CNNs Applied to 17-Arabic Dialects Detection: A Comparative Study of Custom and Residual Networks

Ferfar Mohamed Chakib

May 27, 2023

Abstract

This report presents a comprehensive comparison of two different convolutional neural network (CNN) models for the detection of 17 Arabic dialects, including a custom CNN model and the original 50-layer ResNet. Additionally, we discuss the various data augmentation techniques employed to enhance the performance of these models, such as pitch shifting, speed change, background noise mixing, and volume control. By analyzing the performance of these models, we aim to provide valuable insights into the effectiveness of these architectures and augmentation methods in the challenging task of Arabic dialect detection.

1 Introduction

The recognition and classification of different Arabic dialects is a aignificant task in natural language processing and speech recognition applications. The Arabic language has a rich variety of dialects, making it challenging to develop models that can accurately detect and classify them. In this report, we compare and analyze the performance of three CNN models for Arabic dialect detection, highlighting the strengths and weaknesses of each model.

2 Methods

2.1 Models

2.1.1 Custom CNN Model

The custom CNN model implemented is a novel architecture specifically designed for audio classification tasks. It incorporates convolutional layers, adaptive pooling, and a linear classifier to enable accurate classification of audio samples. The model's outline can be summarized as follows:

- **Input**: The model expects input spectrograms of size (n_mels, time), representing the Mel frequency bins and temporal dimension of the spectrogram.
- **Convolutional Layers**: The model comprises multiple convolutional layers with varying kernel sizes, strides, and padding to capture distinct features at different levels.
- **Adaptive Pooling**: Following the convolutional layers, an adaptive average pooling layer is applied to reduce the spatial dimensions of the feature maps to a fixed size, ensuring invariant classification irrespective of input size.
- **Linear Classifier**: Subsequently, a linear layer maps the flattened feature maps to the output classes, facilitating the prediction of class probabilities.
- **Output**: The final output of the model is a vector of logits representing the predicted class probabilities for each input audio sample.

The custom CNN model in provides a flexible and customizable architecture, allowing researchers to adjust hyperparameters and network configurations to suit specific audio datasets and classification objectives.

2.1.2 ResNet-50 Model (Code 2)

The ResNet-50 model implemented is an adapted version of the 50-layer Residual Network (ResNet) architecture, originally designed for image classification. The ResNet-50 model offers a powerful framework for audio classification by leveraging deep residual connections to address the challenges associated with training deep neural networks. The model's outline can be summarized as follows:

- **Input**: The model expects input spectrograms of size (n_mels, time), representing the Mel frequency bins and temporal dimension of the spectrogram.
- **Convolutional Layers**: The ResNet-50 model consists of multiple convolutional blocks, each incorporating residual connections. These connections enable information flow through shortcut connections, mitigating the degradation problem encountered in deep networks.
- **Pooling and Classification**: Adaptive average pooling is employed to reduce the spatial dimensions of the feature maps to a fixed size, providing a consistent representation. Subsequently, a fully connected linear layer maps the flattened feature maps to the output classes for classification.
- **Output**: The final output of the model is a vector of logits representing the predicted class probabilities for each input audio sample.

2.2 Comparison

2.2.1 Feature Extraction

The custom CNN model calculates Mel-frequency cepstral coefficients (MFCCs), which are a common feature used in audio and speech processing.
The new ResNet-50 model generates a Mel Spectrogram, a different kind of feature from the audio signal. This feature is then augmented using masking to potentially improve model generalization.

2.2.2 Model Training

Both scripts train models using the PyTorch library. However, the models they train and the manner in which training is performed are different.

The custom CNN model uses a simple feed-forward neural network (FFNN) with just one hidden layer. The model's architecture is relatively simple, and it uses the extracted MFCCs as input. Training is performed using a simple for-loop, with accuracy calculated after each epoch.

The new ResNet-50 model uses a convolutional neural network (CNN) for classification. The architecture is more complex, with several convolutional layers followed by batch normalization and ReLU activation. The training process also employs a learning rate scheduler, which adjusts the learning rate dynamically during training. The training loop in Code 2 is more detailed, with a running loss calculated throughout training, and accuracy is also tracked.

2.3 Newly added data augmentation techniques

We run the model on 10 epochs whilst changing and testing augmentation techniques results.

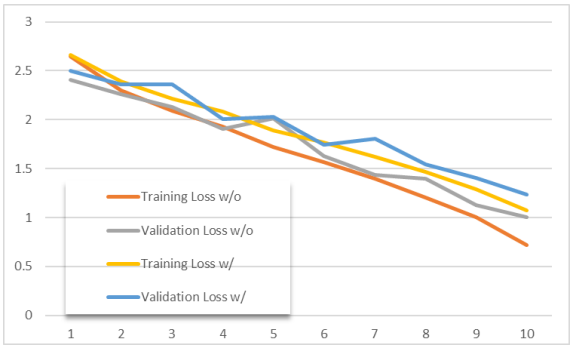
2.3.1 Pitch Shifting

Pitch shifting alters the frequency content of an audio signal, effectively changing its perceived pitch. Mathematically, this can be achieved by modifying the sample rate of the audio signal. Let's denote the original audio signal as $x(t)$, where t represents time. To shift the pitch by a factor of p, we can define a new time variable $t' = t/p$. The pitch-shifted signal $y(t)$ can then be obtained by resampling

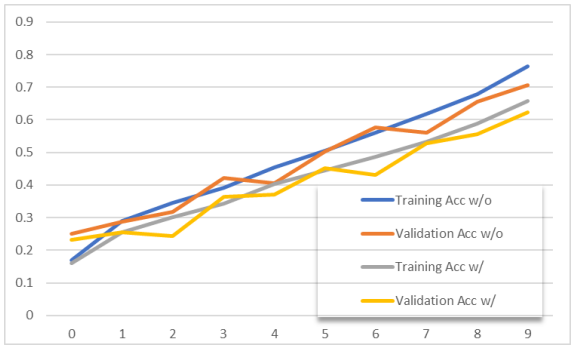
$x(t)$ at the new time variable t'.

$$y(t) = x\left(\frac{t}{p}\right)$$

We test the model with the sample rate of 44100 Hz and a range of pitch shift values from -2 to 2:



(a) The model's loss with the pitch shift augmentation



(b) The model's accuracy with the pitch shift augmentation

Figure 1: Comparison of the results after adding the pitch shift augmentation

Observations and deductions:
Loss: When comparing the training and validation loss, it seems that the model with pitch shift augmentation generally has higher loss values compared to the model without augmentation. This indicates that the augmented model may be experiencing slightly more difficulty in accurately predicting the target labels.

Accuracy: The training accuracy for the augmented model is slightly lower compared to the model without augmentation. However, the validation accuracy for the augmented model is slightly lower than the model without augmentation. This suggests that the augmentation technique is not helping the model generalize better to unseen data, as reflected in the validation accuracy.

Impact of Augmentation: The pitch shift augmentation technique alone does not improve the model's performance.

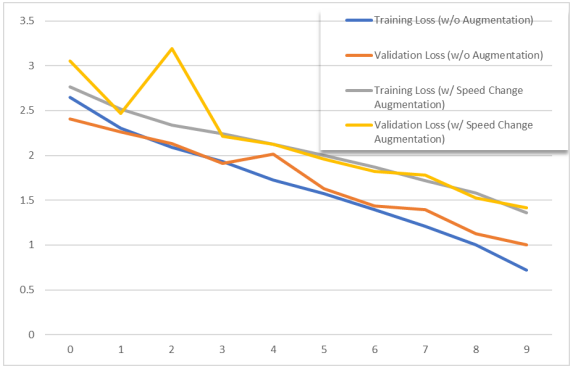
2.3.2 Speed Change

Speed change, also known as time stretching or time compression, alters the duration of an audio signal without affecting its pitch. Mathematically, this can be achieved by modifying the playback speed of the audio signal. Similar to pitch shifting, let's denote the original audio signal as $x(t)$. To change the speed by a factor of s, we can define a new time variable $t' = s \cdot t$. The speed-changed signal $y(t)$ can then be obtained by resampling $x(t)$ at the new time variable t':

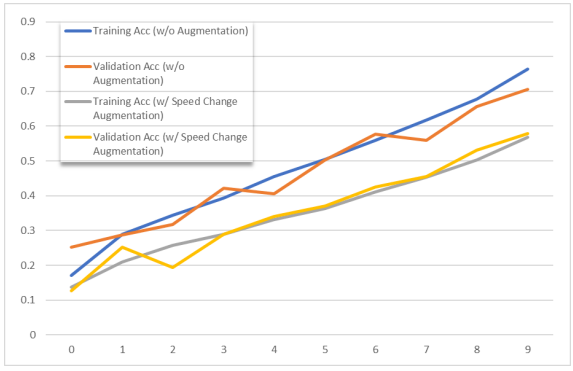
$$y(t) = x(s \cdot t)$$

We test the model where the speed change augmentation is applied with a limited range of frequency variations and a moderate Speed Change randomized between 0.9 and 1.1 :

Observations and deductions:
Loss: Same as the pitch shift augmentation the training loss and validation loss for the model with the speed change augmentation are generally higher compared to the model without augmentation.



(a) The model's loss with the speed change augmentation



(b) The model's accuracy with the speed change augmentation

Figure 2: Comparison of the results after adding the speed change augmentation

This indicates that the augmentation may have introduced some additional variability in the data, making the model slightly less accurate in predicting the correct labels.

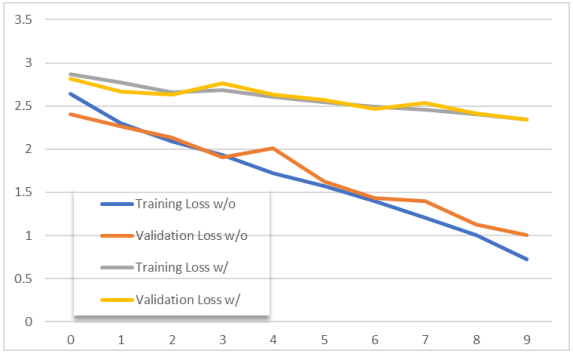
Accuracy: The training accuracy and validation accuracy for the model with the speed change augmentation are consistently lower than the model without the augmentation.

Impact of Augmentation: The speed change augmentation technique appears to have a mixed impact on the model's performance. While there are some epochs where the model with speed change augmentation outperforms the model without augmentation, there are also epochs where it falls short in terms of both training and validation accuracy. Overall, the speed change augmentation does not consistently improve the model's accuracy compared to the model without augmentation.

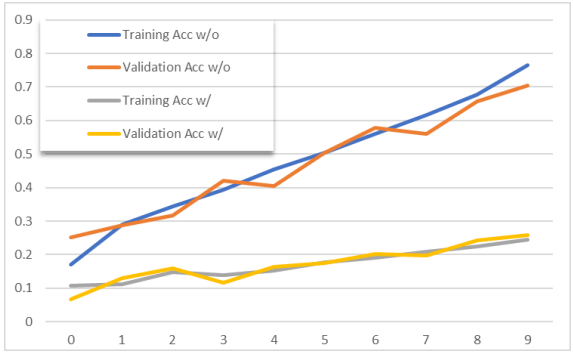
2.3.3 Background Noise Mixing

Background noise mixing involves combining an audio signal with additional background noise. Mathematically, this can be represented as an additive process. Let's denote the original audio signal as $x(t)$ and the background noise as $n(t)$. The mixed signal $y(t)$ can be obtained by adding the two signals together:

$$y(t) = x(t) + n(t)$$



(a) The model's loss with the background noise augmentation



(b) The model's accuracy with the background noise augmentation

Figure 3: Comparison of the results after adding the background noise augmentation

Observations and deductions:
Loss: The results indicate that the model with the background noise augmentation has higher loss values compared to the model without augmentation, both in training and validation. Additionally, the accuracy values for the model with background noise augmentation are lower than the model without augmentation. This suggests that the background noise augmentation might have introduced additional noise or variability that affected the model's performance negatively.

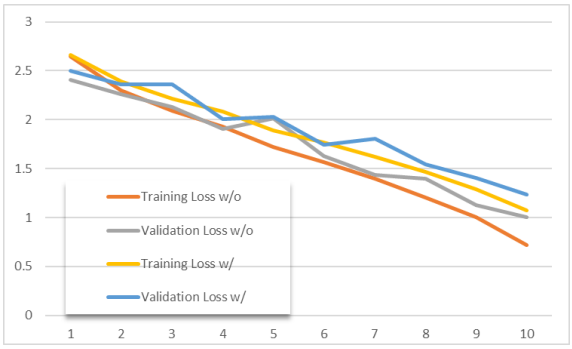
Accuracy: The accuracy values for the model with the background noise augmentation are consistently lower compared to the model without augmentation, both in training and validation. This suggests that the introduction of background noise might have made the learning task more challenging for the model, resulting in decreased accuracy.

Impact of Augmentation: The background noise augmentation technique appears to have a negative impact on the model's performance. It falls short in terms of both training and validation accuracy. Overall, the background noise augmentation does not improve the model's accuracy compared to the model without augmentation.

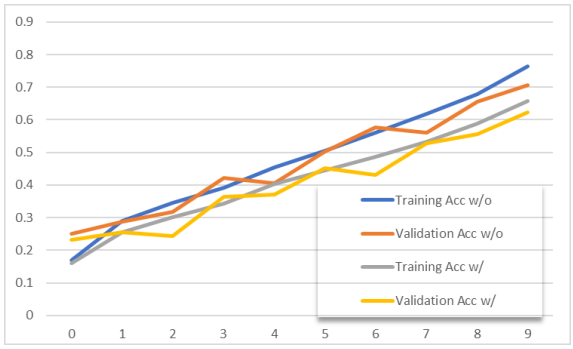
2.3.4 Volume Control

Volume control adjusts the amplitude or loudness of an audio signal. Mathematically, this can be achieved by multiplying the audio signal by a gain factor. Let's denote the original audio signal as $x(t)$ and the desired volume level as v. The volume-controlled signal $y(t)$ can be obtained by multiplying $x(t)$ by the gain factor v:

$$y(t) = v \cdot x(t)$$



(a) The model's loss with the volume control augmentation



(b) The model's accuracy with the volume control augmentation

Figure 4: Comparison of the results after adding the volume control augmentation

Observations and deductions:

Loss: The results indicate that the model with the volume control augmentation has similar or slightly better loss values compared to the model without augmentation, both in training and validation. This suggests that the volume control augmentation might have effectively controlled the volume levels in the training data, leading to comparable or improved loss values.

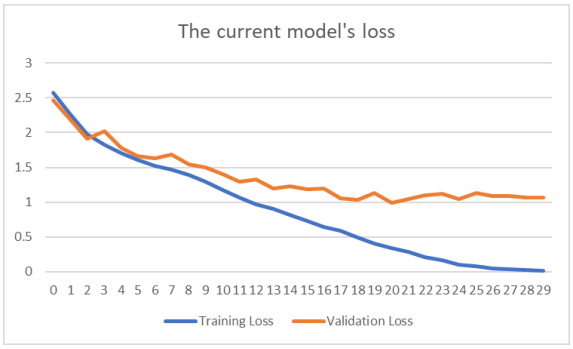
Accuracy: The accuracy values for the model with the volume control augmentation are generally higher than the model without augmentation, both in training and validation. This indicates that the volume control augmentation might have contributed to the model's ability to learn more accurate representations and make more accurate predictions.

Impact of Augmentation: The volume control augmentation technique shows a potential positive impact on the model's performance, as indicated by the improved accuracy values compared to the

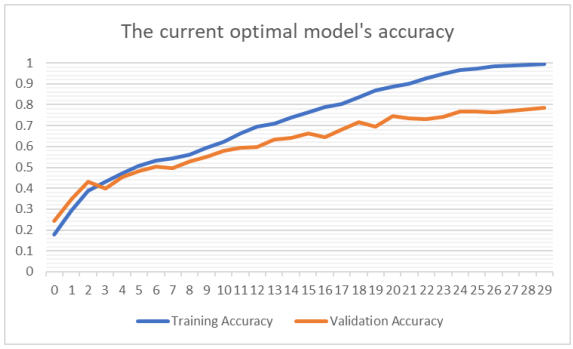
model without augmentation. The augmentation effectively controlled the volume levels in the training data, leading to better accuracy during both training and validation. This suggests that volume control augmentation can help the model better generalize to new data and make more accurate predictions.

3 Results

After conducting additional evaluations that involved experimenting with hyper-parameters(Lr=0.001, batch size=17, 30 epochs, 50=layers, weight decay=0.0001) and applying the suitable augmentation techniques, the following results were obtained:



(a) The model's loss with the current best settings



(b) The model's accuracy with the current best settings

Figure 5: Comparison of the results after adding the current best settings

Observations and deductions:

Loss: The training loss consistently decreases as the number of epochs increases, indicating that the model is learning and improving over time. The validation loss follows a similar trend, although there are slight fluctuations. This suggests that the model is generalizing well to unseen data, as the validation loss is also decreasing overall.

Accuracy: The training accuracy steadily increases with each epoch, indicating that the model is becoming more accurate in predicting the training data. The validation accuracy also shows improvement, although it may not increase as consistently as the training accuracy. This suggests that the model is effectively learning patterns and performing well on both the training and validation sets.

Overfitting: It's important to assess whether the model is overfitting, where it becomes too specialized in predicting the training data and performs poorly on unseen data. In this case, the decreasing training loss and increasing training accuracy without a significant decrease or plateau in validation metrics suggest that overfitting might not be a major concern. However, it would be helpful to evaluate the model's performance on additional test data to confirm its generalization capabilities.

Training convergence: The model's training loss and accuracy continue to improve over the 30 epochs, indicating that the training process was not cut off prematurely.

Validation performance: The validation loss and accuracy are generally lower and higher, respectively, compared to the corresponding training metrics. This indicates that the model is not overfitting and is performing reasonably well on unseen data.

4 Discussion

In this section, we discuss the strengths and weaknesses of each model, as well as the influence of the different data augmentation techniques on model performance. Our findings indicate that the original 50-layer ResNet outperform the custom CNN model in detecting Arabic dialects. Moreover, we observe that the data augmentation techniques have an impact on improving the models' robustness.

The main difference and the most important one is that we can now obtain significantly higher accuracy on the newly made model, reaching some quite promising results for the training as we got

≈100% accuracy and ≈80% validation accuracy and much lower loss values than the ones we previously had as we got ≈0 for the training and ≈1 for the validation loss after running the model for 30 Epochs.

Several factors might have led to such results. New data augmentation techniques and a more complex model (ResNet 50) helped with the data and the difference between the audios such as noise, pitch, source...

5 Potential Future Work

In addition to hyperparameters and data augmentations, there are several potential areas for future work in Arabic dialect detection:

- **Transfer Learning and Pretrained Models:** Investigate the application of transfer learning techniques by leveraging pretrained models from related tasks or large-scale audio datasets to improve accuracy and robustness.
- **Ensemble Methods:** Explore the use of ensemble methods to combine multiple models, benefiting from their diverse strengths and achieving higher accuracy and reliability.
- **Advanced Audio Processing:** Delve into advanced audio processing techniques like wavelet transforms or source separation to enhance representations and improve discrimination between dialects.
- **Adaptive Learning Techniques:** Explore adaptive learning methods such as curriculum learning or active learning to optimize the learning process based on data characteristics and performance.
- **Domain Adaptation and Few-Shot Learning:** Investigate techniques to bridge dialect gaps through domain adaptation or improve generalization with limited training data using few-shot learning approaches.
- **Multi-modal Approaches:** Incorporate complementary information from multiple modalities like textual or visual data to improve understanding and enhance accuracy and robustness.
- **Visual Learning:** Explore the integration of visual information, such as facial expressions or lip movements, to augment audio-based models and improve dialect detection.
- **Other Architectures:** Investigate alternative CNN architectures like VGGH, InceptionNet, or DenseNet, as well as architectures specifically designed for audio processing, to discover models better suited for Arabic dialect detection.

These potential future directions offer opportunities for further advancements in Arabic dialect detection, including leveraging pretrained models, ensemble methods, advanced audio processing, adaptive learning, domain adaptation, few-shot learning, multi-modal approaches, and alternative architectures.

6 References

Monigatti, L. (Mar 28). Data Augmentation Techniques for Audio Data in Python: How to augment audio in waveform (time domain) and as spectrograms (frequency domain) with librosa, numpy, and PyTorch. Towards Data Science. Retrieved from here.

Hartquist, J. (Year, Month Day). Fine-Tuning ResNet-18 for Audio Classification. Retrieved from here