UNSW
AUSTRALIA

SCHOOL OF ELECTRICAL ENGINEERING
AND TELECOMMUNICATIONS

Exploring Transfer Learning for Arabic Dialect Identification

by

Madeline Younes

Student ID: z5208494

Thesis submitted as a requirement for the degree
Bachelor of Engineering (Electrical Engineering)

Submitted: November 21, 2022
Supervisor: Dr Beena Ahmed

Abstract

This thesis presents a novel approach to Dialectal Identification (DID) through leveraging pre-trained speech recognition models for low resource dialects, particularly Arabic. It focuses on designing an umbrella DID which is able to classify four umbrella Arabic dialects, and then it's adaptability to a regional DID with seventeen dialects. Its key aims are to assess the minimum amount of data required to create a viable system through training with varying utterance lengths and amounts of training data. As well as investigate how different design or fine-tuning techniques such as unfreezing encoder layers and inserting downstream layers may improve the performance of the DID. The performance of Arabic DIDs have been limited by the scarcity of available datasets and the linguistic similarities between the dialects. Thereby, there is a need for a system that has low data requirements and is able to take advantage of the existing systems trained on higher resource languages. The method proposed explores wav2vec 2.0 and it's variants as pre-trained models which can be fine-tuned for the Arabic DID downstream task. It investigated the effectiveness of inserting downstream models and other fine-tuning techniques to improve the system's performance. Results found that using XLSR Arabic which was fine-tuned with unfrozen encoder layers at step 50 with 10s utterances produced the highest performance umbrella DID with a weighted average F1-Score of 76%. When adapted to a finer grain task of regional dialectal identification, the DID had a weighted average F1-Score of 58%. Compared to traditional machine learning's accuracy of 85.1% the transfer learning DID was not on par. Although, for some dialects compared to the phonetic approach it produced higher performance results with 10s utterances. With a 96% for Iraqi (IQ) compared to phonemic model's 56% and for Levantine (LEV) Arabic the system's F1-Scores were 81% and 78% respectively. Further work is encouraged on building an Arabic DID using transfer learning techniques as the method shows promise.

Topic Title: Exploring Transfer Learning for Arabic Dialect Identification

Student Name: Madeline Younes **Student ID:** z5208494

A. **Problem statement**

Although voice enabled technology is used globally advancements have not benefited all languages equally. With more diverse languages with several dialects limited in advancements due to data scarcity and the complexity that comes from shared features between the dialects. One language that this problem has become apparent is Arabic. Dialectal Identifiers enable more accurate systems in speech recognition and transcription by allowing a speech model tuned for a particular dialect to be selected at the start of the process. In order to build a high performance Arabic Dialectal Identifier, desirable methods should require low amounts of data and leverage existing speech models. However, no work has been done using this modern method of transfer learning and so there is a need to apply contemporary techniques to Arabic Dialectical Identifiers.

B. **Objective**

Implement an Arabic Dialectal Identifier for both four umbrella dialects and its adaptability to a finer grain task of identifying seventeen regional dialects using transfer learning techniques. Assess the performance of using various pretrained models, utterance lengths, amount of training files, the effect of inserting downstream models and finetuning the encoder layers within the pretrained model. Analyse the results, provide conclusions and next steps.

C. **My solution**

Arabic Dialectal Identifier using XLSR Arabic pretrained model, finetuning the encoder layers within the pretrained model.
Fine tuning with dialectal Arabic grouped into four umbrella dialects and then seventeen regional dialects.
Explored inserting DNN and LSTM downstream models into the model structure.

D. **Contributions** (at most one per line, most important first)

Novel application of modern speech representation architecture.
Novel downstream task using pretrained models.
Detailed assessment of the preposed model's performance for both broader and more fine grained Arabic dialect groups.

E. **Suggestions for future work**

Train & test with datasets that contain codeswitching between dialects and languages.
Apply methods to use contextual information to improve performance.
Explore inserting other downstream models architectures into the model.

While I may have benefited from discussion with other people, I certify that this report is entirely my own work, except where appropriately documented acknowledgements are included.

Signature: __*Madeline Younes*_____ Date: 20 /11 / 2022

Pointers

List relevant page numbers in the column on the left. Be precise and selective: Don't list all pages of your report!

5	Problem Statement
5	Objective

Theory (up to 5 most relevant ideas)

7	Introduction into Dialectical Identification
9	Pretrained models
11	Context Network: Transformer Encoder-Decoders
12	Introduction into Dialectal Arabic

Method of solution (up to 5 most relevant points)

23-25	Preposed architecture
25-28	Implementation of architecture
21	Dataset Design

Contributions (most important first)

25	Novel application of modern speech representation architecture.
29	Novel downstream task using pretrained models.
28-40	Detailed assessment of the preposed model's performance for both broader and more fine grained Arabic dialect groups.
41	Summary of key results of novel application

My work

7-9	Description of key metrics
23-25	Preposed architecture
41	Summary of results

Results

41	Summary of key results of novel application

Conclusion

43	Statement of insights gained through the experiments
43	Suggestions for future research

Literature: (up to 5 most important references)

18	[11] Babu et al. 2021
16	[17] Biadsy et al. 2009
15	[38] Lin et al. 2020
20	[59] Shon et al. 2018

Acknowledgements

I wish to thank Dr Beena Ahmed for her guidance and encouragement she has provided throughout this thesis. I would also like to thank Iain McCowan from Dubber AI for introducing Language and Dialectal Identification as a novel use case for transfer learning. A thank you to Jack Murray and Erin Moss who also completed their theses in partnership with Dubber AI for their support. I am also particularly grateful for the constant love and encouragement from my friends and family throughout the journey of completing this thesis.

Abbreviations

BE Bachelor of Engineering

EE&T School of Electrical Engineering and Telecommunications

LaTeX A document preparation computer program

CNN Convolutional Neural Network

BiLSTM Bidirectional Long Short-Term Memory

LSTM Long Short-Term Memory

DNN Deep Neural Network

RNN Recurrent Neural Network

LID Language Identification

DID Dialectal Identification

ASR Automatic Speech Recognition

NLP Natural Language Processing

E2E End to End

wav2vec wav2vec-base

XLSR wav2vec-large-xlsr-53

XLSR wav2vec-large-xlsr-53-arabic

MSA Modern Standard Arabic

NOR North African Arabic

EGY Egyptian Arabic

LEV Levantine Arabic

GLF Gulf Arabic

Contents

Acknowledgements **3**

Abbreviations **4**

Contents **5**

1 Introduction **7**
1.1 Problem Statement . . . 7
1.2 Thesis Aims . . . 8
1.3 Chapter Outline . . . 8

2 Background: The Fundamentals of Dialect Identification and An Introduction into Dialectal Arabic **9**
2.1 Introduction into Language and Dialect Identifiers . . . 9
2.1.1 Performance Metrics . . . 9
2.2 Introduction into Transfer Learning . . . 11
2.2.1 Pretrained Models . . . 11
2.2.2 Feature Extraction . . . 12
2.2.3 Quantisation . . . 12
2.2.4 Context Network: Transformer Encoder-Decoders . . . 12
2.3 Introduction to Dialectal Arabic . . . 14

3 Literature Review **16**
3.1 An Introduction to LID and DID systems . . . 16
3.2 Traditional Methodologies . . . 17
3.2.1 Phonematic Modelling . . . 17
3.2.2 Traditional Machine Learning . . . 18
3.3 Transfer Learning . . . 18
3.3.1 Pretrained Models . . . 20

4 Methodologies: Design and Implementations **22**
4.1 Dataset . . . 22
4.2 Implementation of Arabic DID System . . . 24
4.2.1 Overall System Design . . . 24
4.2.2 Pre-training . . . 26
4.2.3 Fine-tuning . . . 26
4.2.4 Downstream Network . . . 26

5 Experiments, Results and Analysis **29**
5.1 Preliminary experimentation . . . 29
5.2 Fine-Tuning an Umbrella Arabic DID . . . 30
5.2.1 Assessment of Performance of Pretrained Models . . . 30
5.2.2 Exploring Noise filtering . . . 32
5.2.3 Effect of Utterance Length . . . 34
5.2.4 Fine-tuning with addition of a Downstream Model . . . 35
5.2.5 Fine-tuning with Encoder layers . . . 37
5.3 Counteracting Data Bias . . . 39
5.4 Assessing Robustness . . . 40
5.4.1 Amount of Training Data . . . 40
5.4.2 Adaptability to Regional DID . . . 41
5.5 Summary of Results . . . 42

6 Conclusion **44**
6.1 Future Work . . . 44
6.2 Broader Impact . . . 44

Bibliography **46**

A Appendices **52**
A.1 Appendix A - Additional Fine-Tuning Results . . . 52

Chapter 1

Introduction

1.1 Problem Statement

Although voice enabled technology is used globally advancements have not benefited all languages equally. With languages other than English that have a diverse range of dialects limited in advancements due to data scarcity and the complexity that comes from shared features between the dialects. One language that this problem has become apparent is Arabic. In multicultural societies such as that in Sydney, NSW it is common to have speakers of different dialects interact. There are around 200 thousand Arabic speakers within NSW, majority speaking Levantine dialect of Arabic with then a spread among Egyptian and Gulf Arabic. It would be a common scenario where perhaps for a telehealth consultation the only Arabic speaker available doesn't speak the same dialect as a patient. The speaker may then code switch with the words, phrases etc. they know from one dialect to their main dialect to hold a conversation. Having a system that is able to accurately identify the dialects spoken in certain segments to then produce an accurate transcription would certainly be helpful in that scenario. Dialectal Arabic generally has very limited available datasets, with no commercially available datasets and therefore is considered low resource. Current methods use phoneme recognition or traditional machine learning but both these methods have flaws that limit their ability to reliably recognise Arabic dialects. As phoneme identifiers rely on phonemic differences between the dialects, where there are shared phonemes distinguishing the dialects become a difficult task. While traditional machine learning requires large amounts of labelled training data which is currently unavailable for Arabic dialects. In most voice enabled technology the first step in selecting an automated speech recognition model (ASR) is to first identify its language or dialect. Dialectal Identifiers enable more accurate systems in speech recognition and transcription by allowing a speech models tuned for a particular dialect to be selected at the start of this process. In order to build a high performance Arabic dialectal identifier (DID) desirable methods should require low amounts of data and leverage existing speech models. However, no work has been done using this modern method of transfer learning and so there is a need to apply contemporary techniques to Arabic dialectical identifiers (DID). This thesis will investigate transfer learning as an approach in designing more accurate and reliable Arabic DIDs. The dialectal identifier should accurately distinguish the four umbrella Arabic dialects, and then it's robustness tested by extending it to a finer grain downstream task of distinguishing seventeen regional dialects.

1.2 Thesis Aims

The goal of this thesis is to answer the overarching question: *Can transfer learning be leveraged to be used to improve the performance of the low resource task of Arabic Dialectal Identification?*

The key aims of this thesis are:

- To assess the viability of using transfer learning to improve the accuracy of Arabic Dialectal Identification.
- To investigate the performance of a transfer learning based DID on low resource dialects. Through assessing the minimum amount of data needed to accurately fine tune a DID system. This is in terms of utterance length of the audio files and the amount of files provided.
- To critically analyse which pretrained model and downstream model architecture is able to produce the most accurate DID system.
- To assess the effect of unfreezing and fine-tuning the encoder layers within the pretrained model on the performance of the DID system.
- To explore whether a transfer learning based DID system can be adapted to be accurately applied to a finer set of dialectal groups.

1.3 Chapter Outline

This report is organised as described.

- Chapter 2 details some background information surrounding the impacts of this thesis and describes the unique features of Arabic Dialects.
- Chapter 3 provides a detailed analysis of current LID and DID methodologies. As well as details the literature which supports the use of transfer learning for the application of LID and DID.
- Chapter 4 proposes a methodology for this thesis.
- Chapter 5 explores the experimentation conducted throughout this thesis and analyses the results.
- Chapter 6 draws up conclusions and summarises the key takeaways of the report. As well as providing suggestions for future work and the implications of this thesis on similar research.

Chapter 2

Background: The Fundamentals of Dialect Identification and An Introduction into Dialectal Arabic

This chapter aims to provide a brief explanation of the preliminary background theory that is needed to understand this thesis. An explanation of dialectal identifiers, the metrics that will be used, an introduction into transfer learning and an exploration into wav2vev's structure will be provided. As well as the end portion will provide some additional insight into dialectal Arabic.

2.1 Introduction into Language and Dialect Identifiers

Language Identification (LID) and Dialectal Identification (DID) are specialised audio classifiers. Audio classification is defined as the process of analysing audio segments and then categorising them on a predefined set of labels. For LIDs and DIDs these labels correspond to the languages or dialects being identified. LID systems are the critical first step in selecting the most accurate model to use for Automatic Speech Recognition (ASR), multilingual transcription or other automated speech processing systems. A dialect is a sub-variant of a language which is usually mutually intelligible by other speakers of that language despite the speaker using a different dialect. Dialects evolve within a certain region, area or within a class. Dialect Identification (DID) generally poses some interesting challenges compared to language identification that semi-self supervised systems could provide solutions to. These challenges include that dialects unlike languages are not standardised, generally have very limited labelled datasets and so, are often considered low resource and the differences between different dialects are often not as clear as the differences between languages.

2.1.1 Performance Metrics

Accuracy

Accuracy is the most basic metric for evaluating the performance of classification models, showing the correct predictions to the total number predictions and is formally defined in the Equation 2.1. However, it is an unbalanced metric as the amount of files for each class is not taken into consideration.

$$Accuracy = \frac{TruePostive + FalsePostive}{TruePositive + FalsePostive + TrueNegative + FalseNegative} \tag{2.1}$$

Precision and Recall

Precision or Positive Predictive Value, is a metric that evaluates the amount of predicted outcomes are actually correct, providing an evaluation of relevant data points. It is formally written in the Equation 2.2.

$$Precision = \frac{TruePostive}{TruePositive + FalsePostive} \tag{2.2}$$

Recall, also known as Positive Rate and Sensitivity, takes into consideration the correctly predicted values over those of a specific class. As a metric it provides an assessment of the relevant data and is formally expressed in Equation 2.3.

$$Recall = \frac{TruePostive}{TruePositive + FalseNegative} \tag{2.3}$$

F1-Score

F1-Score, expressed in Equation 2.4 is the harmonic mean of Recall and Precision. It is the most relevant metric for this thesis as tuning the F1-Score allows for a balanced optimisation of both Recall and Precision metrics.

$$F1 - Score = \frac{Precision * Recall}{Precision + Recall} \tag{2.4}$$

Macro and Weighted Average

The macro and weighted averages of all the metrics are computed for this thesis. The macro average calculates the average of each metric without taking into consideration the amount of data for each class. While the weighted average takes into consideration the amount of data points. Weighted average is the provides the most accurate view of performance so is the primary metric used as an assessment in this thesis.

Confusion Matrix

The key plot that will be used to assess performance is a colour map plot of the confusion matrix of the test results. The confusion matrix summarises the number of correct predictions to incorrect predictions for each class. A colourmap is used to plot the matrix with the y-axis corresponding to the true values, the x-axis the predicted and the colour bar the number of files. An example outputted matrix is shown in the Table 2.1 and it's corresponding colour map is shown in Figure 2.1.

	Real Values				
Predicted Values		NOR	EGY	GLF	LEV
	NOR	79	2	12	7
	EGY	4	45	31	20
	GLF	0	3	93	2
	LEV	1	2	11	86

Table 2.1: Example Confusion Matrix

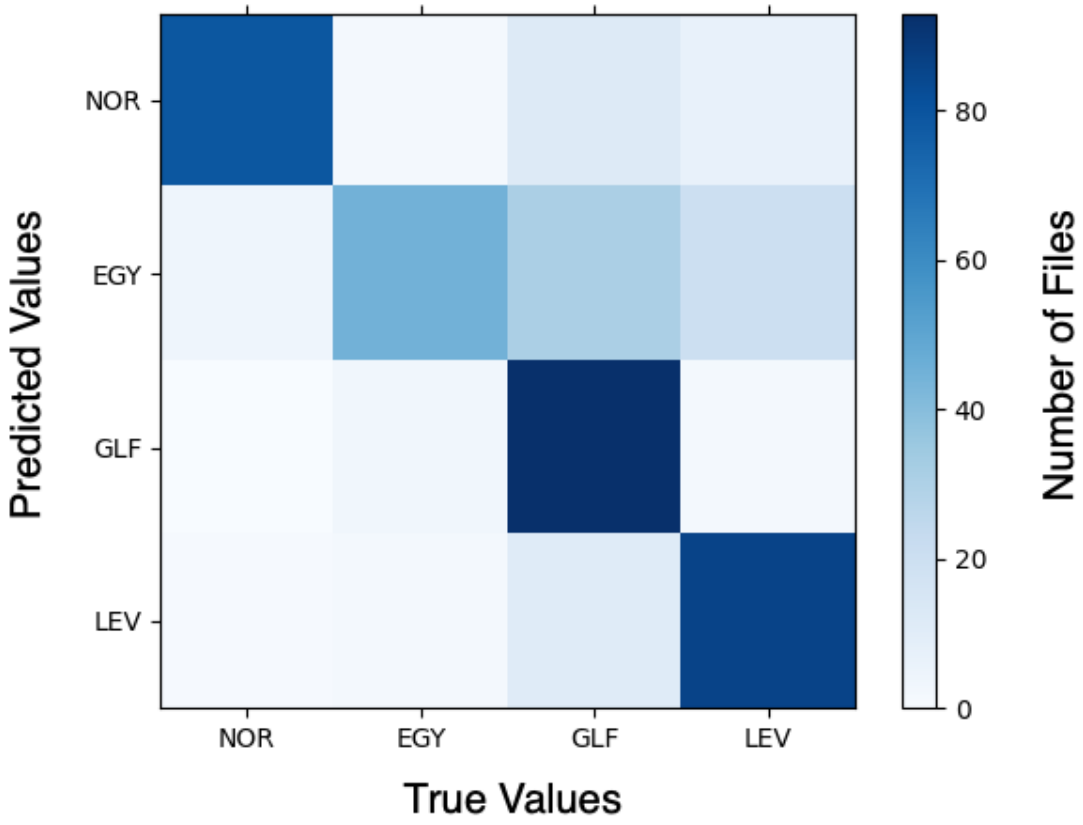


Figure 2.1: Example Colour Map Plot of Confusion Matrix

2.2 Introduction into Transfer Learning

Transfer Learning is an approach to machine learning systems that adapts a model trained to operate on one domain to function on another similar domain. The main advantage of this strategy is the initial model can be trained on a domain with large amounts of data using self supervision techniques. Then adapted for domains and use cases where there is more limited data available. In terms of this thesis transfer learning is applied through using a pretrained model designed for ASR that was trained using large amounts of speech data then finetuned it using a smaller dataset for the downstream task of Arabic Dialect Identification (DID).

2.2.1 Pretrained Models

Wav2Vec 2.0

Wav2Vec 2.0 [14] is a self-supervised transformer based neural network designed by Facebook with the aim of designing a system that could outperform semi-supervised methods with less complexity and then further adapted for other domains. It was trained using 53K hours of unlabeled and 10 mins of labelled English speech from the Librispeech Corpus. Its framework is shown in Figure 2.2. Wav2vec 2.0 was further developed to produce wav2vec XLSR. wav2vec XLSR [11] which used an identical structure to wav2vec 2.0 but was trained with data from multilingual speech corpuses to learn cross-lingual speech representations. It was trained on a total of 53 languages from three different corpuses CommonVoice, Babel and Multilingual

LibriSpeech (MLS). This thesis also tests the performance of wav2vec XLSR Arabic, a version of wav2vec XLSR finetuned using 7.5hrs of Male Voice Levantine Dialectal Arabic Speech specifically Syrian with a Damascus accent. Another fined tuned model of wav2vec 2.0 tested was wav2vec supberb sid which was finetuned for a speaker classification downstream task.

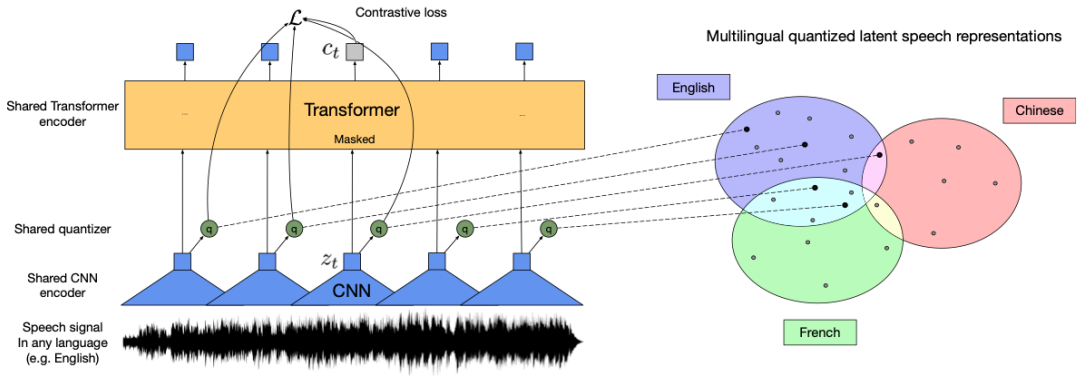


Figure 2.2: High level view of the Framework of XLSR [11].

2.2.2 Feature Extraction

The feature encoder is a 7-layer, 512 channel CNN that translates a waveform into feature vectors, reducing the dimension of the audio to 1D. It does this every 20ms and has a receptive field of 400 samples which is equivalent to 25ms of audio sampled at 16kHz. The feature extractor also normalises the audio before it is then forwarded into the network.

2.2.3 Quantisation

The key challenge of audio related speech data is that it is continuous. Written language can be naturally segmented into words, sentences and subwords while speech does not have this natural sub-unit. The quantisation module addresses the continuous nature of speech data, automatically learning discrete speech units such as phonemes and words. It does this through sampling from the Gumbel-Softmax distribution, the possible units are comprised of codewords which are grouped in codebooks. The speech unit is the concatenation of these codewords. In wav2vec 2.0 there are 2 groups with 320 words in each group and a theoretical maximum of 102400 speech units. The features are multiplied by the quantisation matrix, to produce logits and then converting those into probabilities of a codeword matching an exisitng codeword in the codebook using the Gumbel-Softmax. This module is illustrated in the Figure 2.3.

2.2.4 Context Network: Transformer Encoder-Decoders

At the core of wav2vec is a context network comprised of 12 transformer based encoders. The transformers have an attention based encoder-decoder architecture, the structure enables all frames of an audio clip to be processed simultaneously. The concept of a transformer based network was proposed in the paper ”Attention is All You Need” [66] which aimed to iterate upon the design of sequential Recurrent Neural Networks (RNNs) which could only process one frame of speech at a time. The architecture and flow of data is illustrated in the Figure 2.4.

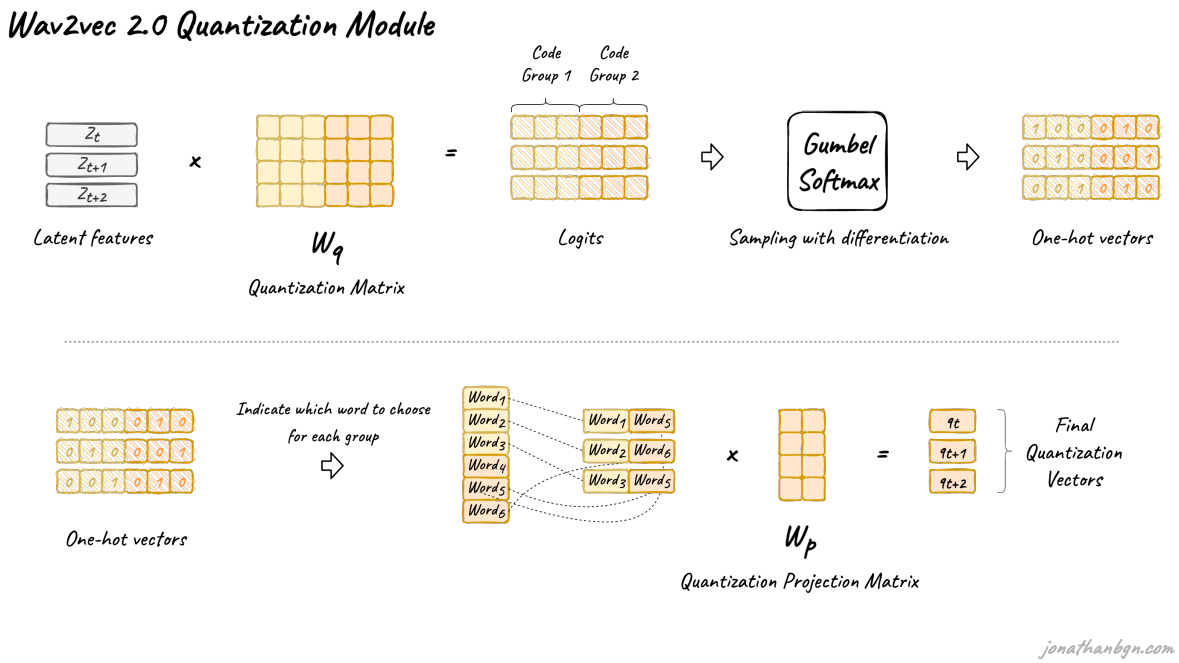


Figure 2.3: High Level view of the operation of the Quantisation Module [33].

The outputs of the feature encoder are fed into these transformers and the high dimensional inputs are then converted into inner embeddings. Within the encoders and decoders are self attenuation layers which observe multiple words in an input sequence at one time. They identify the most relevant features and words within the frames. Ensuring that these portions of the input sequence are given the most weight and focused on. The self attenuation doesn't preserve the ordering of the input frame and so a grouped convolutional layer is used to learn the positional embeddings. During training the input, previous predictions and actual value is fed into the decoder to predict the next output. After using the input and previous prediction, to generate a prediction it compares it with the correct result. The attention vectors are converted into human interpretable outputs at the feed forward, linear and softmax layers.

Self attention, a key feature of the transformer is defined by the Equation 2.2.4. Taking in query Q, keys K and values V as the input. Their matrices are resulted from multiplying the embeddings' matrix X and their corresponding weight matrices W_Q, W_K and W_V.

$$I = X \times W_I \qquad I \in \{Q, K, V\}$$

$$attention(Q, K, V) = softmax(Q \cdot K^T)V$$

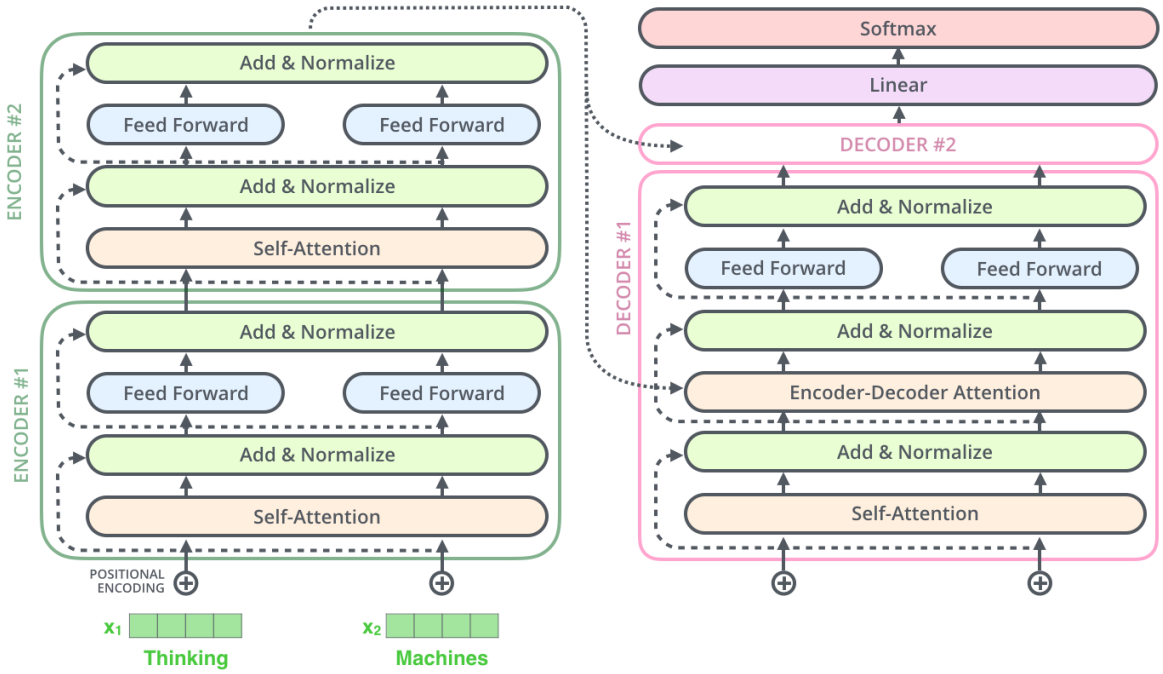


Figure 2.4: High Level view of the encoder-decoder transformer [31].

2.3 Introduction to Dialectal Arabic

This thesis will focus on Arabic dialects as despite being a widely spoken language and dialects being the primary spoken form of Arabic, there are still significant improvements that can be made to Arabic DIDs with current systems achieving the highest accuracy of 86.29%.

Arabic is the official language of 25 countries and has 330 million native speakers. Academically the regional dialects are usually grouped into 5 main groups North African (NOR), Egyptian (EGY), Levantine (LEV), Gulf (GLF) and Modern Standard Arabic (MSA). MSA is taught academically in most Arabic speaking countries and originates from the Gulf region but is not used for general conversation or outside academic setting.

Comparatively to MSA, the lack of standardisation in dialectal Arabic has resulted in more linguist complexities. Dialectal Arabic has a richer morphology and cliticisation system, accounting for circumfix negation, and for attached pronouns to act as indirect objects. As well as this some words are shared but are used for differing functions eg. For example,'Tyb' is used as an adjective in MSA but dialectal as an interjection. North African has the largest amount of dialectical variation within the dialect and is the most different from the other Arabic dialects. Taking influences from French and Berber languages. Egyptian globally is the widely understood dialect due to the Egyptian movie and television industry. Levantine dialects differ slightly in pronunciation and intonation but are equivalent when transcribed. Closely related to Aramaic. Gulf is the form which is most closely related to MSA and preserves many of MSA verb conjugations. Understanding of different dialects depends on an individuals exposure outside their own country. eg. due to the prevalence of Egyptian television and movies, many Arab people can understand the Egyptian Dialect but a Levantine speaker would not be able to understand the Moroccan dialect.

The differences between Arabic dialects are comparable to the differences present in North Germanic languages such as Norwegian, Swedish, Danish or the West Salvic languages eg. Czech, Slovak, Polish. Some linguistic variation between dialectal Arabic include incongruous

morphemes, prepositions verb conjugations, word meanings, phonemes and pronunciation. Some examples of this is shown in the Table 2.2. [17, 18]

In addition to this, the majority of available pretrained models that are used in self supervised or semi supervised systems are trained on English datasets. Arabic has 6 vowels/diphthongs in MSA and 8-10 vowels in most dialects, 28 constants while English has 24 consonants and 22 vowels. [70] As well as this compared to English, Arabic and particularly dialectal Arabic has large amount of morphemes, rendering it unfeasible for the training data to contain all the possible morphemes. So, a system which has been built to operate well for a DID that is linguistically different to English should thereby be robust enough to be applied to other languages with similar complexity.
Hence, the two key challenges in creating an Arabic DID which will be explored in this thesis are:

- Dialectal Arabic is considered low resource, as there are no large commercially available datasets.
- There is a significant amount of complex linguist differences and similarities between Arabic dialects.

More details about the ADI17 dataset to be used is provided in Section 4.1, the dataset contains audio from 17 countries and be divided into the 4 widely spoken major dialect groups (NOR, EGY, LEV, GLF), as MSA is not spoken in general conversation it is not included in the dataset and will not be identified by the DID designed in this thesis.

English/Feature	MSA	LEV	GLF	EGY
Money	nqwd	mSAry	flws	flws
I want	Aryd	bdy	Abɣý	ʕAyz
Now	AlĀn	hlq	AlHyn	dlwqt
When?	mtý?	Aymtý?	mtý?	Amtý
alveolar affricate sound	dj	j	y	g
Handsome	djami:l	jami:l	yami:l	gami:l
consonant sound	Θ	t	Θ	t
Three	Θala:Θa	tla:te	Θala:Θa	tala:ta

Table 2.2: Examples of linguistic differences between Arabic Dialects

Chapter 3

Literature Review

3.1 An Introduction to LID and DID systems

Dialect identification (DID) is a specialised task of Language Identification (LID) which identifies the dialects within a language. It poses more challenges compared to LID, as dialects share many acoustic, linguistic features and speaker characteristics. An accurate LID or DID allows for more specialised models to be used for other speech related tasks including ASR, speech transcription and Natural Learning Processing (NLP). Over time the methodology of LID and DID systems has evolved. Traditionally Phonemic Modelling was used to construct Arabic DIDs which is discussed further in Section 3.2.1. Then traditional machine learning networks were used, which is explored in Section 3.2.2. Current research is exploring the viability of utilising transfer based learning methods for LIDs and DIDs, which is detailed in Section 3.3. Table 3.1 compares the accuracies that were achieved both with traditional machine learning methods and transfer learning. The two Arabic DIDs in the table will be discussed further in this section, and it can be seen that transfer learning using a wav2vec pretrained model has only been tested for LID which was fairly successful with an accuracy of 95.5%

Application	Features	Pretrained Model	Downstream Model	Accuracy	Year, Paper
Arabic DID (17 dialects)	80 dimensional Fbank	Transformer Based Network (trained on ADI17)	CNN	86.29%	(2020), [38]
Arabic DID (5 dialects)	i-vector + FBANK + word + char + phoneme	N/A	E2E CNN+RNN+FC, DNN+SNN (feature extraction)	81.36%	(2018), [58]
LID (English, Spanish, French, German, Russian, and Italian)	Spectrograms	Resnet50	CNN + RESnets	89%	(2019), [54]
LID (Arabic, English, Malay, French, Spanish, German, Persian, and Urdu)	Acoustic features (MFCC + GMM + i-vector)	N/A	ESA-ELM (Enhanced Self- Adjusting Extreme Learning Machine)	96.25%	(2018), [62]
LID (26 languages)	N/A	wav2vec 2.0	pooling layer + linear layer	95.5%	(2021), [45]

Table 3.1: Recent Machine Learning Implementations of LIDs & DIDs.

3.2 Traditional Methodologies

3.2.1 Phonematic Modelling

A phoneme in linguistics is the smallest unit of sound which can convey meaning (for instance, the sound /c/ in cat). Phonematic modelling utilises recognisers to identify the phonemes present within an audio segment. Different dialects usually have different phoneme combinations and so, the identified phonemes are mapped to identify a dialect. In the paper [16] this technique was used to construct an Arabic DID for 4 dialects (Gulf, Iraqi, Levantine, and Egyptian) plus MSA which took advantage of English, Arabic, Japanese phone recognisers to identify the phoneme differences between the dialects as seen in Figure 3.2.1. This method was able to achieve high accuracy levels for identifying MSA with F-Measures above 98% and the highest of the dialects was Egyptian Arabic with an F-Measure of 90.2% with 30s test-utterances. As seen in Figure 3.2.1, phonemic modelling for Arabic struggled when given shorter utterances and had particularly low accuracies for the Gulf dialect. The key challenges with using phoneme modelling is that it relies on the distinguishing phonemes to be present in the test data and for finer regional dialects it requires there to be more shared phonemes between the dialects.

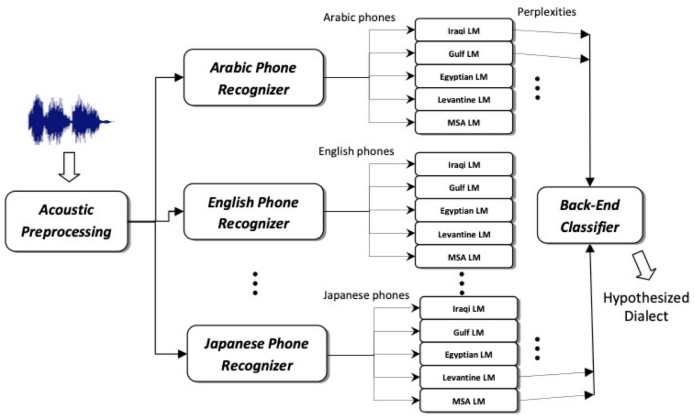


Figure 3.1: Parallel Phone Recognition Followed by Language Modeling (PRLM) for Arabic DID [17].

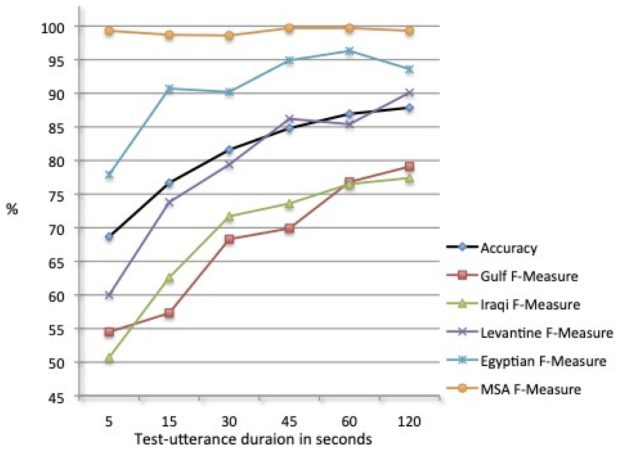


Figure 3.2: The accuracies and F-Measures of the five-way classification task with different test-utterance durations [17].

3.2.2 Traditional Machine Learning

Traditional Machine learning has been used for both LID and DID systems as explored in papers [3,38,42,43,53,58,62]. They operate by extracting key features from the training audio, which could be acoustic and/or linguistic, then using some form of traditional machine learning structure to learn the differences between dialects based on the extracted features. Implemented in the papers [38, 42, 43, 58] are ransformer based networks which are constructed with a similar structure to that shown in Figure 3.3. Simple transformer networks were compared to networks which used a combination of CNN, LSTM networks along with the transformer network. Convolutional Neural Networks (CNN) are composed of three types of layers, a convolutional layer, pooling layer and a fully-connected (FC) layer, with a greater amount of layers the complexity of the network increases. CNNs learn through using filters to detect certain features in the training data and adjusting its weights accordingly. The Arabic DID explored in paper [39] used the ADI17 dataset which will be used in this thesis was able to achieve the highest accuracy of 86.29% when cascaded with a CNN network. In contrast, Bidirectional Long Short Term Memory (BiLSTM) is created from two Recurrent Neural Networks (RNN). It has the ability to combine information from both past and future inputs. The structure of BiLSTM is shown in Figure 3.4. Although, there are no papers showing the effectiveness of using BiLSTMs specifically for Arabic DID, LSTMs were used in [42, 43], transformer based networks and were able to achieve an accuracy of well over 90% for all the ADI17 dialects in [43]. As well as this, the papers [55, 70] explored the use of BiLSTMs in text based Arabic DIDs and paper [63] explored its use in Mandarin/English LID with XLS-R producing a 92.7% accuracy.

3.3 Transfer Learning

Transfer learning is a form of deep machine learning which focuses on adapting a pretrained model to execute a similar task. The pretraining enables rapid learning for modelling the task and often requires less data to achieve a high level of accuracy in the secondary class. For the task of LID and DID, this often means applying a pretrained model designed for speech tasks like ASR, speech synthesis etc. Transfer learning is a relatively new method for optimising

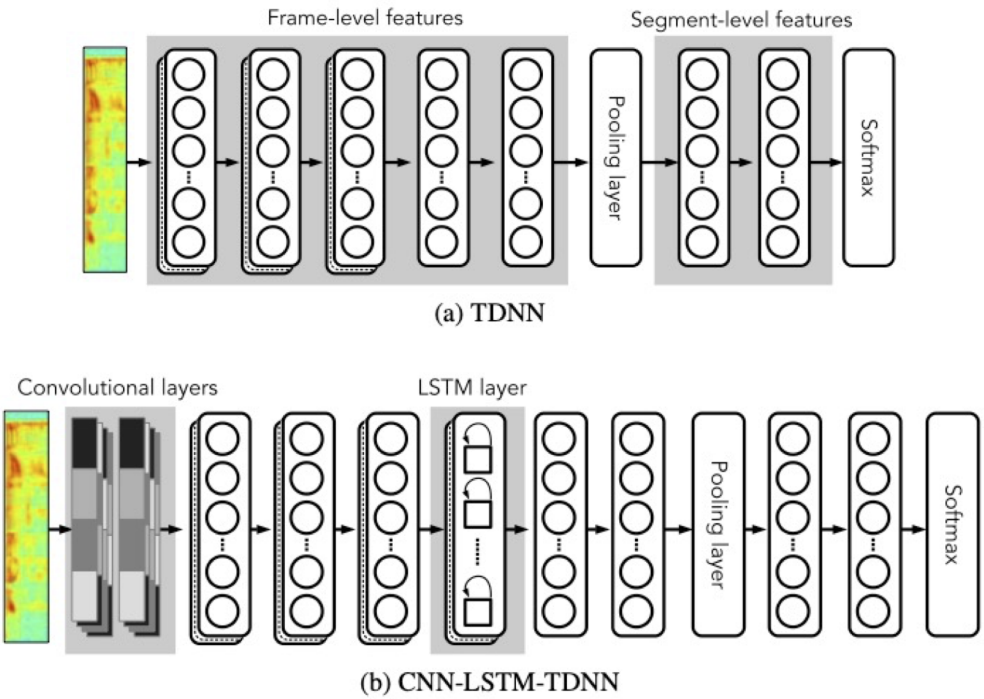


Figure 3.3: Transformer Network based LID [43].

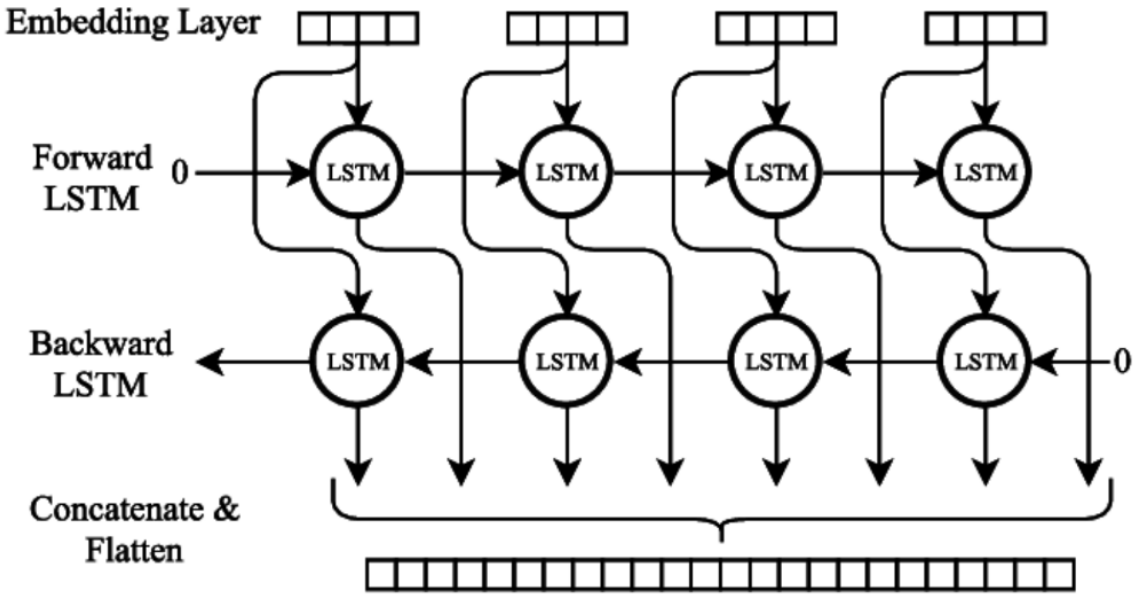


Figure 3.4: BiLTSM Structure [23].

machine learning to complete a task and so, there are not a vast amount of papers exploring its use for some more niche tasks like LID and DID. Papers [26, 29, 45, 68] have shown significant increases in accuracy using pretrained models for tasks such as emotion recognition, ASR and NLP.

In addition to a pretrained model, a transfer learning system often has a downstream model connected to it that allows the model to adapt to more specific tasks. The system can then be trained end to end (E2E) or fine-tuned in portions, tuning the pretrained model, then the downstream. It was found in the paper [68], which explored E2E training for wav2vec and

HuBERT for the tasks Speaker Verification (SV), Intent Classification (IC) and Slot Filling (SF), that on average using E2E training provides more accurate systems. For example, looking at their results for wav2vec, E2E outperformed segmented training by 12.47% in SER, decreased EER by 3.26% in SV, improved accuracy in IC by 39.98% and SF by 36.66%. Thereby, this thesis will employ E2E training for the system.

3.3.1 Pretrained Models

The pretrained models are semi-self supervised machine learning models often designed by large tech companies, then trained on large amounts of unlabeled and small amounts of labelled data. There are several pretrained models available for use for speech processing the main ones that will be explored are HuBERT, wav2vec 2.0 and XLSR developed by Facebook. Wav2vec 2.0 is designed for speech data, consisting of a feature encoder, context network, quantisation module and a contrastive loss layer. Wav2vec is pretrained using a contrastive task, masking a unit in the feature vector then predicting what should be in that unit. In the case where the prediction is wrong a negative score is given and when right a positive, and the network then adjusts its weights accordingly. HuBert is a hidden unit bidirectional and shares a structure with wav2vec 2.0 using a transformer based networks and contrastive based learning, although it uses BERT. BERT is able to process a segment of speech simultaneously learning the surrounding context of a word. It aimed to improve wav2vec through the use of BERT prediction loss and was able to produce up to 19% and 13% relative WER reduction for a 1B parameter model. XLS-R is a fine-tuned variant of wav2vec 2.0, that is trained using data from 128 different languages collected from BABEL, MLS, CommonVoice and VoxPopuli speech corpa. Tuning the model on languages other than English reduced error rates 14-34% relative on average [11]. It has also shown to operate with a higher degree of accuracy on low resource languages compared to other models as shown in Figure 3.3.1. The paper [45] compared the accuracy when using no pretrained model, wav2vec 2.0 and XSL-R on a 26 language LID. The highest accuracy was consistently achieved by XLS-R as seen in Figure 3.3.1, the highest being 95.7% with 100hrs of labelled training data. Hence, this thesis will be using XLS-R and benchmarking it against wav2vec 2.0, HuBERT will not be tested as for the scope of this thesis it is too ambitious to explore more than two pretrained models.

There hasn't been any research into the application of pretrained models for Arabic DIDs but there has been limited research into using wav2vec for LID systems. The papers [11,45,62] demonstrate it as a possible methodology for LID. The paper [45] was able to achieve an accuracy of 95.5% for their 26 language LID, utilising only a simple pooling layer and linear layer as their downstream model as shown in Figure 3.5. So, fine-tuning of the last two layers of wav2vec 2.0 will be the base method explored in this thesis.

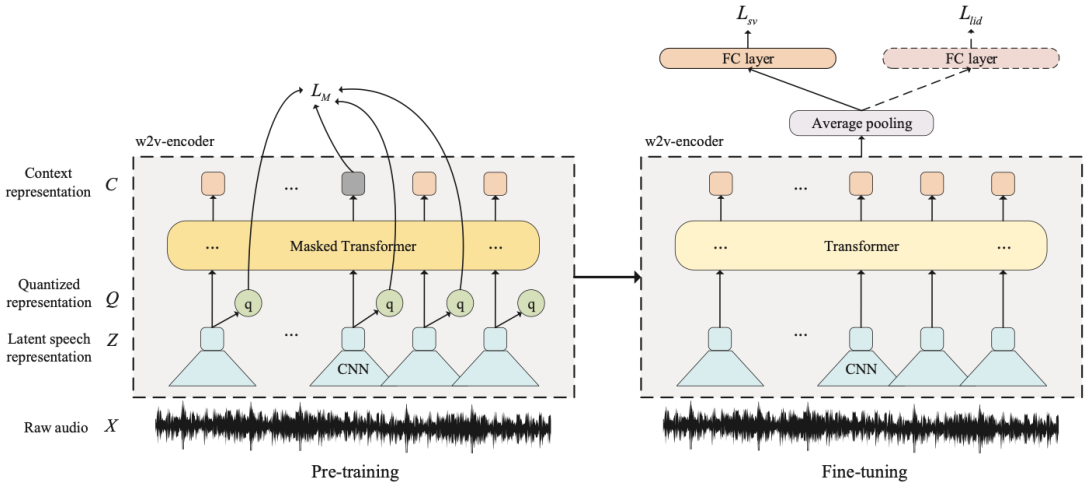


Figure 3.5: wav2vec 2.0 LID [45].

Lbl. / lang	Pre-training	Test Accuracy (%) 0-6s	6-18s	18-∞s	Overall
10 min	None	7.1	9.5	10.6	9.6
	w2v2 En	71.3	73.1	76.1	74.2
	XLSR	85.4	88.8	90.8	**89.2**
1 hour	None	20.2	25.2	29.5	26.5
	w2v2 En	79.3	85.9	89.3	86.5
	XLSR	87.2	92.5	94.8	**92.8**
10 hours	None	48.3	61.9	71.8	64.5
	w2v2 En	86.8	93.3	95.6	93.4
	XLSR	88.2	94.3	96.1	**94.2**
100 hours	None	72.2	84.9	90.7	86.7
	w2v2 En	89.5	95.7	97.3	95.5
	XLSR	90.3	95.9	97.2	**95.7**

Figure 3.6: 26 language LID test accuracy [45].

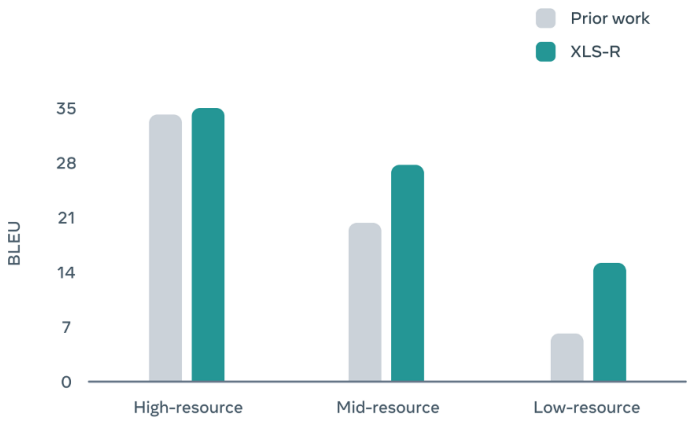


Figure 3.7: XLS-R BLEU Accuracy when Translating to English [1]

Chapter 4

Methodologies: Design and Implementations

This chapter will explore the design and implementation of the Arabic Dialect identification System. As well as the base dataset choice, it's preparation and preprocessing. The experiments conducted throughout this thesis will be briefly introduced and Chapter 5 will further explore them in detail. The code used, the dataset design and results are publicly available on the GitHub repository https://github.com/madelineyounes/thesis.

4.1 Dataset

The dataset to be used in this thesis is the **ADI17** dataset [59]. The **ADI17** dataset consists of audio segments from known Youtube videos with dialects from 17 different Middle Eastern and North African countries. The dataset is divided into training, development and test data groups. The training set contains 3000hrs of audio total while the development and test combined is 57hrs of audio. The specifics of the dataset can be seen in Figure 4.1. The data was collected from around 30 different Youtube channels per country and the primary dialect each Youtube channel used was verified by a human annotator. Using the Youtube channel's dialect audio segment's dialectal label was allocated. The training data relies on this for its labelling, whilst the test and development data was annotated by a human annotator. The audio segments are split into utterances ranging from 10-20s in length, which are small portions of audio generated by segmenting the original audio at silence points. These silence points are usually natural pauses in conversation and a threshold is used to determine how long the silence must be before the audio is split. The creators of the ADI17 dataset have not specified the threshold that they used. The dataset is labelled using 17 regions, a portion of this thesis explores creating a generalised DID of 4 generalised umbrella dialects that encompass this finer set of regional dialects as shown in Figure 4.2. Hence, the of data from each region is taken to construct the training set for the generalised dialects as shown in Figure 4.1. The core challenges with the ADI17 dataset are that the acoustics are unbalanced across each of the dialect regions and the amount of data provided is unbalanced. The amount of noise in each of the regions datasets is shown in Figure 4.3, although its effect on the DID will be mitigated through using channel normalisation and filtering experimented with, as the papers [18, 51] have shown is effective at increasing accuracy of Arabic DIDs. The dataset is also unbalanced in terms of amount of training data for each region as shown in Figure 4.1, with Jordan having the least amount of data. To ensure that training, validation and testing is balanced between all the dialects only a subset of the dataset is used.

A script was written to generate three csv files (test, training and validation) containing the file names and the dialectal label of each file. For the umbrella DID, the script would ensure that equal amounts of files were taken from each regional dialect and then changed the label from the regional one to it's corresponding umbrella dialect. eg. for the umbrella DID test csv 700 files were allocated to the North African umbrella dialect, 175 were from each of the regional dialects (DZA, MAR, MRT, LBY). Within the system pipeline 70% of the data is allocated to training, 20% to validation and 10% to test. A breakdown of this for both the umbrella DID and the regional DID is shown in tables 4.1 and 4.2 respectively.

Country	ABBR	Training (hrs)	Dev (hrs)	Test (hrs)
Egypt	EGY	451.1	1.9	2.1
Sudan	SDN	47.7	0.7	2
Eygpt Total		**498.8**	**2.6**	**4.1**
Algeria	DZA	115.7	0.6	1.9
Mauritania	MRT	456.4	0.5	1.3
Morocco	MAR	57.8	1.1	1.9
Libya	LBY	127.4	2.3	2
North Africa Total		**757.3**	**4.5**	**7.1**
Iraq	IRQ	815.8	1.5	1.9
Kuwait	KWT	108.2	1.2	2
United Arab Emirates	ARE	108.4	2.2	1.8
Yemen	YEM	53.4	1.3	1.8
Saudi Arabia	SAU	186.1	1.2	2.1
Qatar	QAT	62.3	2	1.7
Oman	OMN	58.5	1.7	1.8
Gulf Total		**1392.7**	**11.1**	**13.1**
Jordan	JOR	25.9	1.7	2
Lebanon	LBN	116.8	1.3	1.9
Palestine, State of	PSE	121.4	1.4	2.1
Syrian Arab Republic	SYR	119.5	1.3	2
Levantine Total		**383.6**	**5.7**	**8**

Figure 4.1: ADI17 Dataset Details.

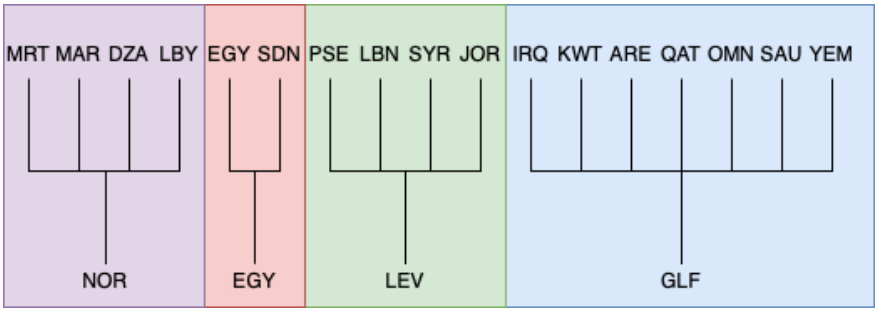


Figure 4.2: Regional to Umbrella Dialect Grouping.

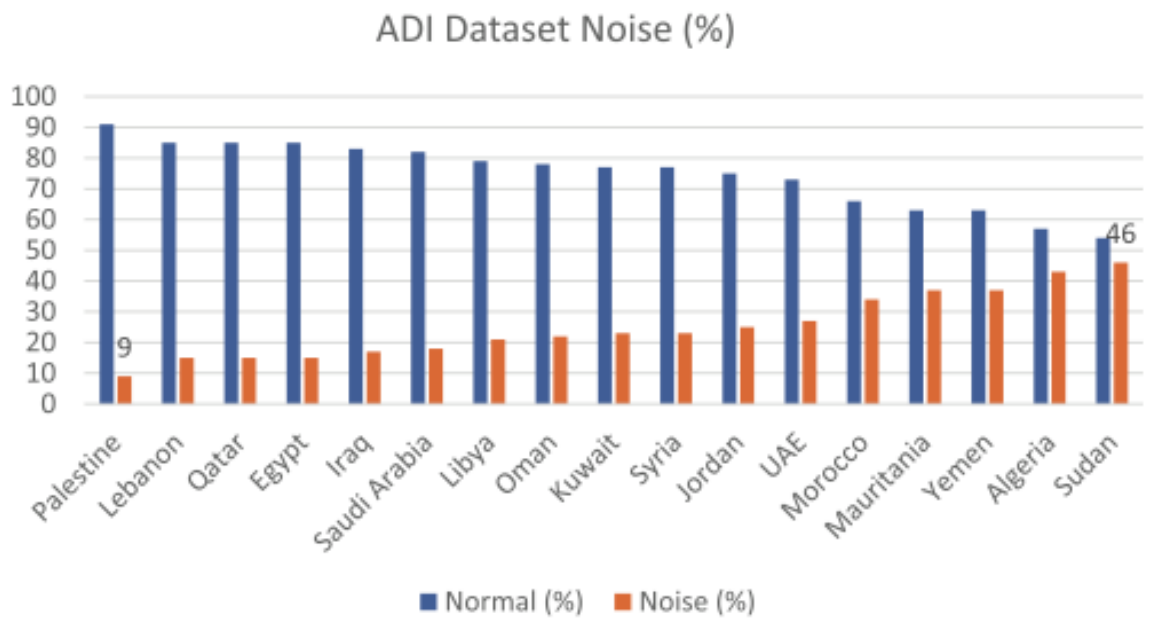


Figure 4.3: ADI17 Dataset Noise levels. [6]

Umbrella DID File Breakdown			
	Split (%)	Number of Files	Total Time (hrs)
Training	70	2800	7.78
Validation	20	800	2.22
Test	10	400	1.11

Table 4.1: Table showing the file breakdown for the Umbrella DID pipeline.

Regional DID File Breakdown			
	Split (%)	Number of Files	Total Time (hrs)
Training	70	11900	33
Validation	20	3400	9.4
Test	10	1700	4.72

Table 4.2: Table showing the file breakdown for the Regional DID pipeline.

4.2 Implementation of Arabic DID System

This section will describe the implementation of the system in Python. The pipeline scripts leverage the Hugging Face Transformer library but expand out the Dataset, DataLoader and Trainer classes rather than only using their default functionality. The pretrained models are also imported using this library.

4.2.1 Overall System Design

The proposed pipeline design for the Arabic DID is illustrated in Figure 4.4. The system takes in three csv files, training, validation and test. The files specify the labels and file paths of the audio files. There is a corresponding dataloader for each set which manages the processing of

the audio files. It extracts the audio features, trims the audio to a specified length, normalises them, then shuffles and groups the data into batches. The Experiment 5.2.2 also tests filtering the data to remove noise. The main portion of the pipeline is the trainer, which manages the fine-tuning of the pretrained model. The model is prepared to be trained at the start of the pipeline with the layers to be trained unfrozen. In all the experiments the feature projector layer and the linear classifier layer are unfrozen. In the experiment 5.2.5 unfreezing and training the encoder layers is explored. While in experiment 5.2.4 a downstream model is inserted into the network, this is discussed further in Section 4.2.4. At each epoch the processed training data is passed through the model calculating the loss, using a cross entropy loss function then updating the layer weights accordingly. The model is then put in evaluation mode, to assess the accuracy at that epoch using the validation dataset.

Once a model is fine-tuned it is then run through a third set of data which has not been used in the training process. Similarly to the main pipeline, the audio features are extracted from the audio file, it is truncated, batched and then fed into the final model. The Python package sklearn.metrics is then given the predictions and true labels to generate the confusion matrices and classification report. A function was written taking advantage of the matplotlib.pyplot library to convert the confusion matrices into colour map plots. This testing pipeline is shown in the Figure 4.5.

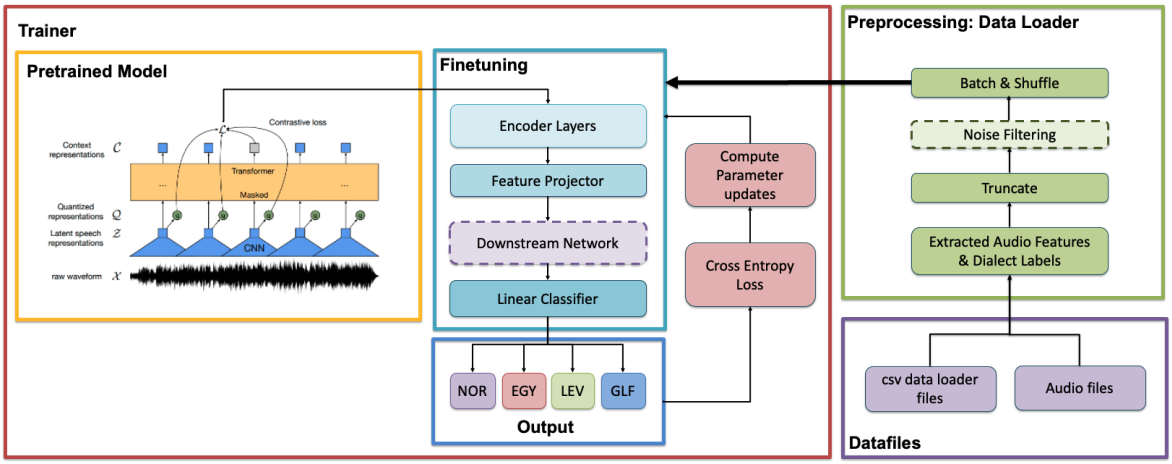


Figure 4.4: Overview of the System.

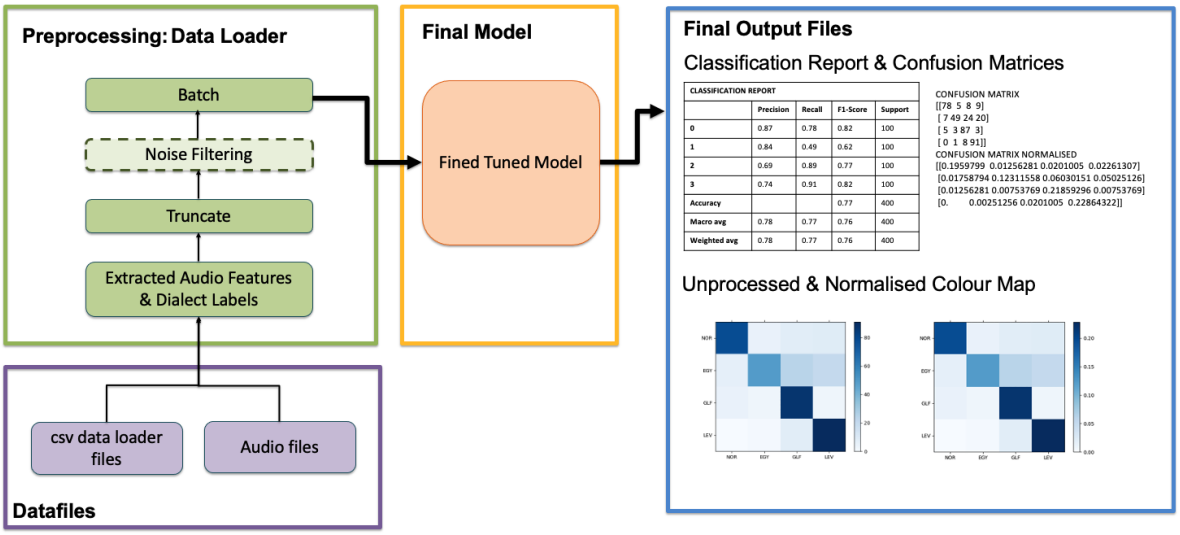


Figure 4.5: Test Pipeline.

4.2.2 Pre-training

All the pre-trained models used in this thesis were trained by Facebook using 16kHz sampled speech audio and were imported from the Hugging Face Transformer library. The advantage of using a pretrained model is the significant amount of resources needed to sufficiently train a wav2vec 2.0 or hubert architecture is not required. Several model choices were explored, they are listed with a summary of their key features in the Table 4.2.2 and their performance was tested in Experiment 5.2.1.

4.2.3 Fine-tuning

Fine-tuning is the process which adapts the pretrained model to the downstream task of Arabic dialectal identification. The pretrained model is initialised with the weights from in its initial training and the final classification layer or downstream network is randomly initialised. So, for the umbrella DID the output of the final layer is 4 and for the regional DID is 17, corresponding to the number of classes each DID have. In the Experiment the CNN encoder layers are unfrozen, so that their weights are also updated.

The meta-paramaters of the trainer are kept consistent for each of the experiments. The learning rate is set to 0.00004, an Adam optimiser is used with a learning rate warm-up for the first 10% of updates. The updates are then linearly decayed for the remainder of the updates. A batch size of 40 per GPU is used thereby when using 2 GPUs the samples processed at one time is 80 and in the case where 3 GPUs are used 120 samples are processed.

Training is performed on the shared UNSW computational cluster Katana[] using 2 to 3 Tesla V100-SXM2 32GB GPUs depending on the experiment being conducted and it's reasource requirements.

4.2.4 Downstream Network

When using transfer learning, for some downstream tasks it can be advantageous to add a network to the structure of a pretrained model, this structure is referred to as a downstream model. To investigate the effectiveness of a downstream model on creating an Arabic DID, a

downstream model is inserted into the wav2vec 2.0 architecture. The model is inserted between the Feature Projector layer, taking in the 256 outputs from it and the final linear layer with the classifier classes. For experiments with a downstream model the weights of a previously successful model are imported onto the head of the model. The pretrained model head is then frozen and only the downstream model with the final linear classifier layer is trained. This thesis tested DNN and LSTM model structures. For the DNN downstream network the number of layers within the model is varied between 3 and 6 layers. The DNN networks are comprised of rectified linear activation function (ReLu) layers. The ReLu layers are a linear piece wise function that outputs the input if it is given a positive value and zero if it isn't. It was chosen to be used in the DNN structures as they are one of the most common activation functions, relatively simple to train and achieve high performances with. Whilst the LSTM architecture is constructed with a Long short-term memory (LSTM) layer that uses information from past and current data samples, a dropout layer which reduces overfitting by randomly setting the inputs to 0 and the softmax layer converts the inputs into probabilities. All the downstream model structures explored are shown in Figure 4.6, in this figure the number of classes is set to 4 for the umbrella DID.

Model	**Key Features**	**Hugging Face Path**
Hubert	**Training:** 960 hrs of English unlabeled speech and 1hr of labelled speech from Librispeech corpa (LS-960). **Structure:** Masked hidden units.	facebook/hubert-base-ls960
wav2vec 2.0	**Training:** 960 hrs of English unlabeled speech and 1hr of labelled speech from Librispeech corpa (LS-960). **Structure:** Masking and contrastive based learning.	facebook/wav2vec2-base
XLS-R	Fined tuned version of wav2vec 2.0. **Training:** 56k hrs unlabeled speech data in 53 languages, from Multilingual LibriSpeech (MLS), Babel and Common Voice speech corpses.	facebook/wav2vec2-large-xlsr-53
XLS-R Arabic	XLS-R variant finetuned on 7.5hrs of male voiced Syrian (Levantine) Arabic with a Damascus accent	elgeish/wav2vec2-large-xlsr-53-arabic
wav2vec sid	wav2vec 2.0 variant finetuned for the downstream task of speaker identification using the VoxCeleb1 corpa.	superb/wav2vec2-base-superb-sid
wav2vec lid	wav2vec 2.0 variant finetuned for the downstream task of language identification.	log0/wav2vec2-base-lang-id

Table 4.3: Summary of the pretrained models used in this thesis.

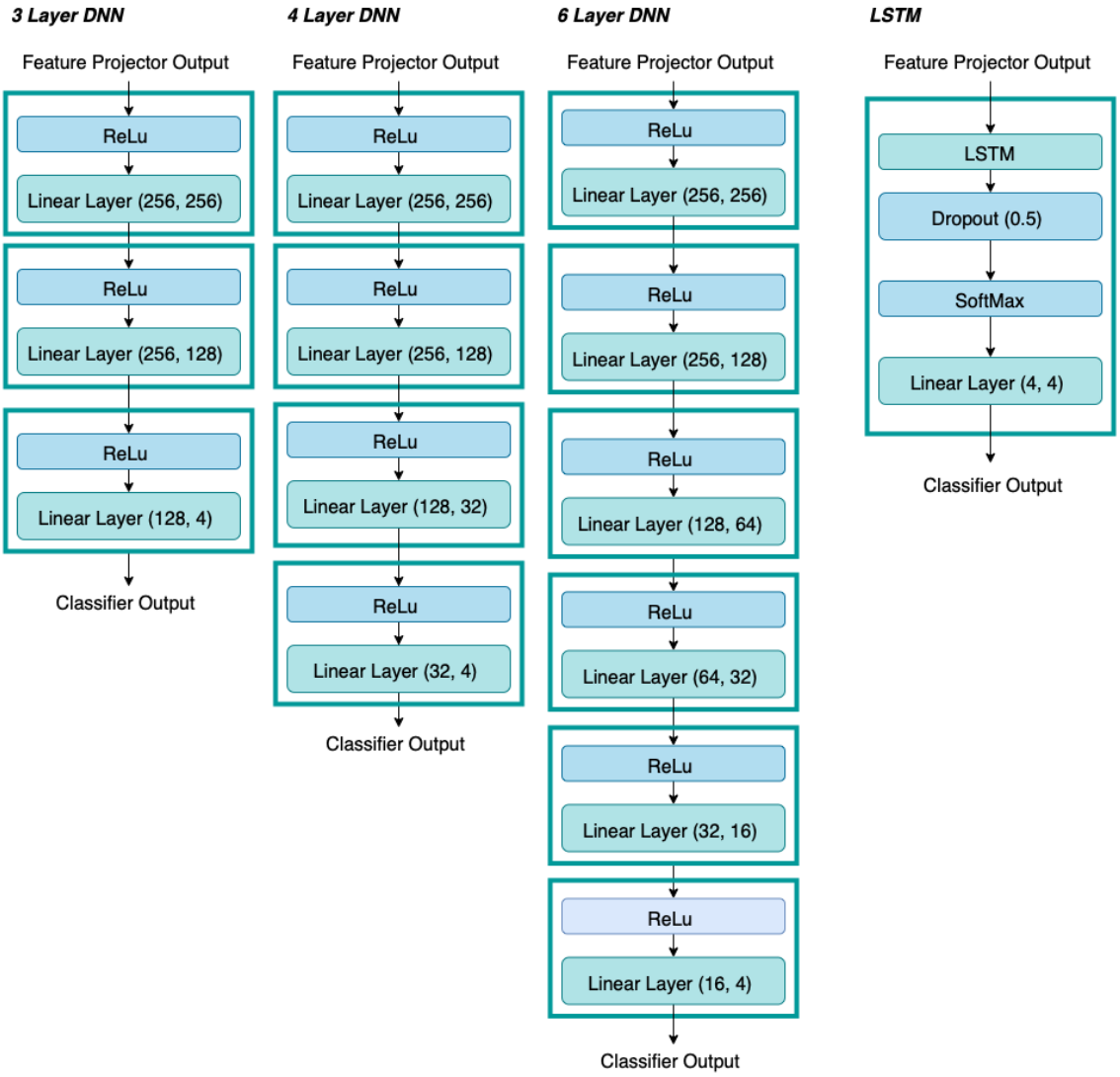


Figure 4.6: Downstream model structures.

Chapter 5

Experiments, Results and Analysis

This chapter presents the experiments conducted to fine-tuning an umbrella Arabic DID, then assessing its robustness and adaptability. The results are analysed and the system setup for each of the experiments are detailed in Chapter 4. The full, detailed results are provided in Appendix A.1.

5.1 Preliminary experimentation

Prior to streamlining the final system pipeline for the Umbrella DID some preliminary testing was performed. These experiments provided vital information that informed some desgin choices for the final system pipeline. As well as confirming some assumptions made when undertaking this thesis.

Testing SpeechBrain LID performance on Arabic Dialects

An out of the box Python package that perform's langauge identification (LID) is the SpeechBrain package. The system has a ECAPA-TDNN architecture which is composed primarily of convolutional and residual layers. The LID was trained using CommonLanguage, a curated subset of data from the open source dataset CommonVoice. The dataset is sourced from voluntarily submitted audio files. In order to get an understanding of the gaps in LIDs in identificating dialectal Arabic as Arabic, the ADI17 dataset was run through SpeechBrain's LID. The LID had an overall accuracy of 87.69%. Investigating the LID's performance on the umbrella dialects it identified Egyptian (EGY) with the highest accuracy of 90.72%, Gulf (GLF) with the lowest accuracy of 85.77% and the dialects were most likely to be misidentified as Somali (so) or Hebrew (iw). Breaking down that performance into the regional dialects Jordian (JOR) Arabic was identified as Arabic with the highest level of accuracy at 95.98% and Yemen (YEM) Arabic was identified with the lowest accuracy of 81.58%. Thereby, showing that despite LID systems working well at identifying dialectal Arabic as Arabic there is still room for improvement. Having a specialised Arabic DID could improve both LID systems and other speech systems.

Amount of Training files

In order to determine what a reasonable amount of data files were for performing basic training of the the DID system a preliminary test was conducted. The resources allocated for the experiment were 2 GPUs, 16 CPUs and 92GB of memory. Wav2vec 2.0 base was used as the pretrained model and the audio was truncated to 5 seconds. Training with 50, 100, 500

and 1000 files per dialect was tested, their validation accuracies were 27.29%, 32.06%, 34.30% and 33.51% respectively. Additionally, the runtime on average was 1hr and did not fluctuate greatly when the files were increased. From this experiment it was determined that training with 500-1000 files per dialectal group would be sufficient. 700 files per dialect was chosen in further testing to be the default training amount.

Batch Size

Another variable that was experimented with to determine a reasonable base amount was batch size. Batching is when training samples are grouped together to be processed parallel to one another at an iteration. Out of the pretrained models that will be experimented with in later tests, the model with the largest amount of parameters to train and thereby requires the most amount of working memory is XLSR. So the experiment was conducted with XLSR Arabic with 2 GPUs, 16 CPUs, 92GB of memory and the audio was truncated to 10 seconds. Using 500 training files the batch sizes 8, 20, 40 were tested. The maximum validation accuracy was 44.4%, 44.6% and 43% respectively. While their runtimes were 2.04hrs, 1.08hrs and 1.04hrs. So, it was determined that the batch size didn't significantly effect the performance of the system, although larger batch sizes did help reduce runtime. It was also confirmed that the larger batch sizes required significantly larger amounts of memory to process, with the largest batch size that could be used without the system running out of working memory being 40. A batch size of 40 was then to be used as the default training amount in further experimentation.

5.2 Fine-Tuning an Umbrella Arabic DID

The focus of the experiments in this section is to determine the design choices and factors which will produce the highest performance Umbrella Dialect Arabic DID. The four umbrella dialects are North Afriacan (NOR), Egyptian (EGY), Gulf (GLF) and Levantine (LEV). The breakdown of these groupings and the data used can be read in Section 4.1.

5.2.1 Assessment of Performance of Pretrained Models

The first formal experiment conducted on the umbrella Arabic DID was trialing the use of different pretrained models. The resources allocated for these tests were 3 GPUs, 24 CPUs with 138GB of memory allocated. The results are presented in the table 5.1, figure 5.1 and figure 5.2. The aims of experimenting with different pretrained models was to determine:

- The effect of structure.
- The significance of using a diverse range of languages in the training data.
- The adaptability of models fine-tuned for similar downstream tasks.

The two base structures tested in this experiment were Hubert and Wav2Vec 2.0, with more details about their architecture provided in the Section 2.2. Hubert outperformed the base wav2vec in all significant metrics and had a weighted average F1-Score that was 15% greater than it. Although, compared to Wav2Vec 2.0, Hubert does not have many variants and so, further testing was conducted on wav2vec variant models. The next objective of this experiment was to compare the base wav2vec with XLSR, a variant trained with 53 languages and a version of XLSR which had been further fined tuned with Levantine Arabic data. The highest

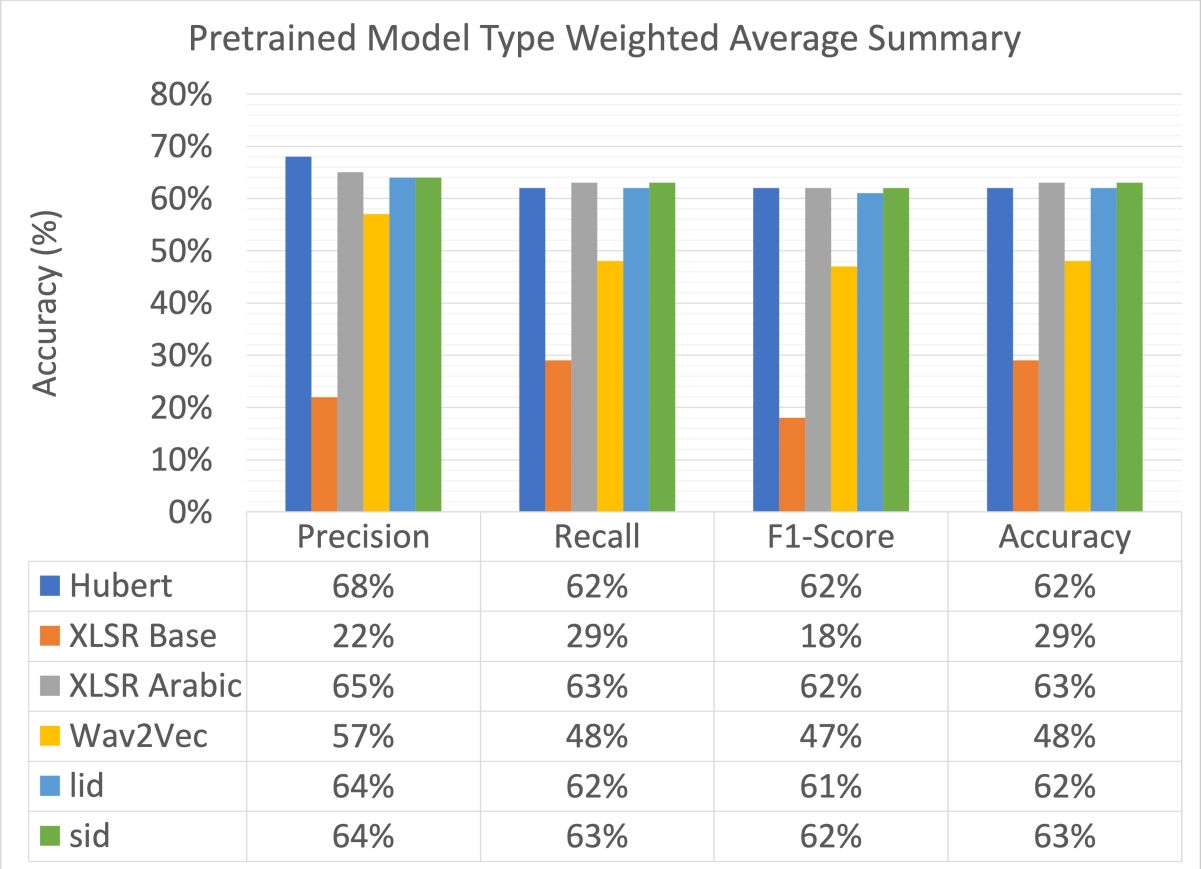
performance was from XLSR Arabic with a weighted average F1-Score of 63%, 14% higher than the base wav2vec and 45% higher than the base XLSR. An interesting behavior of the base XLSR was it's inability to identify Egyptian or Levantine Arabic, predicting the data was Gulf (GLF) Arabic 91.96% of the time. This bias towards Gulf Arabic can be seen in Figure 5.2 and is most probably due to biases within the pretraining dataset. With the data used pretrain XLSR being from the Gulf region or Modern Standard Arabic (MSA) which is most similar to Arabic from the Gulf region. Finally, models which had been fine-tuned for similar downstream tasks were tested. Comparing wav2vec fine-tuned for LID and SID tasks against the base model, both of the downstream task models outperformed the base wav2vec. The wav2vec SID had the highest F1-Scores with a weighted average F1-Score 16% greater than the base wav2vec and comparable to XLSR Arabic. Looking at the results for North African (NOR) Arabic wav2vec SID marginally outperformed XLSR Arabic by 2%. The wav2vec LID also had comparable results to Hubert performing marginally better at identifying North African (NOR) and Gulf (GLF) dialects.

This experiment also gave an indication of the dialects the system excels or struggles at identifying. It was found that across all the pretrained model types the system was able to identify North African with the greatest proficiency and Levantine with the least.

Ergo, from this experiment it was found that using a pretrained model that was already fine-tuned with a similar dataset or for a similar downstream task was more effective than using a base pretrained model. For further experimentation the XLSR Arabic pretrained model was chosen to be used based on these results, as the features used to differentiate the dialects were less likely to be the acoustic features or features of the speaker compared to the wav2vec SID model.

F1-Score (%)							
	Hubert	XLSR	XLSR Arabic	wav2vec	wav2vec sid	wav2vec lid	**Average**
NOR	67	30	70	50	72	68	60
EGY	64	0	64	48	65	60	50
GLF	61	42	62	54	60	63	57
LEV	56	0	54	35	53	51	42
Weighted Average	62	18	63	47	63	61	

Table 5.1: Umbrella DID F1-Score Breakdown of each Dialect for Pretrained Models



	Precision	Recall	F1-Score	Accuracy
Hubert	68%	62%	62%	62%
XLSR Base	22%	29%	18%	29%
XLSR Arabic	65%	63%	62%	63%
Wav2Vec	57%	48%	47%	48%
lid	64%	62%	61%	62%
sid	64%	63%	62%	63%

Figure 5.1: Plot comparing key metrics for different pretrained models.

5.2.2 Exploring Noise filtering

The ADI17 dataset, is sourced from Youtube clips, many of the audio clips contain background noise. As discussed in Section 4.1 adding noise filtering can increase the performance of a DID. The focus of this experiment was to test two filtering methods against the performance of an umbrella DID with no filtering performed on the dataset. The tests were performed with XLSR Arabic as the pretrained model, the audio files truncated to 10 seconds and The first method tested was using the Python package noise reduce. The package over filtered the audio clips, reducing the clarity of the speaker as seen in a sample utterance waveform in figure 5.2.2 and had a final accuracy of 53.3%, 10% less than the unfiltered dataset. The second method tested was a Butterworth bandpass filter with a lower cutoff frequency of 50Hz and an upper cutoff frequency of 500Hz. The frequencies were chosen with the aim to cut off any sound outside the human vocal range. The bandpass filter also didn't perform well in comparison to the unfiltered signal with an accuracy of 43.3%, 20% less than the filtered signal. This reduced accuracy is also due to the distortions filtering causes to the audio signals as shown in figure 5.2.2. Thereby, it was found that despite the literature suggesting noise filtering would improve the umbrella DID's performance it was not found as an effective strategy.

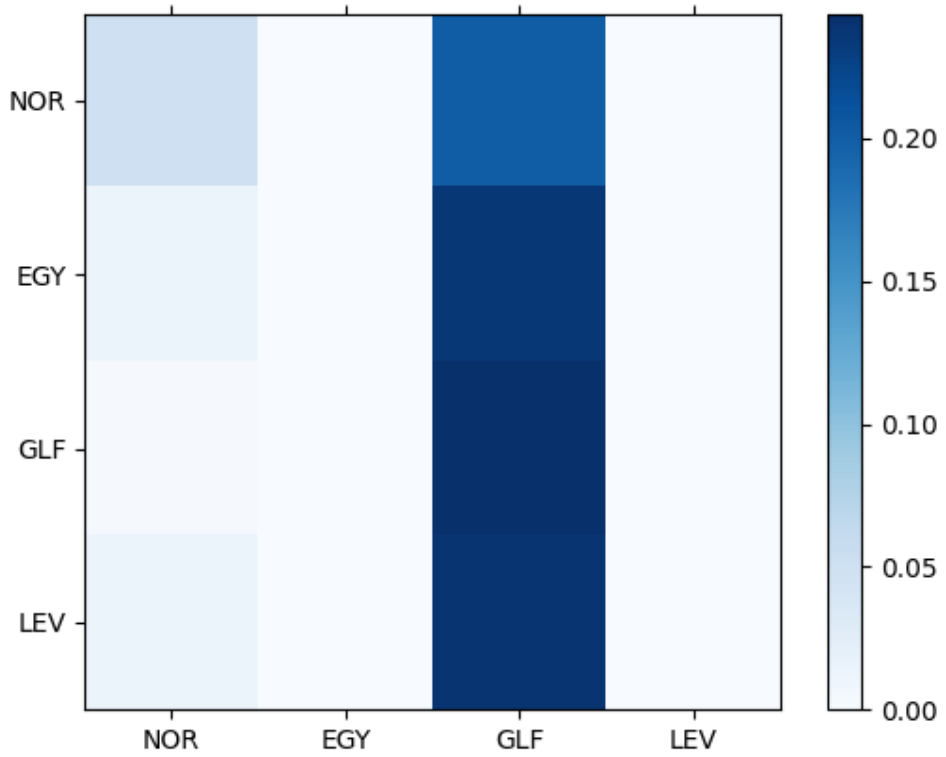


Figure 5.2: XLSR Normalised confusion Matrix Colour map.

Filtering Method	**Accuracy(%)**
Noise Reduce Package	53.3
Band pass Filter	43.3
No Filter	63.2

Table 5.2: Effect of Noise Filtering on Final Accuracy

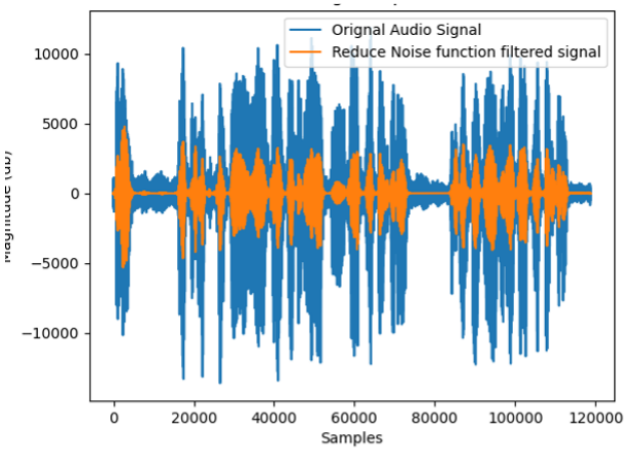


Figure 5.3: Sample Utterance Filtered using Reduce Noise Package

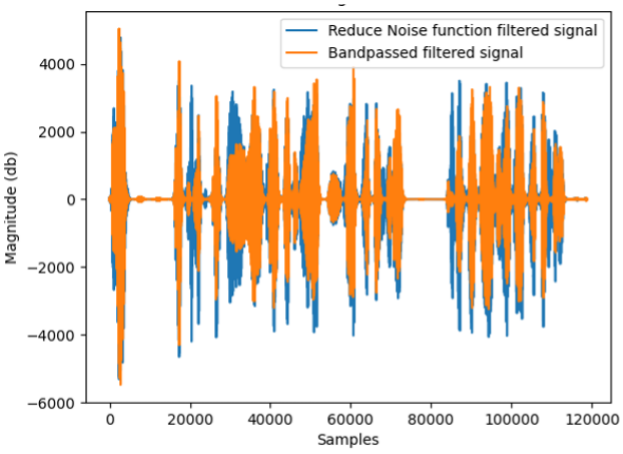


Figure 5.4: Sample Utterance Filtered using Bandpass Filter

5.2.3 Effect of Utterance Length

This experiment aimed to compare the performance of the umbrella DID with varying utterance lengths to find the minimum acceptable training utterance length. The training was performed with 3 GPUs, 24 CPUs with 138GB of memory and 500 files allocated per dialect. The utterances within the dataset range from 10 - 25s, a script was written to select files with audio lengths greater than or equal to 10s for the medium utterance length test and then greater than or equal to 20s for the long utterance length test. The files were then truncated to the specified training lengths of 5s, 10s or 20s. Analysing the results in the Table 5.3 the long and medium utterances had comparable performances. The longer utterances outperformed both short and medium utterances for most of the F1-Scores. With a weighted average F1-Score of 59% it was 8% greater than the medium utterances and 26% greater than the short. Although, for the North African (NOR) dialect the medium utterances marginally performed better with an F1-Score of 57% compared to the long utterance's 53%. Looking at the confusion matrices colour maps shown in figures 5.2.3 and 5.2.3, the long utterances were more likely to be misidentified as Egyptian (EGY). Considering the runtimes of the medium and long utterance training segments shown in figure 5.5. Training with the long utterances have a significantly longer runtime demand of 7.36hrs compared to the medium's 1.13hrs. So, it was found that longer training utterances overall produced a higher performing DID but need significantly more resources to train. It was decided for further testing, particularly with more complex network structures and more fine-tuning layers that the performance of using medium clips had sufficient performance.

	F-1 Score (%)			
	Short (5s)	Medium (10s)	Long (20s)	**Average**
NOR	24	57	53	44.7
EGY	48	56	58	54
GLF	30	50	62	47.3
LEV	30	41	63	44.7
Weighted Average	33	51	59	

Table 5.3: F1-Score for Varying Utterance Lengths.

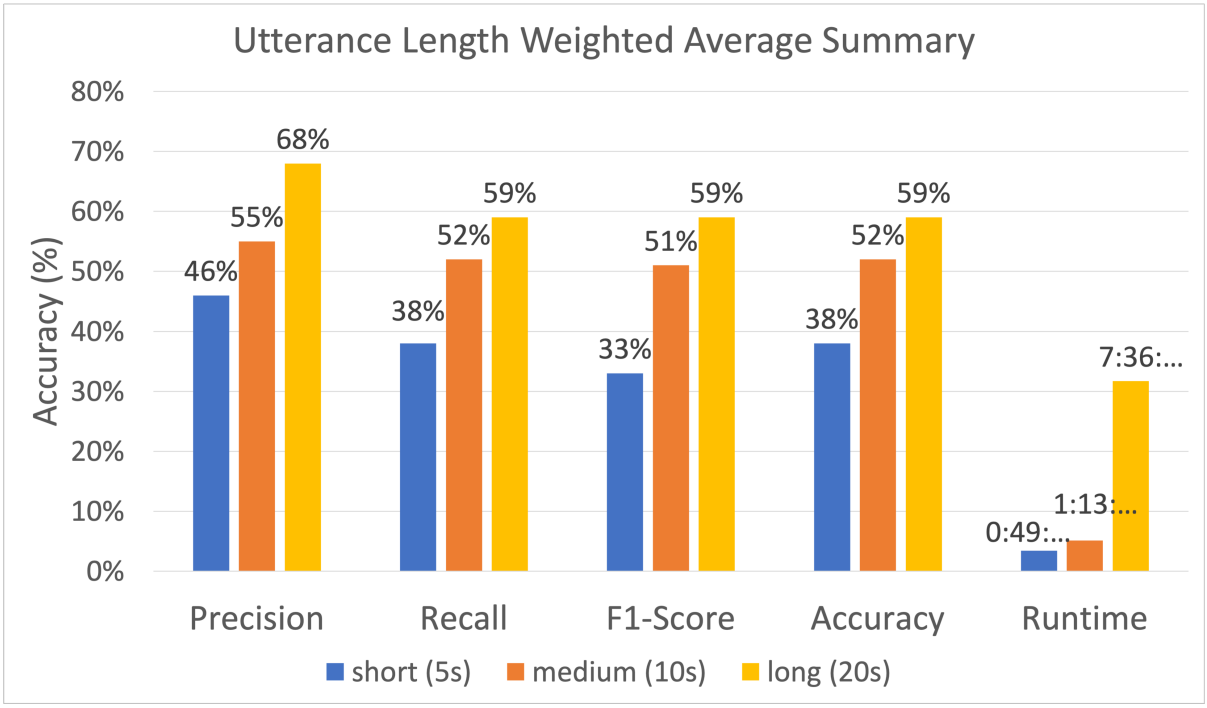


Figure 5.5: Plot comparing key metrics varying training utterance lengths.

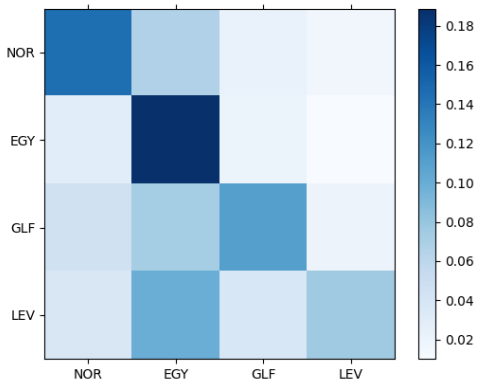


Figure 5.6: Medium Utterances(10s) Normalised Confusion Matrix Colour Map.

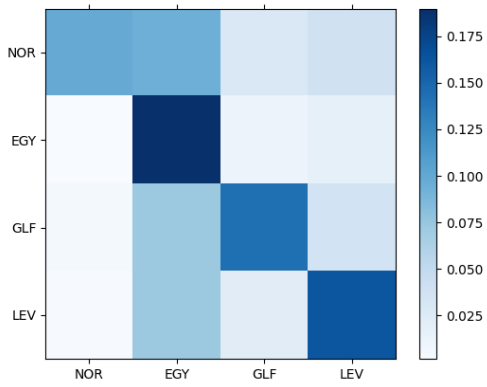


Figure 5.7: Long Utterances(20s) Normalised Confusion Matrix Colour Map.

5.2.4 Fine-tuning with addition of a Downstream Model

The aim of this experiment was to explore the possibility of enhancing the performance of the umbrella DID through the addition of a downstream model. The downstream models tested in this experiment structure is explained in Section 4.2.4. The downstream models were inserted into a XLSR Arabic model and the weights from fine-tuning the model without the downstream model were loaded into the model when it was initalised. The training was performed with 3 GPUs, 24 CPUs with 138GB of memory and 700 files allocated per dialect.The LSTM performs poorly in all metrics as seen in the Figure 5.8, classifying all the test files as Gulf with a F1-Score of 40% as seen in the Table 5.4. The DNN structures outperform the LSTM but underperform compared to having no downstream model at all. Comparing the DNN models of varying layers and more complexities, the simplest 3 layer DNN performs the best overall

with a weighted average F1-Score of 55%. The 4 layer DNN has a marginally higher F1-Score than the 3 and 6 layer for North African (NOR) and Levantine (LEV) dialects. Whilst the results for the 6 layer DNN are comparable to the 3 layer but with lower F1-Scores for Gulf (GLF) and Levantine (LEV) dialects.

Analysing the Figure 5.9, it can be seen compared to having no downstream model, the model with a DNN structure has a steep decrease in it's training loss and increase in its validation loss. The trends in its loss over the epochs show that adding a DNN structure leads to overfitting during the fine-tuning process. So, it can be concluded from these findings that adding a downstream model adds unnecessary complexity to the system causing overfitting and decreasing the performance of the DID.

	F1-Score (%)				
	DNN 3 layers	DNN 4 layers	DNN 6 layers	LSTM	No Downstream Model
NOR	64	66	64	0	70
EGY	51	32	51	0	64
GLF	57	56	53	40	62
LEV	50	52	48	0	54
Weighted Average	55	52	54	10	62

Table 5.4: F1-Score of Varying Downstream Models

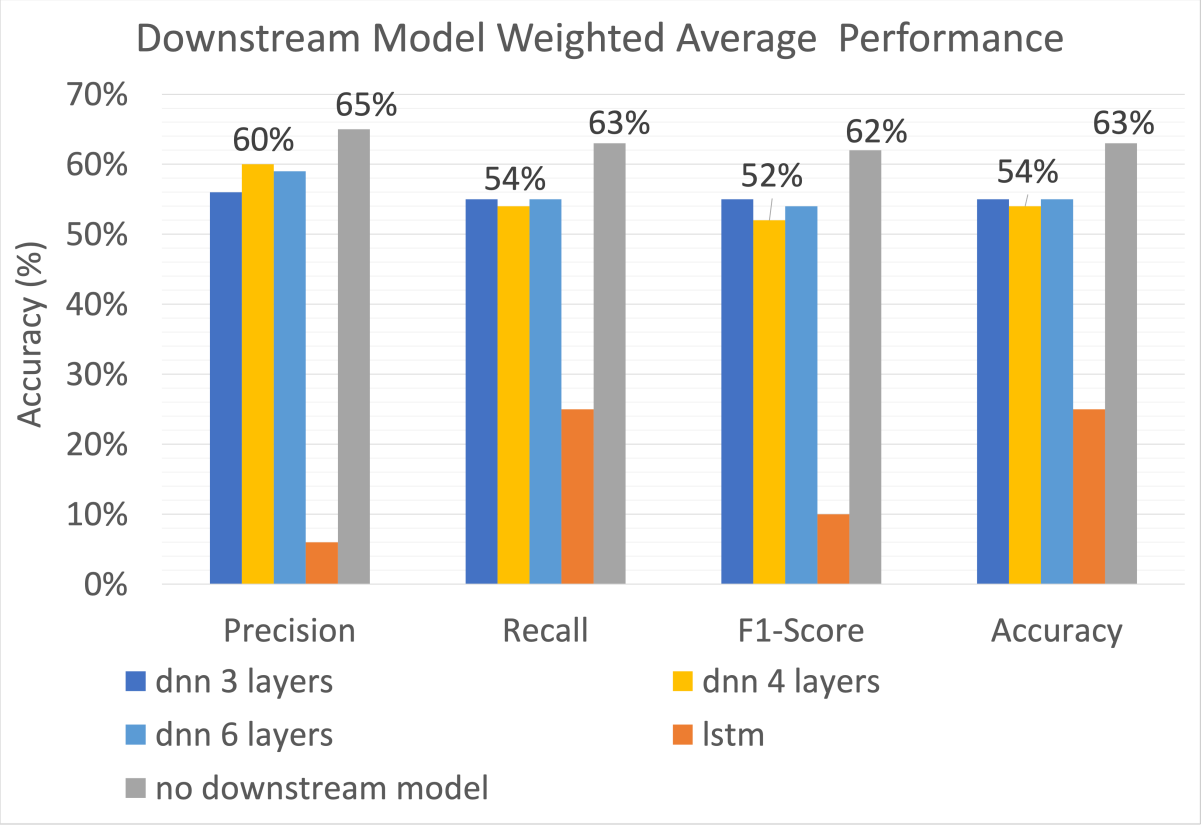


Figure 5.8: Downstream model Summary of key metrics.

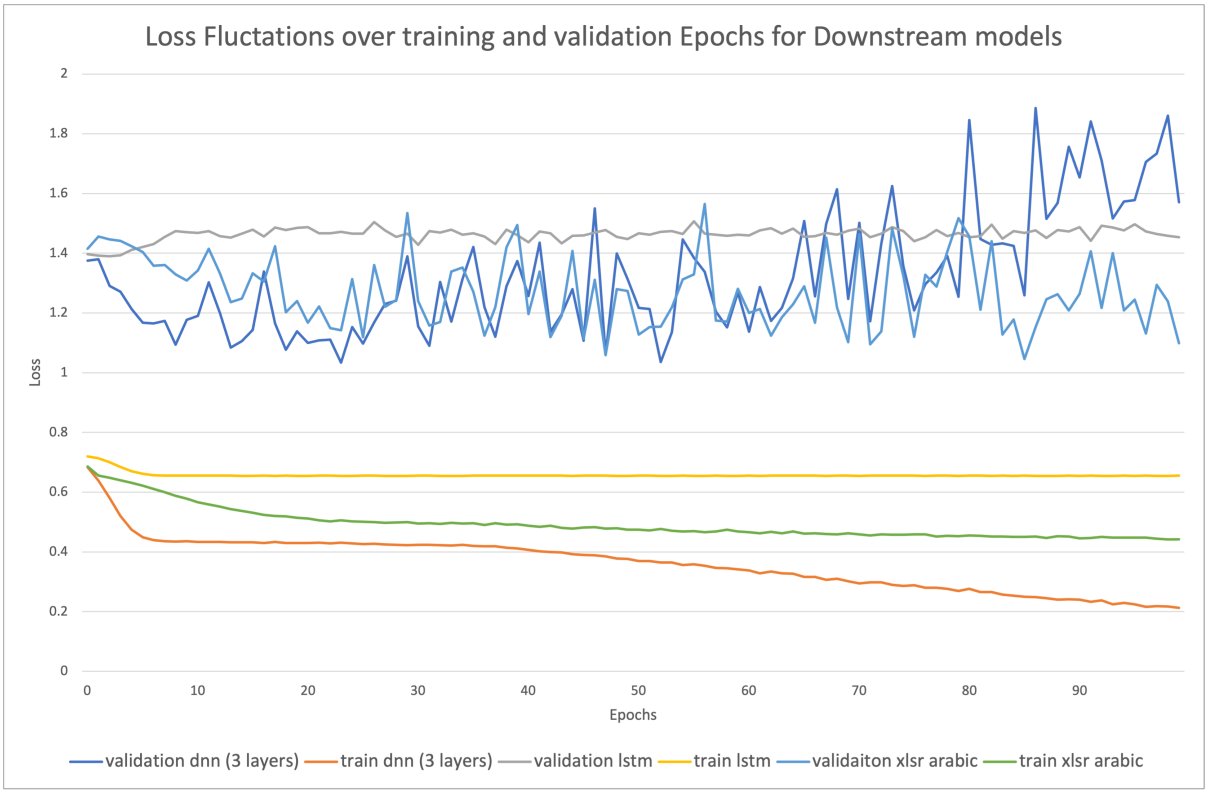
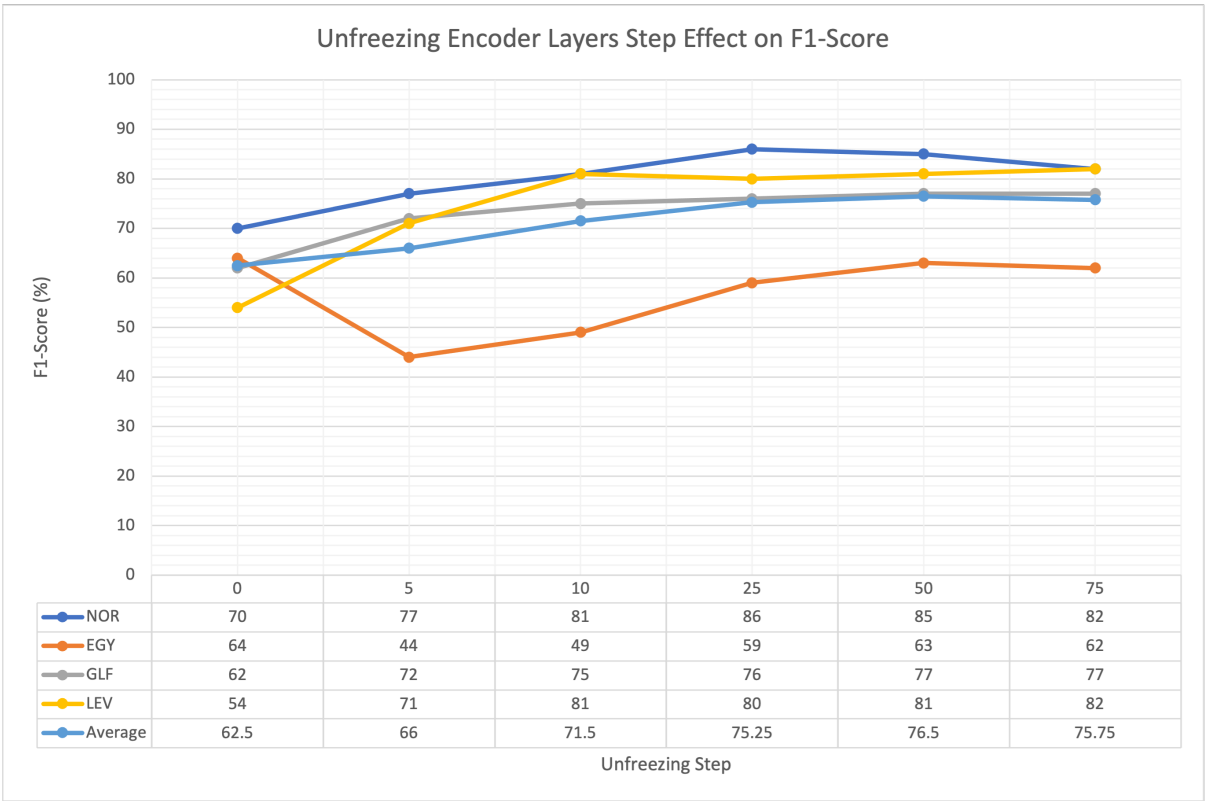


Figure 5.9: Downstream model types change of loss.

5.2.5 Fine-tuning with Encoder layers

A significant portion of the wav2vec pretrained model structure its encoder layers. Wav2vec and thereby XLSR Arabic has 12 encoder layers, in previous experimentation these layers were frozen. This portion of experimentation explores unfreezing those layers gradually while the model is fine-tuning.The training used the XSLR Arabic pretrained model with 3 GPUs, 24 CPUs, 138GB of memory and 700 files allocated per dialect. The step at which the layers are unfrozen is varied in testing and the results are shown in the Figure 5.10. Analysing these results it was found that increases the step also increased the F1-Score until step 50 in which the F1-Score began to plateau and decrease. The highest F1-Scores overall was when the unfreezing step was set to 50 with a weighted average F1-Score of 76%, it performed well on most dialects with the highest being North African (NOR) at 85% and struggling with Egyptian with an F1-Score of 63%. This model was the most effective umbrella DID developed out of all the experiments, it's final classification report is shown in Table 5.2.5 and confusion matrix colour map in Figure 5.11.

	Precision (%)	**Recall (%)**	**F1-Score (%)**	**Support**
NOR	86	84	85	100
EGY	79	52	63	100
GLF	69	86	77	100
LEV	76	86	81	100
Accuracy			77	400
Macro Average	78	77	76	400
Weighted Average	78	77	76	400



	0	5	10	25	50	75
NOR	70	77	81	86	85	82
EGY	64	44	49	59	63	62
GLF	62	72	75	76	77	77
LEV	54	71	81	80	81	82
Average	62.5	66	71.5	75.25	76.5	75.75

Figure 5.10: Plot showing unfreeze encoder layer step effect on F1-Score.

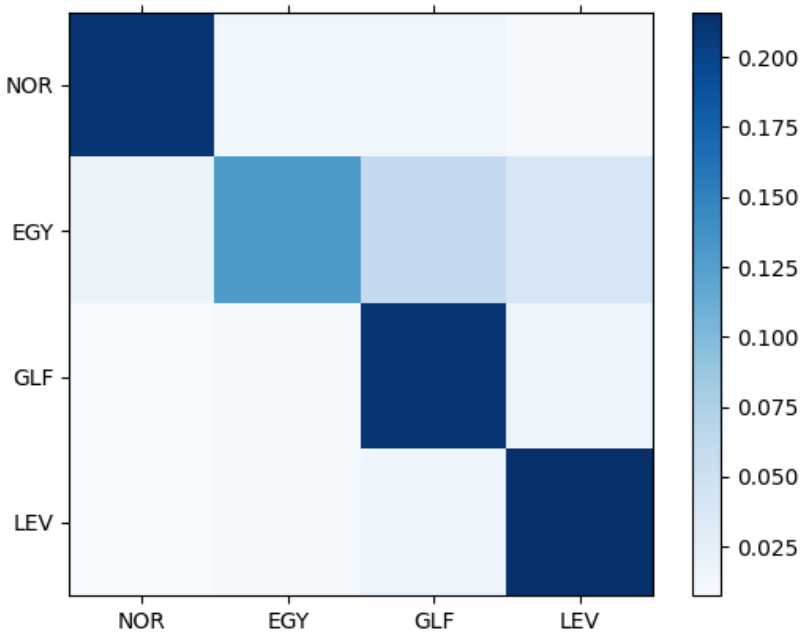


Figure 5.11: Step 50 Umbrella DID Normalised Confusion Matrix Colour Map.

F-1 Score (%)			
	No Adjustments	No LEV	Double EGY
NOR	85	68	69
EGY	63	68	78
GLF	77	73	61
LEV	81	72	66
Weighted Average	76	71	70

Table 5.5: Exploring Counteracting Data Biases

5.3 Counteracting Data Bias

Two tests were conducted to observe the significance of biases from the pretrained model on the final model's performance. As well as testing to see if these biases could be counteracted by providing a disproportionate amount of data per dialect in the fine-tuning files. The training used the XSLR Arabic pretrained model with 3 GPUs, 24 CPUs, 138GB of memory and 700 files allocated for each of the unmodified dialects.

Fine-tuning without Levantine Arabic

Inherited from the XLSR Arabic's pretraining is a data bias towards Levantine Arabic. To test the effects of this bias on the performance of the umbrella DID all training, validation and test files from the Levantine dialect were withheld from the pipeline. The hypothesis was that if the F1-Scores significantly improved, the bias inherited from the pretrained model may have limited the performance of the DID. Looking at the normalised colour map 5.3 and the results in the Table 5.3 it can be seen that removing Levantine did not improve the F1-Score. Removing Levantine decreased the F1-Score of North African (NOR) identification by 17%, misidentifying the audio clips as Egyptian (EGY) 22.8% and Gulf (GLF) 38.9% of the time. Whilst the weighted average F1-Score without Levantine was 71%, 5% less than the standard set of data with an average of 76%. Hence, it was found that the bias from the pretrained model was not significant in determining the final performance of the Umbrella DID.

Fine-tuning with double the amount of Egyptian Arabic

It was observed in Figure 5.11 that the umbrella DID was challenged the most with identifying the Egyptian (EGY) dialect from the other umbrella dialects. It was proposed that doubling the training data given to the system to fine-tune the model could improve its performance. The training data provided for Egyptian was increased from 700 files to 1400. Although, observing the results in the Table 5.3 and the Figure 5.3, it can be seen that this doesn't improve the performance model. Increasing the Egyptian (EGY) data increases the F1-Score of Egyptian from 63% to 78% but the F1-Scores for all the other dialects decrease. The greatest drop in F1-Score can be seen in North African (NOR) going from 85% to 69%. So, increasing the data for a particular class does not improve the system's ability to identify that class.

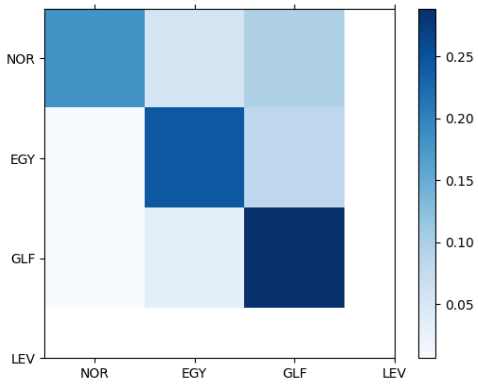


Figure 5.12: No Levantine Normalised Confusion Matrix Colour Map.

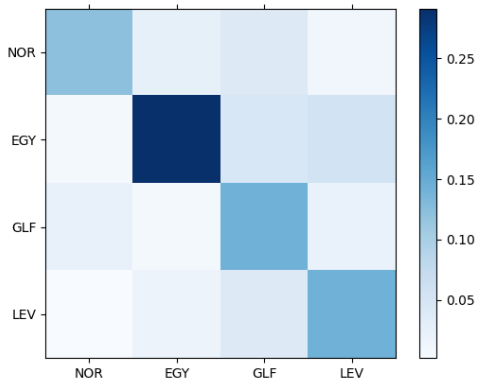


Figure 5.13: Doubled Egyptian Training Data Normalised Confusion Matrix Colour Map.

5.4 Assessing Robustness

Through the process of developing the highest performance Umbrella DID design was the XLSR Arabic model with the encoder layers fined tuned at step 50. Taking this model its robustness is then tested by observing the minimum amount of training files needed to fine tune the model. As well as assessing how well the model performs when the trained model is then further fine-tuned for the downstream task of a regional dialect identifier.

5.4.1 Amount of Training Data

Using 3 GPUs, 24 CPUs, 138GB of memory, the umbrella DID was fine-tuned with file amounts ranging from 25 to 1000. In the Figure 5.14 it can be seen that the F1-Score doubles between 25 and 50 training files per dialect, going from a weighted average F1-Score of 26% to 42%. The F1-Score plateaus slightly and increases slowly between 200 and 400 files. Then significantly increasing at 600 to a weighted average F1-Score of 66% before plateauing again. From this it can be attained that it is sufficient to provide around 600 files per class without the model's performance fluctuating significantly.

	F-1 Score (%)							
Number of Files	**25**	**50**	**100**	**200**	**400**	**600**	**800**	**1000**
NOR	0	33	57	54	63	74	71	74
EGY	40	44	23	29	40	52	57	52
GLF	17	48	52	57	63	70	61	70
LEV	46	43	45	51	53	67	69	67
Weighted Average	26	42	44	48	55	66	65	66

Table 5.6: F1-Scores of Varying Amounts of Training Files

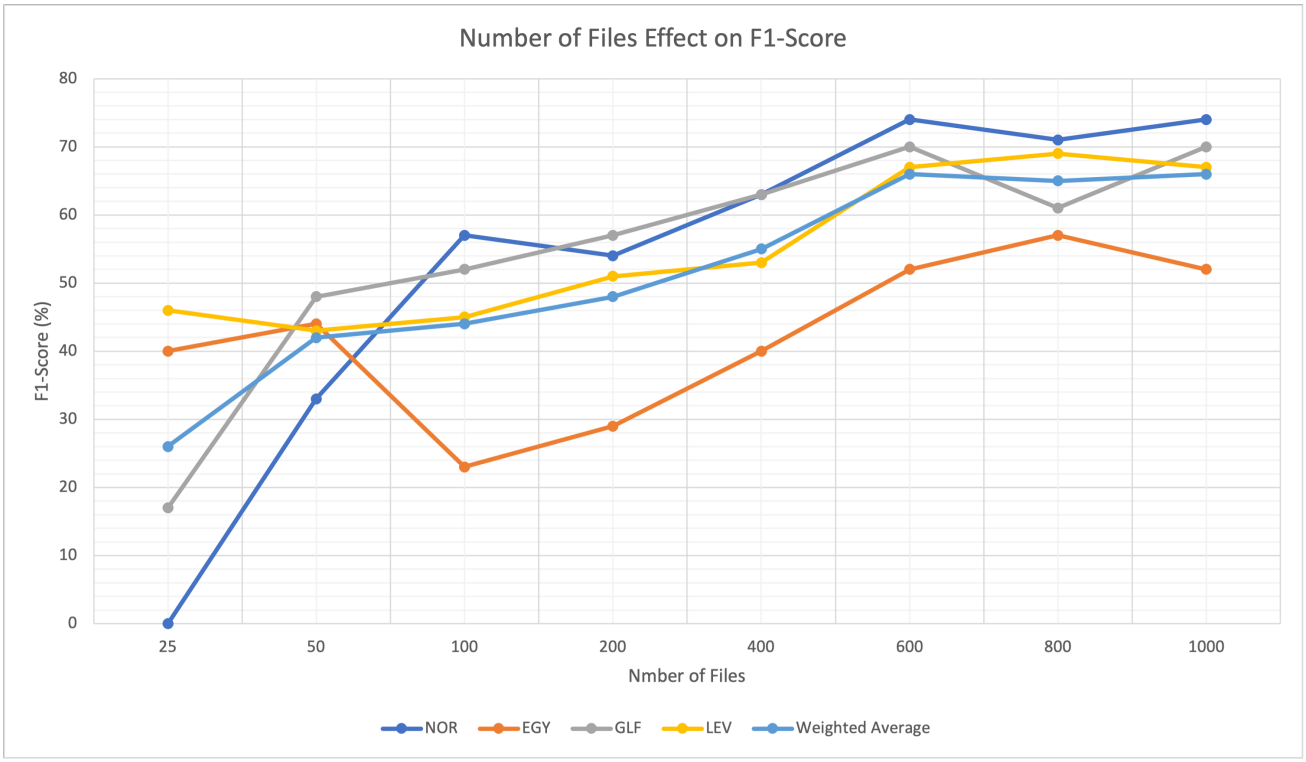


Figure 5.14: Number of Training Files Effect on F1-Score.

5.4.2 Adaptability to Regional DID

The final test conducted was testing the DID's adaptability to being used in a finer grain dialect identifying task. The number of classes on the output layer were increased from 4 to 17. The model was then trained on 3 GPUs, 24 CPUs, 138GB of memory with 700 files per dialect making the total number of training files 11900. As seen in the classification report in Table 5.4.2, the final weighted average F1-Score of the regional DID was 58% with the model performing better on some dialects more than others. The Regional Arabic DID excelled at identifying Iraqi (IRQ) Arabic with an F1-Score of 96% while struggling the most with Egyptian (EGY), Sudan (SDN), Saudi (SAU) and Algerian (DZA). Inspecting the miss identifications 5.15, Saudi (SAU) was most likely to misidentified as KWT, ARE, QAT OMN and YEM, all of which are in the Gulf region. This misidenfication from a dialect regionally close is also seen with Algerian (DZA) which is most likely to be misidentified as Moroccan (MAR). From this it can be inferred that the regional DID while working well for more distinct dialects finds it challenging to differentiate linguistically similar regional dialects.

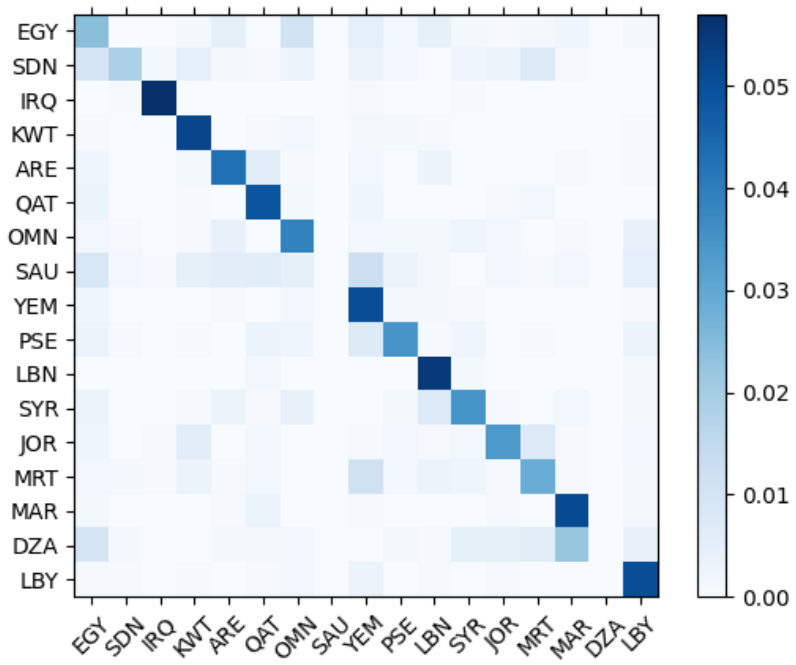


Figure 5.15: Regional DID Normalised Confusion Matrix Colour Map.

5.5 Summary of Results

Comparing the performance of the umbrella and regional DID construct in this thesis to traditional methods. It was found that a pretrained model and a transfer learning approach to designing a regional DID under performed with an accuracy of 62% compared to the paper [38] in where they were able to achieve an accuracy of 85.1% . Although, in comparison to a phonetic approach discussed in the paper [17], the transfer learning approach was able to achieve higher F1-Scores with 10s utterances. As shown in Table 5.5 the transfer learning DID design in this thesis had a higher F1-Score for Levantine (LEV) by 3%, Gulf (GLF) by 21% and Iraqi (IRQ) by 40%. With only Egyptian (EGY) being better represented through the phonetic approach with it's F1-Score of 84% compared to the transfer learning's 63%.

Finally, to summarise the findings of the experimentation conducted during this thesis and presented in this chapter:

1. It was found that a pretrained model that was pretrained with data similar to the downstream task was effective approach to transfer learning, although adapting pretrained models fine-tuned on similar downstream tasks also showed promise. In the case explored in this thesis in designing an Arabic DID, XLSR Arabic, a pretrained model that was pretrained on 53 languages and Levantine Arabic produced the most successful DID model. (Section 5.2.1)

2. Removing noise from input files did not improve the performance of pipelines due to the distortions it caused to the audio. (Section 5.2.2)

3. Longer utterance lengths were more accurate but using files longer than 10s was unsustainable with the resources available as longer audio clips were not able to be processed in a timely manner. (Section 5.2.3)

CLASSIFICATION REPORT				
	Precision	Recall	F1-Score	Support
EGY	32%	41%	36%	100
SDN	74%	32%	45%	100
IRQ	95%	97%	96%	100
KWT	68%	89%	77%	100
ARE	66%	73%	69%	100
QAT	64%	83%	72%	100
OMN	53%	66%	59%	100
SAU	0%	0%	0%	100
YEM	50%	86%	63%	100
PSE	69%	59%	63%	100
LBN	69%	93%	79%	100
SYR	64%	59%	61%	100
JOR	70%	57%	63%	100
MRT	53%	49%	51%	100
MAR	62%	87%	72%	100
DZA	0%	0%	0%	100
LBY	66%	86%	74%	100
Accuracy			62%	1700
Macro Average	56%	62%	58%	1700
Weighted Average	56%	62%	58%	1700

Table 5.7: Regional DID Classification Report

F1-Score (%)		
	Transfer Learning DID	Phonematic Model [17]
LEV	81	78
GLF	77	56
EGY	63	84
IRQ	96	56

Table 5.8: Regional DID Classification Report

4. Unfreezing and fine-tuning encoder layers for the last 50 epochs improved the system significantly. (Section 5.2.5)

5. Including a downstream model showed promise, but the added complexity to the system created diminishing returns in improvement. (Section 5.2.4)

6. Inherited biases from the pretrained don't have a significant effect on the final performance of the Arabic DID systems and training with unbalanced amount of data was not effective at correcting performance differences between the dialect classes. (Section 5.3)

7. The minimum amount of training data needed to effectively train an Arabic DID is around 500 files although significant improvements to the transfer learning DID's performance for this result to be conclusive (Section)

Chapter 6

Conclusion

The paper has investigated the challenges designing a reliable Arabic DID and the need for more accurate systems. It has provided the relevant background information on the theory of DIDs, the wav2vec model and a literature review of current methodologies for LID and DID systems. This thesis has outlined the proposed method, tested it through experimentation and analysed the results. The report has provided some suggestions for future work and discussed the broader impact the thesis may have. Finally, this thesis has shown for the first time that a transfer learning approach could be a potential method of overcoming the challenges of both regional and umbrella Arabic Dialectical Identification (DID).

6.1 Future Work

The work presented in this thesis provides insight into further areas of exploration and future work. Some next steps include:

- Further experiment with downstream model structures other than those explored in Section 5.2.4. Such as adding a CNN or BiLSTM network.
- Conduct more robust testing and training using audio files which contain code switching between dialects.
- Allocate more resources to train with longer audio files.
- Explore adding contextual identification to the system pipeline, by using the probabilities of adjacent utterances from the same audio file. Then taking them into consideration when predicting an utterance's class. This grouping method could increase performance of the DID.
- Test against multilingual datasets, mixed with dialectal Arabic to determine the system's effectiveness at distinguishing Dialectal Arabic from other languages.
- Integrate the Arabic DID with a segmentation system where live audio is able to be segmented into utterances and the dialect present identified.

6.2 Broader Impact

This thesis has shown that transfer learning with a pretrained model designed for one task can be fine-tuned for a loosely related downstream task. It has achieved this by implementing

an Arabic DID with pretrained models originally designed for ASR. Transfer learning can be leveraged to improve the performance of downstream tasks with low resource datasets. While this thesis has a focus for the use case of dialectal Arabic, it could be directly applied to creating a LID or DID for other low resource languages and dialects.

The methods and theory of this thesis should be further examined for other limited resourced use cases and downstream tasks.

Bibliography

[1] XLS-R: Self-supervised speech processing for 128 languages, November.

[2] Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. NADI 2020: The First Nuanced Arabic Dialect Identification Shared Task. *arXiv:2010.11334 [cs]*, November 2020. arXiv: 2010.11334.

[3] Musatafa Abbas Abbood Albadr, Sabrina Tiun, Fahad Taha AL-Dhief, and Mahmoud A. M. Sammour. Spoken language identification based on the enhanced self-adjusting extreme learning machine approach. *PLOS ONE*, 13(4):e0194770, April 2018.

[4] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. The first high-performance self-supervised algorithm that works for speech, vision, and text, January 2022.

[5] David Alfter. Language Segmentation. *arXiv:1510.01717 [cs]*, October 2015. arXiv: 1510.01717.

[6] Zainab Alhakeem and Hong-Goo Kang. Confidence Learning from Noisy Labels for Arabic Dialect Identification. In *2021 36th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, pages 1–4, 2021.

[7] Ahmed Ali, Shammur Chowdhury, Amir Hussein, and Yasser Hifny. Arabic Code-Switching Speech Recognition using Monolingual Data. *arXiv:2107.01573 [cs, eess]*, July 2021. arXiv: 2107.01573.

[8] Ahmed Ali, Suwon Shon, Younes Samih, Hamdy Mubarak, Ahmed Abdelali, James Glass, Steve Renals, and Khalid Choukri. The MGB-5 Challenge: Recognition and Dialect Identification of Dialectal Arabic Speech. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1026–1033, 2019.

[9] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. Common Voice: A Massively-Multilingual Speech Corpus. *arXiv:1912.06670 [cs]*, March 2020. arXiv: 1912.06670.

[10] Kartik Audhkhasi, Andrew Rosenberg, Abhinav Sethy, Bhuvana Ramabhadran, and Brian Kingsbury. End-to-end ASR-free keyword search from speech. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4840–4844, New Orleans, LA, March 2017. IEEE.

[11] Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski,

Alexis Conneau, and Michael Auli. XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale. *arXiv:2111.09296 [cs, eess]*, December 2021. arXiv: 2111.09296.

[12] Alexei Baevski, Michael Auli, and Abdelrahman Mohamed. Effectiveness of self-supervised pre-training for speech recognition. *arXiv:1911.03912 [cs]*, May 2020. arXiv: 1911.03912.

[13] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language. *arXiv:2202.03555 [cs]*, February 2022. arXiv: 2202.03555.

[14] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. *arXiv:2006.11477 [cs, eess]*, October 2020. arXiv: 2006.11477.

[15] Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, and Andrew McCallum. Self-Supervised Meta-Learning for Few-Shot Natural Language Classification Tasks. *arXiv:2009.08445 [cs]*, November 2020. arXiv: 2009.08445.

[16] Fadi Biadsy, Julia Hirschberg, and Daniel Ellis. Dialect and Accent Recognition Using Phonetic-Segmentation Supervectors. pages 745–748, January 2011.

[17] Fadi Biadsy, Julia Hirschberg, and Nizar Habash. Spoken Arabic dialect identification using phonotactic modeling. In *Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages - Semitic '09*, page 53, Athens, Greece, 2009. Association for Computational Linguistics.

[18] Hynek Bořil, Abhijeet Sangwan, and John H. L. Hansen. Arabic dialect identification - "is the secret in the silence?" and other observations. In *Interspeech 2012*, pages 30–33. ISCA, September 2012.

[19] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep Clustering for Unsupervised Learning of Visual Features. *arXiv:1807.05520 [cs]*, March 2019. arXiv: 1807.05520.

[20] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing. *arXiv:2110.13900 [cs, eess]*, January 2022. arXiv: 2110.13900.

[21] Shammur Absar Chowdhury, Amir Hussein, Ahmed Abdelali, and Ahmed Ali. Towards One Model to Rule All: Multilingual Strategy for Dialectal Code-Switching Arabic ASR. *arXiv:2105.14779 [cs, eess]*, July 2021. arXiv: 2105.14779.

[22] Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training. *arXiv:2108.06209 [cs, eess]*, September 2021. arXiv: 2108.06209.

[23] Savelie Cornegruta, Robert Bakewell, Samuel Withey, and Giovanni Montana. Modelling Radiological Language with Bidirectional Long Short-Term Memory Networks. *arXiv:1609.08409 [cs, stat]*, September 2016. arXiv: 1609.08409.

[24] Bilal Dendani, Halima Bahi, and Toufik Sari. Self-Supervised Speech Enhancement for Arabic Speech Recognition in Real-World Environments. *Traitement du Signal*, 38(2):349–358, April 2021.

[25] Zhiyun Fan, Meng Li, Shiyu Zhou, and Bo Xu. Exploring wav2vec 2.0 on speaker verification and language identification. 2020. Publisher: arXiv Version Number: 2.

[26] Nizar Habash, Owen Rambow, Mona Diab, and Reem Kanjawi-Faraj. Guidelines for annotation of Arabic dialectness. In *Proceedings of the LREC Workshop on HLT & NLP within the Arabic world*, pages 49–53, 2008.

[27] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. CNN Architectures for Large-Scale Audio Classification, January 2017. Number: arXiv:1609.09430 arXiv:1609.09430 [cs, stat].

[28] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *arXiv:2106.07447 [cs, eess]*, June 2021. arXiv: 2106.07447.

[29] Amir Hussein, Shinji Watanabe, and Ahmed Ali. Arabic Speech Recognition by End-to-End, Modular Systems and Human. *arXiv:2101.08454 [cs, eess]*, June 2021. arXiv: 2101.08454.

[30] Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. Automatic Language Identification in Texts: A Survey. *arXiv:1804.08186 [cs]*, November 2018. arXiv: 1804.08186.

[31] Jay Alammar. The Illustrated Transformer.

[32] Jonathan Bgn. HuBERT: How to Apply BERT to Speech, Visually Explained, October 2021.

[33] Jonathan Bgn. An Illustrated Tour of Wav2vec 2.0, October 2021.

[34] Muhammad Khalifa, Hesham Hassan, and Aly Fahmy. Zero-Resource Multi-Dialectal Arabic Natural Language Understanding. *International Journal of Advanced Computer Science and Applications*, 12(3), 2021. arXiv: 2104.06591.

[35] Daniel Korzekwa, Jaime Lorenzo-Trueba, Szymon Zaporowski, Shira Calamaro, Thomas Drugman, and Bozena Kostek. Mispronunciation Detection in Non-native (L2) English with Uncertainty Modeling. *arXiv:2101.06396 [cs, eess]*, February 2021. arXiv: 2101.06396.

[36] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv:1909.11942 [cs]*, February 2020. arXiv: 1909.11942.

[37] Hunter Lang and Hoifung Poon. Self-supervised self-supervision by combining deep learning and probabilistic logic. *arXiv:2012.12474 [cs, stat]*, December 2020. arXiv: 2012.12474.

[38] Wanqiu Lin, Maulik Madhavi, Rohan Kumar Das, and Haizhou Li. Transformer-based Arabic Dialect Identification. *arXiv:2011.00699 [eess]*, November 2020. arXiv: 2011.00699.

[39] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*, July 2019. arXiv: 1907.11692.

[40] Ignacio Lopez-Moreno, Javier Gonzalez-Dominguez, David Martinez, Oldřich Plchot, Joaquin Gonzalez-Rodriguez, and Pedro J. Moreno. On the use of deep feedforward neural networks for automatic language identification. *Computer Speech & Language*, 40:46–59, November 2016.

[41] Khaled Lounnas, Mourad Abbas, and Mohamed Lichouri. Building a Speech Corpus based on Arabic Podcasts for Language and Dialect Identification. September 2019.

[42] Xiaoxiao Miao and Ian McLoughlin. LSTM-TDNN with convolutional front-end for Dialect Identification in the 2019 Multi-Genre Broadcast Challenge. *arXiv:1912.09003 [cs, eess]*, December 2019. arXiv: 1912.09003.

[43] Xiaoxiao Miao, Ian McLoughlin, and Yonghong Yan. A New Time-Frequency Attention Mechanism for TDNN and CNN-LSTM-TDNN, with Application to Language Identification. In *Interspeech 2019*, pages 4080–4084. ISCA, September 2019.

[44] Michael Ellis. *Accent Identification for English Speakers*. PhD thesis, University of New South Wales.

[45] Omar Mohamed and Salah A. Aly. Arabic Speech Emotion Recognition Employing Wav2vec2.0 and HuBERT Based on BAVED Dataset. *arXiv:2110.04425 [cs]*, October 2021. arXiv: 2110.04425.

[46] Jama Hussein Mohamud, Lloyd Acquaye Thompson, Aissatou Ndoye, and Laurent Besacier. Fast Development of ASR in African Languages using Self Supervised Speech Representation Learning. *arXiv:2103.08993 [cs, eess]*, March 2021. arXiv: 2103.08993.

[47] Hamdy Mubarak, Amir Hussein, Shammur Absar Chowdhury, and Ahmed Ali. QASR: QCRI Aljazeera Speech Resource – A Large Scale Annotated Arabic Speech Corpus. *arXiv:2106.13000 [cs, eess]*, June 2021. arXiv: 2106.13000.

[48] Scott Novotney, Rich Schwartz, and Sanjeev Khudanpur. Unsupervised Arabic dialect adaptation with self-training. In *Interspeech 2011*, pages 541–544. ISCA, August 2011.

[49] Pascal Fivian and Dominique Reiser. *Speech Classification using wav2vec 2.0*. PhD thesis, ZHAW School of Engineering, Switzerland, June 2021.

[50] Charles Perreault and Sarah Mathew. Dating the Origin of Language Using Phonemic Diversity. *PLoS ONE*, 7(4):e35289, April 2012.

[51] Jouni Pohjalainen, Fabien Fabien Ringeval, Zixing Zhang, and Björn Schuller. Spectral and Cepstral Audio Noise Reduction Techniques in Speech Emotion Recognition. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 670–674, Amsterdam The Netherlands, October 2016. ACM.

[52] G. Ramesh, C. Shiva Kumar, and K. Sri Rama Murty. Self-Supervised Phonotactic Representations for Language Identification. In *Interspeech 2021*, pages 1514–1518. ISCA, August 2021.

[53] Shauna Revay and Matthew Teschke. Multiclass Language Identification using Deep Learning on Spectral Images of Audio Signals. *arXiv:1905.04348 [cs, eess]*, May 2019. arXiv: 1905.04348.

[54] Mohammad Salameh, Houda Bouamor, and Nizar Habash. Fine-Grained Arabic Dialect Identification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1332–1344, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[55] Younes Samih, Mohammed Attia, Mohamed Eldesouki, Ahmed Abdelali, Hamdy Mubarak, Laura Kallmeyer, and Kareem Darwish. A Neural Architecture for Dialectal Arabic Segmentation. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 46–54, Valencia, Spain, 2017. Association for Computational Linguistics.

[56] Younes Samih, Suraj Maharjan, Mohammed Attia, Laura Kallmeyer, and Thamar Solorio. Multilingual Code-switching Identification via LSTM Recurrent Neural Networks. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 50–59, Austin, Texas, 2016. Association for Computational Linguistics.

[57] Sanket Shah, Sunayana Sitaram, and Rupeshkumar Mehta. FirstWorkshop on Speech Processing for Code-switching in Multilingual Communities: Shared Task on Code-switched Spoken Language Identification. pages 24–28. Microsoft Research India, Microsoft Corporation, October 2020.

[58] Suwon Shon, Ahmed Ali, and James Glass. Convolutional Neural Networks and Language Embeddings for End-to-End Dialect Recognition. *arXiv:1803.04567 [cs, eess]*, April 2018. arXiv: 1803.04567.

[59] Suwon Shon, Ahmed Ali, Younes Samih, Hamdy Mubarak, and James Glass. ADI17: A Fine-Grained Arabic Dialect Identification Dataset. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8244–8248, Barcelona, Spain, May 2020. IEEE.

[60] Suwon Shon, Wei-Ning Hsu, and James Glass. Unsupervised Representation Learning of Speech for Dialect Identification. *arXiv:1809.04458 [cs, eess]*, September 2018. arXiv: 1809.04458.

[61] Guan-Lip Soon, Nur-Hana Samsudin, and Dennis Lim. Evaluating the Effect of Multiple Filters in Automatic Language Identification without Lexical Knowledge. *International Journal of Advanced Computer Science and Applications*, 11(10), 2020.

[62] Andros Tjandra, Diptanu Gon Choudhury, Frank Zhang, Kritika Singh, Alexis Conneau, Alexei Baevski, Assaf Sela, Yatharth Saraf, and Michael Auli. Improved Language Identification Through Cross-Lingual Self-Supervised Learning. *arXiv:2107.04082 [cs, eess]*, October 2021. arXiv: 2107.04082.

[63] Liang-Hsuan Tseng, Yu-Kuan Fu, Heng-Jui Chang, and Hung-yi Lee. Mandarin-English Code-switching Speech Recognition with Self-supervised Speech Representation Models. *arXiv:2110.03504 [cs, eess]*, October 2021. arXiv: 2110.03504.

[64] Michael D. Tyler and Anne Cutler. Cross-language differences in cue use for speech segmentation. *The Journal of the Acoustical Society of America*, 126(1):367–376, July 2009.

[65] Jörgen Valk and Tanel Alumäe. VoxLingua107: a Dataset for Spoken Language Recognition. *arXiv:2011.12998 [eess]*, November 2020. arXiv: 2011.12998.

[66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, December 2017. arXiv:1706.03762 [cs].

[67] Anshul Wadhawan. Dialect Identification in Nuanced Arabic Tweets Using Farasa Segmentation and AraBERT. *arXiv:2102.09749 [cs]*, February 2021. arXiv: 2102.09749.

[68] Yingzhi Wang, Abdelmoumene Boumadane, and Abdelwahab Heba. A Fine-tuned Wav2vec 2.0/HuBERT Benchmark For Speech Emotion Recognition, Speaker Verification and Spoken Language Understanding. *arXiv:2111.02735 [cs, eess]*, November 2021. arXiv: 2111.02735.

[69] Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I. Jeff Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu-Hsien Huang, Wei-Cheng Tseng, Ko-tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung-yi Lee. SUPERB: Speech processing Universal PERformance Benchmark. *arXiv:2105.01051 [cs, eess]*, October 2021. arXiv: 2105.01051.

[70] Omar F. Zaidan and Chris Callison-Burch. Arabic Dialect Identification. *Computational Linguistics*, 40(1):171–202, March 2014.

[71] Chiyu Zhang and Muhammad Abdul-Mageed. No Army, No Navy: BERT Semi-Supervised Learning of Arabic Dialects. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 279–284, Florence, Italy, 2019. Association for Computational Linguistics.

[72] Qian Zhang and John H. L. Hansen. Language/Dialect Recognition Based on Unsupervised Deep Learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(5):873–882, 2018.

[73] Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. A Survey on Negative Transfer. *arXiv:2009.00909 [cs, stat]*, August 2021. arXiv: 2009.00909.

Appendix A

Appendices

A.1 Appendix A - Additional Fine-Tuning Results

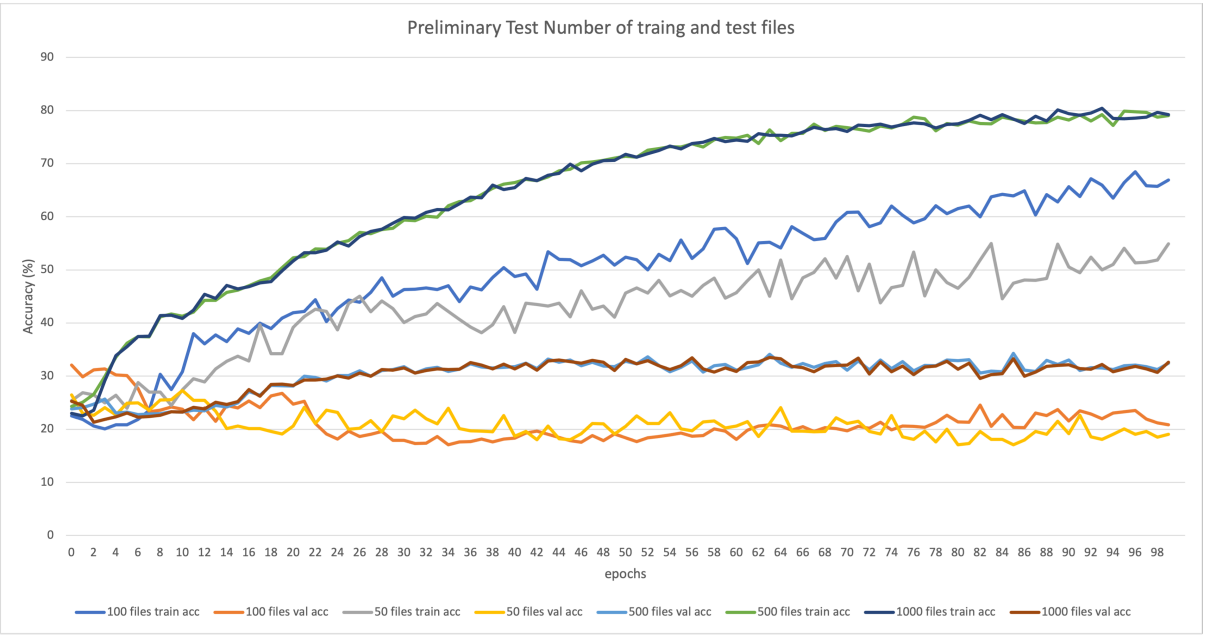


Figure A.1: Preliminary Testing Number of Files.

	No Levantine				Double Egyptian			
	precision	recall	f1-score	support	precision	recall	f1-score	support
NOR	93%	54%	68%	100	78%	61%	69%	100
EGY	93%	54%	68%	100	85%	72%	78%	200
GLF	73%	73%	73%	100	53%	72%	61%	100
LEV	61%	88%	72%	100	61%	71%	66%	100
accuracy			71%	300			70%	500
macro avg	76%	72%	71%	300	69%	69%	68%	500
weighted avg	76%	71%	71%	300	73%	70%	70%	500

Table A.1: Classification Reports: Counteracting Bias

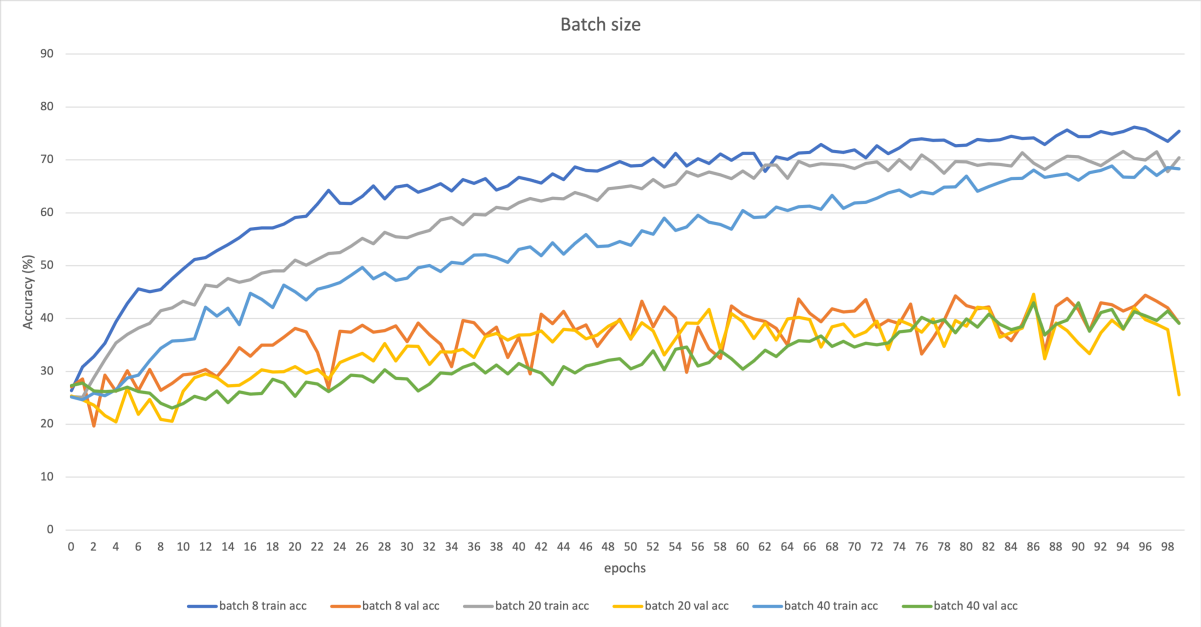


Figure A.2: Preliminary Testing Batch Size.

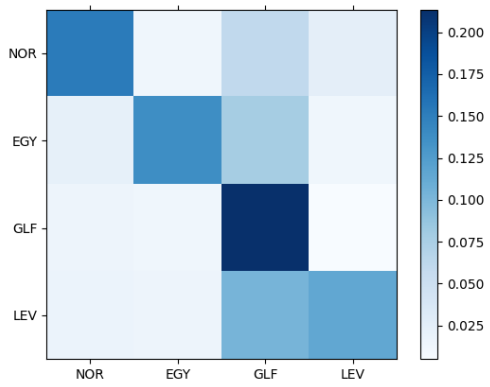


Figure A.3: Hubert
Normalised Confusion Matrix Colour Map.

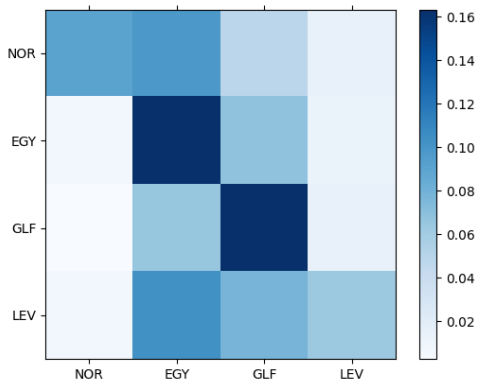


Figure A.4: Wav2vec 2.0 Base
Normalised Confusion Matrix Colour Map.

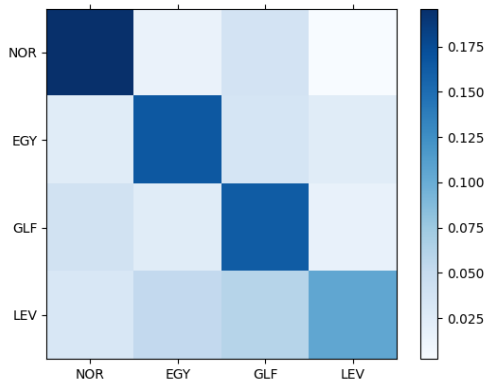


Figure A.5: Wav2vec SID
Normalised Confusion Matrix Colour Map.

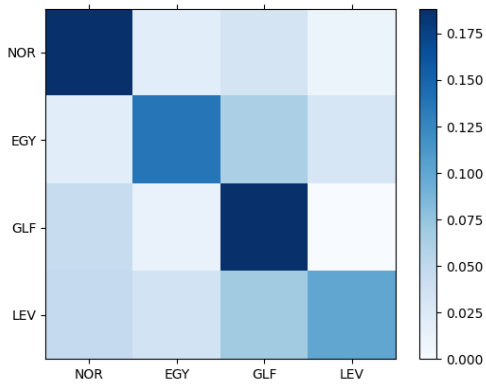


Figure A.6: Wav2vec LID
Normalised Confusion Matrix Colour Map.

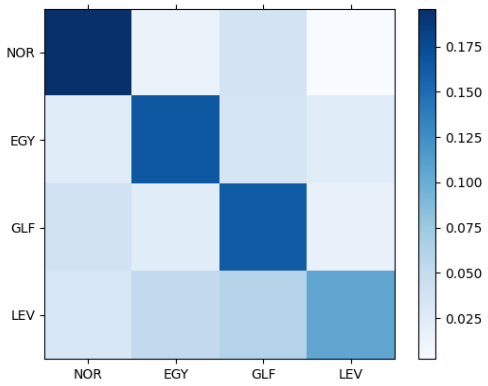


Figure A.7: Wav2vec SID
Normalised Confusion Matrix Colour Map.

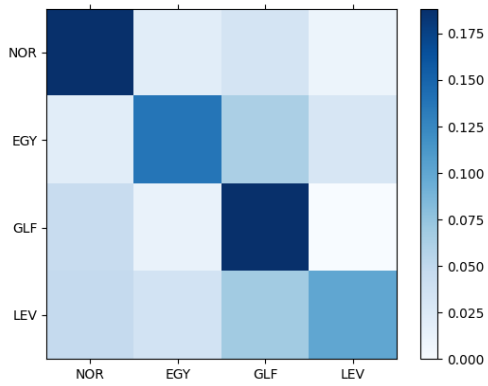


Figure A.8: Wav2vec LID
Normalised Confusion Matrix Colour Map.

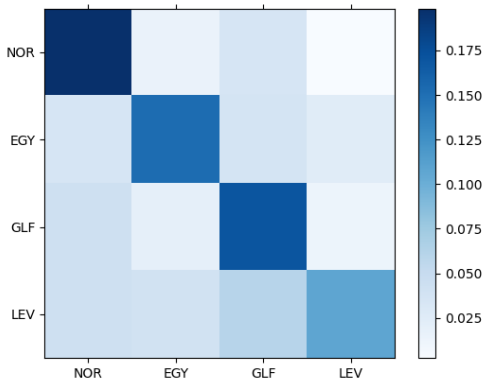


Figure A.9: XLSR Arabic
Normalised Confusion Matrix Colour Map.

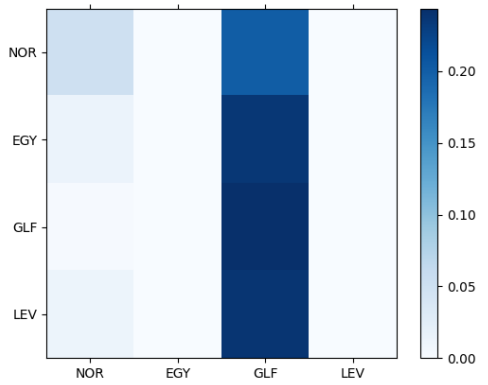


Figure A.10: XLSR
Normalised Confusion Matrix Colour Map.

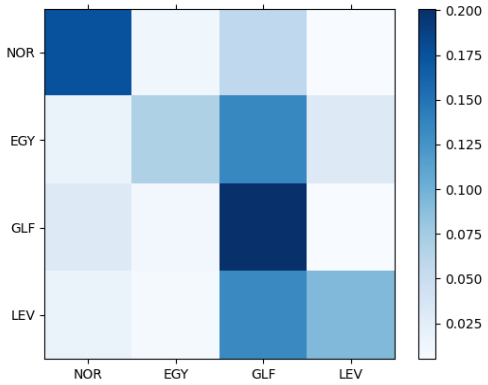


Figure A.11: DNN 3 layer
Normalised Confusion Matrix Colour Map.

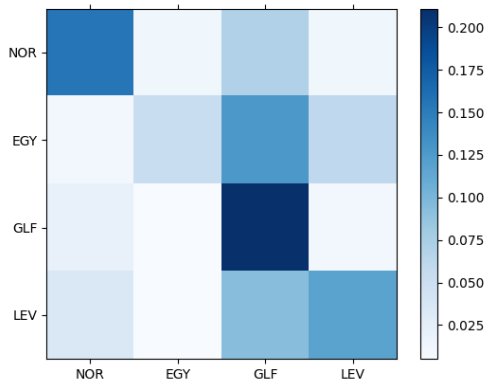


Figure A.12: DNN 4 layer
Normalised Confusion Matrix Colour Map.

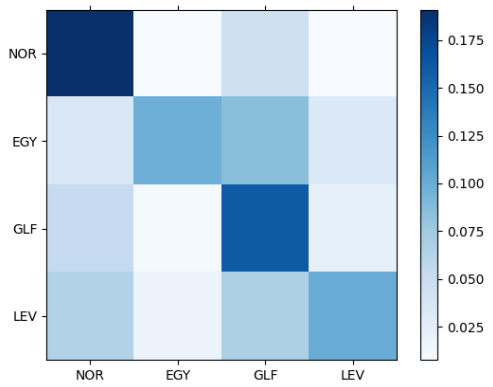


Figure A.13: DNN 6 layer
Normalised Confusion Matrix Colour Map.

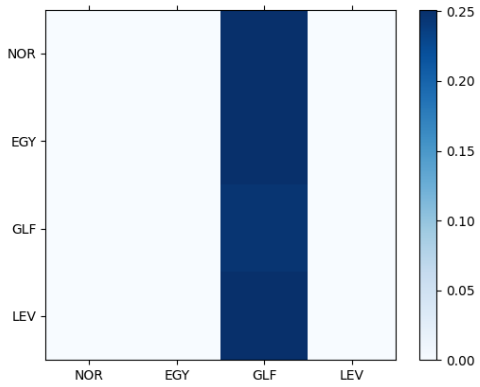


Figure A.14: LSTM
Normalised Confusion Matrix Colour Map.

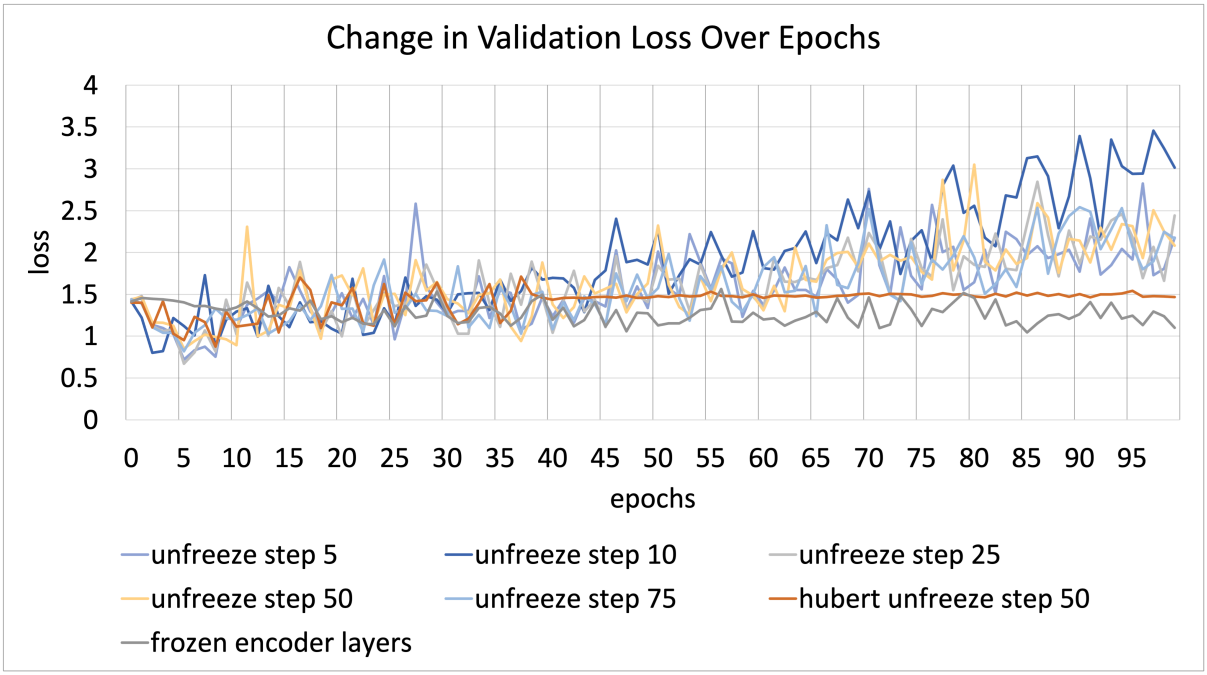


Figure A.15: Unfreezing encoder layers change in validation loss.

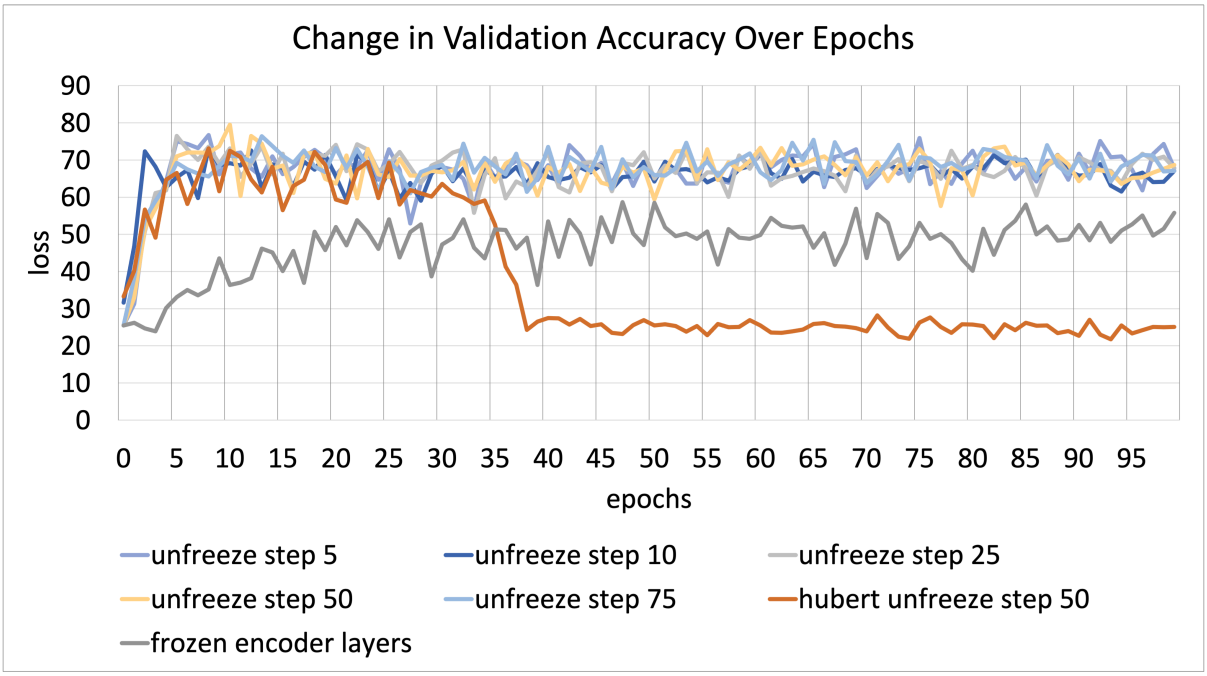


Figure A.16: Unfreezing encoder layers change in validation accuracy.

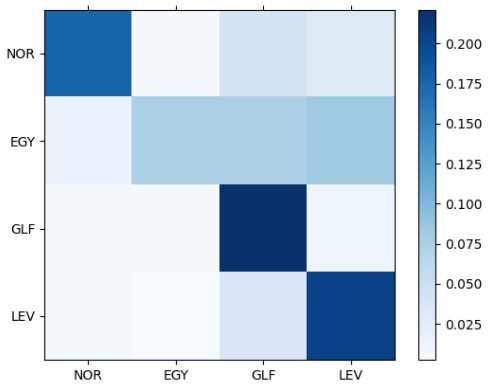


Figure A.17: Unfreeze step 5
Normalised Confusion Matrix Colour Map.

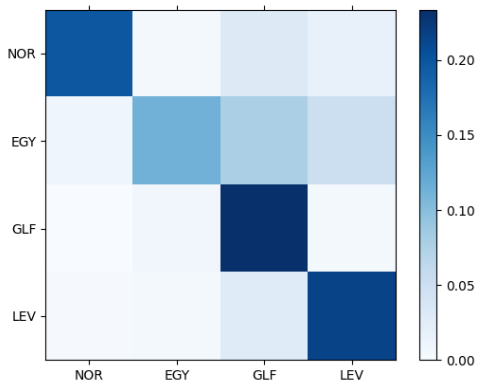


Figure A.18: Unfreeze step 25
Normalised Confusion Matrix Colour Map.

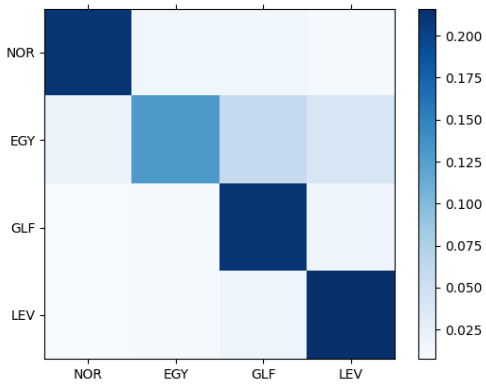


Figure A.19: Unfreeze step 50
Normalised Confusion Matrix Colour Map.

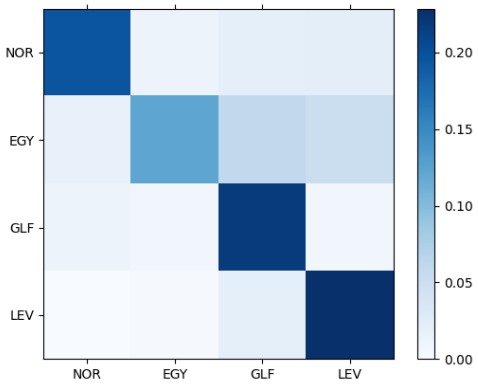


Figure A.20: Unfreeze step 75
Normalised Confusion Matrix Colour Map.

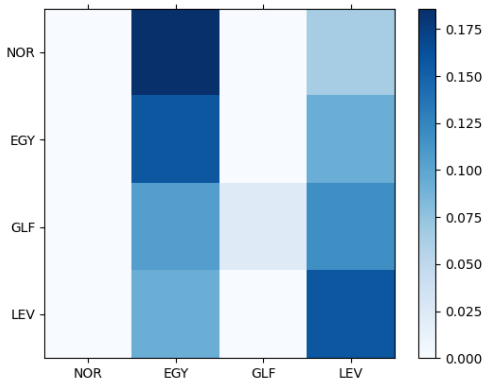


Figure A.21: 25 training Files
Normalised Confusion Matrix Colour Map.

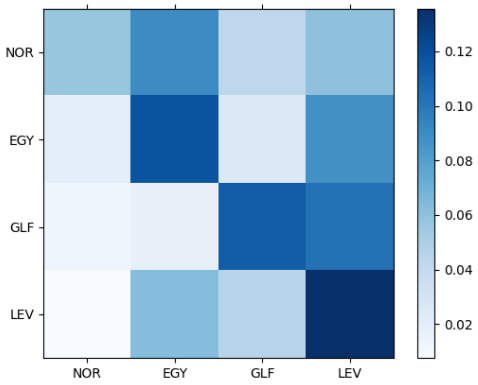


Figure A.22: 50 training Files
Normalised Confusion Matrix Colour Map.

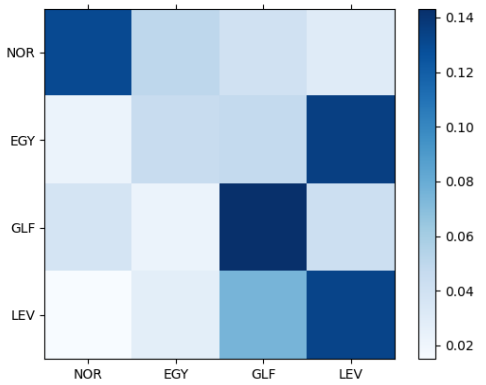


Figure A.23: 100 training Files
Normalised Confusion Matrix Colour Map.

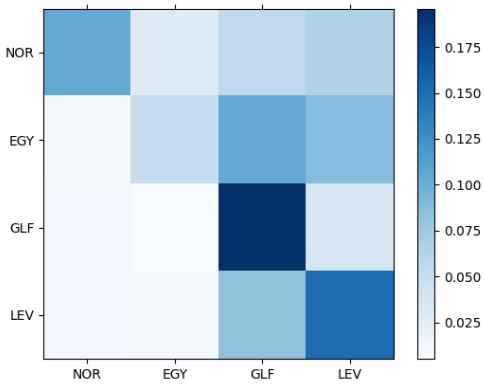


Figure A.24: 200 training Files
Normalised Confusion Matrix Colour Map.

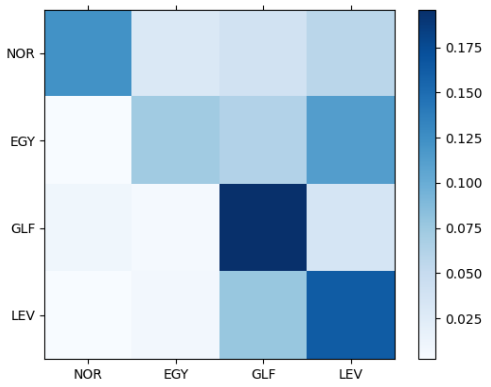


Figure A.25: 400 training Files Normalised Confusion Matrix Colour Map.

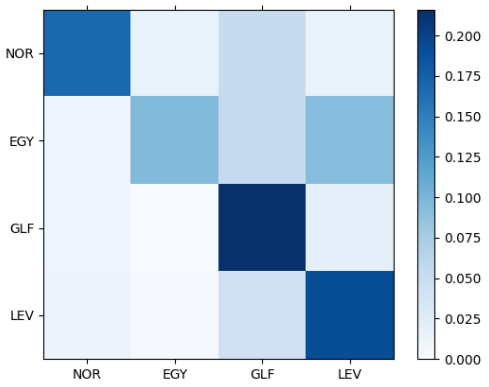


Figure A.26: 600 training Files Normalised Confusion Matrix Colour Map.

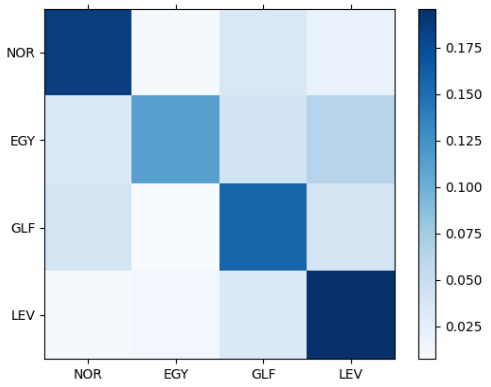


Figure A.27: 800 training Files Normalised Confusion Matrix Colour Map.

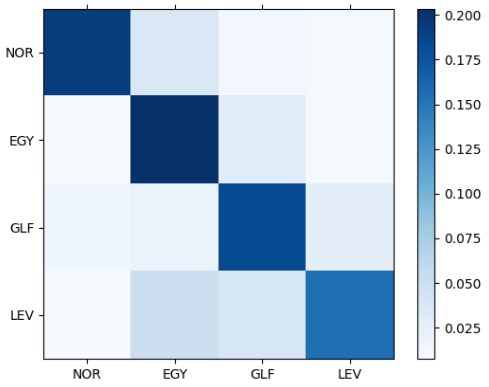


Figure A.28: 1000 training Files Normalised Confusion Matrix Colour Map.