Classification of spoken Arabic dialects using deep learning

Hassan Tamer, Salma Ahmed Sherif, Ahmed AbdelMoneim, Mostafa A.Sultan
computer and communication engineering department
Alexandria University

***Abstract*—Automatic classification of Arabic dialects from audio data presents significant challenges due to the language's rich diversity and complex phonetic variations. This paper introduces a novel deep learning approach to address this problem, leveraging pretrained models to analyze spectrograms and Mel-Frequency Cepstral Coefficients (MFCCs). By exploring various model architectures and data augmentation techniques, we aim to achieve high accuracy in classifying Arabic dialects, contributing to the development of more sophisticated language processing tools for the Arabic language.**

***Index Terms*—Deep Learning, Neural Networks, Machine Learning, Data Science, audio recognition**

I. Introduction

The Arabic language features a rich diversity of dialects, posing challenges for automatic speech recognition and natural language processing. Despite growing interest, the literature lacks comprehensive studies on classifying Arabic dialects using deep learning techniques. This project aims to fill this gap by utilizing pretrained deep learning models to classify Arabic dialects based on audio features.

We employ pre-trained models to analyze audio spectrograms and Mel-Frequency Cepstral Coefficients (MFCCs). Spectrograms visually represent the audio signal's frequency spectrum over time, capturing pronunciation and phonetic nuances unique to each dialect. MFCCs effectively represent the short-term power spectrum of sound, widely used in speech recognition.

Our approach seeks to achieve high accuracy in dialect classification, enhancing Arabic-specific speech recognition systems. The novelty of our work lies in applying deep learning to this relatively unexplored area, contributing to the development of more sophisticated language processing tools for the Arabic language and improving communication across diverse Arabic-speaking communities.

II. Related Work

The nature of the task overlaps with some other areas including language identification and accent detection. However, most of the work done and published in the literature tackling the dialect detection models was on different accents in English and different languages. [**?**] [**?**] Previous studies have focused on Arabic dialect identification using machine learning models such as K-Nearest Neighbor, Random Forest, Multi-Layer Perceptron, and Artificial Neural Networks, achieving accuracies ranging from 34% to 76%. Deep learning models, specifically Convolutional Neural Networks (CNNs), have been utilized for Arabic speech dialect classification, reaching an accuracy of 83% by converting speech into images using spectrogram features. [**?**] The fusion of multiple classifiers, combining acoustic and linguistic features, has shown promising results with a classification accuracy of 82.44% for identifying five Arabic dialects. [**?**] All the previously mentioned papers use written text scrapped from tweets or relatively limited datasets to identify written text origin. For the previously mentioned reasons, this paper was introduced as a novel approach to collect the data and build a model similar to English accent detection for different Arab countries' accents for various classes.

III. Data

The data we used was audio records from popular podcasts from different countries and regions in the Arab world because we believed that the podcasts did not have white noise and tackled different dialects and topics with a wide variety of speakers. We scrapped the data from YouTube podcasts and this was done under copyright fair use policies applied in their terms of service, with due respect to the user privacy and complying with relevant laws and regulations and the US government approval [**?**] The data was then divided into 5 classes similar to the classification used in MIT paper on written dialect [**?**] As shown in Table I, The final state of the data collected and demonstrated in the table below with evenly distributed data between ages and genders.

Dataset	Dialect	Minutes
Train	Moroccan	551.6567
	Egyptian	472.2843
	Gulf	470.2781
	Levantine	453.8284
	MSA	421.8423
Test	Moroccan	214.9315
	Egyptian	162.1384
	Gulf	82.6734
	Levantine	82.7774
	MSA	124.3439

TABLE I
Dataset statistics in minutes for different dialects.

IV. Methodology

A. Data Collection

The audio data for Arabic dialects was collected from YouTube using the YouTube Data API and Pydub libraries. We targeted channels and videos tagged with specific dialect

names to ensure a diverse dataset and ensuring that the data used is evenly distributed with no bias towards a certain category.

B. Data Preprocessing

The collected audio files were cleaned to remove the first 2 to 3 minutes to exclude the intro which usually is just noise and music not the targeted dataset. in addition, Spleeter was used to enhance sample quality and separate noise. Segmentation was performed to break down long recordings into manageable segments of 1 minute each. Audio normalization was applied to ensure consistent volume levels across all samples.

1) Data Augmentation: To enhance the robustness and generalizability of our models, we applied several data augmentation techniques to artificially expand our dataset. These techniques include:

- **time stretching & shifting**: which alters the speed of an audio signal without affecting its pitch and shifting the audio signal slightly forward or backward in time.
- **pitch shifting**: which changes the pitch of the audio signal without altering its duration.
- **adding background noise**:which helps the model become more robust to real-world noisy environments.
- **random cropping**: which involves selecting random segments from longer audio recordings.

other techniques could be used like volume perturbation, which involves randomly adjusting the volume of the audio signal; These techniques were implemented using the Python library `librosa` and generated on-the-fly during the training process, significantly increasing the diversity of our training data and leading to improved model performance and robustness.

2) Feature Extraction: Feature extraction is a crucial step in processing audio data for classification task and We used two primary feature extraction techniques using `librosa` [**?**] :

- **Spectrograms**: Audio signals were converted into spectrograms using a window size of 25 ms with a 10 ms overlap. This provided a visual representation of the frequency spectrum over time.
- **MFCCs**: MFCCs, on the other hand, is used in speech processing and capture the power spectrum of an audio signal on a Mel scale, which approximates the human ear's response more closely. We extracted 13 MFCCs from each audio segment, along with their first and second derivatives, resulting in 39-dimensional feature vectors. [**?**]

These features were used as inputs to various deep learning models. For Convolutional Neural Networks (CNNs), the spectrograms were fed as input images, allowing the CNNs to exploit their spatial structure and learn local patterns relevant to different dialects. The CNNs were particularly effective in capturing the time-frequency characteristics embedded in the spectrograms as it will be discussed in the next models section.

For transformer-based models, which are adept at handling sequential data and capturing long-range dependencies, we used the MFCC features. The transformer models, equipped with self-attention mechanisms, could effectively model the temporal relationships and dependencies within the audio signal.

C. Model Selection

We explored three distinct approaches for classifying Arabic dialects from audio data:

- **Image-Based Models**: Initially, we applied models designed for image classification to spectrograms generated from audio signals. While these models capture spatial patterns, their performance was limited due to their primary focus on visual features.
- **Audio Feature-Based Models**: We then utilized models specifically designed for processing audio data, which extract features like log mel spectrograms. These models better captured the unique auditory characteristics of Arabic dialects, resulting in improved performance.
- **Transformer-Based Models**: Our final approach involved transformer-based models, known for handling sequential data and capturing long-range dependencies. By processing raw audio data directly, these models effectively captured the temporal patterns of dialects, outperforming other approaches. [**?**]

These approaches provided valuable insights into the strengths and weaknesses of different model types, guiding our methodology towards improved performance in dialect classification.

D. Training Process

The dataset was split into 80% training, 20% validation, and test sets. The test set was used to detect how the models did in the training and make hyperparameter tuning to optimize learning rate, batch size, and the number of epochs to avoid overfitting as it was the main problem faced because models were too complex. The Adam optimizer was used for training due to its efficiency in handling sparse gradients.

E. Evaluation

The model's performance was evaluated using key metrics including accuracy, precision, recall, and F1-score, along with support values for each class. Additionally, a confusion matrix was generated to visualize the model's predictions and identify patterns of misclassification across different classes. These metrics provide a comprehensive overview of the model's performance and check which dialects were the most troublesome and how we could address them

V. Models Used

In this section, we outline the models utilized in our project:

A. DenseNet and MobileNet

We initially employed DenseNet and MobileNet architectures due to their efficiency in handling image-related tasks. Despite being primarily designed for images, we repurposed them for audio spectrogram processing. However, their performance was suboptimal for audio tasks as they lacked specific adaptations for auditory features.

B. VGGish and YAMNet

To address the limitations of image-based models, we adopted VGGish and YAMNet, which are specialized for audio data processing. VGGish converts audio inputs into log mel spectrograms, while YAMNet is proficient in sound classification.

C. Transformer-Based Approach

As a future enhancement, we plan to explore transformer-based models for audio classification as they can handle sequential data effectively, and hold promise for capturing temporal patterns and nuances in languages as they were proposed [**?**] and we hoped to do the same with dialects. This represents a promising direction for improving the performance and robustness of our classification system.

VI. Results

The models did differently on the date used and we tried to make the dataset larger to avoid overfitting and to get the best of the results we had. Densenet did very poorly as it was overfitting the data as the differences in the pitches and phases weren't noticeable at the end. and you can see in figure 1 as in the training the accuracy was suspiciously high and was an indication of overfitting although the evaluation metrics were fine.

- **V**GGish

TABLE II
VGGish Classification Report

Class	Precision	Recall	F1-score	Support
Egyptian	1.00	0.77	0.87	31
Khalijy	1.00	1.00	1.00	28
Levantine	0.83	1.00	0.91	34
Moroccan	0.96	0.88	0.92	25
MSA	0.92	1.00	0.96	24
Accuracy		0.93		
Macro avg		0.94 / 0.93 / 0.93		
Weighted avg		0.94 / 0.93 / 0.93		

this was the first stage of the training but DenseNet and VGGish resulted in poor accuracies around 10% testing accuracies.

The other stage was enlarging the dataset and making the Yamnet model more robust to overfitting as we proceeded with the next training and testing stage.

The resulting accuracies as demonstrated in the table III and the confusion matrix in 2

- **Y**amnet
  However, the testing accuracies were not as high for various reasons

On the other hand, the testing accuracies were as follows

-

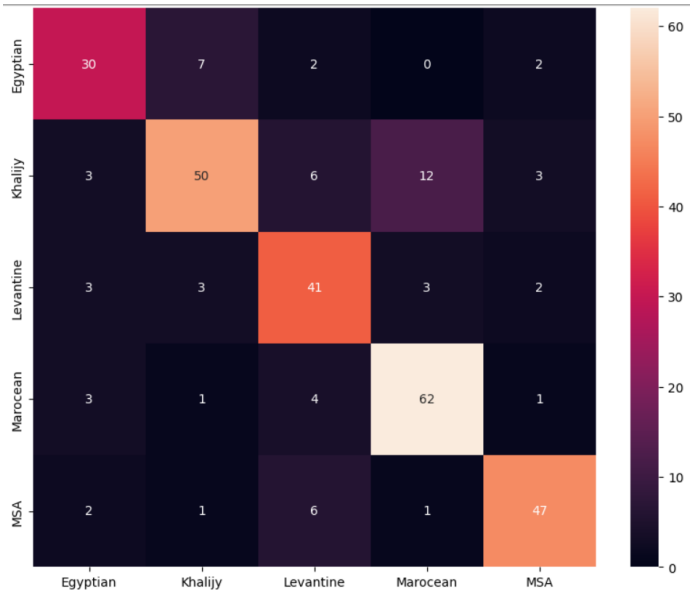

Fig. 1. VGGish confusion matrix.

TABLE III
Yamnet Classification Report

Class	Precision	Recall	F1-score	Support
Egyptain	0.84	0.60	0.70	99
Khalijy	0.89	0.82	0.85	94
Levantine	0.70	0.90	0.79	77
Moroccan	0.82	0.89	0.86	94
MSA	0.80	0.86	0.83	91
Accuracy		0.81		
Macro avg		0.81 / 0.81 / 0.80		
Weighted avg		0.81 / 0.81 / 0.80		

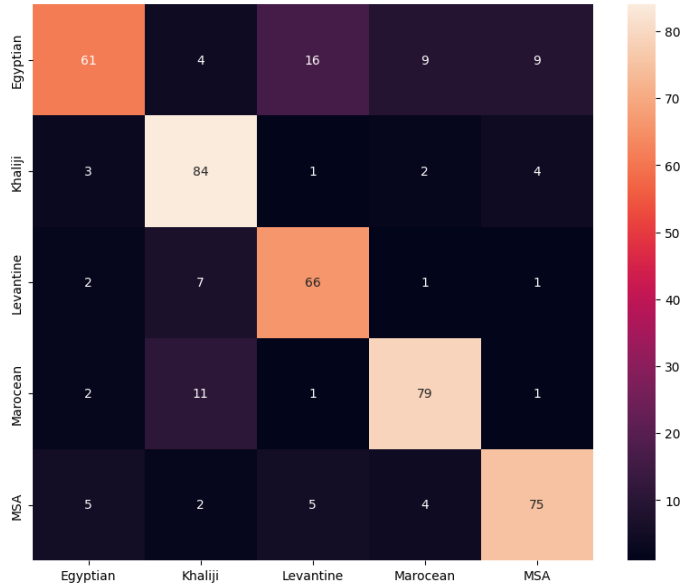

Fig. 2. Yamnet confusion matrix for Training

TABLE IV
TESTING DATA REPORT

Class	Precision	Recall	F1-score	Support
Egyptain	0.41	0.30	0.35	83
Khaligy	0.65	0.61	0.63	84
Levantine	0.36	0.60	0.45	84
Moroccan	0.61	0.48	0.54	112
MSA	0.48	0.46	0.47	93
Accuracy		0.49		
Macro avg		0.50 / 0.49 / 0.49		
Weighted avg		0.51 / 0.49 / 0.49		

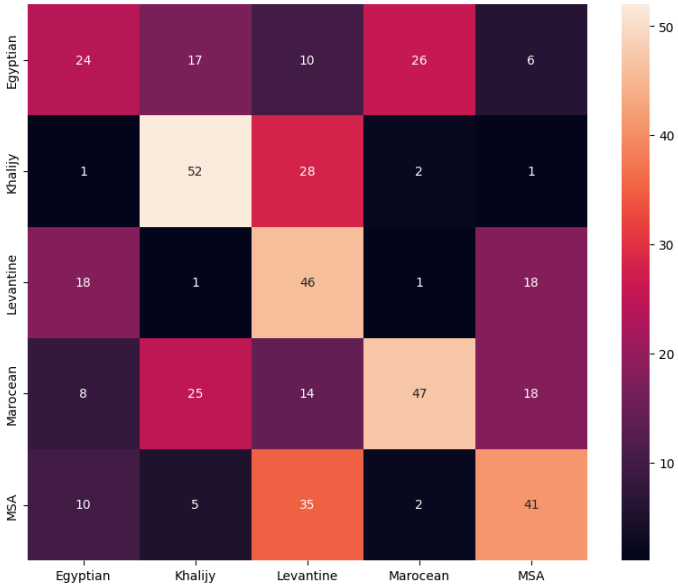

Fig. 3. Yamnet confusion matrix for testing

VII. DISCUSSION

The training phase of our model yielded promising accuracies, averaging around 80%, indicating that the model has learned meaningful patterns from the training data. However, during testing, the accuracy dropped to approximately 50%, suggesting that the model struggled to generalize well to unseen data. Several factors could contribute to this performance gap.

One potential explanation is overfitting during training, where the model learns to memorize the training data rather than generalize from it. This could occur if the model architecture is too complex relative to the size of the training dataset, or if the training process lacks appropriate regularization techniques.

Additionally, the discrepancy between training and testing accuracies may also indicate a mismatch between the distributions of the training and testing data. If the testing data significantly differs from the training data in terms of characteristics or noise levels, the model may fail to generalize effectively.

To improve the model's performance, several strategies we explored and could do more in this regards:

1) **Regularization Techniques**: Implement techniques such as dropout, weight regularization, or early stopping to mitigate overfitting during training.
2) **Data Augmentation**: Increase the diversity of the training dataset through techniques such as random cropping, scaling, or adding noise to audio samples. This can help the model learn more robust features and generalize better to unseen data.
3) **Hyperparameter Tuning**: Explore different model architectures, learning rates, and optimization algorithms to find the optimal configuration for the task.
4) **Ensemble Learning**: Combine predictions from multiple models to leverage diverse perspectives and improve overall performance.

VIII. FUTURE ENHANCEMENTS

Incorporating transformer-based models, such as Wav2Vec, offers a promising approach to enhancing our model's performance and overcoming challenges. Transformers excel in capturing long-range dependencies and hierarchical features from audio data, addressing issues with subtle temporal patterns in Arabic dialects.

By fine-tuning pre-trained transformer models like Wav2Vec, we can leverage transfer learning to bootstrap our model's learning process, leading to faster convergence and better generalization. Transformers also feature selective attention mechanisms, improving robustness to noise and variability in the data.

In summary, integrating transformer-based models into our framework holds the potential to significantly improve accuracy and robustness in classifying Arabic dialects from audio data.

IX. CONCLUSION

In conclusion, this paper presented a comprehensive exploration of deep learning techniques for classifying Arabic dialects from audio data. By leveraging pretrained models and advanced feature extraction methods, we demonstrated promising results in accurately identifying dialect variations. Our findings highlight the potential of deep learning to address the challenges inherent in Arabic language processing, paving the way for the development of more robust language recognition systems tailored to diverse Arabic-speaking communities.

Moving forward, further research is needed to refine our models, address limitations in generalization, and explore additional avenues for improving performance. By continuing to innovate in this field, we can contribute to the advancement of language processing technologies and foster greater communication and understanding across linguistic boundaries.