from LLDs like for example concatenated MFCCs.

The first one is more recent. Thanks to the development of convolutional neural networks and improvements in computational power feature extraction directly from the waveform is now a possibility. Convolutional layers basically represent learned filters. They are applied as time convolutions with a certain shift across the time axis. Those time convolutions learn the best fitted low level filter banks for a certain application. On top of those learned convolutional filters, one can apply any other high–level filter approach like Neural Networks (NNs). One example of this is the Convolutional, Long-Short-Term Memory Deep Neural Network (CLDNN) [49]. Here convolutions are used to create frame level features which are passed on to a RNN as can be seen in figure 4.1. Something similar could be imagined using the Wavenet approach [66].

The other approach is more well–known and adopted. It consists of creating more meaningful higher–level features from concatenated MFCCs. This finds application in feature extraction or voice modeling approaches as well as for example performed in [23], [32], [33] and [1].

For the task at hand learning of features was taken into consideration but was discarded. Since this thesis focuses on transfer learning and trained feature sets are by nature specifically adapted to the data set that they were trained on.

5. Simple Progressive Networks for Para-linguistic Cross–Task Learning

In this chapter preliminary experiments using the ASC-Inclusion data set are described. First the ASC-Inclusion data pre–processing steps are described. In section 5.2 the four models used for evaluation are described. This includes a simple Neural Network (NN), two different approaches for pre–training and fine–tuning and finally the progressive NN. In section 5.3 first hyperparameter tuning is described and later one cross–validation and cross–culture evaluation results are presented. Those are discussed in section 5.4. The four models are compared to each other and cross–culture evaluation is evaluated to test if features transfer onto an unseen language or culture subset.

5.1. Data

As the database for this test, the ASC-Inclusion data set as described in section 2.2 was used. It was chosen because it describes emotion recognition on acoustic data from Autistic Spectrum Disorder (ASC) children and has already been evaluated before. This way baselines for comparison are available.

5.1.1. Data Splits

The distribution of utterances over the different classes used in the following experiments per language can be found in table 5.1. In order to create binary arousal and valence scores, the emotion classes were converted following the conventions in the *Emotion in the Speech of Children with Autism Spectrum Conditions: Prosody and Everything Else* [10]. Underrepresented classes were up-sampled in the training data splits.

Hyperparameter tuning for the different architectures was performed using seven-fold cross–validation. The English and Hebrew data set were used for cross–validation. The folds were semi-randomly selected, so that at least two individuals from each culture were present and two of the target class. The folds can be found in tables 5.2 and 5.3. M and F encode for a male or female participant whereas the E or H in the end encodes the corresponding culture.

	English	Hebrew	Swedish
High Arousal	301	302	414
Low Arousal	212	227	315
High Valence	389	320	436
Low Valence	124	209	293

Table 5.1.: Data Distribution Across Languages as the Number of Utterances

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7
F5E	F6E	M4E	M1E	M2E	M6H	M5H
M4H	M7H	M3H	M2H	F7H	M8E	F7E
M1H	F10E	F12E	M17H	M14H	F11H	M13H
M10H	M17E	F13E	F9E	F11E	F16H	F8H
F12H	M9H	M14E	F15H	M15E	M3E	M18E

Table 5.2.: Cross–Validation Folds for Gender Recognition

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7
F5E	F6E	M4E	M1E	M2E	M6H	M5H
M4H	M7H	M3H	M2H	F7H	M8E	F7E
M1H	F10E	F12E	F13E	M14H	M9H	M13H
M17E	F11E	M17H	F9E	M10H	F16H	F8H
F12H	F11H	M14E	F15H	M15E	M3E	M18E

Table 5.3.: Cross–Validation Folds for Typicality, Arousal and Valence Recognition

Additionaly cross–culture evaluation was performed as a measure of evaluation. The training was performed on two languages and the third language served as test set. This very obviously leads to comparativley high errors as has already been seen in *Towards Cross-lingual Automatic Diagnosis of Autism Spectrum Condition in Children's Voices* [28]. Maximilian Schmitt evaluated the potential for cross–culture learning with the **ASC-Inclusion!** (**ASC-Inclusion!**) data in his paper. Sadly they only provide Unweighted Average Recall (UAR) metrics in their evaluation which can vary strongly, depending on the class and does not provide evidence if the system failed to predict one class completely. For example, misclassifying all individuals in the test set to be Typically Developing (TD). Possible causes for the failure to extend features to a different culture are also given in this paper. Nevertheless cross–culture evaluation gives a lot of insight into the generalization of the learned features and was therefore chosen. One has to keep in mind though, that cross–culture evaluation scores are not comparable to intra–culture evaluation scores which are most likely higher.

5.1.2. Data Preparation

Functionals over the full utterance were extracted using the OpenSMILE toolkit [13]. The functionals for the standardized feature sets extended Geneva Minimalistic Acoustic Parameter Set (eGeMAPS) and Computational Paralinguistics Challenge (COMPARE) 2016 were used as described in chapter 4. In order to normalize the feature values, the mean and standard deviation per feature were calculated over the training data set. Features of the training, validation and test set were then normalized to have zero mean and one as standard deviation.

To discourage bias in the training data set is up sampled to contain an equal number of both aim classes. This was performed by adding random copies of the under-represented class. The same was done for the validation set during hyperparameter tuning.

5.2. Models

All models used in this trial were neural networks with two hidden layers of variable sizes. All neurons used Rectified Linear Unit (RELU) activation functions. Dropout as described in [61] was used as one strategy to prevent overfitting. First the main simple models were tuned and evaluated. One for each task and culture combination was saved to serve as a starting point for transfer learning. The specifications and evaluations of all the pretrained models can be found in more detail in the attachment. All models were optimized using the Adam Optimizer and an additional L2 regularizer [31]. The softmax cross–entropy loss as given in equation 5.3 was optimized during

$$p_j = \frac{e^{a_i}}{\sum_{k=1}^{N} e_k^a} \tag{5.1}$$

The Softmax Function

$$H(y, p) = -\sum_i y_i log(p_i) \tag{5.2}$$

The Softmax Cross–Entropy Loss

training. This was chosen since the softmax changes a given vector to a vector in the range of (0,1) where all entries add up to 1. This allows for a probabilistic interpretation of the values.

5.2.1. Simple Models

The simple model consisted of two hidden layers of variable size determined during hyperparameter tuning. All neurons used the RELU activation function given in equation 5.4. A model can be found in figure 5.1 and the corresponding equation for the logit output and the final output is given in equation 5.5. Where x_i are the input features, D is the number of features, M the number of first hidden layer cells and N the number of second hidden layer cells. The logit output a_l has a dimension of two for the binary classification tasks described here.

5.2.2. Progressive Network

The progressive NNs were constructed from the pretrained NNs by extending it with a completely new, additional NN, as described in chapter 3.4. This new NN was enhanced by having the outputs of every layer of the pretrained network as inputs for its own corresponding layers. This structure is represented in figure 3.7. A more detailed description and first evaluation of this structure on emotional classification tasks is given in [18].

The first column of the progressive network looks the same, as the simple NN. The

$$E(w) = -\sum_i y_i log(p_i) + \lambda * \sum_j w_j^2 \tag{5.3}$$

The Softmax Cross–Entropy Loss with L2 regularization

$$h_{RELU}(x) = RELU(x) = max(x, 0) \tag{5.4}$$

The RELU Activation Function

$$a_l(x, w) = \sum_{i=1}^{N}(w_{lk}^{(3)} h_{RELU}(\sum_{i=1}^{M}(w_{kj}^{(2)} h_{RELU}(\sum_{i=1}^{D}(w_{ji}^{(1)} x_i + w_{j0}^{(1)})) + w_{k0}^{(2)})) + w_{l0}^{(3)}) \tag{5.5a}$$

$$y_l(x, w) = argmax(a_l) \tag{5.5b}$$

Equation of the Simple Neural Network Used. First their Logit Computation than their Actual Output.

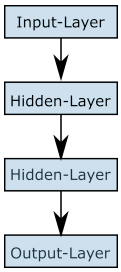


Figure 5.1.: Architecture of the Simple Network.

$$z_j^{(k)} = h(w_j^{(k)} z_{j-1}^{(k)} + \sum_{i<k} U_j^{(k:i)} z_{j-1}^{(i)}) \tag{5.6a}$$

Equation of the Progressive Neural Network used.

$$E(w,u) = -\sum_i y_i log(p_i) + \lambda * (\sum_j w_j^2 + \sum_j u_j^2) \tag{5.7}$$

The Softmax Cross–Entropy Loss for Progressive Neural Networks with L2 Regularization

following columns can be mathematically described as can be found in equation 5.6. The error function to be minimized for classification changes to equation 5.7.

5.2.3. Fine–Tuning

The two most classical approaches were chosen for the fine–tuning approach as already described in chapter 3.2. One of them is to use the weights of the pretrained network to initialize the new network and start training from there. The network is thus able to make use of some already learned features or choose to alter them to its needs. The corresponding equations are the same as for the simple networks as presented in section 5.5.

This approach has been seen to have trouble in bigger networks, not being able to make sense of some more complicated features and therefore not using them during transfer learning. This can be seen in immensely shifting weights all over the network during fine–tuning. Ideally only the last layers or some units are reshaped that much during training while the rest stays quite similar to what they were before.

The second approach tackles this problem by keeping the weights of the higher level layers fixed thus only allowing the network to make changes to the lower layers. It has proven succesfull especially for bigger NNs or Convolutional Neural Network (CNN) architectures and closely related datasets [26], [18]. This approach does not provide as much freedom for the new task to adapt and therefore was not as successful with the small simple networks presented. Here as well the equation is the same as for the simple network 5.5 but only $w_{kl}^{(3)}$ and $wk0^{(3)}$ are adapted during the learning process.

Architectures	(100, 50), (50, 25), (25, 5), (10, 3), (5, 3)
Dropouts	0.5, 0.6, 0.8., 1
Learning rates	0.001, 0.002, 0.004

Table 5.4.: Hyperparamters Tested for the Simple NN

Expanding architectures	(25, 5), (10, 3), (5, 3)
Dropouts	0.5, 0.6, 0.8., 1
Learning rates	0.001, 0.002, 0.004

Table 5.5.: Hyperparamters Tested for the Extending Column of the Progressive Network

5.3. Evaluation

After the process of hyperparameter tuning the models were evaluated using seven-fold cross–validation using the joint English and Hebrew data set. The models were also cross–culture evaluated using the same hyperparameters. The models were trained in two of the three languages and later evaluated using the last language.

5.3.1. Hyperparameter Tuning

For the simple NNs different architectures, learning rates and dropouts were tested as can be found in table 5.4. For the tuning of the eGeMAPS feature set the (100, 50) architecture was discarded since from previous experiments it did not bring more precision than any other architecture. In the end, the smaller architectures always gave better generalization results. Most networks used the (5, 3) or (10, 3) architectures, one of the Computational Paralinguistics Challenge (ComParE) networks used the (25, 5) architecture. The hyperparameters used for every network can be found in the attachment and is included in the specification of the pretrained networks.

The progressive NN were tuned with a slightly different set of possible hyperparameters which can be found in table 5.5. The final hyperparameters can be found in the attachment.

For fine–tuning slightly different parameters were tested which can be found in table 5.6. The basic architecture was obviously the same as for the corresponding simple network. A smaller learning rate was added.

For the process of hyperparameter tuning all combinations of parameters were tested

Dropouts	0.5, 0.6, 0.8., 1
Learning rates	0.0005, 0.001, 0.002, 0.004

Table 5.6.: Hyperparamters tested for fine–tuning

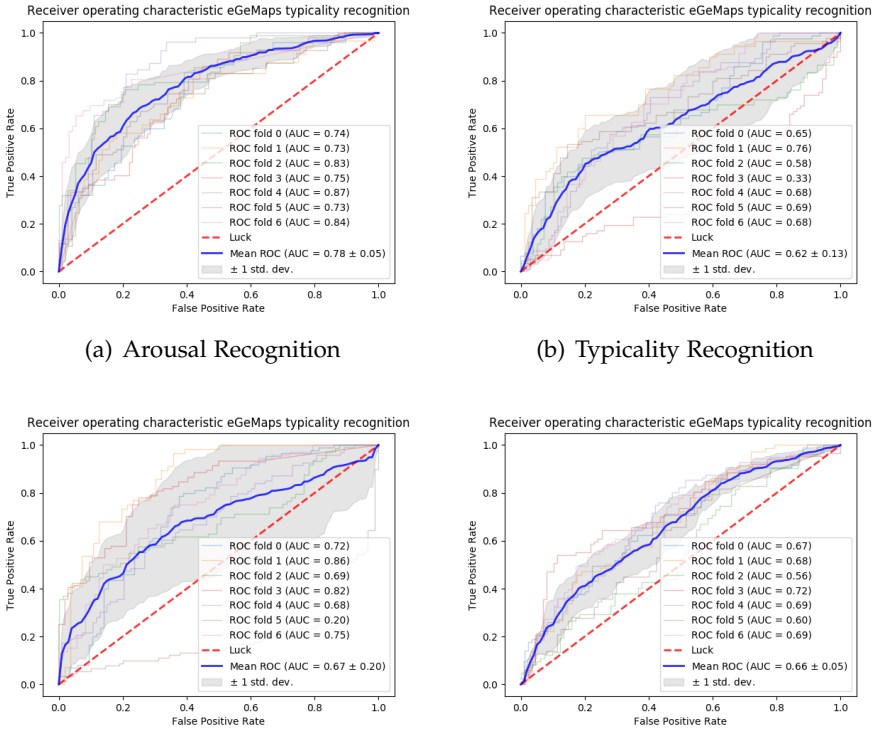
once. This was done with the seven-fold English-Hebrew cross–validation. Validation sets were also up sampled to contain an equal number of samples from both classes. From there a first selection of promising hyperparameters was made. Since from one run no valuable conclusions can be made, five runs were conducted for all promising parameters. The mean accuracy of all those runs was used to select the best set of hyperparameters.

5.3.2. Cross–Validation

To perform cross–validation five runs were performed for each model measuring the mean cross–validation accuracy. Receiver Operator Curves (ROCs) and Area Under Curves (AUCs) scores were also computed for all simple models. Those can be found in figures 5.2 and 5.3. Here it becomes visible, that there is a high variation in accuracy for the different folds for gender and typicality recognition. This happened even though both were performed with different training splits which were balanced for the target classes and cultures. One way to prevent this problem would be a simple Leave One Speaker Out (LOSO) evaluation instead of cross–validation.

The evaluation results in comparison for the different tasks can be seen in figures 5.4 and 5.5. One can see, that in this case fine–tuning sometimes yields worse results than normally trained simple networks. This might be due to local minima in the fitting process. The bad performance of only tuning the last layer can be explained easily. This happens because of the very low number of tunable parameters (either six or ten) in the last layer. With this little freedom the network was not able to learn new feature interpretations or leverage advantages of the old features. This approach generally yields better results for far bigger networks. Progressive neural networks overall provide an average accuracy that is comparable or better than the one of simple networks. On the downside progressive networks have more parameters. In the tested models around 60 more parameters, about half of them frozen. The biggest models contained up to 200 more parameters than the original. Still, this improve in performance was not seen during hyperparameter tuning. There none of the networks performed better with a bigger number of parameters. The confusion matrices of all cross–validations can be found as .npz files on the attached DVD.

Another interesting result is the little difference in the performance of ComParE and



(a) Arousal Recognition (b) Typicality Recognition

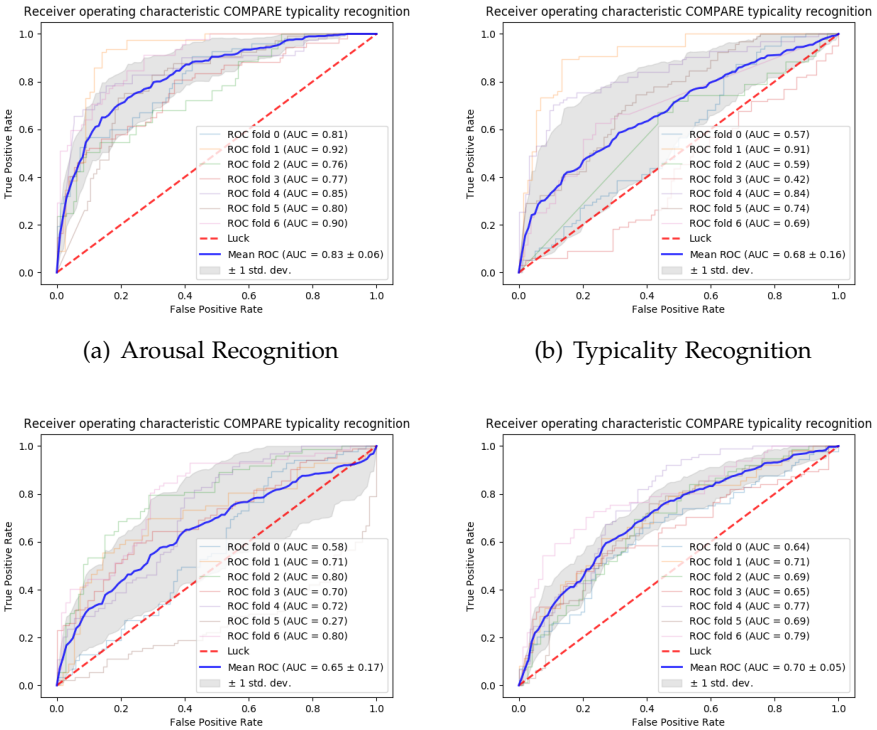
(c) Gender Recognition (d) Valence Recognition

Figure 5.2.: ROC Curves for the Cross–Validation of Simple Networks with eGeMAPS Features

eGeMAPS features. Those vary greatly in their number of features, but performance does not improve greatly while using the brute force feature set. This promotes the usage of the expert eGeMAPS feature set.

5.3.3. Cross–Culture Evaluation

Cross–culture evaluation was performed by training on two language sets and evaluating the model on the third language. Those utterances were up to that moment completely unknown to the model. Since there exist remarkable differences between the cultures performance of the models evidently dropped. It is remarkable that some of the models display a very low F1 score which is due to placing almost all samples in the same class. Schmitt who tested cross–culture learning on typicality detection [28] also didn't reach higher scores. Since he only used UAR one cannot see, if he had a



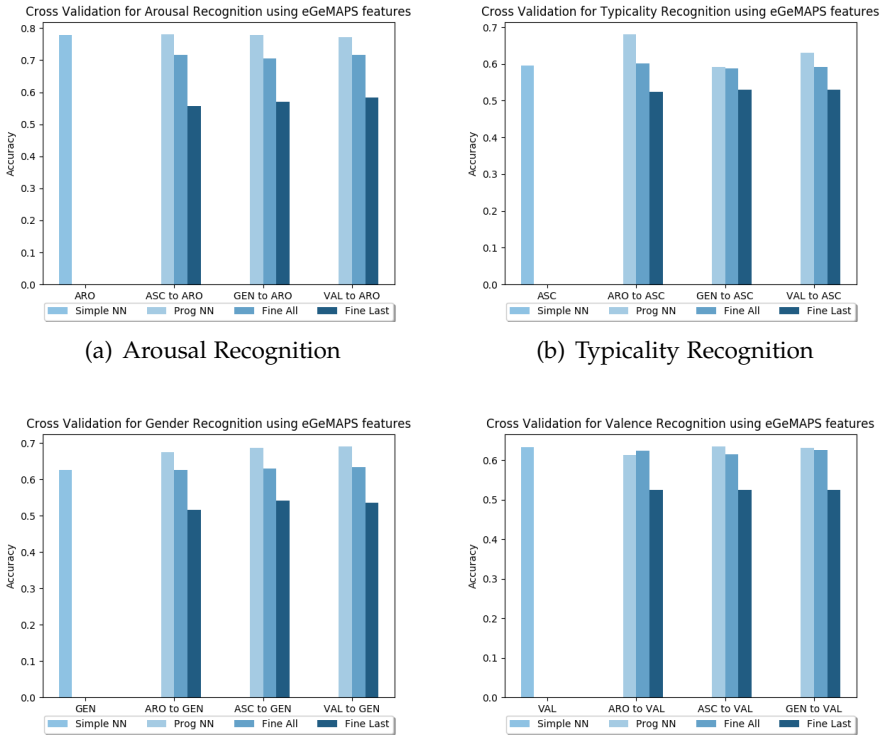
(a) Arousal Recognition

(b) Typicality Recognition

(c) Gender Recognition

(d) Valence Recognition

Figure 5.3.: ROC Curves for the Cross–Validation of Simple Networks with ComParE Features



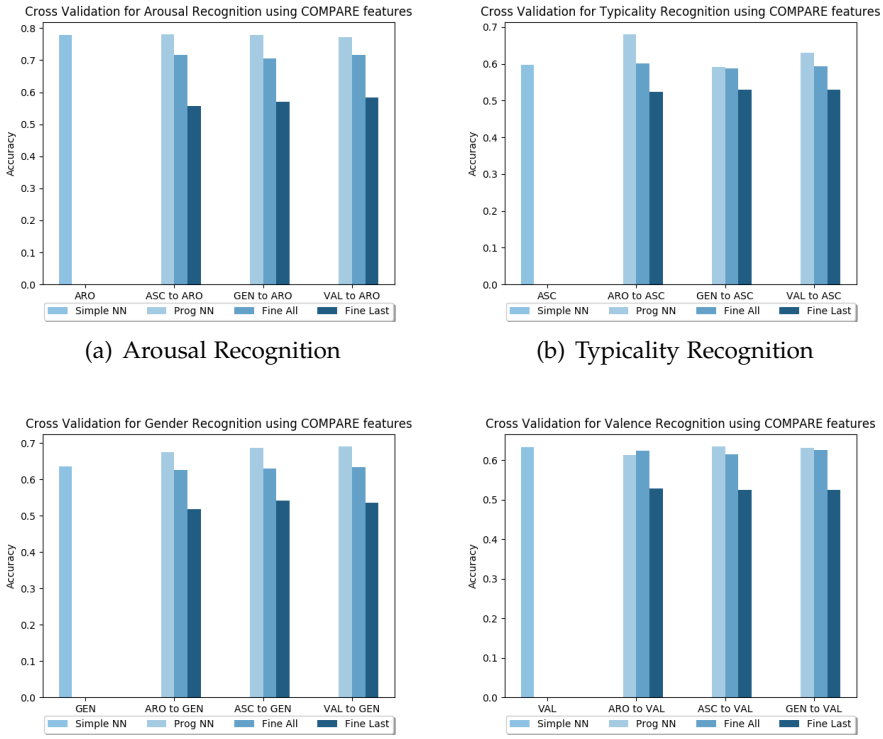
(a) Arousal Recognition

(b) Typicality Recognition

(c) Gender Recognition

(d) Valence Recognition

Figure 5.4.: Cross–Validation Results using eGeMAPS Features



(a) Arousal Recognition (b) Typicality Recognition

(c) Gender Recognition (d) Valence Recognition

Figure 5.5.: Cross–Validation Results using ComParE Features

Training Cultures	English and Hebrew	Hebrew and Swedish	Swedish and English
ASC			
UAR	0.5915	0.4857	0.5932
F1 Score	0.5341	0.3168	0.414
Gender			
UAR	0.5433	0.5389	0.5608
F1 Score	0.4971	0.5629	0.5965
Arousal			
UAR	0.6007	0.5384	0.6371
F1 Score	0.581	0.1624	0.6019
Valence			
UAR	0.5126	0.5307	0.5715
F1 Score	0.3335	0.4441	0.5423

Table 5.7.: UAR and F1 Scores for Simple Networks Training with eGeMAPS Features. Evaluation on the Corresponding New Language.

similar problem. Here, the ComParE feature set outperforms the eGeMAPS feature set on gender recognition. This could be due to more general features which can be found using some features that are not present in the expert feature set. But, this improvement can not be leveraged to the other tasks.

It was hoped that cross–task learning could leverage culture irrelevant features from one task to another. Unfortunately cross–task learning did not improve the performance. This can be seen for example in figure 5.6. Similar results can be reported for other transfer learning techniques and tasks. Those can be found in the attachment.

5.4. Discussion

From the presented results it can be concluded that progressive neural networks does provide an improvement compared to fine–tuning. Those are only opposed by the increasing number of parameters (tunable and frozen). Nevertheless this approach shall be evaluated further and compared to simple learning and fine–tuning of all parameters using more complicated network structures. They will as well be evaluated on cross–corpus learning to see how well they perform on that transfer task.

The brute force ComParE feature set did not improve the results by much except for cross–culture gender recognition. Since this is not the main aim of the project, efforts can now be focused on training with the eGeMAPS feature set. Nevertheless

Training Cultures	English and Hebrew	Hebrew and Swedish	Swedish and English
ASC			
UAR	0.5511	0.5084	0.5692
F1 Score	0.562	0.4845	0.4068
Gender			
UAR	0.5829	0.5129	0.552
F1 Score	0.7387	0.678	0.6825
Arousal			
UAR	0.5498	0.5859	0.5661
F1 Score	0.5073	0.3749	0.4438
Valence			
UAR	0.5191	0.5146	0.5587
F1 Score	0.4014	0.4752	0.4972

Table 5.8.: UAR and F1 Scores for Simple Networks Training with ComParE Features. Evaluation on the Corresponding New Language.

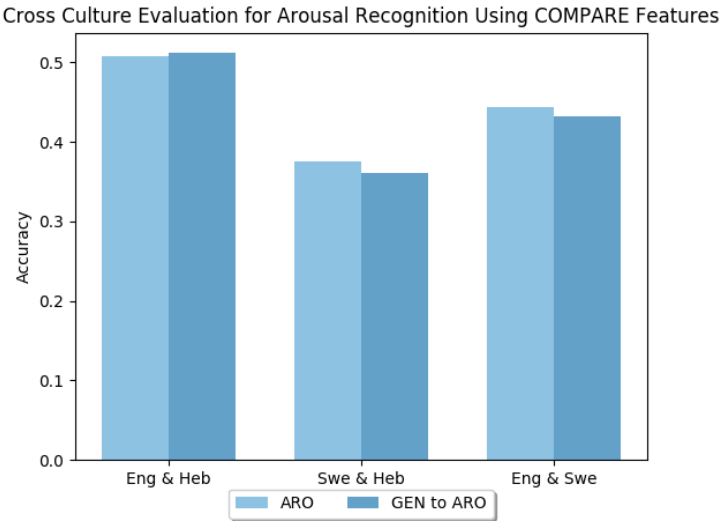


Figure 5.6.: Comparison of Simple Network and Prog Network on Cross–Culture Arousal Recognition

basic comparisons should be repeated for recurrent architectures. This can be done by comparing simple network performances and performing transfer learning on ComParE features only if they seem to provide better results in the simple architectures.

Leveraging culture independent features through transfer learning did not provide better and more stable results. This emphasizes the necessity to include the focus culture group in the training set for emotion recognition and shall be taken into account.

6. Emotion Recognition on ASC-Inclusion Data

In this chapter the implemented emotion recognition on ASC-Inclusion and the corresponding results are described and analyzed. First in section 6.1 emotion recognition using functional features and Neural Networks (NNs) is evaluated. In section 6.2 Recurrent Neural Networks (RNNs) on ASC-Inclusion Low Level Descriptor (LLD) features are described and evaluated. Finally, cross–corpus learning with a Recurrent Progressive Neural Network (R-PNN) is evaluated with a DE-ENIGMA RNN as the initial column. The results are compared in section 6.4.

6.1. Neural Networks and Functional Features

ASC-Inclusion Functionals as already used in chapter 5 were applied in those first experiments. Since Cross–Culture Evaluation has not proven to be helpful all three cultures (Swedish, English and Hebrew) were used in the emotion recognition experiment.

6.1.1. Data Preparation

For the emotion recognition on the ASC-Inclusion Functionals, the same data preparation techniques as in chapter 5 were applied. Five-Fold Cross–validation was applied. The corresponding folds are quasi randomly selected, containing at least three children of each culture and at least three female speakers. They can be found in table 6.1 where F and M encode female and male speakers whereas the letter in the end encodes the corresponding culture. The distribution of emotional utterances by culture is already given in table 2.2 in section 2.2.

6.1.2. Evaluation

For hyperparameter tuning the same parameters as in section 5.3.1 were used and explored in an extensive grid search. Promising parameter sets were then tested five times to provide more reliable measures. In the end the architecture presented in table 6.2 was chosen. This model reached an average accuracy of 0.25619 in five runs.

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
F6E	F5E	M4E	M1E	M2E
M7H	M4H	M3H	M2H	F7H
M1H	F10E	F12E	M17H	M14H
M17E	M10E	F9E	F13E	F11E
F12H	M9H	M14E	F15H	M15E
F8H	F11H	F16H	M5H	F7E
M8E	M3E	M13H	M18E	M6H
F16S	F14S	F12S	F11S	F17S
M1S	M10S	M13S	M15S	M18S
M19S	M2S	M20S	M3S	M4S
M5S	M6S	M7S	M8S	
			M9S	

Table 6.1.: Cross–Validation Folds for Emotion Recognition ASC-Inclusion

Architecture	(100, 50)
Dropout	0.8.
Learning rate	0.001
Number of Epochs	110

Table 6.2.: Hyperparameters for the Simple ASC NN

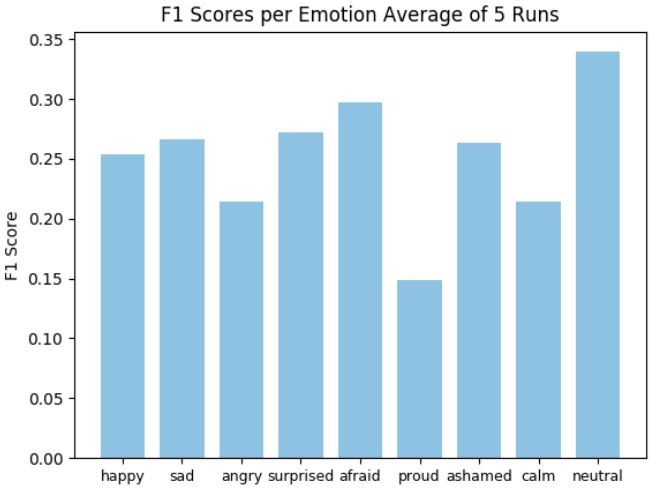


Figure 6.1.: ASC F1 Scores by Emotion.

The average F-1 Scores per emotion can be found in figure 6.1. One can see that some were very nicely recognized whereas some uncommon emotions like 'proud' and 'calm' were not recognized as reliably. A main problem in this task was overfitting due to the small amount of training data. With early stopping applied training accuracies were still around 68% and evaluation accuracies around 25%. The corresponding confusion matrices can be found in the attachment.

Another interesting point of investigation is the error distribution across cultures. For this experiment the Swedish data scores are lower. Example measures out of one run can be seen in figure 6.2. This can also be seen in [11]. The higher Unweighted Average Recall (UAR) scores achieved there are not comparable to the ones presented here, since they were achieved for training on only one culture and for one typicality but the training across cultures shows the same trend with lower scores for the Swedish culture and a notable difference between Typically Developing (TD) and Autistic Spectrum Disorder (ASC) children. This can be seen in figure 6.3. Even the low difference between ASC and TD accuracy in the Swedish data set is comparable to the findings by Erik Marchi.

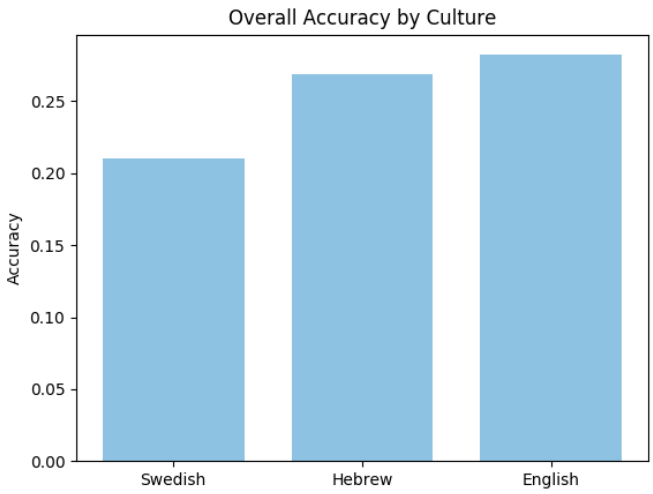


Figure 6.2.: ASC Accuracy by Culture

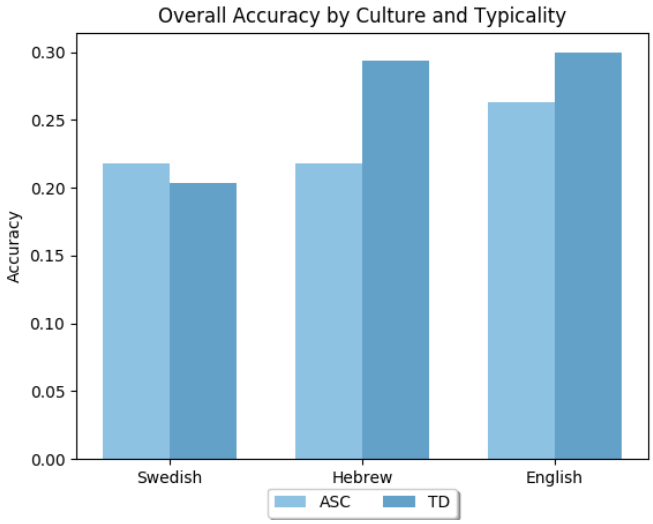


Figure 6.3.: ASC Accuracy by Culture and Typicality

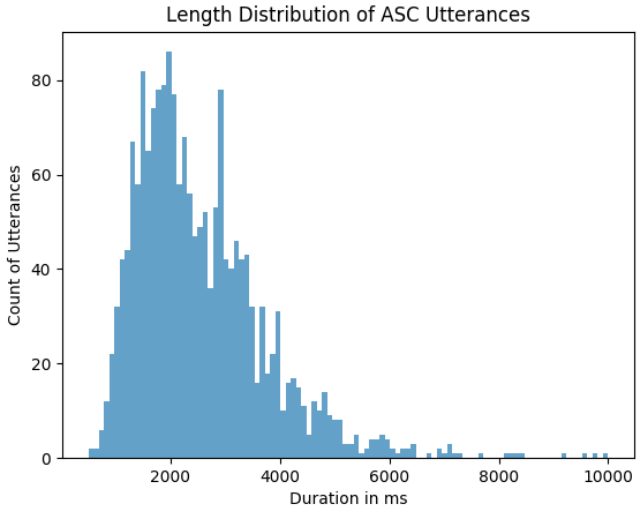


Figure 6.4.: Length Distribution of ASC-Inclusion Utterances, cut off at 10s, Three Utterances not Displayed

6.2. Recurrent Neural Networks

Additionally, a RNN was trained on LLD features of the ASC-Inclusion data. The categorical features for every utterance posed an additional problem in the training process. Different techniques were tested and evaluated to cope with this challenge.

6.2.1. Data Preparation

From the emotional utterances in the ASC-Inclusion data set the LLDs were extracted using the Opensmile toolkit. Those were divided into five data splits as described in table 6.1. These utterances were of varying length as can be seen in figure 6.4. To enhance computational speed and create more data samples utterances longer than 4 seconds were split into smaller ones for training purposes. Before processing the data, underrepresented classes in the training data were upsampled. The features were normalized to zero mean and one standard deviation regarding the training data split.

6.2.2. Model

The computation of the output logits of the recurrent version of the ASC-Inclusion Neural Network follows the equation 6.1. Where the output of $h(a_j^t)$ is computed

$$h(a_j^t) = \sigma(W_o[h(a_j^{t-1}, x_t] + b_o) * tanh(C_t) \tag{6.1a}$$

$$z_j^t = h(a_j^t) \tag{6.1b}$$

$$y_k^t = \sum_{i=0}^{M} (w_{kj}^{(2)} z_j^t) \tag{6.1c}$$

Output of a recurrent NN

following the Long Short Term Memory (LSTM) equations given in section 3.3. There is an output computed for every timestep in the sequence (every 10ms). On the other hand, only one label per utterance is available. To handle this setting different strategies for error and final output computation were explored. Just as for the simple NN the Softmax Cross–Entropy was optimized. Figure 6.5 shows the different input strategies for the Softmax Function.

The *Last Output* technique only collects the last output of the RNN and reduces its softmax cross entropy or takes the Argmax to predict the corresponding emotion. This does not harness the full power of the RNN since it only takes the last output and just neglects the ones before. It also gets more difficult for longer utterances. The *All Outputs* technique takes every 10 ms frame into consideration and back–propagates those errors. This not only leads to a stronger weight on long utterances but also neglects changes in emotional display over the time of the utterance. Small parts of one utterance could include a different emotional style. The *Mean Output* technique accounts for the problem of weighing longer utterances more as it gives every utterance the same weight without considering the length. It is also superior to taking the mean of the *All Outputs* for one utterance. Because the logit output still contains a numerical value in comparison to the binary output of the Argmax. This is comparable to a kind of certainty measure still included which can be considered during averaging. Those three techniques were compared, and the results can be found in figure 6.6. The figure has been created with an architecture of (20, 10), a learning rate of 0.002 and 0.75 dropout probability for one fold. The accuracy difference between the *Last Output* optimization and the other two techniques is very well visible. From those results and some additional experiments, the *Mean Output* strategy was chosen for the final evaluation.

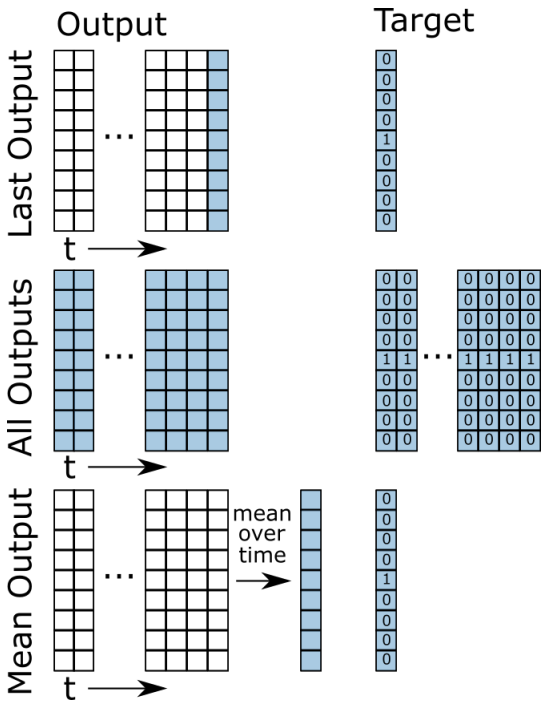
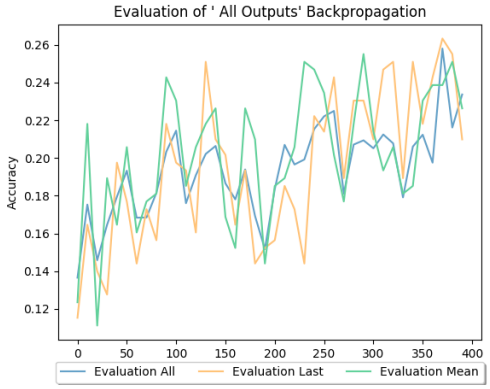
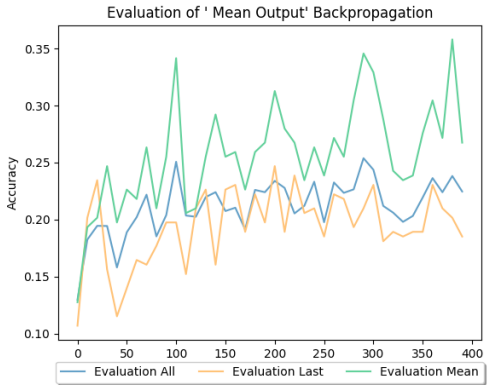


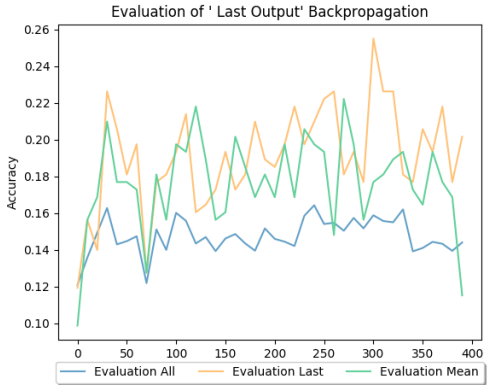
Figure 6.5.: Different Paradigms for the Error Computation



(a) 'All Outputs' Optimization



(b) 'Mean Outputs' Optimization



(c) 'Last Output' Optimization

Figure 6.6.: Evaluation of the Different Error Strategies

Architecture	(30, 20)
Dropout	0.75
Learning rate	0.003
Number of Epochs	600
Batch size	50

Table 6.3.: Hyperparameters for the ASC RNN

6.2.3. Evaluation

Due to constraints on time and computational power no extensive hyperparameter search was performed for the RNN. The chosen parameters can be found in table 6.3. Over five runs an average accuracy of 0.2574 was reached. This is sure to be improved by more careful hyperparameter search. One can find the F1 scores per emotion in figure 6.7. One can see, that the distribution over the emotions is comparable to the one reached with a normal NN.

The accuracy divided by culture and typicality can be found in figures 6.8 and 6.9 for one example run. Here it is interesting to note, that in contrast to the results from the simple NN the RNN is better at classifying ASC samples. For the Hebrew data set it is even better than for TD samples. This hints towards the usefulness of RNNs when working on ASC data. The Swedish data set seems to be the most difficult as has already been seen in the simple NN approach.

6.3. Cross–Corpus Learning with Recurrent Progressive Neural Networks

R-PNNs are tested as strategy of cross–corpus transfer learning using a network trained on the DE-ENIGMA data set as the initial column and training an extensible column on ASC-Inclusion data.

6.3.1. Data Preparation

For the ASC-Inclusion data set the same data preprocessing steps as used in section 6.2 are used. For the training of the initial column extended Geneva Minimalistic Acoustic Parameter Set (eGeMAPS) LLD features were extracted as described in section 7.2. Utterances longer than four seconds were split. Those were normalized to zero mean and one standard deviation.

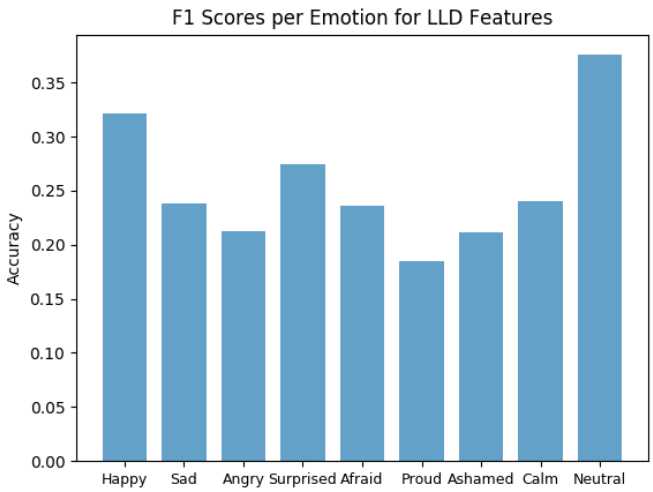


Figure 6.7.: ASC F1 Scores by Emotion for LLD Features.

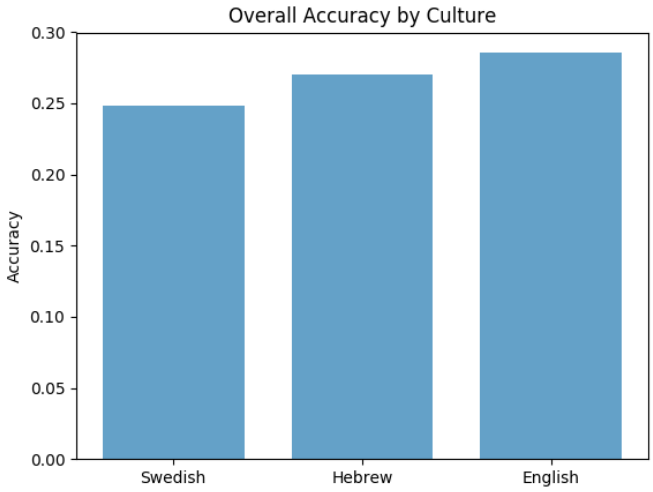


Figure 6.8.: ASC Accuracy by Culture for LLD Features

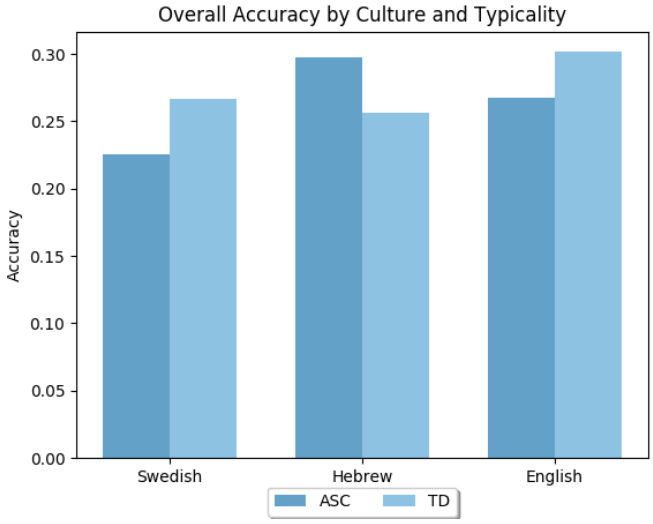


Figure 6.9.: ASC Accuracy by Culture and Typicality for LLD Features

6.3.2. Model

Recurrent Progressive Neural Networks can be seen as an extension of Progressive Neural Networks as described in [48]. An overview can already be found in chapter 3.5 and a visualization in figure 3.8. This architecture can in principal be used with any combination of activation functions and recurrent cells. Here the standard LSTM cell was used which is described in section 3.3. The output equation of the recurrent progressive network is given in equation 6.2. The error function for the classification task is given in equation 6.3.

The first column used the same hyperparameters as the Multi Task Learning (MTL) RNN presented in section 7.2 and was trained on all available DE-ENIGMA data at once.

6.3.3. Evaluation

As for the other RNNs no extensive hyperparameter search was performed. The parameters for the extensive column used can be found in table 6.4.

An overall accuracy score of 20.61 was reached. The F1 Scores per emotion are shown in figure 6.10. Since the DE-ENIGMA data set only contains utterances by children with autism it would be expected that performance improves especially for the autistic children within the ASC data set. To evaluate this the results were also evaluated for

$$h(a_j^t) = \sigma(W_o[h(a_j^{t-1}, x_t] + b_o) * tanh(C_t) \tag{6.2a}$$

$$z_j^t = h(a_j^t + \sum_{i<k} U_j^{(k:i)} z_{j-1}^{(i)}) \tag{6.2b}$$

$$y_k^t = \sum_{i=0}^{M} (w_{kj}^{(2)} z_j^t) \tag{6.2c}$$

Output of a Progressive Recurrent NN

$$E(w, u, w_{LSTM}) = -\sum_i y_i log(p_i) + \lambda * (\sum_j w_j^2 + \sum_j u_j^2) \tag{6.3}$$

Error Function for the Classification R-PNN

	ASC-Inclusion
Architecture	(20, 10)
Dropout	0.8
Learning rate	0.004
Number of Epochs	100

Table 6.4.: Hyperparameters for the Extensive Column of the R-PNN on ASC-Inclusion

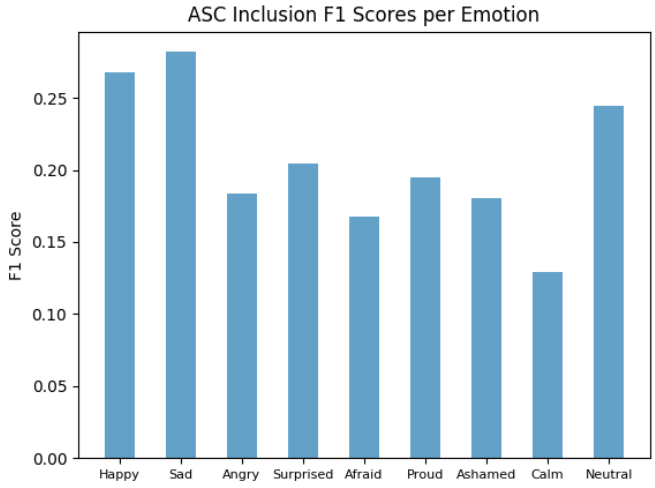


Figure 6.10.: F1 Scores by Emotion for Cross Corpus Learning.

	Simple NN	RNN	R-PNN
Accuracy	0.2562	0.2574	0.2061

Table 6.5.: Cross–Validation Accuracies of the Different Models on ASC-Inclusion

their accuracy per culture and by typicality. This evaluation can be found in figure 6.12 and figure 6.11.

6.4. Discussion

All results are summarized in table 6.5. RNN evaluation did not yet notably improve upon the simple functional results. This could be due to hyperparameter tuning which was not performed due to time constraints. It is nevertheless interesting to see the change in accuracy regarding ASC children from figure 6.3 to 6.9. The scores from Cross–Corpus learning don't reach the scores achieved during simple RNN nor NN training. This could be due to bad hyperparameters since those were not tuned especially. The greatest difference in hyperparameters was the number of steps performed - 100 for cross culture and 600 for normal RNNs. Maybe a more careful tuning process could lead to improvements and real knowledge transfer.

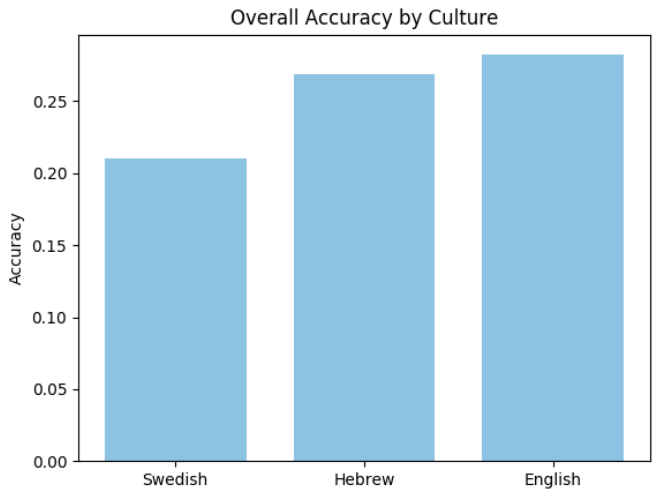


Figure 6.11.: ASC-Inclusion Accuracy by Culture for Cross Corpus Learning.

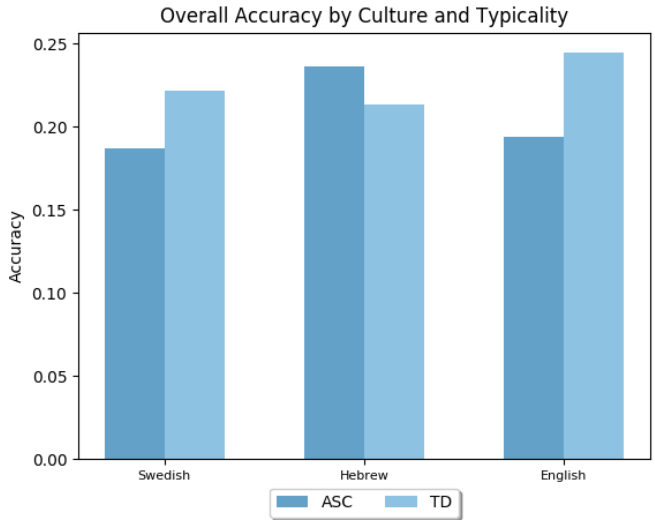


Figure 6.12.: ASC-Inclusion Accuracy by Culture and Typicality.

7. Emotion Recognition on DE-ENIGMA Data

In this chapter the implemented emotion recognition on the DE-ENIGMA data set and the corresponding results are described and analyzed. First in section 7.1 emotion recognition using functional features and NNs is evaluated. In section 7.2 RNNs on LLD features are described and evaluated. This includes a MTL approach training to predict arousal and valence measures jointly with one shared network. In section 7.3 R-PNNs are used for cross–task learning from arousal to valence and vice versa. Finally, cross–corpus learning with a R-PNN is evaluated with the ASC-Inclusion RNN as the initial column. The results are compared in section 7.5.

7.1. Neural Networks and Functional Features

As a first baseline emotion recognition on DE-ENIGMA functionals was performed. In opposition to the ASC-Inclusion categorical annotations the ones of the DE-ENIGMA data set are continuous valence and arousal annotations. Those were averaged to one value for the simple NN application.

7.1.1. Data Preparation

The extended Geneva Minimalistic Acoustic Parameter Set (eGEMAPS) functionals for the DE-ENIGMA data set were created using the OpenSMILE toolkit. The start and end times of the child utterances were extracted from the diarization annotations and fed into the extraction algorithm. The functionals are created for exactly each utterance duration. A mean arousal and valence score for each utterance was provided. The extracted features were normalized to zero mean and one standard deviation regarding the training split. The four data splits can be found in table 7.1. They were pseudo randomly created, ensuring two Serbian children per data split. Culture is indicated by the letter up front. B encodes for British and S for Serbian subjects.

Fold 1	Fold 2	Fold 3	Fold 4
S031	B028	B065	B060
B044	S042	B009	S014
S018	B026	B056	B030
B017	B042	B023	B016
B057	B038	B034	B066
S004	B006	B001	B012
B021	S007	S013	B007
B063	B018	S017	B045
B003	S040		S024

Table 7.1.: Cross–Validation Folds for Emotion Recognition DE-ENIGMA

$$E(w) = \frac{1}{2}\sum_{n=1}^{N} ||y(x_n, w) - t_n||^2 + \lambda * \sum_j w_j^2 \tag{7.1}$$

Mean Squared Error with L2 Regularization

7.1.2. Model

In contrast to the ASC-Inclusion data annotations for DE-ENIGMA are valence and arousal scores. Those range between -1 and 1, indicating low or high arousal and valence. During hyperparameter tuning different neural networks were fitted for arousal scores only, valence scores only and both combined, giving the same weight to both scores. This is basically a MTL approach harnessing the relationship between the valence and arousal scores. The networks were tuned using the mean squared error and L2 regularization. The corresponding equation is given in equation 7.1.

A full grid search was performed for hyperparameter tuning. The parameters can be found in table 7.2. This was done for all three set ups - arousal, valence and both - to investigate the difference and possible improvements from combining both.

Architectures	(100, 50), (50, 25), (25, 5), (10, 3), (5, 3)
Dropouts	0.5, 0.6, 0.8., 1
Learning rates	0.001, 0.002, 0.004, 0.006

Table 7.2.: Hyperparameters tested for the emotion NN

$$RMSE(x) = \sqrt{(y(x_n, w) - t_n)^2} \tag{7.2}$$

Root Mean Squared Error

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}(x_i - \bar{x})^2 * \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{7.3}$$

Pearson Correlation Coefficient

7.1.3. Evaluation

As performance measures the Root Mean Squared Error (RMSE), Concordance Correlation Coefficient (CCC) and Pearson Correlation Coefficient (PCC) were used. The equation for RMSE is given in equation 7.2. The CCC and PCC were used to investigate the correlation of the predictions with the given targets. Those are computed following equations 7.3 and 7.4. These scores are given for arousal and valence in tables 7.4 and 7.5.

The chosen hyperparameters for each configuration are given in table 7.3. The corresponding scores can be found in the tables 7.4 and 7.5. Those are the average of five cross–validations. The result of 0.13 RMSE for data in an interval of -1 and 1 is not as high as expected. It is as well noted that the error did not reduce when applied to both parameters at once. It has been shown previously, that a combination of valence and arousal during training can improve the error measure [39]. This and the overall, comparatively low score hint towards great difficulty within the data set. This could be due to problems within the annotations or within the averaging of the annotation for the utterances.

7.2. Recurrent Neural Networks

To exploit the full knowledge of the time continuous annotations a RNN was trained on the data set to fit the annotation curves.

$$p_c = \frac{2r\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2 + (\mu_x - \mu_y)^2} \tag{7.4}$$

Concordance Correlation Coefficient

	Arousal	Valence	Combined
Architecture	(10, 3)	(5, 3)	(10, 3)
Dropout	0.8	0.8	0.8
Learning rate	0.002	0.006	0.006
Number of Epochs	230	210	210

Table 7.3.: Hyperparameters for the DE-ENIGMA NN

	RMSE	CCC	PCC
One Only	0.1332	0.1129	0.2089
Combined	0.1306	0.0814	0.2174

Table 7.4.: Results for DE-ENIGMA Arousal Only and Combined

	RMSE	CCC	PCC
One Only	0.1422	0.0438	0.1031
Combined	0.1441	0.0429	0.1129

Table 7.5.: Results for DE-ENIGMA Valence Only and Combined

7.2.1. Data Preparation

For the recurrent neural network, the DE-ENIGMA data was first completely converted to LLD eGeMAPS features. Those were created for each session at once. From there only child utterances were selected using the diarization annotations. The corresponding valence and arousal annotations were added as the target. The extracted utterances were then filtered by their duration. For emotion recognition comparatively short utterance lengths were admitted with a cut off at 0.5 seconds. Samples shorter than 0.5 seconds were neglected. This short length is due to the nature of the DE-ENIGMA data. The interaction concept of the therapy session was focused on teaching emotions. The children were shown faces by their therapist, the robot or pictures. They had to name or imitate those emotions. Due to this approach many single word utterances naming an emotion were collected which are of very short length. Another problem regarding the diarization annotation is the vocal trait of some of the children to use comparatively long inter word silences. For example, in the vocalizations of child B001 the diarization cut off after every single word instead of after a full utterance. This shall be improved in future work on the data set but could not be altered for this thesis. Another problem lies within the nature of ASC which often shows as language impairment, leading to very little and short speech segments. Utterances longer than four seconds were split into shorter parts. All data was normalized to a zero mean and one standard deviation derived from the training data. In figure 7.1 one can see the original utterance length distribution of the data set.

7.2.2. Model

For the DE-ENIGMA RNN the same output function applies as given in equation 6.1. In contrast to the categorical output in the ASC-Inclusion data set it is not passed to a softmax function. Instead it is directly treated as the output. It then is compared to the target data which is a continuous annotation as can be seen in equation 7.5.

As correlation measures PCC and CCC for the predicted utterances with respect to the gold standard annotation were computed as in section 7.1.1. PCC gives the linear correlation of two samples according to equation 7.3. The CCC scales their correlation with their mean squared difference. It was chosen because it is connected to the gold standard computation and often used as a standard measurement. This way the results can be compared to other works.

7.2.3. Evaluation

Just as for the recurrent version of the ASC-Inclusion network, no extensive hyperparameter tuning was performed for the same reasons. The hyperparameters can

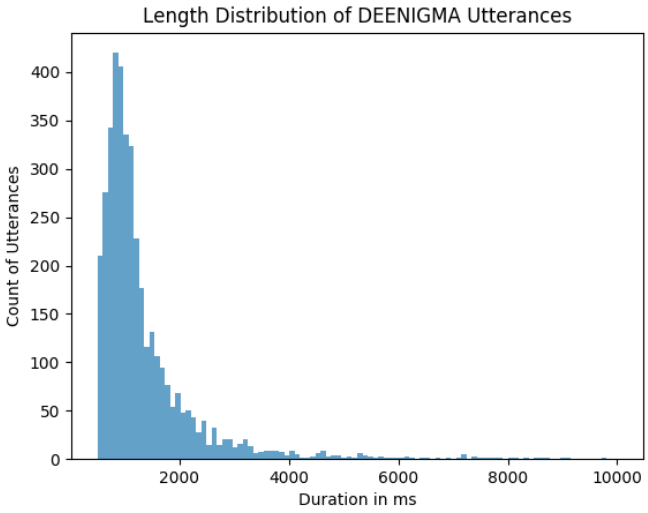


Figure 7.1.: The Distribution of Utterance Duration in the DE-ENIGMA Data Set. The Histogram is cut off at 10 s Excluding Two Longer Utterances.

$$E(w) = \frac{1}{2} \sum_{n=1}^{N} \sum_{t=0}^{T} ||y(x_n^t, w) - t_n^t||^2 + \lambda * \sum_j w_j^2 \tag{7.5}$$

Mean Squared Error with L2 Regularization for the RNN

	Arousal	Valence	Combined
Architecture	(75, 50)	(75, 50)	(75, 50)
Dropout	0.8	0.8	0.8
Learning rate	0.004	0.004	0.004
Number of Epochs	90	110	100

Table 7.6.: Hyperparameters for the DE-ENIGMA RNN

	RMSE	CCC	PCC
One Only	0.1354	0.1243	0.1669
Combined	0.1466	0.1006	0.1150

Table 7.7.: Results for DE-ENIGMA Continuous Arousal Only and Combined

be found in table 7.6. It must be considered for the evaluation of the DE-ENIGMA RNN that this data set is very difficult. Even during the simple NN experiments no considerably high scores could be achieved.

To improve upon the scores annotation delay was taken into account as described in [27]. Therefore, the annotation labels were shifted by 2 seconds. This could help since it takes into consideration the time passed while the annotators see and realize the action on the screen until they move the joystick to indicate their annotation. This technique shifted the results in arousal recognition from a CCC of 0.1430 to 0.1240 with the same architecture as specified in table 7.6. Nevertheless, this shift can be considered an additional hyperparameter which can be tuned and most likely can improve the scores. All following experiments were performed with this 2 second shift.

All final results can be found in tables 7.7 and 7.8. The complete results per fold can be found in the attachment or on the attached DVD.

As expected the MTL reaches slightly higher scores. The MTL CCC scores for arousal did not change much but the valence CCC score improved in MTL training. During MTL the valence score achieved improved, but the arousal score fell a bit in comparison. This is probably due to differences in difficulty between those tasks. Arousal seems to be fitted earlier and starts to overfit while valence scores still rise. This could be solved using different weights for the back–propagation. Those results could unfortunately not be tested for significance since not enough data points were available at the time of writing and could not be created due to time constraints. This creates a very interesting and easy starting point for future work on this project.

	RMSE	CCC	PCC
One Only	0.1569	0.06781	0.0908
Combined	0.1600	0.1038	0.1240

Table 7.8.: Results for DE-ENIGMA Continuous Valence Only and Combined

$$E(w, u, w_{LSTM}) = \frac{1}{2} \sum_{n=1}^{N} ||y(x_n, w, u, w_{LSTM}) - t_n||^2 + \lambda * (\sum_j w_j^2 + \sum_j u_j^2) \tag{7.6}$$

Error function for the Regression R-PNN

7.3. Cross–Task Learning with Recurrent Progressive Neural Networks

Progressive Recurrent Neural Networks are hoped to increase performance for cross–task learning. Simple Progressive Neural Networks have already shown promising results on cross–task learning in chapter 5. For this reason, they are evaluated in R-PNNs as well on the transfer of knowledge from continuous valence to arousal recognition and vice versa. Their performance will then be compared to MTL and RNNs trained only on valence or arousal.

7.3.1. Data Preparation

This was done using four–fold cross–validation as in section 7.2 and the same data preprocessing stack. One set of the RNNs trained in section 7.2 were used as the first columns, thereby ensuring that the network was not trained on data from those speakers before. Otherwise it could draw additional knowledge from the dependencies between valence and arousal scores. The pretrained networks can as well be found in the attachment.

7.3.2. Model

Recurrent Progressive Neural Networks is basically constructed as the cross–corpus R-PNN in section 6.3. The main difference is that here both columns are trained on a regression task. The error function for the regression R-PNN is given in equation 7.6.

	Aro->Val	Val->Aro
Architecture	(20, 10)	(20, 10)
Dropout	0.8.	0.8
Learning rate	0.004	0.004
Number of Epochs	50	50

Table 7.9.: Hyperparameters for the Extensive Column of the Cross–Task R-PNN

	RMSE	CCC	PCC
Aro -> Val	0.1548	0.0711	0.1033
Val -> Aro	0.1318	0.1192	0.1748

Table 7.10.: Results for DE-ENIGMA Continuous Prediction Cross–Task Learning

7.3.3. Evaluation

For the second column training the hyperparameters presented in table 7.9 were used. The results are presented in table 7.10 and CCC scores are presented in figure 7.2 in comparison to the scores achieved in section 7.2.

The scores for both, arousal and valence, slightly improved in this trial. The increase in the valence scores during MTL was not seen in cross–task R-PNNs.

7.4. Cross–Corpus Learning with Recurrent Progressive Neural Networks

Cross–Corpus Learning is tested for the DE-ENIGMA data set as well using R-PNNs. The first column is trained on the ASC-Inclusion data corpus while the second column is trained on the DE-ENIGMA data set.

7.4.1. Data Preparation

The same data pre–processing stack and data folds as in section 7.2 are used for this evaluation. For the ASC-Inclusion data the full data set was taken as training data and normalized to zero mean and one standard deviation.

	DE-ENIGMA
Architecture	(15, 8)
Dropout	0.8
Learning rate	0.004
Number of Epochs	100

Table 7.11.: Hyperparameters for the Extensive Column of the DE-ENIGMA R-PNN

	RMSE	CCC	PCC
Arousal	0.1355	0.1009	0.1443
Valence	0.1559	0.0625	0.0887

Table 7.12.: Results for DE-ENIGMA Continuous Prediction Cross–Corpus Learning

7.4.2. Evaluation

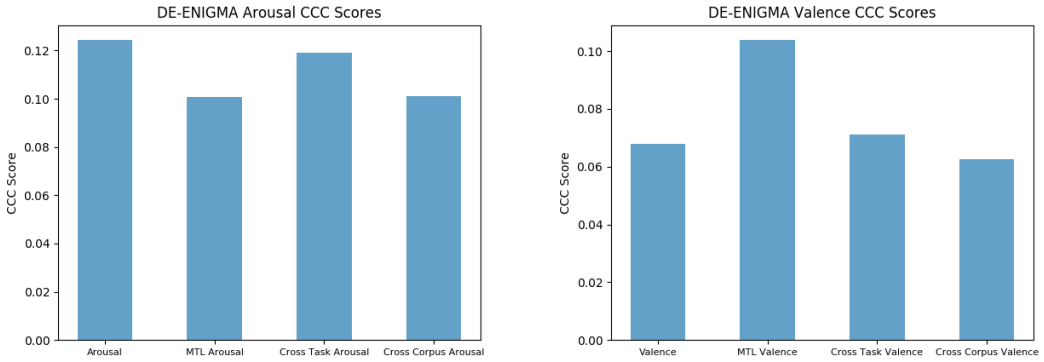
As for the other RNNs no extensive hyperparameter search was performed. Pretrained networks trained on all available data with the hyperparameters used in chapter **??** were used as the base column. The parameters for the extensive column can be found in table 7.11.

The results for the DE-ENIGMA data set are given in table 7.12. The scores are comparable to the ones reached during simple RNN training. Neither the arousal nor the valence score improved in comparison to simple RNN training. Those scores were nevertheless reached with a much smaller architecture. Even if the non-adaptable parameters of the first column are considered those still sum up to less than the parameters of the (75, 50) architectures used in section 7.2.

7.5. Discussion

In figure 7.2 one can see, that all transfer learning approaches, MTL, cross–task learning and cross–corpus learning, improved or are equal to the basic RNN scores. All those techniques seem promising but a more extended test including hyperparameter tuning would be needed to show their significance and a more in–depth comparison. This was not performed during this thesis due to time constraints but can easily be done with the attached code and framework.

As for all cross–corpus learning techniques standard problems are faced during cross–corpus evaluation. The most obvious of them is the difference in annotation. A



(a) DE-ENIGMA CCC Scores Arousal

(b) DE-ENIGMA CCC Scores Valence

Figure 7.2.: DE-ENIGMA CCC Score Comparison

regression task is performed for the DE-ENIGMA data set whereas the ASC-Inclusion data set poses a classification problem. The R-PNN provides an easy solution for this since the output of the columns are independent from each other. Since arousal, valence and categorical emotions are highly connected it is hoped, that knowledge can thus be transferred from one corpus to another. The other main challenge is the difference in the data collection process. Not only do both data sets highly differ in background noise and recording conditions. They also result from a very different setting. In the ASC-Inclusion data set the children stage emotional utterances. The DE-ENIGMA data is more comparable to real world data with short, natural utterances. Those problems could cause transfer learning to be less successful than hoped for. Here, no increase in performance could be seen.

8. Conclusion

In the following chapter everything learned from the experiments performed during this thesis will be summarized and final conclusions will be drawn. Additionally, different options for future improvements will be given. First emotion recognition on the ASC-Inclusion data set will be summarized in section 8.1.1 and be compared to results from other researchers on this data set. Afterwards the results of the DE-ENIGMA emotion recognition will be summed up in section 8.1.2. For this data set techniques for future improvement will be given in section 8.1.3. Finally, Progressive Neural Networks and Recurrent Progressive Neural Networks (R-PNNs) will be evaluated in section 8.2 for cross–task and cross–corpus learning. In section 8.2.2 an overview of possible next steps and application possibilities is given.

8.1. Emotion Recognition

After all the research done lately in emotion recognition and the improvements made on emotion recognition in general it is still considered a difficult problem. This not only stems from the different definitions of emotion but also from the point of view. Emotion can be seen as something that emerges inside and is very subjective or as done in this thesis as a state of mind portrayed by a human. This can be done as some kind of enactment like in the ASC-Inclusion data or during natural conversation as in the DE-ENGIMA dataset. Those definitions and additional information on their use in acoustic emotion recognition are described in [51]. Even we as humans are not able to always predict the correct emotion and generally reach an accuracy of 60-80% in acoustic categorical emotion recognition [3].

8.1.1. Emotion Recognition on ASC-Inclusion Data

For the ASC-Inclusion data set there already exist benchmarks and publications working on the data. But it must be mentioned that this combination of Swedish, Hebrew and English data has not yet been jointly evaluated before.

As a comparison to the Neural Network (NN) and progressive NN trained for typicality recognition and binary arousal and valence annotations the achieved results are overall worse than the ones achieved in *Emotion in the Speech of Children with Autism*

Spectrum Conditions: Prosody and Everything Else [11]. There only the Hebrew data set was considered. Those scores are not simply comparable since they were achieved only on the Hebrew part of the data set and for the binary arousal and valence scores and the emotion recognition only for a subgroup of the children (Typically Developing (TD) or Autistic Spectrum Disorder (ASC) only). Additionally, those results were a Leave One Speaker Out (LOSO) evaluation whereas the ones from this thesis are 7–fold cross–validation results on the joint English and Hebrew data set. Additionally, different feature sets were used.

In another paper on the ASC-Inclusion dataset it was shown that multiple cultures pose an additional problem and therefore might be to blame for this slightly worse score [55]. This can as well be seen in the difficulties with cross–culture evaluation. Additionally, from the evaluation of emotion recognition with only the Hebrew part of the data set it can already be seen that the emotions *proud* and *calm* are considered difficult. They as well have a low score on their prediction of the emotion *sad* which was not the case during the experiments presented here.

In the paper *Typicality and Emotion in the Voice of Children with Autism Spectrum Condition: Evidence Across Three Languages* all three cultures of the data set are evaluated separately in the emotion recognition task. Those scores are not comparative since the models were fitted on only the data from TD children from one language set. But the results already hint towards the increased difficulty in emotion recognition on the Swedish part of the data set with a decrease in Unweighted Average Recall (UAR) of at least 4 % in comparison to emotion recognition tasks on the other two languages. This difficulty was as well observed during the experiments presented here.

Overall the scores on emotion recognition and the other tasks are comparable to the ones achieved in previous works or slightly worse for the reasons presented above. Trends from previous works were mostly seen in the presented experiments as well. Additionally, it was shown that a slight improvement was made using Cross–Task Progressive Neural Networks. Cross–culture evaluation was tested for all tasks but did not prove representative or useful for the ASC-Inclusion data. It would be an interesting research topic to evaluate if the trends seen here are significant and what conclusions can be drawn from this for culture independent emotion recognition.

8.1.2. Emotion Recognition on DE-ENIGMA Data

This work is the first baseline produced on the DE-ENIGMA data set. Up to now only preliminary experiments using Support Vector Machines (SVMs) were run on the data but not yet published. Those worked with time windows and extended Geneva Minimalistic Acoustic Parameter Set (eGeMAPS) Functionals over 3 seconds and an average Concordance Correlation Coefficient (CCC) score of 0.07115 and 0.180781 for

valence and arousal respectively were achieved. The computation and evaluation of those scores were quite different from the evaluation and computation here. Especially considering the short utterance lengths by the children in comparison to the three second windows used. Therefore, a direct comparison to this baseline is not possible.

The scores reached in section 7.2 for Recurrent Neural Networks (RNNs) are not very high compared to acoustic emotion recognition scores on TD adults or children. Nevertheless, they can serve as a baseline for future recognition systems trained on the DE-ENIGMA data set as this data set is especially challenging. Possible improvements could stem from the transfer learning approaches presented in this thesis.

There are some problems that complicate acoustic emotion recognition on the DE-ENIGMA data. First the annotation process itself is not focused on acoustic emotion recognition since annotators saw the video stream whilst annotating the data. Therefore, they might have incorporated additional knowledge from the facial expression or body language displayed by the children. This knowledge is not included in the acoustic data and therefore cannot easily be predicted.

Another problem has already been presented in section 7.2 due to highly differing language scores between the participants and their vocal activity during the sessions the data varies strongly. Additionally, short utterances are more difficult to be labeled with a certain emotional score than longer ones. Errors in diarization engrave this problem.

The difficulty and a possible solution to the annotation delay has already been given in section 7.2.3. This delay is well known and can easily be explained. Additional tuning of the delay as another hyperparameter could enhance the results.

Last but not least the main difficulty of this dataset is the nature of ASC. The participants are impaired in their social ability to display and understand emotions. Therefore, it is even for humans not trivial to infer their emotions from their facial expressions and their vocalizations. Human recognition rates of 60-80 % on TD people [3] don't transfer to ASC participants and must be assumed to be lower.

8.1.3. Future Improvement Possibilities

To improve emotion recognition scores on the DE-ENIGMA data set multiple strategies can be proposed.

The first and most simple improvement can stem from extensive hyperparameter tuning. The architecture, learning rate, dropout probability and annotation bias shift are among the hyperparameters to be fitted. Additionally, different recurrent gates and activation functions can be evaluated. For example, the Gated Recurrent Unit (GRU) cell could be easily incorporated into the design. Additional filtering of the output and target data could as well improve the scores and reliability of the predictions.

The code attached to this thesis already provides the possibility for LOSO evaluation. This can improve the scores reached and provide more insight into possible dependencies between language scores, speech duration and the evaluation scores. This was not done during this thesis due to time constraints.

Another interesting trial would be emotion recognition on culture dependent subsets. As can already be seen on the ASC-Inclusion results culture and language plays an important role in emotion recognition. Therefore, models trained on only one part of the data set can be expected to perform better. Those could additionally profit from a cross–culture transfer learning approach using R-PNNs to transfer knowledge from one culture to the other.

The most obvious and probably greatest improvement can be expected from an incorporation of the camera data into the recognition system. The additional modality could help with the recognition of emotions which are mostly apparent in the facial expression or body language of the participants. Additional knowledge could be extracted as well from the time before and after an utterance. This could help to put the very short utterances into context.

Another interesting approach would be to use cross–corpus learning with a big TD data set. It would be favorable for this data set to consist of speech data from children since this is more correlated to the DE-ENIGMA data set. If this transfer is done through a R-PNN or another progressive NN structure, the relevance of the previously learned features could as well be evaluated through inspection of the transferring U-weights. This could lead to additional insight in differences in emotion display in acoustic data between TD and ASC children.

8.2. (Recurrent) Progressive Neural Networks for Transfer Learning

Progressive Neural Networks were only presented in 2016 [47] and therefore have not yet been evaluated and applied for a lot of tasks. They were already tested for cross–task and cross–corpus learning on paralinguistic tasks in [18]. Recurrent Progressive Neural Networks have never been presented before to the author's knowledge.

8.2.1. Results of this Evaluation

The results presented in [18] for simple progressive NN were found to be true in this thesis as well. They were tested on gender, typicality, and binary arousal and valence recognition. There they improved the performance more than normal pre–training and fine–tuning. Those results can be found in chapter 5. The results are coherent with

the ones presented in their paper. The results as well showed improvement over the standard pre–training and fine–tuning approach and a simple NN.

Recurrent Progressive Neural Networks have never been presented before to the authors knowledge. They provide a new way of transfer learning for RNNs. Most approaches are based on Multi Task Learning (MTL) which is difficult for different output parameters. Those two approaches were tested and evaluated on the DE-ENIGMA data set for arousal and valence. Predicting them with one RNN each, jointly using a MTL approach and each with a R-PNN. Here both, MTL and cross–task R-PNNs showed slight improvements which seem promising but should still be tested for significance.

They were as well evaluated in a cross–corpus learning tasks for the DE-ENIGMA and ASC-Inclusion data set. There the different nature of the utterances - staged versus natural - and the different recording conditions and quality complicated the knowledge transfer. In the results here, no improvement could be seen in comparison to simple RNN training. This does not imply an overall problem with R-PNNs but could be due to the difficulty of the data, the hyperparameters or the differences between the data sets.

8.2.2. Future Work

As a first and most basic step, the results presented here should be evaluated for significance. This can be done with the code attached to this thesis. The results of LOSO evaluations would provide enough data to perform reliable significance tests.

Progressive Neural Networks provide another very interesting possibility for analysis that has not been taken in this thesis. The transferring weights U can be analyzed to see which features are considered for the second column. It would be helpful to know if low or high–level features are mostly transferred. This could provide additional insight into the differences in the data set and the knowledge transferred. Additionally, the complete set of those transferring weights in comparison to the weights inside the second column can be taken as a metric to measure how much knowledge can be transferred between the columns for different tasks or data sets.

The transfer learning technique of R-PNNs still needs to be evaluated on more tasks and settings. The experiments here already show the potential in this transfer learning technique. Bigger data sets and more correlated corpora could provide additional benchmarks for the performance of those networks. Other recurrent cells could to be tested in the context of R-PNNs. It would as well be insightful to apply R-PNNs to a recent competition data set to get a better comparison to the baseline and other state of the art learning techniques. This could be done for example in one of the Audio/ Visual Emotion Recognition Challenge (AVEC) competitions.

List of Figures

List of Tables

List of equations

Bibliography

[1] Shahin Amiriparian et al., eds. *Sequence to sequence autoencoders for unsupervised representation learning from audio*.

[2] American Psychiatric Association. *Diagnostic and Statistical Manual of Mental Disorders*. American Psychiatric Association, 2013. ISBN: 0-89042-555-8. DOI: 10.1176/appi.books.9780890425596.

[3] Rainer Banse and Klaus R. Scherer. "Acoustic profiles in vocal emotion expression." In: *Journal of Personality and Social Psychology* 70.3 (1996), p. 614. ISSN: 1939-1315.

[4] Tanja Bänziger, Sona Patel, and Klaus R. Scherer. "The role of perceived voice and speech characteristics in vocal emotion communication." In: *Journal of nonverbal behavior* 38.1 (2014), pp. 31–52.

[5] Simon Baron-Cohen, Alan M. Leslie, and Uta Frith. "Does the autistic child have a "theory of mind" ?" In: *Cognition* 21.1 (1985), pp. 37–46. ISSN: 00100277. DOI: 10.1016/0010-0277(85)90022-8.

[6] Christopher M. Bishop. *Pattern recognition and machine learning*. Corrected at 8th printing 2009. Information science and statistics. New York, NY: Springer, 2009. ISBN: 9781493938438.

[7] Paul Boersma and David Weenink. *Praat: Doing phonetics by computer (Version 5.1. 04)[Computer program]. Retrieved April 4, 2009*. 2009.

[8] Carlos Busso, Sungbok Lee, and Shrikanth Narayanan. "Analysis of emotionally salient aspects of fundamental frequency for emotion detection." In: *IEEE Transactions on Audio, Speech, and Language Processing* 17.4 (2009), pp. 582–596. ISSN: 1558-7916.

[9] Junyoung Chung et al. "Gated Feedback Recurrent Neural Networks." In: 2015, pp. 2067–2075. ISBN: 1938-7228.

[10] Erik Marchi, Björn Schuller, Anton Batliner, Shimrit Fridenzon, Shahar Tal, Ofer Golan. *Emotion in the Speech of Children with Autism Spectrum Conditions: Prosody and Everything Else*. 2012.

[11] Erik Marchi, Simon Baron-cohen, Ofer Golan, Prerna Arora. *Typicality and Emotion in the Voice of Children with Autism Spectrum Condition: Evidence Across Three Languages*.

[12] Florian Eyben. *Real-time speech and music classification by large audio feature space extraction*. Springer, 2015.

[13] Florian Eyben, Martin Wöllmer, and Björn Schuller. "Opensmile: The Munich Versatile and Fast Open-source Audio Feature Extractor." In: *Proceedings of the 18th ACM International Conference on Multimedia*. MM '10. New York, NY, USA: ACM, 2010, pp. 1459–1462. ISBN: 978-1-60558-933-6. DOI: 10.1145/1873951.1874246.

[14] Florian Eyben et al. "The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing." In: *IEEE Transactions on Affective Computing* 7.2 (2016), pp. 190–202. ISSN: 1949-3045. DOI: 10.1109/TAFFC.2015.2457417.

[15] Gunnar Fant, Johan Liljencrants, and Qi-guang Lin. "A four-parameter model of glottal flow." In: *STL-QPSR* 4.1985 (1985), pp. 1–13.

[16] Uta Frith and Francesca Happé. "Autism: Beyond "theory of mind"." In: *Cognition* 50.1-3 (1994), pp. 115–132. ISSN: 00100277. DOI: 10.1016/0010-0277(94)90024-8.

[17] F. A. Gers, J. Schmidhuber, and F. Cummins. *Learning to forget: Continual prediction with LSTM*. 1999. DOI: 10.1049/cp_19991218-af1,.

[18] John Gideon et al. *Progressive Neural Networks for Transfer Learning in Emotion Recognition*. 2017.

[19] Temple Grandin. "How People with Autism Think." In: *Learning and cognition in autism*. Ed. by Eric Schopler and Gary B. Mesibov. [Place of publication not identified]: Springer-Verlag New York, 2013, pp. 137–156. ISBN: 978-1-4899-1286-2. DOI: 10.1007/978-1-4899-1286-2_8.

[20] Klaus Greff et al. "LSTM: A Search Space Odyssey." In: *IEEE transactions on neural networks and learning systems* 28.10 (2017), pp. 2222–2232. ISSN: 2162-2388. DOI: 10.1109/TNNLS.2016.2582924.

[21] Gerhard Hagerer et al. "Robust Laughter Detection for Wearable Wellbeing Sensing." In: *Proceedings of the 2018 International Conference on Digital Health - DH '18*. Ed. by Patty Kostkova et al. New York, New York, USA: ACM Press, 2018, pp. 156–157. ISBN: 9781450364935. DOI: 10.1145/3194658.3194693.

[22] Kaiming He et al. "Deep residual learning for image recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[23] Geoffrey Hinton et al. "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups." In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 82–97. ISSN: 1053-5888. DOI: 10.1109/MSP.2012.2205597.

[24] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory." In: *Neural Computation* 9.8 (1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735.

[25] Patricia Howlin et al. *Teaching children with autism to mind-read: A practical guide for teachers and parents*. Repr. Chichester u.a.: Wiley, 2000. ISBN: 0-471-97623-7.

[26] Gang Hua et al., eds. *Best Practices for Fine-Tuning Visual Classifiers to New Domains: Computer Vision – ECCV 2016 Workshops*. Springer International Publishing, 2016. ISBN: 978-3-319-49409-8.

[27] Zhaocheng Huang et al. "An investigation of annotation delay compensation and output-associative fusion for multimodal continuous emotion prediction." In: *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge*. 2015, pp. 41–48.

[28] *Interspeech 2016*. Interspeech. ISCA, 2016.

[29] Kenzo Ishizaka and James L. Flanagan. "Synthesis of voiced sounds from a two-mass model of the vocal cords." In: *Bell system technical journal* 51.6 (1972), pp. 1233–1268.

[30] Tom Johnstone. *The effect of emotion on voice production and speech acoustics*. 2017. DOI: 10.31237/osf.io/qd6hz.

[31] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*.

[32] Quoc V. Le. "Building high-level features using large scale unsupervised learning." In: *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. 2013, pp. 8595–8598.

[33] Jinkyu Lee and Ivan Tashev. *High-level Feature Representation using Recurrent Neural Network for Speech Emotion Recognition*. 2015.

[34] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. "Recurrent Neural Network for Text Classification with Multi-Task Learning." In: *CoRR* abs/1605.05101 (2016).

[35] Packwood Lynn. *DE-ENIGMA progress report YR1: Deliverable D:7.2*. 2017.

[36] Erik Marchi et al. "Voice Emotion Games: Language and Emotion in the Voice of Children with Autism Spectrum Conditio." In: *Proceedings of the 3rd International Workshop on Intelligent Digital Games for Empowerment and Inclusion (IDGEI 2015) as part of the 20th ACM International Conference on Intelligent User Interfaces, IUI 2015*. 2015, 9–pages.

[37] Warren S. McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." In: *The Bulletin of Mathematical Biophysics* 5.4 (1943), pp. 115–133. ISSN: 0007-4985. DOI: 10.1007/BF02478259.

[38] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning." In: *Nature* 518.7540 (2015), p. 529. ISSN: 0028-0836. DOI: 10.1038/nature14236.

[39] Srinivas Parthasarathy and Carlos Busso. "Jointly Predicting Arousal, Valence and Dominance with Multi-Task Learning." In: *Interspeech 2017*. ISCA: ISCA, 2017, pp. 1103–1107. DOI: 10.21437/Interspeech.2017-1494.

[40] Elizabeth Pellicano and David Burr. "When the world becomes 'too real': A Bayesian explanation of autistic perception." In: *Trends in cognitive sciences* 16.10 (2012), pp. 504–510. DOI: 10.1016/j.tics.2012.08.009.

[41] Daniel J. Ricks and Mark B. Colton. "Trends and considerations in robot-assisted autism therapy." In: *IEEE International Conference on Robotics and Automation (ICRA), 2010*. Piscataway, NJ: IEEE, 2010, pp. 4354–4359. ISBN: 978-1-4244-5038-1. DOI: 10.1109/ROBOT.2010.5509327.

[42] Fabien Ringeval et al., eds. *AVEC'17: Proceedings of the 7th Annual Workshop on Audio/Visual Emotion Challenge : October 23, 2017, Mountain View, CA, USA*. New York, New York: The Association for Computing Machinery, 2017. ISBN: 9781450355025. DOI: 10.1145/3133944.

[43] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." In: *Nature* 323.6088 (1986), pp. 533–536. ISSN: 0028-0836. DOI: 10.1038/323533a0.

[44] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge." In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252. ISSN: 1573-1405. DOI: 10.1007/s11263-015-0816-y.

[45] James A. Russell. "A circumplex model of affect." In: *Journal of Personality and Social Psychology* 39.6 (1980), pp. 1161–1178. ISSN: 1939-1315. DOI: 10.1037/h0077714.

[46] James A. Russell and Merry Bullock. "Multidimensional scaling of emotional facial expressions: Similarity from preschoolers to adults." In: *Journal of Personality and Social Psychology* 48.5 (1985), pp. 1290–1298. ISSN: 1939-1315. DOI: 10.1037/0022-3514.48.5.1290.

[47] Andrei A. Rusu et al. *Progressive Neural Networks*. 2016.

[48] Andrei A. Rusu et al. "Progressive Neural Networks." In: *CoRR* abs/1606.04671 (2016).

[49] Tara N. Sainath et al. "Learning the speech front-end with raw waveform CLDNNs." In: *Sixteenth Annual Conference of the International Speech Communication Association*. 2015.

[50] Brian Scassellati, Henny Admoni, and Maja Matarić. "Robots for use in autism research." In: *Annual review of biomedical engineering* 14 (2012), pp. 275–294. ISSN: 1545-4274. DOI: 10.1146/annurev-bioeng-071811-150036.

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

[52] Klaus R. Scherer et al. "Comparing the acoustic expression of emotion in the speaking and the singing voice." In: *Computer Speech & Language* 29.1 (2015), pp. 218–235. ISSN: 08852308. DOI: 10.1016/j.csl.2013.10.002.

[53] Annett Schirmer and Ralph Adolphs. "Emotion Perception from Face, Voice, and Touch: Comparisons and Convergence." In: *Trends in cognitive sciences* 21.3 (2017), pp. 216–228. DOI: 10.1016/j.tics.2017.01.001.

[54] Harold Schlosberg. "The description of facial expressions in terms of two dimensions." In: *Journal of Experimental Psychology* 44.4 (1952), pp. 229–237. ISSN: 0022-1015. DOI: 10.1037/h0055778.

[55] Maximilian Schmitt et al. "Towards cross-lingual automatic diagnosis of autism spectrum condition in children's voices." In: *Speech Communication; 12. ITG Symposium; Proceedings of*. 2016, pp. 1–5.

[56] Björn Schuller et al. "ASC-Inclusion: Interactive emotion games for social inclusion of children with Autism Spectrum Conditions." In: *Proceedings 1st International Workshop on Intelligent Digital Games for Empowerment and Inclusion (IDGEI 2013) held in conjunction with the 8th Foundations of Digital Games 2013 (FDG)(B. Schuller, L. Paletta, and N. Sabouret, eds.), Chania, Greece*. 2013.

[57] Bjorn Schuller et al. *The INTERSPEECH 2013 Computational Paralinguistics Challenge: Social Signals, Conflict, Emotion, Autism*. 2013.

[58] Björn Schuller et al. "The INTERSPEECH 2016 Computational Paralinguistics Challenge: Deception, Sincerity and Native Language." In: *Interspeech 2016*. Interspeech. ISCA, 2016, pp. 2001–2005. DOI: 10.21437/Interspeech.2016-129.

[59] David Silver et al. "Mastering the game of Go with deep neural networks and tree search." In: *Nature* 529.7587 (2016), p. 484. ISSN: 0028-0836. DOI: 10.1038/nature16961.

[60] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." In: *CoRR* abs/1409.1556 (2014).

[61] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting." In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[62] Brad H. Story. *Mechanisms of Voice Production*. 2015. DOI: 10.1002/9781118584156.ch3.

[63] Brad H. Story and Ingo R. Titze. "Voice simulation with a body-cover model of the vocal folds." In: *The Journal of the Acoustical Society of America* 97.2 (1995), pp. 1249–1260.

[64] Ingo R. Titze. "Nonlinear source–filter coupling in phonation: Theory." In: *The Journal of the Acoustical Society of America* 123.4 (2008), pp. 1902–1915.

[65] Silvan S. Tomkins. *Affect Imagery Consciousness: The Complete Edition (Two Volumes)*. New York: Springer Pub. Co, 2008. ISBN: 0826144098.

[66] Aäron van den Oord et al. "WaveNet: A Generative Model for Raw Audio." In: *CoRR* abs/1609.03499 (2016).

[67] Bernard Widrow and Marcian E. Hoff. *Adaptive switching circuits*. 1960.

[68] Rui Xia and Yang Liu. "A Multi-Task Learning Framework for Emotion Recognition Using 2D Continuous Space." In: *IEEE Transactions on Affective Computing* 8.1 (2017), pp. 3–14. ISSN: 1949-3045. DOI: 10.1109/TAFFC.2015.2512598.

[69] Serdar Yildirim et al. "An acoustic study of emotions expressed in speech." In: *Eighth International Conference on Spoken Language Processing*. 2004.

A. Simple Progressive Neural Networks

A.1. Hyperparameters

	ASC	Gender	Arousal	Valence
Architecture	(5,3)	(10,3)	(5,3)	(10,3)
Dropout	0.5	0.6	0.8	0.8
Learning rate	0.004	0.001	0.001	0.002
Number of Epochs	80	70	50	200

Table A.1.: Hyperparameters chosen for the simple neural nets with eGeMAPS features

	Gender to ASC	Arousal to ASC	Valence to ASC
Architecture of extensible column	(5,3)	(25,5)	(5,3)
Dropout	0.8	1	0.6
Learning rate	0.004	0.006	0.001
Number of Epochs	50	90	70

Table A.2.: Hyperparameters chosen for the progressive neural nets extensible column for ASC recognition with eGeMAPS features

	ASC to Gender	Arousal to Gender	Valence to Gender
Architecture of extensible column	(10,3)	(10,3)	(5,3)
Dropout	0.8	0.6	0.8
Learning rate	0.001	0.002	0.006
Number of Epochs	160	130	120

Table A.3.: Hyperparameters chosen for the progressive neural nets extensible column for gender recognition with eGeMAPS features

	ASC to Arousal	Gender to Arousal	Valence to Arousal
Architecture of extensible column	(5,3)	(25,5)	(5,3)
Dropout	0.6	0.8	0.5
Learning rate	0.006	0.001	0.002
Number of Epochs	50	70	70

Table A.4.: Hyperparameters chosen for the progressive neural nets extensible column for arousal recognition with eGeMAPS features

	ASC to Valence	Gender to Valence	Arousal to Valence
Architecture of extensible column	(5,3)	(5,3)	(5,3)
Dropout	0.5	1	0.5
Learning rate	0.004	0.002	0.002
Number of Epochs	120	60	140

Table A.5.: Hyperparameters chosen for the progressive neural nets extensible column for valence recognition with eGeMAPS features

	Gender to ASC	Arousal to ASC	Valence to ASC
Dropout	1	0.5	1
Learning rate	0.0005	0.002	0.002
Number of Epochs	50	70	170

Table A.6.: Hyperparameters chosen for the finetuning all layers for ASC recognition with eGeMAPS features

	Gender to ASC	Arousal to ASC	Valence to ASC
Dropout	1	1	1
Learning rate	0.006	0.001	0.002
Number of Epochs	40	120	130

Table A.7.: Hyperparameters chosen for the finetuning the last layer for ASC recognition with eGeMAPS features

	ASC to Gender	Arousal to Gender	Valence to Gender
Dropout	0.8	0.8	0.8
Learning rate	0.001	0.002	0.0005
Number of Epochs	60	90	230

Table A.8.: Hyperparameters chosen for the finetuning all layers for gender recognition with eGeMAPS features

	ASC to Gender	Arousal to Gender	Valence to Gender
Dropout	1	1	0.6
Learning rate	0.004	0.006	0.001
Number of Epochs	120	40	150

Table A.9.: Hyperparameters chosen for the finetuning the last layer for gender recognition with eGeMAPS features

	ASC to Arousal	Gender to Arousal	Valence to Arousal
Dropout	0.6	0.5	0.6
Learning rate	0.0005	0.0005	0.0005
Number of Epochs	110	180	110

Table A.10.: Hyperparameters chosen for the finetuning all layers for arousal recognition with eGeMAPS features

	ASC to Arousal	Gender to Arousal	Valence to Arousal
Dropout	0.8	0.8	1
Learning rate	0.006	0.002	0.006
Number of Epochs	60	100	50

Table A.11.: Hyperparameters chosen for the finetuning the last layer for arousal recognition with eGeMAPS features

	ASC to Valence	Gender to Valence	Arousal to Valence
Dropout	0.8	0.6	0.8
Learning rate	0.0005	0.001	0.001
Number of Epochs	120	80	60

Table A.12.: Hyperparameters chosen for the finetuning all layers for valence recognition with eGeMAPS features

	ASC to Valence	Gender to Valence	Arousal to Valence
Dropout	1	0.8	0.8
Learning rate	0.001	0.002	0.004
Number of Epochs	200	160	210

Table A.13.: Hyperparameters chosen for the finetuning the last layer for valence recognition with eGeMAPS features

	ASC	Gender	Arousal	Valence
Architecture	(5,3)	(10,3)	(10,3)	(25,5)
Dropout	0.5	0.6	0.5	1
Learning rate	0.002	0.001	0.001	0.001
Number of Epochs	115	100	155	140

Table A.14.: Hyperparameters chosen for the simple neural nets with COMPARE features

A.2. Pretrained Architectures

	English and Hebrew	English and Swedish	Hebrew and Swedish
Architecture	(5, 3)		
Dropout	0.5		
Learning rate	0.004		
Number of Epochs	80		
UAR	0.580	0.689	0.556
Recall ASC	0.578	0.360	0.275
Precision ASC	0.536	0.556	0.504
F1 Score ASC	0.557	0.427	0.356
Recall TD	0.582	0.855	0.782
Precision TD	0.623	0.725	0.572
F1 Score TD	0.579	0.611	0.508

Table A.15.: Evaluation of the pretrained ASC simple net with eGeMAPS features

	English and Hebrew	English and Swedish	Hebrew and Swedish
Architecture	(5, 3)		
Dropout	0.5		
Learning rate	0.004		
Number of Epochs	80		
UAR	0.580	0.689	0.556
Recall ASC	0.578	0.360	0.275
Precision ASC	0.536	0.556	0.504
F1 Score ASC	0.557	0.427	0.356
Recall TD	0.582	0.855	0.782
Precision TD	0.623	0.725	0.572
F1 Score TD	0.579	0.611	0.508

Table A.16.: Evaluation of the pretrained gender recognition simple net with eGeMAPS features

	English and Hebrew	English and Swedish	Hebrew and Swedish
Architecture	(5, 3)		
Dropout	0.8		
Learning rate	0.001		
Number of Epochs	50		
UAR	0.575	0.639	0.766
Recall ASC	0.713	0.775	0.217
Precision ASC	0.482	0.529	0.54
F1 Score ASC	0.574	0.629	0.31
Recall TD	0.482	0.55	0.941
Precision TD	0.714	0.789	0.79
F1 Score TD	0.575	0.639	0.585

Table A.17.: Evaluation of the pretrained arousal recognition simple net with eGeMAPS features

	English and Hebrew	English and Swedish	Hebrew and Swedish
Architecture	(10, 3)		
Dropout	0.8		
Learning rate	0.006		
Number of Epochs	200		
UAR	0.621	0.577	0.534
Recall ASC	0.625	0.595	0.532
Precision ASC	0.524	0.506	0.447
F1 Score ASC	0.57	0.547	0.488
Recall TD	0.619	0.563	0.532
Precision TD	0.711	0.649	0.62
F1 Score TD	0.616	0.575	0.53

Table A.18.: Evaluation of the pretrained valence recognition simple net with eGeMAPS features

	English and Hebrew	English and Swedish	Hebrew and Swedish
Architecture	(5, 3)		
Dropout	0.5		
Learning rate	0.002		
Number of Epochs	115		
UAR	0.530	0.66	0.559
Recall ASC	0.735	0.326	0.838
Precision ASC	0.49	0.492	0.504
F1 Score ASC	0.588	0.392	0.63
Recall TD	0.36	0.829	0.335
Precision TD	0.619	0.708	0.72
F1 Score TD	0.522	0.578	0.543

Table A.19.: Evaluation of the pretrained ASC simple net with COMPARE features

	English and Hebrew	English and Swedish	Hebrew and Swedish
Architecture	(10, 3)		
Dropout	1		
Learning rate	0.001		
Number of Epochs	100		
UAR	0.647	0.597	0.547
Recall ASC	0.728	0.792	0.823
Precision ASC	0.79	0.621	0.567
F1 Score ASC	0.758	0.696	0.671
Recall TD	0.395	0.326	0.196
Precision TD	0.318	0.53	0.463
F1 Score TD	0.555	0.55	0.473

Table A.20.: Evaluation of the pretrained gender recognition simple net with COMPARE features

	English and Hebrew	English and Swedish	Hebrew and Swedish
Architecture	(10, 3)		
Dropout	0.5		
Learning rate	0.001		
Number of Epochs	155		
UAR	0.545	0.601	0.70
Recall ASC	0.505	0.459	0.516
Precision ASC	0.442	0.495	0.405
F1 Score ASC	0.471	0.476	0.454
Recall TD	0.571	0.694	0.758
Precision TD	0.632	0.663	0.831
F1 Score TD	0.536	0.577	0.623

Table A.21.: Evaluation of the pretrained arousal recognition simple net with COMPARE features

	English and Hebrew	English and Swedish	Hebrew and Swedish
Architecture	(25, 5)		
Dropout	1		
Learning rate	0.001		
Number of Epochs	140		
UAR	0.528	0.561	0.511
Recall ASC	0.346	0.476	0.321
Precision ASC	0.441	0.489	0.389
F1 Score ASC	0.388	0.482	0.351
Recall TD	0.667	0.626	0.645
Precision TD	0.573	0.614	0.574
F1 Score TD	0.502	0.551	0.479

Table A.22.: Evaluation of the pretrained valence recognition simple net with COMPARE features

A.3. Cross Culture Evaluation eGeMAPS

Type	Name	Language	Architectu	Learning	Dropout	NumSteps	CrossValidation	Accuracy	TP_Rate	PPV	TN_Rate	NPV	UAR	F1_Score
simple	ASC	EH	5-3-2	0.004	0.5	80	0.596634615	0.5986	0.5117	0.566	0.6713	0.6238	0.5915	0.5341
simple	ASC	SH	5-3-2	0.004	0.5	80	0.596634615	0.5101	0.2576	0.42	0.7137	0.5434	0.4857	0.3168
simple	ASC	SE	5-3-2	0.004	0.5	80	0.596634615	0.6733	0.3483	0.524	0.8382	0.7179	0.5932	0.414
simple	GEN	EH	10-3-2	0.001	0.6	70	0.635416667	0.4676	0.3962	0.815	0.6904	0.2928	0.5433	0.4971
simple	GEN	SH	10-3-2	0.001	0.6	70	0.635416667	0.5382	0.5333	0.602	0.5444	0.4767	0.5389	0.5629
simple	GEN	SE	10-3-2	0.001	0.6	70	0.635416667	0.5626	0.5714	0.646	0.5502	0.4812	0.5608	0.5965
simple	ARO	EH	5-3-2	0.001	0.8	50	0.779347826	0.5743	0.7352	0.481	0.4663	0.7248	0.6007	0.581
simple	ARO	SH	5-3-2	0.001	0.8	50	0.779347826	0.7665	0.0968	0.614	0.9799	0.7731	0.5384	0.1624
simple	ARO	SE	5-3-2	0.001	0.8	50	0.779347826	0.617	0.733	0.511	0.5413	0.7569	0.6371	0.6019
simple	VAL	EH	10-3-2	0.002	0.8	200	0.633333333	0.546	0.2673	0.457	0.758	0.5764	0.5126	0.3335
simple	VAL	SH	10-3-2	0.002	0.8	200	0.633333333	0.5448	0.4491	0.448	0.6123	0.6143	0.5307	0.4441
simple	VAL	SE	10-3-2	0.002	0.8	200	0.633333333	0.5671	0.6022	0.498	0.5407	0.6457	0.5715	0.5423
prog	ASC_to_GEN	EH	10-3-2	0.001	0.8	170	0.686111111	0.4239	0.3125	0.809	0.7712	0.2651	0.5418	0.4481
prog	ASC_to_GEN	SH	10-3-2	0.001	0.8	170	0.686111111	0.5706	0.5844	0.628	0.5529	0.5099	0.5686	0.6032
prog	ASC_to_GEN	SE	10-3-2	0.001	0.8	170	0.686111111	0.5703	0.7263	0.61	0.3529	0.4811	0.5396	0.6627
prog	ASC_to_ARO	EH	5-3-2	0.006	0.6	50	0.780434783	0.584	0.7205	0.488	0.4922	0.7242	0.6063	0.5818
prog	ASC_to_ARO	SH	5-3-2	0.006	0.6	50	0.780434783	0.755	0.2137	0.502	0.9275	0.7898	0.5706	0.2607
prog	ASC_to_ARO	SE	5-3-2	0.006	0.6	50	0.780434783	0.6095	0.7507	0.505	0.5172	0.7612	0.634	0.603
prog	ASC_to_VAL	EH	5-3-2	0.004	0.5	120	0.634567901	0.5337	0.3292	0.446	0.6894	0.575	0.5093	0.376
prog	ASC_to_VAL	SH	5-3-2	0.004	0.5	120	0.634567901	0.4359	0.8892	0.415	0.1166	0.5244	0.5029	0.5346
prog	ASC_to_VAL	SE	5-3-2	0.004	0.5	120	0.634567901	0.504	0.7811	0.459	0.2957	0.6208	0.5384	0.5702
prog	GEN_to_ASC	EH	5-3-2	0.004	0.8	50	0.591346154	0.6176	0.4988	0.597	0.7169	0.6313	0.6078	0.5422
prog	GEN_to_ASC	SH	5-3-2	0.004	0.8	50	0.591346154	0.5265	0.2463	0.444	0.7525	0.5533	0.4994	0.3152
prog	GEN_to_ASC	SE	5-3-2	0.004	0.8	50	0.591346154	0.6832	0.2955	0.555	0.8798	0.7115	0.5876	0.3836
prog	GEN_to_ARO	EH	25-5-2	0.001	0.8	70	0.779347826	0.5993	0.6464	0.502	0.5677	0.7049	0.607	0.5646
prog	GEN_to_ARO	SH	25-5-2	0.001	0.8	70	0.779347826	0.7448	0.1734	0.434	0.927	0.7789	0.5502	0.2425
prog	GEN_to_ARO	SE	25-5-2	0.001	0.8	70	0.779347826	0.5881	0.7187	0.486	0.5028	0.7329	0.6107	0.5795
prog	GEN_to_VAL	EH	5-3-2	0.002	1	60	0.632098765	0.5421	0.3029	0.455	0.7242	0.5773	0.5135	0.3597
prog	GEN_to_VAL	SH	5-3-2	0.002	1	60	0.632098765	0.4975	0.4717	0.405	0.5156	0.5825	0.4937	0.4338
prog	GEN_to_VAL	SE	5-3-2	0.002	1	60	0.632098765	0.5484	0.5987	0.479	0.5106	0.6303	0.5546	0.5307
prog	ARO_to_ASC	EH	25-5-2	0.006	1	90	0.680288462	0.6	0.5617	0.562	0.632	0.6332	0.5969	0.5606
prog	ARO_to_ASC	SH	25-5-2	0.006	1	90	0.680288462	0.5439	0.2572	0.48	0.775	0.5644	0.5161	0.3311

prog	ARO_to_ASC	SE	25-5-2	0.006	1	90	0.680288462	0.6554	0.3135	0.49	0.8288	0.7043	0.5711	0.3776
prog	ARO_to_GEN	EH	10-3-2	0.002	0.6	130	0.675	0.432	0.319	0.821	0.7842	0.2713	0.5516	0.4537
prog	ARO_to_GEN	SH	10-3-2	0.002	0.6	130	0.675	0.5846	0.584	0.644	0.5853	0.5246	0.5847	0.6109
prog	ARO_to_GEN	SE	10-3-2	0.002	0.6	130	0.675	0.5505	0.5562	0.63	0.5425	0.4669	0.5494	0.5898
prog	ARO_to_VAL	EH	5-3-2	0.002	0.5	140	0.613580247	0.528	0.3273	0.437	0.6807	0.571	0.504	0.3703
prog	ARO_to_VAL	SH	5-3-2	0.002	0.5	140	0.613580247	0.5844	0.3524	0.531	0.7478	0.61	0.5501	0.3816
prog	ARO_to_VAL	SE	5-3-2	0.002	0.5	140	0.613580247	0.538	0.6965	0.479	0.4189	0.6325	0.5577	0.5619
prog	VAL_to_ASC	EH	5-3-2	0.001	0.6	70	0.629807692	0.5513	0.5042	0.519	0.5907	0.5802	0.5474	0.4917
prog	VAL_to_ASC	SH	5-3-2	0.001	0.6	70	0.629807692	0.5166	0.2598	0.432	0.7236	0.5479	0.4917	0.3239
prog	VAL_to_ASC	SE	5-3-2	0.001	0.6	70	0.629807692	0.6786	0.3264	0.536	0.8573	0.7156	0.5918	0.4023
prog	VAL_to_GEN	EH	5-3-2	0.006	0.8	120	0.690277778	0.4409	0.3482	0.8	0.7299	0.2649	0.5391	0.4813
prog	VAL_to_GEN	SH	5-3-2	0.006	0.8	120	0.690277778	0.5653	0.5524	0.63	0.5818	0.5041	0.5671	0.5868
prog	VAL_to_GEN	SE	5-3-2	0.006	0.8	120	0.690277778	0.5393	0.5084	0.631	0.5824	0.461	0.5454	0.5585
prog	VAL_to_ARO	EH	5-3-2	0.002	0.5	70	0.77173913	0.6169	0.6177	0.52	0.6163	0.7059	0.617	0.5644
prog	VAL_to_ARO	SH	5-3-2	0.002	0.5	70	0.77173913	0.7532	0.2645	0.489	0.909	0.7951	0.5868	0.3399
prog	VAL_to_ARO	SE	5-3-2	0.002	0.5	70	0.77173913	0.6129	0.7837	0.507	0.5012	0.7807	0.6425	0.6153
fine_all	ASC_to_GEN	EH	5-3-2	0.001	0.8	60	0.630567608	0.4695	0.3924	0.809	0.7102	0.2744	0.5513	0.521
fine_all	ASC_to_GEN	SH	5-3-2	0.001	0.8	60	0.630567608	0.5218	0.5191	0.588	0.5253	0.4577	0.5222	0.5485
fine_all	ASC_to_GEN	SE	5-3-2	0.001	0.8	60	0.630567608	0.5612	0.5097	0.66	0.633	0.4814	0.5714	0.574
fine_all	ASC_to_ARO	EH	5-3-2	0.0005	0.6	110	0.716625538	0.5551	0.7754	0.469	0.4071	0.7296	0.5913	0.5835
fine_all	ASC_to_ARO	SH	5-3-2	0.0005	0.6	110	0.716625538	0.7674	0.0871	0.639	0.9843	0.772	0.5357	0.1492
fine_all	ASC_to_ARO	SE	5-3-2	0.0005	0.6	110	0.716625538	0.617	0.745	0.511	0.5334	0.7628	0.6392	0.6057
fine_all	ASC_to_VAL	EH	5-3-2	0.004	0.6	140	0.614731825	0.5333	0.2994	0.441	0.7114	0.5718	0.5054	0.3519
fine_all	ASC_to_VAL	SH	5-3-2	0.004	0.6	140	0.614731825	0.5585	0.4656	0.482	0.6239	0.6338	0.5447	0.4241
fine_all	ASC_to_VAL	SE	5-3-2	0.004	0.6	140	0.614731825	0.5612	0.6643	0.5	0.4838	0.6442	0.574	0.5612
fine_all	GEN_to_ASC	EH	10-3-2	0.0005	1	50	0.58749778	0.5554	0.4605	0.512	0.6348	0.5858	0.5477	0.4827
fine_all	GEN_to_ASC	SH	10-3-2	0.0005	1	50	0.58749778	0.5111	0.4079	0.449	0.5944	0.5488	0.5011	0.4102
fine_all	GEN_to_ASC	SE	10-3-2	0.0005	1	50	0.58749778	0.6412	0.3904	0.51	0.7684	0.6921	0.5794	0.4092
fine_all	GEN_to_ARO	EH	10-3-2	0.0005	0.5	180	0.706046786	0.5863	0.7078	0.49	0.5046	0.7206	0.6062	0.5788
fine_all	GEN_to_ARO	SH	10-3-2	0.0005	0.5	180	0.706046786	0.7721	0.1153	0.66	0.9815	0.777	0.5484	0.191
fine_all	GEN_to_ARO	SE	10-3-2	0.0005	0.5	180	0.706046786	0.6157	0.7464	0.51	0.5303	0.7628	0.6384	0.6053
fine_all	GEN_to_VAL	EH	10-3-2	0.001	0.6	80	0.625873981	0.5222	0.38	0.439	0.6304	0.5651	0.5052	0.3862
fine_all	GEN_to_VAL	SH	10-3-2	0.001	0.6	80	0.625873981	0.5281	0.5783	0.47	0.4927	0.5873	0.5355	0.4743

fine_all	GEN_to_VAL	SE	10-3-2	0.001	0.6	80	0.625873981	0.544	0.6956	0.491	0.4301	0.6241	0.5629	0.5607
fine_all	ARO_to_ASC	EH	5-3-2	0.002	0.5	70	0.601499265	0.5846	0.5289	0.545	0.6312	0.6193	0.5801	0.5322
fine_all	ARO_to_ASC	SH	5-3-2	0.002	0.5	70	0.601499265	0.5308	0.3332	0.471	0.6901	0.5559	0.5117	0.3616
fine_all	ARO_to_ASC	SE	5-3-2	0.002	0.5	70	0.601499265	0.6836	0.3472	0.559	0.8541	0.721	0.6007	0.4222
fine_all	ARO_to_GEN	EH	5-3-2	0.002	0.8	90	0.62569297	0.4691	0.3824	0.819	0.7395	0.2787	0.561	0.5178
fine_all	ARO_to_GEN	SH	5-3-2	0.002	0.8	90	0.62569297	0.5548	0.5691	0.615	0.5364	0.492	0.5528	0.5878
fine_all	ARO_to_GEN	SE	5-3-2	0.002	0.8	90	0.62569297	0.5654	0.5981	0.635	0.5199	0.4827	0.559	0.6141
fine_all	ARO_to_VAL	EH	5-3-2	0.001	0.8	60	0.623630538	0.5362	0.3317	0.449	0.6918	0.5767	0.5118	0.3805
fine_all	ARO_to_VAL	SH	5-3-2	0.001	0.8	60	0.623630538	0.5653	0.5146	0.484	0.601	0.6466	0.5578	0.4821
fine_all	ARO_to_VAL	SE	5-3-2	0.001	0.8	60	0.623630538	0.5766	0.6269	0.506	0.5387	0.6575	0.5828	0.5594
fine_all	VAL_to_ASC	EH	10-3-2	0.002	1	170	0.59262765	0.6036	0.5497	0.567	0.6486	0.6331	0.5992	0.5574
fine_all	VAL_to_ASC	SH	10-3-2	0.002	1	170	0.59262765	0.525	0.2651	0.446	0.7345	0.5535	0.4998	0.3306
fine_all	VAL_to_ASC	SE	10-3-2	0.002	1	170	0.59262765	0.6771	0.3315	0.535	0.8524	0.7161	0.5919	0.4053
fine_all	VAL_to_GEN	EH	10-3-2	0.0005	0.8	230	0.633350505	0.4134	0.2953	0.807	0.7819	0.263	0.5386	0.4285
fine_all	VAL_to_GEN	SH	10-3-2	0.0005	0.8	230	0.633350505	0.5622	0.6142	0.611	0.4956	0.4994	0.5549	0.6112
fine_all	VAL_to_GEN	SE	10-3-2	0.0005	0.8	230	0.633350505	0.5641	0.649	0.621	0.4457	0.4771	0.5474	0.6335
fine_all	VAL_to_ARO	EH	10-3-2	0.0005	0.6	110	0.716018342	0.5712	0.7488	0.479	0.4518	0.7287	0.6003	0.5838
fine_all	VAL_to_ARO	SH	10-3-2	0.0005	0.6	110	0.716018342	0.7698	0.1605	0.615	0.964	0.7838	0.5622	0.2336
fine_all	VAL_to_ARO	SE	10-3-2	0.0005	0.6	110	0.716018342	0.627	0.7278	0.528	0.5612	0.763	0.6445	0.6052
fine_last	ASC_to_GEN	EH	5-3-2	0.004	1	120	0.542230602	0.4385	0.3821	0.748	0.6147	0.2467	0.4984	0.478
fine_last	ASC_to_GEN	SH	5-3-2	0.004	1	120	0.542230602	0.5021	0.4024	0.6	0.6298	0.4248	0.5161	0.4579
fine_last	ASC_to_GEN	SE	5-3-2	0.004	1	120	0.542230602	0.4902	0.4812	0.561	0.5027	0.4165	0.4919	0.489
fine_last	ASC_to_ARO	EH	5-3-2	0.006	0.8	60	0.555904531	0.4831	0.7584	0.424	0.2982	0.6335	0.5283	0.5318
fine_last	ASC_to_ARO	SH	5-3-2	0.006	0.8	60	0.555904531	0.5548	0.5137	0.291	0.5679	0.7946	0.5408	0.3392
fine_last	ASC_to_ARO	SE	5-3-2	0.006	0.8	60	0.555904531	0.5142	0.6751	0.44	0.4091	0.6543	0.5421	0.5213
fine_last	ASC_to_VAL	EH	5-3-2	0.001	1	200	0.524825193	0.4914	0.6121	0.436	0.3995	0.5697	0.5058	0.4944
fine_last	ASC_to_VAL	SH	5-3-2	0.001	1	200	0.524825193	0.4815	0.6406	0.418	0.3694	0.5753	0.505	0.4764
fine_last	ASC_to_VAL	SE	5-3-2	0.001	1	200	0.524825193	0.5095	0.63	0.442	0.4189	0.6021	0.5244	0.4918
fine_last	GEN_to_ASC	EH	10-3-2	0.006	1	40	0.52975802	0.489	0.6193	0.445	0.3801	0.5394	0.4997	0.4791
fine_last	GEN_to_ASC	SH	10-3-2	0.006	1	40	0.52975802	0.4977	0.6769	0.455	0.3532	0.596	0.515	0.533
fine_last	GEN_to_ASC	SE	10-3-2	0.006	1	40	0.52975802	0.5155	0.4882	0.36	0.5293	0.6354	0.5088	0.3757
fine_last	GEN_to_ARO	EH	10-3-2	0.002	0.8	100	0.571098578	0.4754	0.7263	0.416	0.3069	0.6248	0.5166	0.5203
fine_last	GEN_to_ARO	SH	10-3-2	0.002	0.8	100	0.571098578	0.4947	0.671	0.277	0.4386	0.8146	0.5548	0.3864

fine_last	GEN_to_ARO	SE	10-3-2	0.002	0.8	100	0.571098578	0.5743	0.6565	0.482	0.5206	0.7034	0.5885	0.5462
fine_last	GEN_to_VAL	EH	10-3-2	0.002	0.8	160	0.524508139	0.4698	0.706	0.431	0.2901	0.5552	0.4981	0.5298
fine_last	GEN_to_VAL	SH	10-3-2	0.002	0.8	160	0.524508139	0.5158	0.6736	0.464	0.4047	0.6143	0.5391	0.5085
fine_last	GEN_to_VAL	SE	10-3-2	0.002	0.8	160	0.524508139	0.511	0.6542	0.457	0.4033	0.597	0.5287	0.5212
fine_last	ARO_to_ASC	EH	5-3-2	0.001	1	120	0.523644481	0.5048	0.4705	0.444	0.5335	0.5583	0.502	0.4373
fine_last	ARO_to_ASC	SH	5-3-2	0.001	1	120	0.523644481	0.5066	0.6105	0.458	0.4229	0.6036	0.5167	0.5093
fine_last	ARO_to_ASC	SE	5-3-2	0.001	1	120	0.523644481	0.49	0.5079	0.342	0.4809	0.6548	0.4944	0.3893
fine_last	ARO_to_GEN	EH	5-3-2	0.006	1	40	0.517180093	0.4528	0.4187	0.768	0.5593	0.2258	0.489	0.4845
fine_last	ARO_to_GEN	SH	5-3-2	0.006	1	40	0.517180093	0.5207	0.5337	0.59	0.504	0.4386	0.5188	0.5329
fine_last	ARO_to_GEN	SE	5-3-2	0.006	1	40	0.517180093	0.4972	0.4393	0.581	0.5778	0.4377	0.5086	0.4798
fine_last	ARO_to_VAL	EH	5-3-2	0.004	0.8	210	0.52854824	0.4874	0.6594	0.439	0.3565	0.5776	0.5079	0.5112
fine_last	ARO_to_VAL	SH	5-3-2	0.004	0.8	210	0.52854824	0.4608	0.8189	0.423	0.2086	0.6028	0.5138	0.5518
fine_last	ARO_to_VAL	SE	5-3-2	0.004	0.8	210	0.52854824	0.4762	0.7753	0.452	0.2513	0.6242	0.5133	0.5348
fine_last	VAL_to_ASC	EH	10-3-2	0.002	1	130	0.529896631	0.4901	0.5352	0.461	0.4524	0.534	0.4938	0.4718
fine_last	VAL_to_ASC	SH	10-3-2	0.002	1	130	0.529896631	0.4626	0.6035	0.427	0.3489	0.4736	0.4762	0.479
fine_last	VAL_to_ASC	SE	10-3-2	0.002	1	130	0.529896631	0.5611	0.368	0.353	0.659	0.6753	0.5135	0.3308
fine_last	VAL_to_GEN	EH	10-3-2	0.001	0.6	150	0.535380522	0.6088	0.71	0.751	0.2932	0.4	0.5016	0.6768
fine_last	VAL_to_GEN	SH	10-3-2	0.001	0.6	150	0.535380522	0.5062	0.6503	0.537	0.3218	0.4097	0.4861	0.5648
fine_last	VAL_to_GEN	SE	10-3-2	0.001	0.6	150	0.535380522	0.4709	0.4523	0.519	0.4968	0.3987	0.4746	0.4177
fine_last	VAL_to_ARO	EH	10-3-2	0.006	1	50	0.583285778	0.4833	0.7744	0.424	0.2876	0.6556	0.531	0.5441
fine_last	VAL_to_ARO	SH	10-3-2	0.006	1	50	0.583285778	0.53	0.479	0.257	0.5463	0.7412	0.5127	0.2891
fine_last	VAL_to_ARO	SE	10-3-2	0.006	1	50	0.583285778	0.5134	0.6555	0.44	0.4206	0.6937	0.5381	0.495

A.4. Cross Culture Evaluation COMPARE

Type	Name	Language	Architecture	Learning Rate	Dropout	NumSteps	CrossValidation	Accuracy	TP_Rate	PPV	TN_Rate	NPV	UAR	F1_Score
simple	ASC	EH	5-3-2	0.002	0.5	115	0.692307692	0.5418	0.6551	0.496	0.4471	0.6157	0.5511	0.562
simple	ASC	SH	5-3-2	0.002	0.5	115	0.692307692	0.5058	0.5319	0.4524	0.4849	0.5655	0.5084	0.4845
simple	ASC	SE	5-3-2	0.002	0.5	115	0.692307692	0.6236	0.4028	0.4695	0.7356	0.6876	0.5692	0.4068
simple	GEN	EH	10-3-2	0.001	1	100	0.722222222	0.6354	0.685	0.8049	0.4808	0.3308	0.5829	0.7387
simple	GEN	SH	10-3-2	0.001	1	100	0.722222222	0.553	0.8396	0.5715	0.1862	0.4581	0.5129	0.678
simple	GEN	SE	10-3-2	0.001	1	100	0.722222222	0.587	0.7646	0.6179	0.3394	0.5103	0.552	0.6825
simple	ARO	EH	10-3-2	0.001	0.5	150	0.779347826	0.542	0.5894	0.4467	0.5101	0.6501	0.5498	0.5073
simple	ARO	SH	10-3-2	0.001	0.5	150	0.779347826	0.6411	0.479	0.3662	0.6928	0.7804	0.5859	0.3749
simple	ARO	SE	10-3-2	0.001	0.5	150	0.779347826	0.5987	0.411	0.4908	0.7213	0.6533	0.5661	0.4438
simple	VAL	EH	25-5-2	0.001	1	140	0.675308642	0.5402	0.3638	0.4564	0.6744	0.5834	0.5191	0.4014
simple	VAL	SH	25-5-2	0.001	1	140	0.675308642	0.5086	0.5491	0.4283	0.4801	0.6015	0.5146	0.4752
simple	VAL	SE	25-5-2	0.001	1	140	0.675308642	0.5652	0.5123	0.4931	0.605	0.6263	0.5587	0.4972
prog	ASC_to_GEN	EH	10-3-2	0.001	1	90	0.709722222	0.6565	0.7147	0.8102	0.4751	0.3513	0.5949	0.7577
prog	ASC_to_GEN	SH	10-3-2	0.001	1	90	0.709722222	0.5287	0.8073	0.5559	0.172	0.3976	0.4896	0.6569
prog	ASC_to_GEN	SE	10-3-2	0.001	1	90	0.709722222	0.5716	0.7653	0.6044	0.3018	0.4805	0.5335	0.6751
prog	ASC_to_ARO	EH	5-3-2	0.002	0.8	100	0.773913043	0.5469	0.5324	0.4472	0.5567	0.6414	0.5445	0.4804
prog	ASC_to_ARO	SH	5-3-2	0.002	0.8	100	0.773913043	0.668	0.4484	0.356	0.738	0.808	0.5932	0.3939
prog	ASC_to_ARO	SE	5-3-2	0.002	0.8	100	0.773913043	0.5839	0.3722	0.4696	0.7222	0.6384	0.5472	0.4103
prog	ASC_to_VAL	EH	10-3-2	0.001	0.8	100	0.665432099	0.5451	0.3771	0.4655	0.6729	0.5876	0.525	0.4145
prog	ASC_to_VAL	SH	10-3-2	0.001	0.8	100	0.665432099	0.5285	0.4642	0.4364	0.5738	0.6052	0.519	0.4386
prog	ASC_to_VAL	SE	10-3-2	0.001	0.8	100	0.665432099	0.5732	0.4749	0.5042	0.647	0.622	0.561	0.486
prog	GEN_to_ASC	EH	10-3-2	0.002	0.8	120	0.632692308	0.5595	0.6901	0.5146	0.4504	0.6392	0.5702	0.5854
prog	GEN_to_ASC	SH	10-3-2	0.002	0.8	120	0.632692308	0.4957	0.5677	0.4491	0.4377	0.5546	0.5027	0.4976
prog	GEN_to_ASC	SE	10-3-2	0.002	0.8	120	0.632692308	0.6531	0.2506	0.4802	0.8573	0.6929	0.5539	0.3248
prog	GEN_to_ARO	EH	5-3-2	0.001	0.8	190	0.758695652	0.5481	0.5915	0.4524	0.519	0.6548	0.5553	0.5119
prog	GEN_to_ARO	SH	5-3-2	0.001	0.8	190	0.758695652	0.6982	0.3694	0.3852	0.8031	0.8016	0.5862	0.3613
prog	GEN_to_ARO	SE	5-3-2	0.001	0.8	190	0.758695652	0.5856	0.4048	0.4712	0.7037	0.6451	0.5543	0.4328
prog	GEN_to_VAL	EH	5-3-2	0.001	0.8	160	0.67037037	0.5376	0.3975	0.4608	0.6442	0.5843	0.5208	0.421
prog	GEN_to_VAL	SH	5-3-2	0.001	0.8	160	0.67037037	0.5142	0.5887	0.4352	0.4618	0.6179	0.5252	0.4961
prog	GEN_to_VAL	SE	5-3-2	0.001	0.8	160	0.67037037	0.562	0.5132	0.4901	0.5987	0.6223	0.5559	0.4986
prog	ARO_to_ASC	EH	10-3-2	0.001	0.8	80	0.679807692	0.5473	0.6446	0.5014	0.466	0.6161	0.5553	0.562
prog	ARO_to_ASC	SH	10-3-2	0.001	0.8	80	0.679807692	0.4834	0.4821	0.4294	0.4845	0.5368	0.4833	0.447

prog	ARO_to_ASC	SE	10-3-2	0.001	0.8	80	0.679807692	0.6684	0.2742	0.5211	0.8684	0.7029	0.5713	0.3521
prog	ARO_to_GEN	EH	10-3-2	0.001	0.6	90	0.708333333	0.6599	0.7212	0.8108	0.4689	0.357	0.5951	0.7596
prog	ARO_to_GEN	SH	10-3-2	0.001	0.6	90	0.708333333	0.5357	0.7948	0.5639	0.204	0.4043	0.4994	0.6575
prog	ARO_to_GEN	SE	10-3-2	0.001	0.6	90	0.708333333	0.5924	0.7834	0.6192	0.3262	0.5211	0.5548	0.6906
prog	ARO_to_VAL	EH	25-5-2	0.001	0.6	210	0.715277778	0.5487	0.3806	0.474	0.6766	0.5909	0.5286	0.4123
prog	ARO_to_VAL	SH	25-5-2	0.001	0.6	210	0.715277778	0.5205	0.4269	0.4198	0.5864	0.5945	0.5066	0.407
prog	ARO_to_VAL	SE	25-5-2	0.001	0.6	210	0.715277778	0.5569	0.5291	0.49	0.5778	0.6205	0.5534	0.5011
prog	VAL_to_ASC	EH	5-3-2	0.001	0.8	50	0.650961538	0.5307	0.7027	0.4879	0.3869	0.6358	0.5448	0.5675
prog	VAL_to_ASC	SH	5-3-2	0.001	0.8	50	0.650961538	0.4706	0.5105	0.4151	0.4384	0.5277	0.4744	0.4461
prog	VAL_to_ASC	SE	5-3-2	0.001	0.8	50	0.650961538	0.6463	0.3051	0.4658	0.8194	0.6995	0.5622	0.3642
prog	VAL_to_GEN	EH	5-3-2	0.002	0.8	60	0.676388889	0.6455	0.7005	0.8091	0.474	0.3602	0.5873	0.7419
prog	VAL_to_GEN	SH	5-3-3	0.002	0.8	60	0.676388889	0.5394	0.8066	0.5628	0.1973	0.4477	0.502	0.6608
prog	VAL_to_GEN	SE	5-3-4	0.002	0.8	60	0.676388889	0.5826	0.813	0.6064	0.2615	0.5061	0.5373	0.6923
prog	VAL_to_ARO	EH	25-5-2	0.001	0.8	110	0.792391304	0.5366	0.5659	0.4402	0.517	0.6403	0.5414	0.494
prog	VAL_to_ARO	SH	25-5-2	0.001	0.8	110	0.792391304	0.7057	0.3161	0.3855	0.8298	0.793	0.573	0.3318
prog	VAL_to_ARO	SE	25-5-2	0.001	0.8	110	0.792391304	0.5936	0.3775	0.481	0.7347	0.6442	0.5561	0.4216
fine_all	ASC_to_GEN	EH	5-3-2	0.002	1	140	0.62904737	0.6126	0.6565	0.7972	0.4757	0.3112	0.5661	0.7162
fine_all	ASC_to_GEN	SH	5-3-2	0.002	1	140	0.62904737	0.5435	0.8052	0.5665	0.2084	0.4415	0.5068	0.6626
fine_all	ASC_to_GEN	SE	5-3-2	0.002	1	140	0.62904737	0.5773	0.7471	0.6128	0.3407	0.4914	0.5439	0.6724
fine_all	ASC_to_ARO	EH	5-3-2	0.0005	0.6	140	0.732955476	0.5405	0.5512	0.4419	0.5333	0.6402	0.5422	0.4888
fine_all	ASC_to_ARO	SH	5-3-2	0.0005	0.6	140	0.732955476	0.7088	0.3879	0.405	0.8111	0.807	0.5995	0.3861
fine_all	ASC_to_ARO	SE	5-3-2	0.0005	0.6	140	0.732955476	0.5907	0.3679	0.4781	0.7363	0.6416	0.5521	0.4106
fine_all	ASC_to_VAL	EH	5-3-2	0.002	0.8	80	0.640862308	0.5405	0.3181	0.454	0.7097	0.5787	0.5139	0.3633
fine_all	ASC_to_VAL	SH	5-3-2	0.002	0.8	80	0.640862308	0.5135	0.4858	0.4247	0.5329	0.5954	0.5094	0.4464
fine_all	ASC_to_VAL	SE	5-3-2	0.002	0.8	80	0.640862308	0.5609	0.533	0.4884	0.5818	0.625	0.5574	0.5084
fine_all	GEN_to_ASC	EH	10-3-2	0.001	0.8	80	0.651842671	0.5553	0.6355	0.5101	0.4882	0.6166	0.5619	0.5646
fine_all	GEN_to_ASC	SH	10-3-2	0.001	0.8	80	0.651842671	0.5135	0.4987	0.4581	0.5254	0.5664	0.512	0.4725
fine_all	GEN_to_ASC	SE	10-3-2	0.001	0.8	80	0.651842671	0.6512	0.3292	0.485	0.8145	0.7057	0.5719	0.3842
fine_all	GEN_to_ARO	EH	10-3-2	0.0005	0.8	90	0.726984738	0.5277	0.6089	0.4398	0.4732	0.6297	0.541	0.5045
fine_all	GEN_to_ARO	SH	10-3-2	0.0005	0.8	90	0.726984738	0.7154	0.3331	0.3942	0.8373	0.7991	0.5852	0.3471
fine_all	GEN_to_ARO	SE	10-3-2	0.0005	0.8	90	0.726984738	0.5934	0.3789	0.4817	0.7334	0.6443	0.5562	0.4225
fine_all	GEN_to_VAL	EH	10-3-2	0.002	0.8	60	0.645808925	0.5428	0.36	0.4603	0.6819	0.5854	0.5209	0.3951
fine_all	GEN_to_VAL	SH	10-3-2	0.002	0.8	60	0.645808925	0.5386	0.416	0.4392	0.6249	0.6033	0.5205	0.4246

fine_all	GEN_to_VAL	SE	10-3-2	0.002	0.8	60	0.645808925	0.5482	0.5652	0.4827	0.5354	0.6114	0.5503	0.5086
fine_all	ARO_to_ASC	EH	25-5-2	0.0005	0.6	90	0.648861416	0.537	0.7319	0.4973	0.3741	0.6146	0.553	0.588
fine_all	ARO_to_ASC	SH	25-5-2	0.0005	0.6	90	0.648861416	0.4347	0.4114	0.3779	0.4535	0.4858	0.4324	0.3917
fine_all	ARO_to_ASC	SE	25-5-2	0.0005	0.6	90	0.648861416	0.6722	0.3101	0.526	0.8558	0.71	0.583	0.3874
fine_all	ARO_to_GEN	EH	25-5-2	0.002	1	90	0.620464941	0.6517	0.7185	0.8011	0.4435	0.3387	0.581	0.7565
fine_all	ARO_to_GEN	SH	25-5-2	0.002	1	90	0.620464941	0.555	0.8069	0.575	0.2324	0.4711	0.5197	0.6703
fine_all	ARO_to_GEN	SE	25-5-2	0.002	1	90	0.620464941	0.5588	0.6903	0.6087	0.3756	0.4634	0.5329	0.6442
fine_all	ARO_to_VAL	EH	25-5-2	0.0005	0.8	130	0.647169787	0.545	0.3571	0.4641	0.6879	0.5853	0.5225	0.3995
fine_all	ARO_to_VAL	SH	25-5-2	0.0005	0.8	130	0.647169787	0.5376	0.4156	0.4372	0.6236	0.6029	0.5196	0.4231
fine_all	ARO_to_VAL	SE	25-5-2	0.0005	0.8	130	0.647169787	0.5679	0.5229	0.497	0.6017	0.6269	0.5623	0.5087
fine_all	VAL_to_ASC	EH	10-3-2	0.002	0.5	120	0.645298035	0.552	0.628	0.5058	0.4884	0.6203	0.5582	0.5547
fine_all	VAL_to_ASC	SH	10-3-2	0.002	0.5	120	0.645298035	0.4945	0.4607	0.4273	0.5218	0.56	0.4913	0.4309
fine_all	VAL_to_ASC	SE	10-3-2	0.002	0.5	120	0.645298035	0.6514	0.3298	0.4785	0.8145	0.7056	0.5722	0.3879
fine_all	VAL_to_GEN	EH	10-3-2	0.0005	0.6	130	0.628524251	0.6519	0.7083	0.8095	0.4757	0.346	0.592	0.7528
fine_all	VAL_to_GEN	SH	10-3-2	0.0005	0.6	130	0.628524251	0.5478	0.876	0.5623	0.1276	0.4527	0.5018	0.6845
fine_all	VAL_to_GEN	SE	10-3-2	0.0005	0.6	130	0.628524251	0.5843	0.7659	0.615	0.3312	0.5099	0.5486	0.6809
fine_all	VAL_to_ARO	EH	10-3-2	0.0005	0.8	110	0.72222086	0.5497	0.5549	0.4511	0.5461	0.6468	0.5505	0.4967
fine_all	VAL_to_ARO	SH	10-3-2	0.0005	0.8	110	0.72222086	0.7109	0.3355	0.3913	0.8306	0.7978	0.583	0.3509
fine_all	VAL_to_ARO	SE	10-3-2	0.0005	0.8	110	0.72222086	0.5885	0.3478	0.4687	0.7456	0.6372	0.5467	0.3974
fine_last	ASC_to_GEN	EH	5-3-2	0.004	1	140	0.555880953	0.597	0.6938	0.7514	0.2949	0.2898	0.4944	0.6911
fine_last	ASC_to_GEN	SH	5-3-2	0.004	1	140	0.555880953	0.5105	0.6021	0.5561	0.3933	0.4667	0.4977	0.5316
fine_last	ASC_to_GEN	SE	5-3-2	0.004	1	140	0.555880953	0.5095	0.5396	0.5785	0.4674	0.4265	0.5035	0.5352
fine_last	ASC_to_ARO	EH	5-3-2	0.004	1	60	0.535306523	0.4679	0.7126	0.4067	0.3034	0.6136	0.508	0.5066
fine_last	ASC_to_ARO	SH	5-3-2	0.004	1	60	0.535306523	0.4959	0.5411	0.2476	0.4815	0.7742	0.5113	0.3144
fine_last	ASC_to_ARO	SE	5-3-2	0.004	1	60	0.535306523	0.4911	0.7086	0.4214	0.3491	0.6351	0.5288	0.523
fine_last	ASC_to_VAL	EH	5-3-2	0.001	1	50	0.50672148	0.4661	0.7381	0.4271	0.2592	0.5561	0.4986	0.523
fine_last	ASC_to_VAL	SH	5-3-2	0.001	1	50	0.50672148	0.4655	0.733	0.4209	0.2771	0.5225	0.505	0.5079
fine_last	ASC_to_VAL	SE	5-3-2	0.001	1	50	0.50672148	0.4703	0.8009	0.4369	0.2219	0.5802	0.5114	0.5569
fine_last	GEN_to_ASC	EH	10-3-2	0.004	0.8	60	0.507787421	0.4569	0.9102	0.4485	0.0778	0.5534	0.494	0.5944
fine_last	GEN_to_ASC	SH	10-3-2	0.004	0.8	60	0.507787421	0.4552	0.9568	0.4491	0.0507	0.5796	0.5037	0.6091
fine_last	GEN_to_ASC	SE	10-3-2	0.004	0.8	60	0.507787421	0.3577	0.9433	0.3377	0.0607	0.5518	0.502	0.4963
fine_last	GEN_to_ARO	EH	10-3-2	0.0005	1	190	0.53583981	0.4606	0.6962	0.4028	0.3023	0.5898	0.4993	0.4953
fine_last	GEN_to_ARO	SH	10-3-2	0.0005	1	190	0.53583981	0.3735	0.7718	0.2549	0.2465	0.6327	0.5092	0.3649

fine_last	GEN_to_ARO	SE	10-3-2	0.0005	1	190	0.53583981	0.4696	0.7512	0.4128	0.2856	0.6386	0.5184	0.5208
fine_last	GEN_to_VAL	EH	10-3-2	0.004	1	110	0.507444196	0.4527	0.7851	0.4235	0.1998	0.5308	0.4924	0.5392
fine_last	GEN_to_VAL	SH	10-3-2	0.004	1	110	0.507444196	0.4532	0.7689	0.415	0.2309	0.553	0.4999	0.5328
fine_last	GEN_to_VAL	SE	10-3-2	0.004	1	110	0.507444196	0.4866	0.7004	0.4416	0.3258	0.5989	0.5131	0.525
fine_last	ARO_to_ASC	EH	25-5-2	0.004	1	40	0.533264721	0.4953	0.7611	0.4683	0.273	0.5722	0.5171	0.5714
fine_last	ARO_to_ASC	SH	25-5-2	0.004	1	40	0.533264721	0.4819	0.7485	0.4533	0.2669	0.5418	0.5077	0.5452
fine_last	ARO_to_ASC	SE	25-5-2	0.004	1	40	0.533264721	0.4331	0.6978	0.3397	0.2989	0.6856	0.4983	0.4395
fine_last	ARO_to_GEN	EH	25-5-2	0.0005	0.8	180	0.521741893	0.6328	0.7627	0.7516	0.2277	0.3182	0.4952	0.7274
fine_last	ARO_to_GEN	SH	25-5-2	0.0005	0.8	180	0.521741893	0.524	0.8451	0.5477	0.1129	0.4089	0.479	0.6627
fine_last	ARO_to_GEN	SE	25-5-2	0.0005	0.8	180	0.521741893	0.511	0.588	0.5841	0.4036	0.4259	0.4958	0.5503
fine_last	ARO_to_VAL	EH	25-5-2	0.004	1	210	0.517354982	0.4556	0.7619	0.4262	0.2225	0.5485	0.4922	0.5431
fine_last	ARO_to_VAL	SH	25-5-2	0.004	1	210	0.517354982	0.4503	0.7925	0.4196	0.2093	0.5631	0.5009	0.5108
fine_last	ARO_to_VAL	SE	25-5-2	0.004	1	210	0.517354982	0.4735	0.7507	0.4361	0.2652	0.5733	0.5079	0.5395
fine_last	VAL_to_ASC	EH	10-3-2	0.001	1	100	0.533813062	0.4722	0.7509	0.4508	0.239	0.5413	0.495	0.5546
fine_last	VAL_to_ASC	SH	10-3-2	0.001	1	100	0.533813062	0.5023	0.7926	0.4725	0.2683	0.6342	0.5304	0.58
fine_last	VAL_to_ASC	SE	10-3-2	0.001	1	100	0.533813062	0.406	0.8124	0.3413	0.2	0.6361	0.5062	0.4766
fine_last	VAL_to_GEN	EH	10-3-2	0.004	1	260	0.512240331	0.6405	0.7708	0.76	0.2339	0.2807	0.5024	0.7424
fine_last	VAL_to_GEN	SH	10-3-2	0.004	1	260	0.512240331	0.5552	0.908	0.5636	0.1036	0.4716	0.5058	0.6931
fine_last	VAL_to_GEN	SE	10-3-2	0.004	1	260	0.512240331	0.531	0.6977	0.5791	0.2986	0.4062	0.4982	0.621
fine_last	VAL_to_ARO	EH	10-3-2	0.0005	1	210	0.530949758	0.4252	0.8751	0.4012	0.1229	0.629	0.499	0.5474
fine_last	VAL_to_ARO	SH	10-3-2	0.0005	1	210	0.530949758	0.3542	0.7976	0.2449	0.2129	0.7516	0.5052	0.3666
fine_last	VAL_to_ARO	SE	10-3-2	0.0005	1	210	0.530949758	0.4588	0.8325	0.4113	0.2147	0.6458	0.5236	0.5484

B. DE-ENIGMA Results

Arousal

Fold	CCC	PCC	SCC	RMSE
0	0.079684149	0.1224152	0.0968399	0.1340024
1	0.223786720	0.3078807	0.2522965	0.1426455
2	0.090077962	0.1005266	0.0854629	0.1246366
3	0.103803785	0.1368618	0.1243721	0.1404913
MEAN	0.124338154	0.166921086	0.13974285	0.135443912
STD	0.05805045	0.08240477	0.06650484	0.00700384

Arousal Without Shift

Fold	CCC	PCC	SCC	RMSE
0	0.0782428	0.11213433	0.07216467	0.13679138
1	0.23964909	0.39967106	0.32776973	0.14546141
2	0.109658	0.12702692	0.10278985	0.13540479
3	0.1445857	0.16666765	0.12439811	0.14720563
Mean	0.1430339	0.20137499	0.15678059	0.1412158
STD	0.06051599	0.11620829	0.10044994	0.005178

MTL Arousal

Fold	CCC	PCC	SCC	RMSE
0	0.10896756	0.11927482	0.10478461	0.14750595
1	0.11259405	0.12577725	0.1288383	0.16539678
2	0.10107129	0.11411867	0.06243591	0.13221983
3	0.07965641	0.10093556	0.10851796	0.14143949
MEAN	0.10057233	0.11502657	0.1011442	0.14664051
STD	0.01277425	0.00912422	0.02415016	0.01211975

Cross Task Arousal

Fold	CCC	PCC	SCC	RMSE
0	0.06773512	0.09989051	0.10258186	0.13384316
1	0.2062165	0.33589344	0.23978443	0.14218496
2	0.13438099	0.16413896	0.1366935	0.11391544
3	0.06830064	0.09925932	0.09508561	0.13716662
MEAN	0.11915831	0.17479556	0.14353635	0.13177754
STD	0.05710014	0.09667288	0.0577392	0.0107317

Cross Corpus Arousal

Fold	CCC	PCC	SCC	RMSE
0	0.07012447	0.08547147	0.04948351	0.14061563
1	0.17017678	0.27138848	0.18644332	0.14430831
2	0.11166626	0.14085831	0.11953615	0.12225021
3	0.05162204	0.0793174	0.0624895	0.13469764
MEAN	0.10089739	0.14425891	0.10448812	0.13546795
STD	0.04552646	0.07721213	0.05415831	0.00836589

Valence

Fold	CCC	PCC	SCC	RMSE
0	0.09133145	0.10134004	0.09262683	0.13708853
1	0.05093226	0.07378375	0.07757588	0.1944619
2	0.05223241	0.07061253	0.02907353	0.15221792

	3	0.07684193	0.11739091	0.12391459	0.14371823
MEAN		0.06783451	0.09078181	0.08079771	0.15687165
STD		0.01704664	0.01946313	0.03422267	0.02235547

MTL Valence

Fold	CCC	PCC	SCC	RMSE
0	0.1783881	0.19419063	0.2127857	0.14067897
1	0.10073832	0.12844888	0.09101662	0.19721748
2	0.05791173	0.07183978	0.04496853	0.15834837
3	0.07822742	0.10158925	0.09766346	0.14374956
MEAN	0.10381639	0.12401713	0.11160858	0.1599986
STD	0.04564112	0.04519249	0.06183891	0.02250138

Cross Task Valence

Fold	CCC	PCC	SCC	RMSE
0	0.15597601	0.21396704	0.21372777	0.1256565
1	0.03347662	0.05924155	0.08513031	0.1990636
2	0.04129269	0.05893529	0.06275309	0.1514771
3	0.05378545	0.08097593	0.08375093	0.14297846
MEAN	0.07113269	0.10327995	0.11134052	0.15479392
STD	0.04951697	0.06452699	0.05977468	0.02720018

Cross Corpus Valence

Fold	CCC	PCC	SCC	RMSE
0	0.14530578	0.19293533	0.203106	0.13486043
1	0.01024449	0.01744611	0.10788125	0.19732379
2	0.08993949	0.13582092	0.15786839	0.14738937
3	0.00450055	0.00852315	0.01232798	0.14401747
MEAN	0.06249757	0.08868138	0.12029591	0.15589776
STD				