CIS 530
Advanced Data Mining
(Fall 2024)

< Project Report >

END TO END SYSTEM

Regional Language Identification via Audio Accent and Transcript Analysis

Submitted by:

Ayush Tiwari	*02133961*
Trina Xavier	*02102403*
Vinay Shetye	*02018051*

December 13, 2024

1 *Abstract*

Understanding regional dialects in audio is critical for advancing inclusivity in voice technologies. This study focuses on designing a machine learning model to classify regional dialects through audio feature extraction and transcript analysis. Techniques such as Mel-Frequency Cepstral Coefficients (MFCCs), Mel Spectrogram, and Chroma were leveraged to capture phonetic and tonal characteristics of speech. Neural networks, including CNNs, were trained for classification. Challenges such as data imbalance, high feature dimensionality, and computational resource limitations were addressed through augmentation, dimensionality reduction, and GPU acceleration. This report presents the methodology, results, and insights, demonstrating that dialect classification models can support applications in voice assistants, customer service systems, and sociolinguistic studies. Fig1. `Steps of Audio Analysis with Machine Learning` and Fig2. `Dialect Identification Workflow – From Feature Extraction to Classification` guide the process visually.

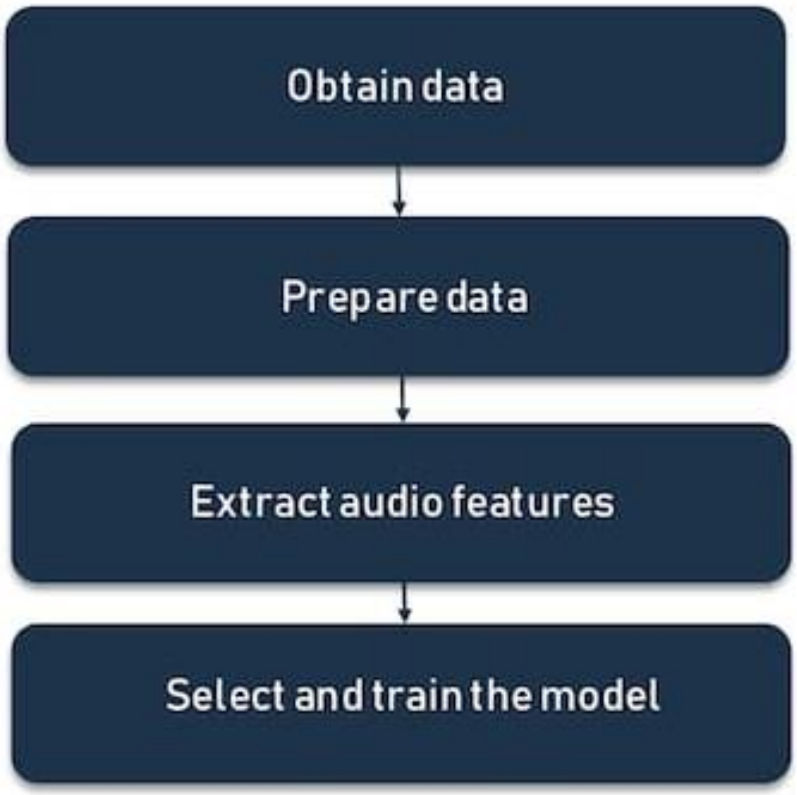


Fig1. Steps of audio analysis with machine learning

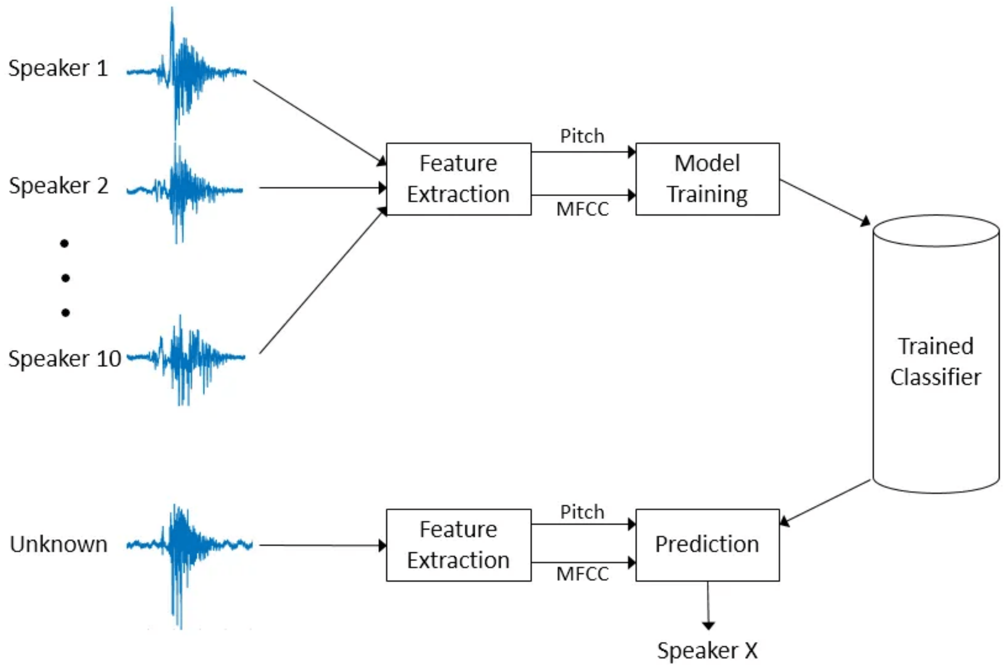


Fig2. Dialect Identification Workflow - From Feature Extraction to Classification

This project not only provides a foundation for dialect identification but also explores how linguistic diversity can be better understood and modeled, fostering advancements in human-computer interaction technologies and regional linguistic research.

2 <u>Introduction</u>

Recognizing regional dialects is essential in today's globalized communication landscape. Variations in accents and dialects can influence the performance of voice-based technologies like virtual assistants, automated customer service systems, and forensic linguistics applications. These dialectical nuances often pose challenges for AI systems to understand and respond accurately to user inputs. Therefore, this project aims to classify English dialects by analyzing audio features and linguistic transcripts, ultimately contributing to more inclusive and adaptive technologies.

The dataset utilized in this study comprises approximately 18,000 audio samples, segmented into regional subsets such as Scottish, Welsh, Irish, Northern, Southern, and Midlands English. These samples are annotated for gender diversity, ensuring a broad representation of voices. Key challenges, including high feature dimensionality, computational complexity, and accent variability, shaped the research design and methodology.

Applications of dialect recognition extend beyond consumer technology. In forensic linguistics, for instance, dialect detection can assist in criminal investigations by providing clues about a speaker's geographical origin. In sociolinguistics, it can support the study of how language evolves across regions and social groups. Fig1. `Steps of Audio Analysis with Machine Learning` (above) provides an overview of the systematic workflow, encompassing data acquisition, preprocessing, feature extraction, and model training.

3 Objective, Goal, Scope

Data Preprocessing:

The objective of this project is to build an audio classification model as a digital forensic application, enabling accurate identification of English dialects from audio recordings to support linguistic profiling, speaker attribution, and regional origin analysis in forensic investigations.

Goal:

The primary goal of this project is to develop an audio classification model capable of identifying English dialects from audio recordings. The model aims to aid digital forensic investigations by providing dialect classification, which can be used to infer geographic and cultural contexts in forensic cases.

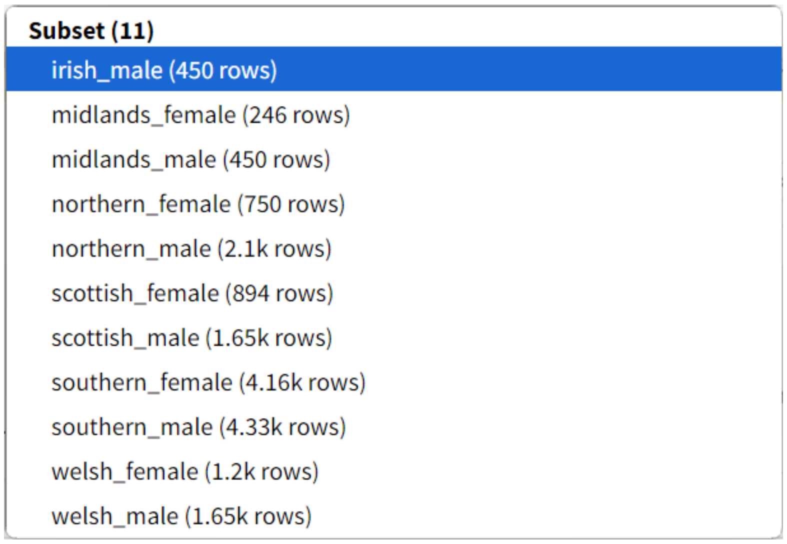
Scope:

Included:

- Preprocessing and normalizing raw audio data to ensure consistent input to the model.
- Training and validating a Wav2Vec2-based audio classifier on English dialects such as Scottish, Welsh, and Irish.
- Implementing padding/truncation to handle variable-length audio inputs.
- Tracking model performance through training and validation metrics (loss, accuracy).
- Saving the trained model for future deployment.
- Testing the model on unseen audio files to evaluate generalization.

Excluded:

- Non-English dialects or languages outside the dataset.
- Extensive data augmentation or noise injection techniques.
- Deployment of the model to a production environment and building a web tool for it (future scope).


Subset (11)

irish_male (450 rows)

midlands_female (246 rows)

midlands_male (450 rows)

northern_female (750 rows)

northern_male (2.1k rows)

scottish_female (894 rows)

scottish_male (1.65k rows)

southern_female (4.16k rows)

southern_male (4.33k rows)

welsh_female (1.2k rows)

welsh_male (1.65k rows)

Fig3. List of Data Sets (approx.18,000)

4 Dataset Characteristics

The dataset utilized contains approximately 18,000 audio samples as reflected in Fig3. List of Data Sets`, segmented into regional subsets such as Irish, Scottish, Welsh, Northern, Southern, and Midlands English. Gender diversity was ensured with male and female samples, ensuring balanced representation across dialects. Each audio file's duration was standardized to 10 seconds, with shorter files padded and longer files truncated to maintain uniformity.

The balanced dataset ensures that the model is trained on diverse accents and speaking styles, providing a robust foundation for classification. Fig4. `Converting Data to Numerical Vectors` depicts the preprocessing and vectorization process, ensuring consistency across the dataset and enabling seamless integration with the classification models.

In addition to ensuring diverse representation, the dataset's segmentation facilitates a granular analysis of dialect-specific characteristics. This approach aids in identifying subtle phonetic and tonal variations that define regional accents, enriching the model's understanding of linguistic diversity.

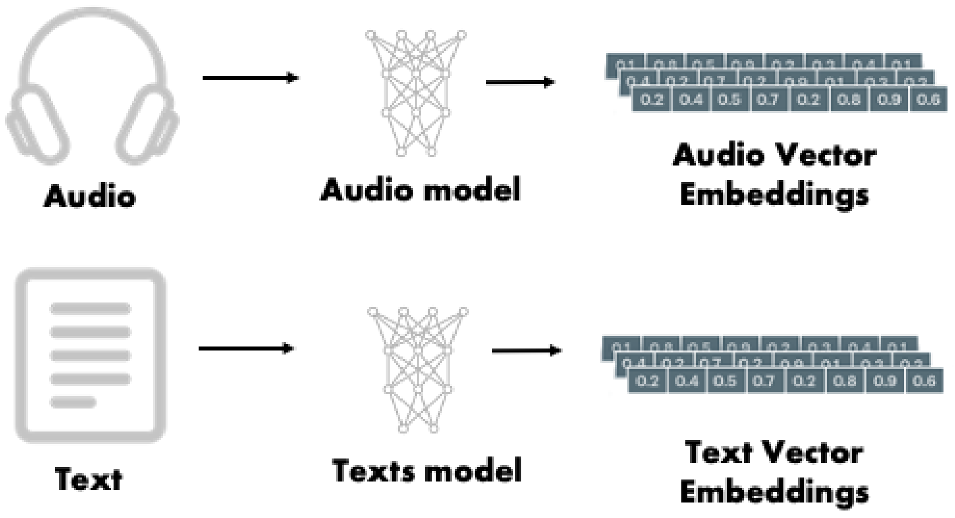


Fig4. Converting Data samples to Numerical Vectors to feed & train model

5 Methods and Methodology

Data Preprocessing:

1. **Audio Resampling**:
   - All audio files were resampled to a consistent sampling rate of 16 kHz to standardize input.
   - This ensured standardization and compatibility with the Wav2Vec2 model.

2. **Padding and Truncation:**
   - Audio files shorter than 10 seconds were padded, while longer files were truncated to ensure uniform input length.
   - This avoided loss of data while maintaining model compatibility.

3. **Noise Reduction:**
   - Background noise and irrelevant sounds were filtered out to enhance the clarity of speech signals.

4. **Label Encoding:**
   - Dataset labels were encoded to match the dialect categories.

Feature Extraction:

Four primary features were extracted to capture the phonetic and tonal characteristics of speech:

- **MFCCs:** These coefficients mimic human auditory perception and encode the speech's short-term power spectrum. Fig5. `MFCC Coefficients` illustrates the extracted MFCCs.
- **Mel Spectrograms:** These represent the audio's frequency content on a mel scale, highlighting pitch variations over time. Fig6. `Mel Spectrogram` showcases this feature.
- **Chroma Features:** These encode pitch class distribution and harmonic properties, as shown in Fig7. `Chroma Features`.
- **Spectrograms:** These visualize amplitude changes across frequencies over time, depicted in Fig8. `An example of a spectrum plot`.

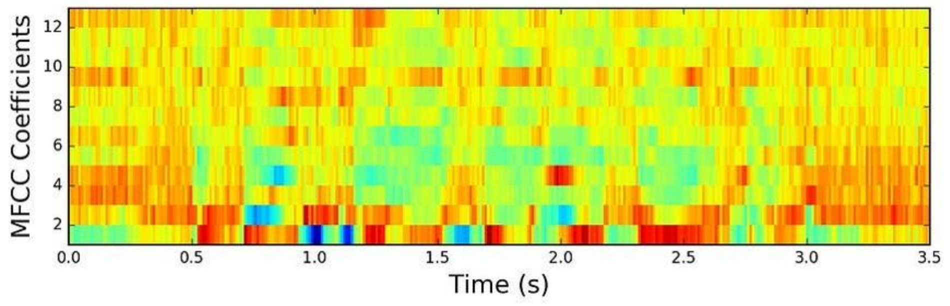


Fig5. MFCC Coefficients

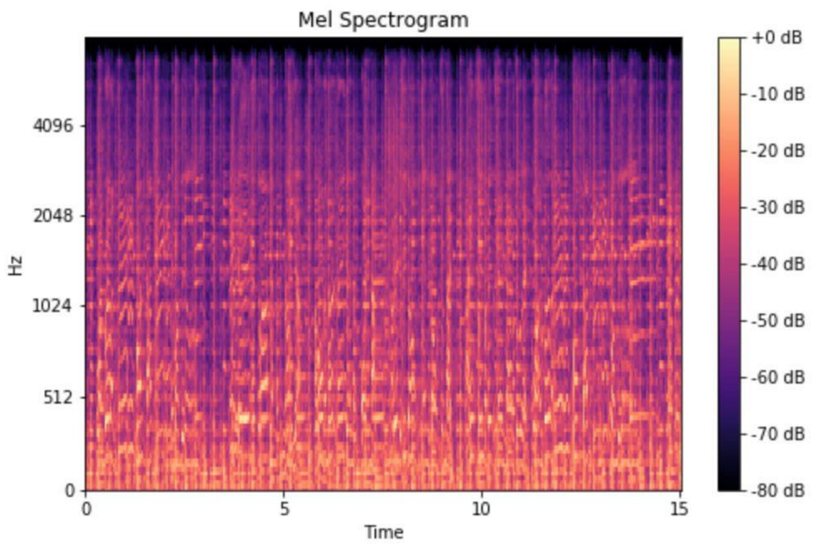


Fig6. Mel Spectrogram

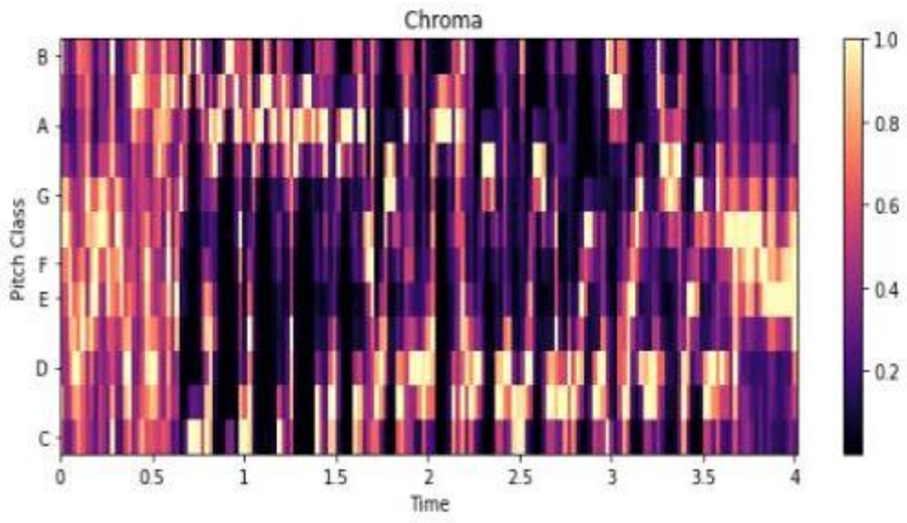


Fig7. Chroma Features

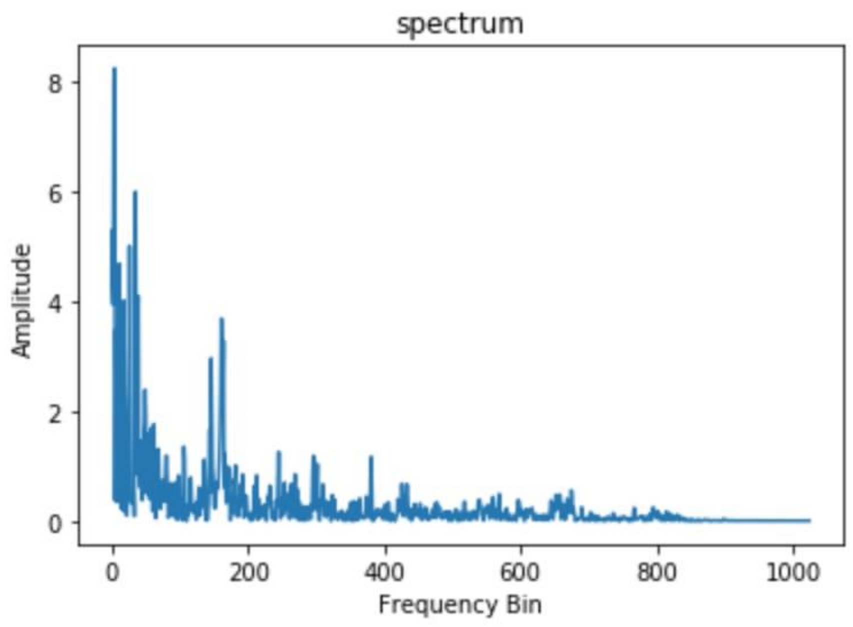
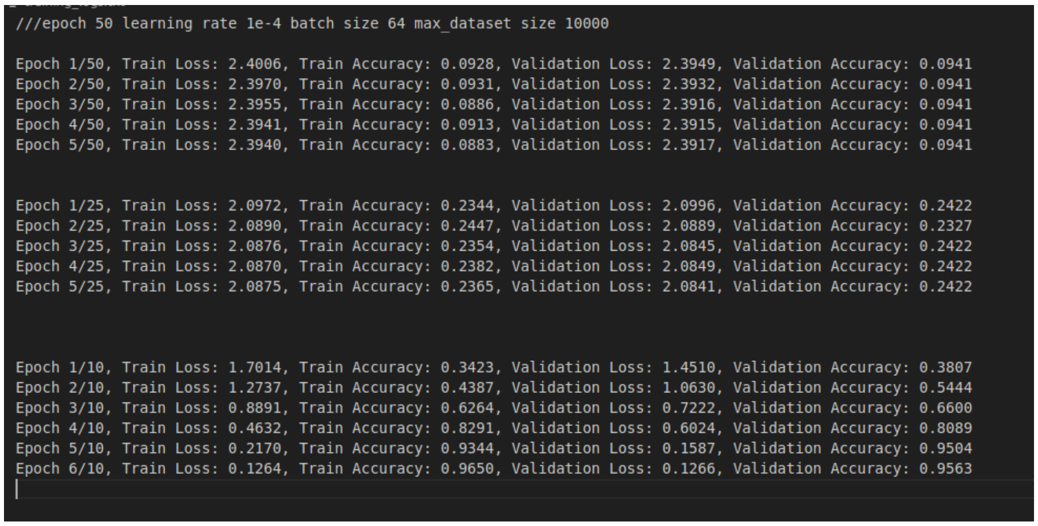


Fig8. An example of a spectrum plot

6 Model Training

The classification model, based on a convolutional neural network (CNN), was trained using feature vectors. NVIDIA GPUs were leveraged for computational acceleration, enabling efficient processing of large datasets. Training logs, seen in Fig9. `Training Logs and Epoch Performance`, provided insights into model performance.



```
///epoch 50 learning rate 1e-4 batch size 64 max_dataset size 10000

Epoch 1/50, Train Loss: 2.4006, Train Accuracy: 0.0928, Validation Loss: 2.3949, Validation Accuracy: 0.0941
Epoch 2/50, Train Loss: 2.3970, Train Accuracy: 0.0931, Validation Loss: 2.3932, Validation Accuracy: 0.0941
Epoch 3/50, Train Loss: 2.3955, Train Accuracy: 0.0886, Validation Loss: 2.3916, Validation Accuracy: 0.0941
Epoch 4/50, Train Loss: 2.3941, Train Accuracy: 0.0913, Validation Loss: 2.3915, Validation Accuracy: 0.0941
Epoch 5/50, Train Loss: 2.3940, Train Accuracy: 0.0883, Validation Loss: 2.3917, Validation Accuracy: 0.0941


Epoch 1/25, Train Loss: 2.0972, Train Accuracy: 0.2344, Validation Loss: 2.0996, Validation Accuracy: 0.2422
Epoch 2/25, Train Loss: 2.0890, Train Accuracy: 0.2447, Validation Loss: 2.0889, Validation Accuracy: 0.2327
Epoch 3/25, Train Loss: 2.0876, Train Accuracy: 0.2354, Validation Loss: 2.0845, Validation Accuracy: 0.2422
Epoch 4/25, Train Loss: 2.0870, Train Accuracy: 0.2382, Validation Loss: 2.0849, Validation Accuracy: 0.2422
Epoch 5/25, Train Loss: 2.0875, Train Accuracy: 0.2365, Validation Loss: 2.0841, Validation Accuracy: 0.2422



Epoch 1/10, Train Loss: 1.7014, Train Accuracy: 0.3423, Validation Loss: 1.4510, Validation Accuracy: 0.3807
Epoch 2/10, Train Loss: 1.2737, Train Accuracy: 0.4387, Validation Loss: 1.0630, Validation Accuracy: 0.5444
Epoch 3/10, Train Loss: 0.8891, Train Accuracy: 0.6264, Validation Loss: 0.7222, Validation Accuracy: 0.6600
Epoch 4/10, Train Loss: 0.4632, Train Accuracy: 0.8291, Validation Loss: 0.6024, Validation Accuracy: 0.8089
Epoch 5/10, Train Loss: 0.2170, Train Accuracy: 0.9344, Validation Loss: 0.1587, Validation Accuracy: 0.9504
Epoch 6/10, Train Loss: 0.1264, Train Accuracy: 0.9650, Validation Loss: 0.1266, Validation Accuracy: 0.9563
```

Fig9. Training Logs and Epoch Performance

The model's architecture included multiple convolutional and pooling layers, optimized for extracting spatial and temporal patterns from feature representations. Dropout and regularization techniques mitigated overfitting, ensuring robust generalization.

1. **Model Architecture:**
   - The Wav2Vec2 model from Hugging Face Transformers was fine-tuned for sequence classification.
   - Pretrained on large multilingual datasets such as LibriSpeech and CommonVoice, Wav2Vec2 employs self-supervised learning techniques.

2. **Training Process:**
   - Loss and accuracy were tracked across 20 epochs.
   - Early stopping was implemented to prevent overfitting.

3. **Workflow Illustration:**
   - As depicted in Fig2. `Dialect Identification Workflow – From Feature Extraction to Classification`, the Wav2Vec2 architecture combines CNN-based feature extraction with transformer-based learning for masked representations.

Tools and Frameworks:

- Programming Language: Python.
- Libraries: PyTorch, Librosa, Torchaudio, Transformers (Hugging Face), TQDM, NumPy.
- Hardware: NVIDIA GPU for accelerated training.

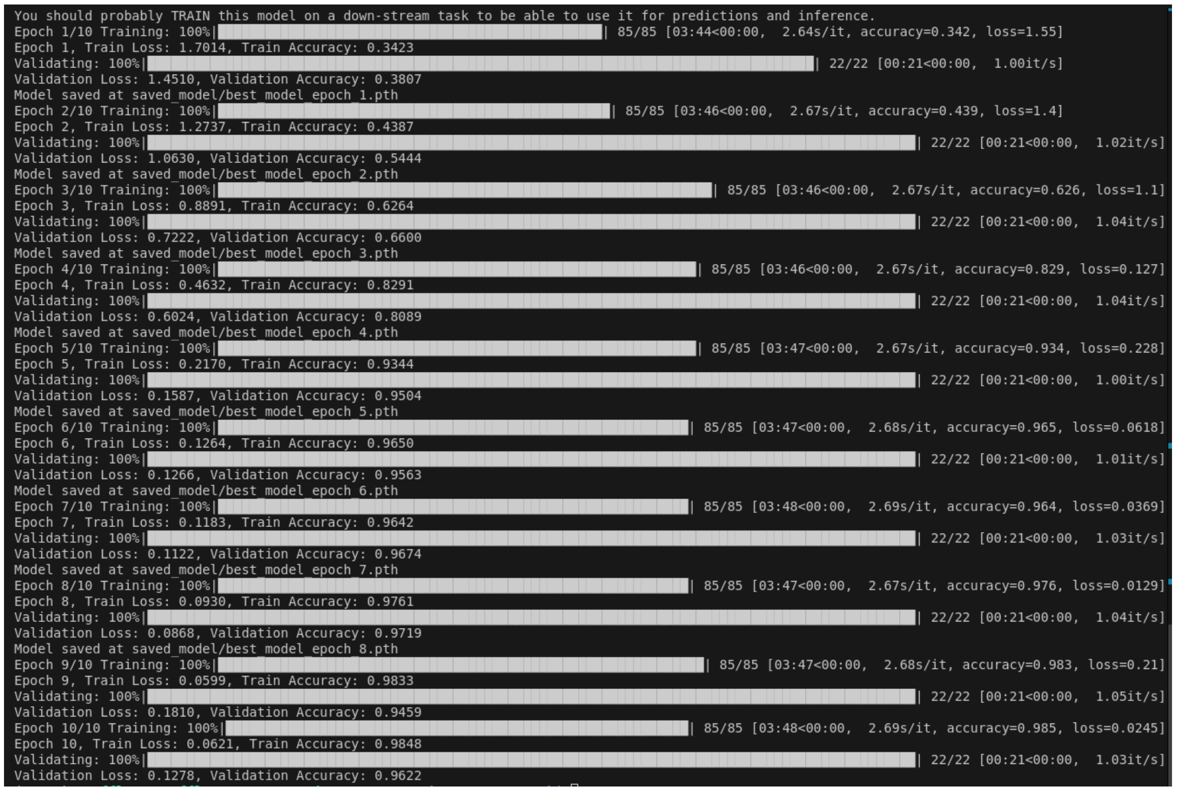
Validation:

1. **Train-Test Split:**
   - An 80-20 split was implemented to assess generalization.
   - Validation loss and accuracy metrics were logged for analysis.

2. **Logs:**
   - Training logs (e.g., `Training Logs and Epoch Performance`) were used to monitor progress and identify bottlenecks.

Model Training and Testing:

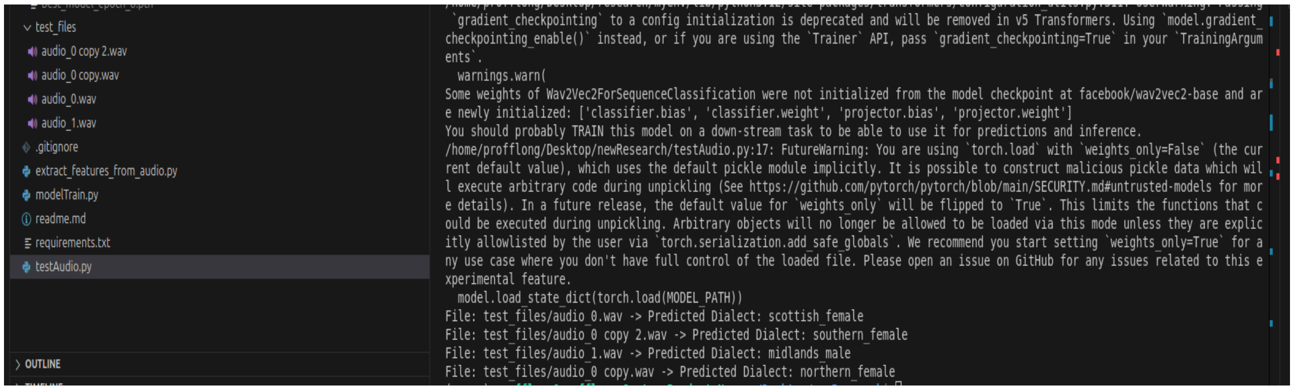
1. **Model Checkpoints:**
   - Models were saved at each epoch, with the best-performing model determined by validation accuracy.



```
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
Epoch 1/10 Training: 100%|████████████████████████████████| 85/85 [03:44<00:00,  2.64s/it, accuracy=0.342, loss=1.55]
Epoch 1, Train Loss: 1.7014, Train Accuracy: 0.3423
Validating: 100%|████████████████████████████████████████| 22/22 [00:21<00:00,  1.00it/s]
Validation Loss: 1.4510, Validation Accuracy: 0.3807
Model saved at saved_model/best_model_epoch_1.pth
Epoch 2/10 Training: 100%|████████████████████████████████| 85/85 [03:46<00:00,  2.67s/it, accuracy=0.439, loss=1.4]
Epoch 2, Train Loss: 1.2737, Train Accuracy: 0.4387
Validating: 100%|████████████████████████████████████████| 22/22 [00:21<00:00,  1.02it/s]
Validation Loss: 1.0630, Validation Accuracy: 0.5444
Model saved at saved_model/best_model_epoch_2.pth
Epoch 3/10 Training: 100%|████████████████████████████████| 85/85 [03:46<00:00,  2.67s/it, accuracy=0.626, loss=1.1]
Epoch 3, Train Loss: 0.8891, Train Accuracy: 0.6264
Validating: 100%|████████████████████████████████████████| 22/22 [00:21<00:00,  1.04it/s]
Validation Loss: 0.7222, Validation Accuracy: 0.6600
Model saved at saved_model/best_model_epoch_3.pth
Epoch 4/10 Training: 100%|████████████████████████████████| 85/85 [03:46<00:00,  2.67s/it, accuracy=0.829, loss=0.127]
Epoch 4, Train Loss: 0.4632, Train Accuracy: 0.8291
Validating: 100%|████████████████████████████████████████| 22/22 [00:21<00:00,  1.04it/s]
Validation Loss: 0.6024, Validation Accuracy: 0.8089
Model saved at saved_model/best_model_epoch_4.pth
Epoch 5/10 Training: 100%|████████████████████████████████| 85/85 [03:47<00:00,  2.67s/it, accuracy=0.934, loss=0.228]
Epoch 5, Train Loss: 0.2170, Train Accuracy: 0.9344
Validating: 100%|████████████████████████████████████████| 22/22 [00:21<00:00,  1.00it/s]
Validation Loss: 0.1587, Validation Accuracy: 0.9504
Model saved at saved_model/best_model_epoch_5.pth
Epoch 6/10 Training: 100%|████████████████████████████████| 85/85 [03:47<00:00,  2.68s/it, accuracy=0.965, loss=0.0618]
Epoch 6, Train Loss: 0.1264, Train Accuracy: 0.9650
Validating: 100%|████████████████████████████████████████| 22/22 [00:21<00:00,  1.01it/s]
Validation Loss: 0.1266, Validation Accuracy: 0.9563
Model saved at saved_model/best_model_epoch_6.pth
Epoch 7/10 Training: 100%|████████████████████████████████| 85/85 [03:48<00:00,  2.69s/it, accuracy=0.964, loss=0.0369]
Epoch 7, Train Loss: 0.1183, Train Accuracy: 0.9642
Validating: 100%|████████████████████████████████████████| 22/22 [00:21<00:00,  1.03it/s]
Validation Loss: 0.1122, Validation Accuracy: 0.9674
Model saved at saved_model/best_model_epoch_7.pth
Epoch 8/10 Training: 100%|████████████████████████████████| 85/85 [03:47<00:00,  2.67s/it, accuracy=0.976, loss=0.0129]
Epoch 8, Train Loss: 0.0930, Train Accuracy: 0.9761
Validating: 100%|████████████████████████████████████████| 22/22 [00:21<00:00,  1.04it/s]
Validation Loss: 0.0868, Validation Accuracy: 0.9719
Model saved at saved_model/best_model_epoch_8.pth
Epoch 9/10 Training: 100%|████████████████████████████████| 85/85 [03:47<00:00,  2.68s/it, accuracy=0.983, loss=0.21]
Epoch 9, Train Loss: 0.0599, Train Accuracy: 0.9833
Validating: 100%|████████████████████████████████████████| 22/22 [00:21<00:00,  1.05it/s]
Validation Loss: 0.1810, Validation Accuracy: 0.9459
Epoch 10/10 Training: 100%|███████████████████████████████| 85/85 [03:48<00:00,  2.69s/it, accuracy=0.985, loss=0.0245]
Epoch 10, Train Loss: 0.0621, Train Accuracy: 0.9848
Validating: 100%|████████████████████████████████████████| 22/22 [00:21<00:00,  1.03it/s]
Validation Loss: 0.1278, Validation Accuracy: 0.9622
```

Fig10. Model Checkpoints & Train Results

2. **Prediction Pipeline:**
   - A custom pipeline was developed to test unseen audio files. Predictions for dialect classification are illustrated in Fig11. `Prediction Results on Test Audio Files`.



```
 `gradient_checkpointing` to a config initialization is deprecated and will be removed in v5 Transformers. Using `model.gradient_
checkpointing_enable()` instead, or if you are using the `Trainer` API, pass `gradient_checkpointing=True` in your `TrainingArgum
ents`.
  warnings.warn(
Some weights of Wav2Vec2ForSequenceClassification were not initialized from the model checkpoint at facebook/wav2vec2-base and ar
e newly initialized: ['classifier.bias', 'classifier.weight', 'projector.bias', 'projector.weight']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
/home/profflong/Desktop/newResearch/testAudio.py:17: FutureWarning: You are using `torch.load` with `weights_only=False` (the cur
rent default value), which uses the default pickle module implicitly. It is possible to construct malicious pickle data which wil
l execute arbitrary code during unpickling (See https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for mor
e details). In a future release, the default value for `weights_only` will be flipped to `True`. This limits the functions that c
ould be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explic
itly allowlisted by the user via `torch.serialization.add_safe_globals`. We recommend you start setting `weights_only=True` for a
ny use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this e
xperimental feature.
  model.load_state_dict(torch.load(MODEL_PATH))
File: test_files/audio_0.wav -> Predicted Dialect: scottish_female
File: test_files/audio_0 copy 2.wav -> Predicted Dialect: southern_female
File: test_files/audio_1.wav -> Predicted Dialect: midlands_male
File: test_files/audio_0 copy.wav -> Predicted Dialect: northern_female
```

Fig11. Prediction Results on Test Audio Files

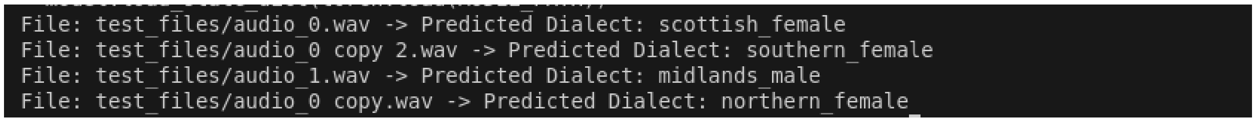
7 <u>Results</u>

Training Results:

- Peak training accuracy: ~98% by the 10th epoch, as seen in Fig9. `Model Checkpoints & Train Results`.
- Validation accuracy stabilized around ~96%, indicating slight overfitting.
- Logs such as `Epoch Results – Validation Accuracy and Train Accuracy` depict the performance trends.

Generalization Results:

1. **Dataset Performance:**
   - The model successfully classified test audio files from the dataset.
   -



```
File: test_files/audio_0.wav -> Predicted Dialect: scottish_female
File: test_files/audio_0 copy 2.wav -> Predicted Dialect: southern_female
File: test_files/audio_1.wav -> Predicted Dialect: midlands_male
File: test_files/audio_0 copy.wav -> Predicted Dialect: northern_female
```

Fig12. Performance Evaluation

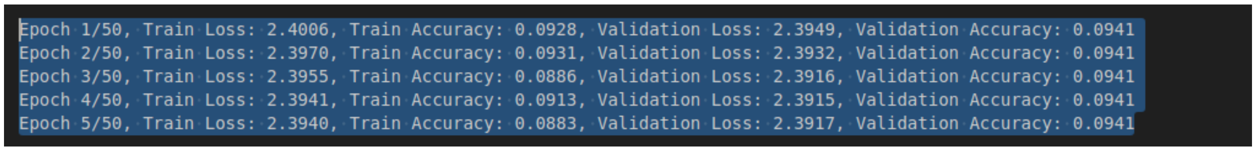
2. **Limitations:**
   - Struggled with dialects not represented in the training set (e.g., Indian English).

3. **Goal Achievement:**
   - The model performs well on known dialects but requires additional data or augmentation for better generalization to unseen dialects.
   - Future work should address expanding datasets and improving augmentation techniques.

8 Challenges

Challenge 1: Low Accuracy with Default Wav2Vec2 Settings:

- Initial attempts using the default Wav2Vec2 model yielded low accuracy
- After parameter tuning and feature engineering, performance improved significantly.



```
Epoch 1/50, Train Loss: 2.4006, Train Accuracy: 0.0928, Validation Loss: 2.3949, Validation Accuracy: 0.0941
Epoch 2/50, Train Loss: 2.3970, Train Accuracy: 0.0931, Validation Loss: 2.3932, Validation Accuracy: 0.0941
Epoch 3/50, Train Loss: 2.3955, Train Accuracy: 0.0886, Validation Loss: 2.3916, Validation Accuracy: 0.0941
Epoch 4/50, Train Loss: 2.3941, Train Accuracy: 0.0913, Validation Loss: 2.3915, Validation Accuracy: 0.0941
Epoch 5/50, Train Loss: 2.3940, Train Accuracy: 0.0883, Validation Loss: 2.3917, Validation Accuracy: 0.0941
```

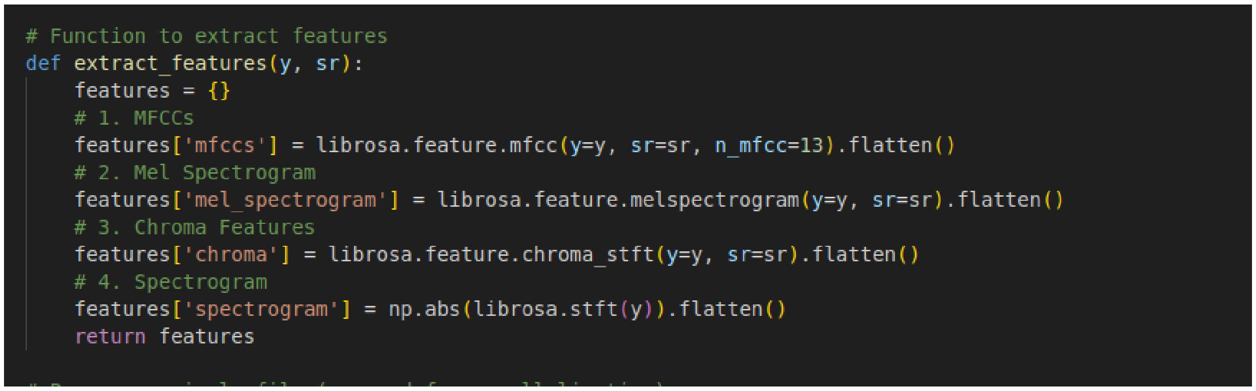
Fig13. Low Accuracy with Default Wav2Vec2 Settings

Challenge 2: High Feature Dimensionality:

- The internal features of Wav2Vec2, including MFCCs, spectrograms, and chroma, were extracted using Librosa and TorchAudio.
- Feature sets, visualized in `MFCC Coefficients` and `Mel Spectrogram`, and required extensive preprocessing.
- Managing these large datasets (up to 7GB per feature set) posed computational challenges.

Definitions:

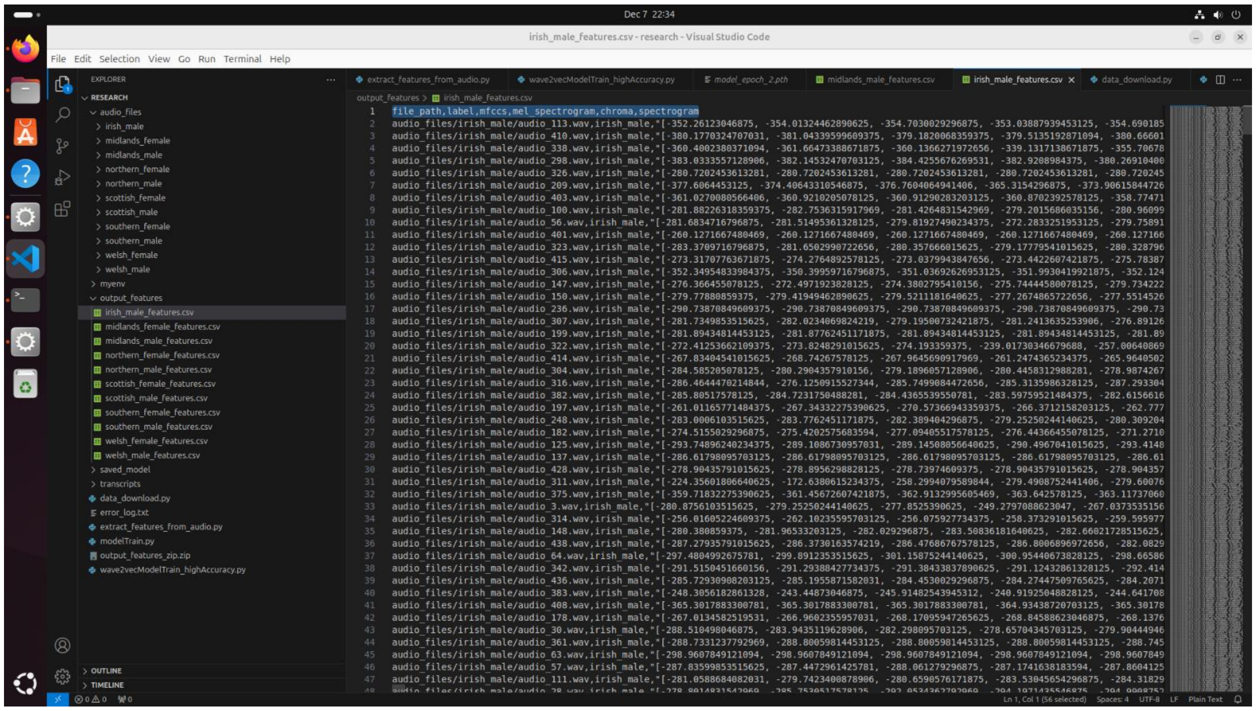
1. **MFCC (Mel-Frequency Cepstral Coefficients):** Encodes timbral features and speech characteristics.
2. **Mel Spectrogram:** Represents the sound's frequency content on a mel scale.
3. **Spectrogram:** Visualizes the amplitude of frequencies over time.
4. **Chroma:** Captures pitch class distribution and harmonic properties.



```python
# Function to extract features
def extract_features(y, sr):
    features = {}
    # 1. MFCCs
    features['mfccs'] = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13).flatten()
    # 2. Mel Spectrogram
    features['mel_spectrogram'] = librosa.feature.melspectrogram(y=y, sr=sr).flatten()
    # 3. Chroma Features
    features['chroma'] = librosa.feature.chroma_stft(y=y, sr=sr).flatten()
    # 4. Spectrogram
    features['spectrogram'] = np.abs(librosa.stft(y)).flatten()
    return features
```

Fig14. High Feature Dimensionality

We used Librosa and TorchAudio to extract these features, as demonstrated in the screenshots.



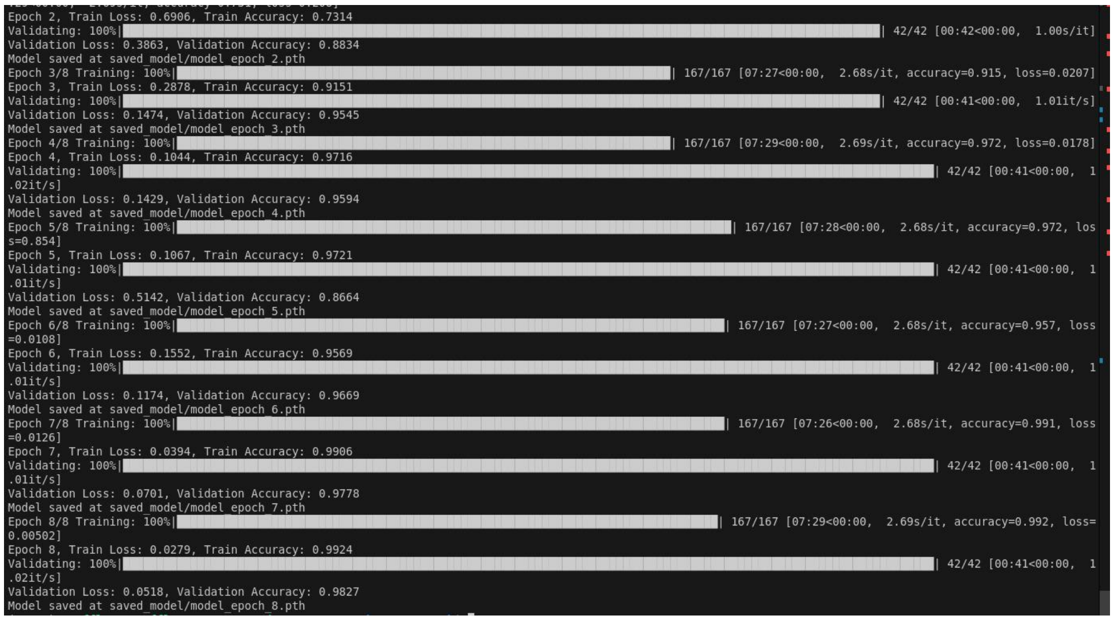
```
file_path,label,mfccs,mel_spectrogram,chroma,spectrogram
audio_files/irish_male/audio_113.wav,irish_male,"[-352.26123046875, -354.01324462890625, -354.7030029296875, -353.03887939453125, -354.690185
audio_files/irish_male/audio_410.wav,irish_male,"[-380.1770324707031, -381.04339599609375, -379.1820068359375, -379.5135192871094, -380.66601
audio_files/irish_male/audio_338.wav,irish_male,"[-360.4002380371094, -361.66473388671875, -360.1366271972656, -339.1317138671875, -355.70678
audio_files/irish_male/audio_298.wav,irish_male,"[-383.0333557128906, -382.1453247070312 5, -384.4255676269531, -382.9208984375, -380.26910400
audio_files/irish_male/audio_326.wav,irish_male,"[-280.7202453613281, -280.7202453613281, -280.7202453613281, -280.7202453613281, -280.720245
audio_files/irish_male/audio_209.wav,irish_male,"[-377.6064453125, -374.40643310546875, -376.7604064941406, -365.3154296875, -373.90615844726
audio_files/irish_male/audio_403.wav,irish_male,"[-361.0270080566406, -360.9210205078125, -360.91290283203125, -360.8702392578125, -358.77471
audio_files/irish_male/audio_100.wav,irish_male,"[-281.88226318359375, -282.7536315917969, -281.4264831542969, -279.201568603515 6, -280.96099
audio_files/irish_male/audio_56.wav,irish_male,"[-281.6834716796875, -281.51495361328125, -279.81927490234375, -272.2833251953125, -279.75891
audio_files/irish_male/audio_401.wav,irish_male,"[-260.1271667480469, -260.1271667480469, -260.1271667480469, -260.1271667480469, -260.127166
audio_files/irish_male/audio_323.wav,irish_male,"[-283.3709716796875, -281.6502990722656, -280.357666015625, -279.1777954101562 5, -280.328796
audio_files/irish_male/audio_415.wav,irish_male,"[-273.31707763671875, -274.2764892578125, -273.0379943847656, -273.4422607421875, -275.78387
audio_files/irish_male/audio_306.wav,irish_male,"[-352.34954833984375, -350.39959716796875, -351.03692626953125, -351.9930419921875, -352.124
audio_files/irish_male/audio_147.wav,irish_male,"[-276.366455078125, -272.4971923828125, -274.3802795410156, -275.7444458007812 5, -279.734222
audio_files/irish_male/audio_150.wav,irish_male,"[-279.77880859375, -279.41949462890625, -279.5211181640625, -277.2674865722656, -277.5514526
audio_files/irish_male/audio_236.wav,irish_male,"[-290.73870849609375, -290.73870849609375, -290.73870849609375, -290.73870849609375, -290.73
audio_files/irish_male/audio_307.wav,irish_male,"[-281.7349853515625, -282.0234069824219, -279.19500732421875, -281.2413635253906, -276.89126
audio_files/irish_male/audio_199.wav,irish_male,"[-281.89434814453125, -281.87762451171875, -281.89434814453125, -281.89434814453125, -281.89
audio_files/irish_male/audio_322.wav,irish_male,"[-272.41253662109375, -273.8248291015625, -274.193359375, -239.01730346679688, -257.00640869
audio_files/irish_male/audio_414.wav,irish_male,"[-267.83404541015625, -268.74267578125, -267.9645690917969, -261.247436523437 5, -265.9640502
audio_files/irish_male/audio_304.wav,irish_male,"[-284.585205078125, -280.2904357910156, -279.1896057128906, -280.4458312988281, -278.9874267
audio_files/irish_male/audio_316.wav,irish_male,"[-286.4644470214844, -276.1250915527344, -285.7499084472656, -285.3135986328125, -287.293304
audio_files/irish_male/audio_382.wav,irish_male,"[-285.80517578125, -284.7231750488281, -284.4365539550781, -283.59759521484375, -282.6156616
audio_files/irish_male/audio_197.wav,irish_male,"[-261.01165771484375, -267.34332275390625, -270.57366943359375, -266.3712158203125, -262.777
audio_files/irish_male/audio_248.wav,irish_male,"[-283.0006103515625, -283.7762451171875, -282.389404296875, -279.2525024414062 5, -280.309204
audio_files/irish_male/audio_182.wav,irish_male,"[-274.5155029296875, -275.4202575683594, -277.09405517578125, -276.44366455078125, -271.2710
audio_files/irish_male/audio_125.wav,irish_male,"[-293.74896240234375, -289.1086730957031, -289.14508056640625, -290.4967041015625, -293.4148
audio_files/irish_male/audio_137.wav,irish_male,"[-286.61798095703125, -286.61798095703125, -286.61798095703125, -286.61798095703125, -286.61
audio_files/irish_male/audio_428.wav,irish_male,"[-278.90435791015625, -278.8956298828125, -278.73974609375, -278.90435791015625, -278.904357
audio_files/irish_male/audio_311.wav,irish_male,"[-224.35601806640625, -172.6380615234375, -258.2994079589844, -279.4908752441406, -279.60076
audio_files/irish_male/audio_375.wav,irish_male,"[-359.71832275390625, -361.45672607421875, -362.9132995605469, -363.642578125, -363.11737060
audio_files/irish_male/audio_3.wav,irish_male,"[-280.8756103515625, -279.25250244140625, -277.8525390625, -249.2797088623047, -267.0373535156
audio_files/irish_male/audio_314.wav,irish_male,"[-256.01605224609375, -262.10235595703125, -256.075927734375, -258.373291015625, -259.595977
audio_files/irish_male/audio_148.wav,irish_male,"[-280.380859375, -281.96533203125, -282.029296875, -283.5083618164062 5, -282.66021728515625,
audio_files/irish_male/audio_438.wav,irish_male,"[-287.27935791015625, -286.3730163574219, -286.47686767578125, -286.8006896972656, -282.0829
audio_files/irish_male/audio_64.wav,irish_male,"[-297.4804992675781, -299.8912353515625, -301.15875244140625, -300.95440673828125, -298.66586
audio_files/irish_male/audio_342.wav,irish_male,"[-291.5150451660156, -291.29388427734375, -291.38433837890625, -291.12432861328125, -292.414
audio_files/irish_male/audio_436.wav,irish_male,"[-285.72930908203125, -285.1955871582031, -284.4530029296875, -284.27447509765625, -284.2071
audio_files/irish_male/audio_383.wav,irish_male,"[-248.3056182861328, -243.44873046875, -245.91482543945312, -240.91925048828125, -244.641708
audio_files/irish_male/audio_408.wav,irish_male,"[-365.3017883300781, -365.3017883300781, -365.3017883300781, -364.93438720703125, -365.30178
audio_files/irish_male/audio_178.wav,irish_male,"[-267.0134582519531, -266.9602355957031, -268.17095947265625, -268.84588623046875, -268.1376
audio_files/irish_male/audio_30.wav,irish_male,"[-288.510498046875, -283.9435119628906, -282.298095703125, -278.65704345703125, -279.90444946
audio_files/irish_male/audio_361.wav,irish_male,"[-288.7331237792969, -288.80059814453125, -288.80059814453125, -288.80059814453125, -288.745
audio_files/irish_male/audio_63.wav,irish_male,"[-298.9607849121094, -298.9607849121094, -298.9607849121094, -298.9607849121094, -298.9607849
audio_files/irish_male/audio_57.wav,irish_male,"[-287.83599853515625, -287.4472961425781, -288.061279296875, -287.1741638183594, -287.8604125
audio_files/irish_male/audio_111.wav,irish_male,"[-281.0588684082031, -279.7423400878906, -280.6590576171875, -283.53045654296875, -284.31829
```

Dec 7 22:34
irish_male_features.csv - research - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
RESEARCH
audio_files
irish_male
midlands_female
midlands_male
northern_female
northern_male
scottish_female
scottish_male
southern_female
southern_male
welsh_female
welsh_male
myenv
output_features
irish_male_features.csv
midlands_female_features.csv
midlands_male_features.csv
northern_female_features.csv
northern_male_features.csv
scottish_female_features.csv
scottish_male_features.csv
southern_female_features.csv
southern_male_features.csv
welsh_female_features.csv
welsh_male_features.csv
saved_model
transcripts
data_download.py
error_log.txt
extract_features_from_audio.py
modelTrain.py
output_features_zip.zip
wave2vecModelTrain_highAccuracy.py
OUTLINE
TIMELINE
extract_features_from_audio.py
wave2vecModelTrain_highAccuracy.py
model_epoch_2.pth
midlands_male_features.csv
irish_male_features.csv
data_download.py
output_features > irish_male_features.csv
Ln 1, Col 1 (56 selected) Spaces: 4 UTF-8 LF Plain Text

Fig15. Extracted Features for subsets

Challenge 3: Overfitting:

- Training accuracy exceeded validation accuracy, as seen in Fig 16. `Overfitting Trends`.
- Regularization techniques such as early stopping and reduced learning rates mitigated this issue.

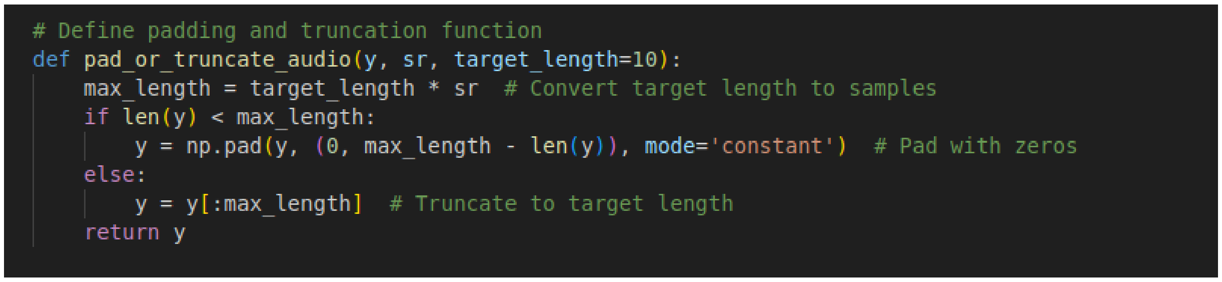


```
Epoch 2, Train Loss: 0.6906, Train Accuracy: 0.7314
Validating: 100%|████████████████████████████████████████████████████████████| 42/42 [00:42<00:00,  1.00s/it]
Validation Loss: 0.3863, Validation Accuracy: 0.8834
Model saved at saved_model/model_epoch_2.pth
Epoch 3/8 Training: 100%|████████████████████████████████████████| 167/167 [07:27<00:00,  2.68s/it, accuracy=0.915, loss=0.0207]
Epoch 3, Train Loss: 0.2878, Train Accuracy: 0.9151
Validating: 100%|████████████████████████████████████████████████████████████| 42/42 [00:41<00:00,  1.01it/s]
Validation Loss: 0.1474, Validation Accuracy: 0.9545
Model saved at saved_model/model_epoch_3.pth
Epoch 4/8 Training: 100%|████████████████████████████████████████| 167/167 [07:29<00:00,  2.69s/it, accuracy=0.972, loss=0.0178]
Epoch 4, Train Loss: 0.1044, Train Accuracy: 0.9716
Validating: 100%|████████████████████████████████████████████████████████████████| 42/42 [00:41<00:00,  1
.02it/s]
Validation Loss: 0.1429, Validation Accuracy: 0.9594
Model saved at saved_model/model_epoch_4.pth
Epoch 5/8 Training: 100%|█████████████████████████████████████████████| 167/167 [07:28<00:00,  2.68s/it, accuracy=0.972, los
s=0.854]
Epoch 5, Train Loss: 0.1067, Train Accuracy: 0.9721
Validating: 100%|████████████████████████████████████████████████████████████████| 42/42 [00:41<00:00,  1
.01it/s]
Validation Loss: 0.5142, Validation Accuracy: 0.8664
Model saved at saved_model/model_epoch_5.pth
Epoch 6/8 Training: 100%|████████████████████████████████████████████| 167/167 [07:27<00:00,  2.68s/it, accuracy=0.957, loss
=0.0108]
Epoch 6, Train Loss: 0.1552, Train Accuracy: 0.9569
Validating: 100%|████████████████████████████████████████████████████████████████| 42/42 [00:41<00:00,  1
.01it/s]
Validation Loss: 0.1174, Validation Accuracy: 0.9669
Model saved at saved_model/model_epoch_6.pth
Epoch 7/8 Training: 100%|████████████████████████████████████████████| 167/167 [07:26<00:00,  2.68s/it, accuracy=0.991, loss
=0.0126]
Epoch 7, Train Loss: 0.0394, Train Accuracy: 0.9906
Validating: 100%|████████████████████████████████████████████████████████████████| 42/42 [00:41<00:00,  1
.01it/s]
Validation Loss: 0.0701, Validation Accuracy: 0.9778
Model saved at saved_model/model_epoch_7.pth
Epoch 8/8 Training: 100%|███████████████████████████████████████████| 167/167 [07:29<00:00,  2.69s/it, accuracy=0.992, loss=
0.00502]
Epoch 8, Train Loss: 0.0279, Train Accuracy: 0.9924
Validating: 100%|████████████████████████████████████████████████████████████████| 42/42 [00:41<00:00,  1
.02it/s]
Validation Loss: 0.0518, Validation Accuracy: 0.9827
Model saved at saved_model/model_epoch_8.pth
```

Fig16. Overfitting Trends

Challenge 4: Variable Audio Lengths:

- A consistent 10-second duration was achieved through padding and truncation.
- Relevant code snippets ensured uniformity while preserving information.



```python
# Define padding and truncation function
def pad_or_truncate_audio(y, sr, target_length=10):
    max_length = target_length * sr  # Convert target length to samples
    if len(y) < max_length:
        y = np.pad(y, (0, max_length - len(y)), mode='constant')  # Pad with zeros
    else:
        y = y[:max_length]  # Truncate to target length
    return y
```

Fig17. Setting for Padding & Truncating Samples

9 Conclusion

This project demonstrates the feasibility of using Wav2Vec2 for dialect classification in digital forensics. Key findings include:

1. Model Efficacy:

- High accuracy with sufficient training data per dialect.

2. Overfitting Mitigation:

- Effective use of regularization and validation monitoring.

3. Dataset Importance:

- Performance highly dependent on diverse and representative datasets.

Future Work:

1. Expanding datasets to include more dialects and languages.
2. Implementing advanced augmentation techniques to improve generalization.
3. Developing real-time applications for voice assistants and forensic tools.

With further improvements, this model can become a valuable tool in digital forensics, providing insights into geographic and cultural contexts from audio data.

10 GitHub Link (Here: Dialect Recognition)

The complete project codebase, including all scripts for data preprocessing, feature extraction, model training, and evaluation, is hosted on GitHub. This repository provides access to.

- **Setup Instructions:** Detailed README files guide users on how to set up the environment, install dependencies, and run the scripts.
- **Code Files:** Python scripts for preprocessing, feature engineering, model training, and testing.
- **Model Checkpoints:** Saved models from different training epochs for reuse or further fine-tuning.
- **Training Logs:** Logs containing detailed insights into model performance across epochs.
- **Sample Data:** Examples of audio files and their associated features to help users get started.

11 Related Work

Several studies have explored speech and dialect recognition, leveraging advancements in machine learning and audio processing technologies. Techniques such as PCA, SVD, and kernel methods have been widely used for data dimensionality reduction, facilitating efficient processing of high-dimensional datasets. Ait-Sahalia and Xiu (2015) highlighted PCA's application in high-frequency financial data analysis, demonstrating its potential in extracting significant features for classification tasks.

The emergence of self-supervised learning frameworks, such as Wav2Vec2, has revolutionized speech processing. As depicted in Fig3. `Self-Supervised Pretraining Workflow`, Wav2Vec2 employs convolutional neural networks (CNNs) for feature extraction and transformers for learning masked representations. These models, pre-trained on multilingual datasets like VoxLingua and CommonVoice, have proven effective in speech recognition and classification tasks.

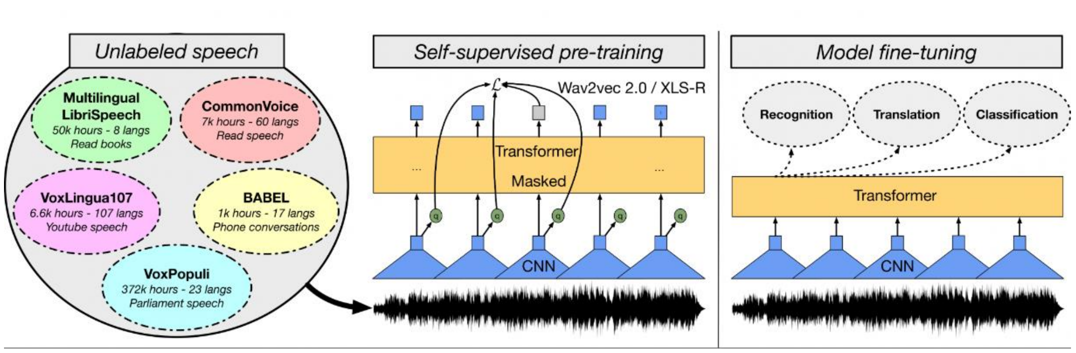


Fig3. Self-Supervised Pretraining Workflow

Additionally, clustering approaches have been extensively applied to audio datasets. Rajkovi (2000) highlighted clustering's potential for identifying patterns in financial data, and similar methods have been adapted for linguistic tasks. Momeni et al. (2015) explored clustering algorithms to group entities based on performance and behavior. These foundational studies guided the implementation of clustering and feature reduction methods in this project.

12 References

1. **Ait-Sahalia, Y., & Xiu, D. (2015)**. Principal Component Analysis of High-Frequency Data. *The Review of Financial Studies*, *28*(6), 1503–1538. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2666352
2. **Baevski, A., Zhou, H., Mohamed, A., & Auli, M. (2020)**. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. *Advances in Neural Information Processing Systems (NeurIPS)*. https://arxiv.org/abs/2006.11477
3. **Chiu, C. C., Sainath, T. N., Wu, Y., et al. (2018)**. State-of-the-art speech recognition with sequence-to-sequence models. *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. https://arxiv.org/abs/1712.01769
4. **Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015)**. Librispeech: An ASR corpus based on public domain audio books. *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. https://www.danielpovey.com/files/2015_icassp_librispeech.pdf
5. **Tjandra, A., Sakti, S., & Nakamura, S. (2017)**. Listening while speaking: Speech chain by deep learning. *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. https://arxiv.org/abs/1707.04879
6. **VoxLingua107: Dataset for Spoken Language Recognition**. (2020). https://www.researchgate.net/publication/346475235_VoxLingua107_a_Dataset_for_Spoken_Language_Recognition
7. **Hugging English Dialects**. https://huggingface.co/datasets/ylacombe/english_dialects
8. **Librosa: Python Package for Audio Analysis**. Accessed at: https://librosa.org/