ELTE
EÖTVÖS LORÁND
TUDOMÁNYEGYETEM

Speech recognition using class-based neural networks (Arabic dialect classification)

Hangfelismerés és osztályozas neurális hálók segítségével (Arab nyelvjárási osztályozás)

Ferfar Mohamed Chakib

Advisors: Pásztor Adél, Lukács András.

01/06/2023

TABLE OF CONTENTS

01

INTRODUCTION

Recapitulation of the events and progress made in project 1.

02

Our approach

Elaborating on the changes we will implement this time and the rationale behind them.

03

Results and observations

Showing what we got so far using our current approach

04

Lessons learned and conclusion

Discussing the main takeaways from our project so far and the possibilities to improve

01

INTRODUCTION

Recapitulation of the events and progress made in project 1.

INTRODUCTION

The first part of the project explored the use of Convolutional Neural Networks (CNNs) for recognizing and classifying Arabic dialects. The goal was to develop an accurate method by preprocessing the data, training CNNs, and evaluating different architectures and hyperparameters. Arabic dialects were diverse and complex, making classification challenging. The results showed less than 50% accuracy, suggesting the need for a larger and more diverse dataset and more sophisticated models.

02

Our approach

Elaborating on the changes we will implement this time and the rationale behind them.

What changes?

- During the concluding phase of the initial part of our project, we deliberated on several potential methods to enhance the outcomes and draw nearer to our objective. Among these, two approaches stood out: adjusting hyperparameters and implementing data augmentation. Now, as we proceed, we will address these strategies.

What are hyperparameters?

In the context of machine learning and deep learning, hyperparameters are parameters whose values are set before the learning process begins, as opposed to the parameters of the model, which are learned during the training process.

Those could be :

Learning Rate: This is one of the most critical hyperparameters. It determines the step size taken in the gradient descent process when updating the weights. A large learning rate might make the learning process faster but can overshoot the minimum point. A small learning rate might require more time to converge and might get stuck in local minima.

Batch Size: This is the number of training samples used in one iteration. Larger batch sizes use more memory but can lead to more accurate estimates of the gradient, while smaller batch sizes can make the training process faster.

Number of Epochs: An epoch is a single pass through the entire training dataset during the training process. The number of epochs is the number of times the learning algorithm will work through the entire training dataset.

Number of Layers and Neurons: These hyperparameters define the structure of the neural network. They greatly influence the complexity of the model and its capacity to learn intricate patterns. However, too many layers or neurons can lead to overfitting.

Activation Function: The activation function determines the output of a neuron given a set of inputs. Popular activation functions include ReLU, sigmoid, and tanh.

Regularization Parameters: These include dropout rate, weight decay coefficients etc., that are used to prevent overfitting.

Data augmentation techniques

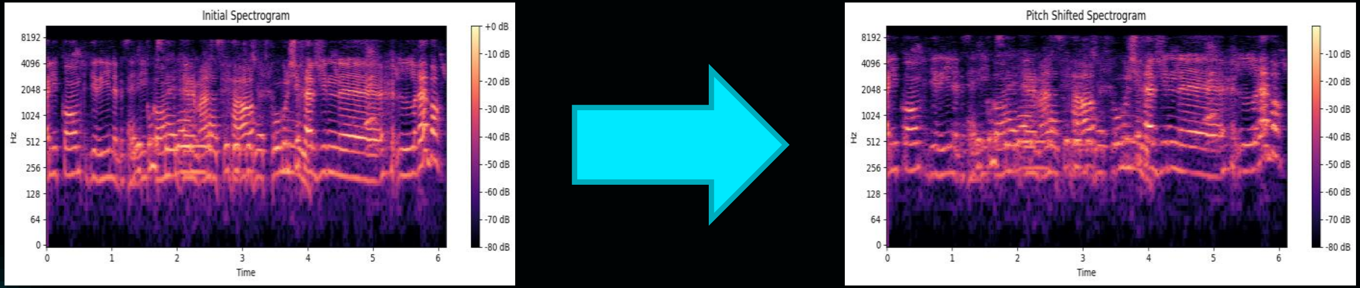
Data augmentation is a strategy that enables practitioners to significantly increase the diversity of data available for training models, without actually collecting new data.

Data augmentation techniques are designed to create, enhance, or otherwise alter data in a way that creates additional, valuable training material.

Which ones did we use?

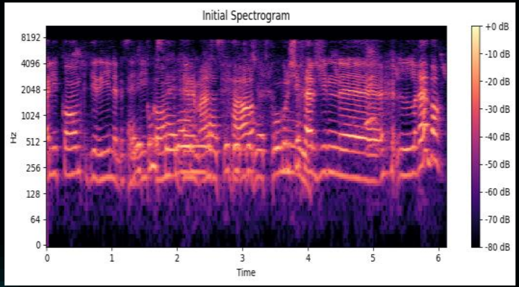
We will describe the purpose and effects of each data augmentation technique and provide a visual example by showcasing a spectrogram of one of our audio files.

Pitch Shifting: changing the pitch without changing the speed. This can mimic variations in the speaking pitch among different speakers.

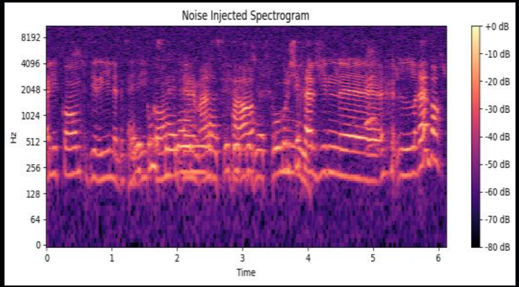


Which ones did we use?

Noise Injection: Adding random noise to the audio data. This can help the model become more robust to noise, which is especially useful if the test data might include noisy samples.



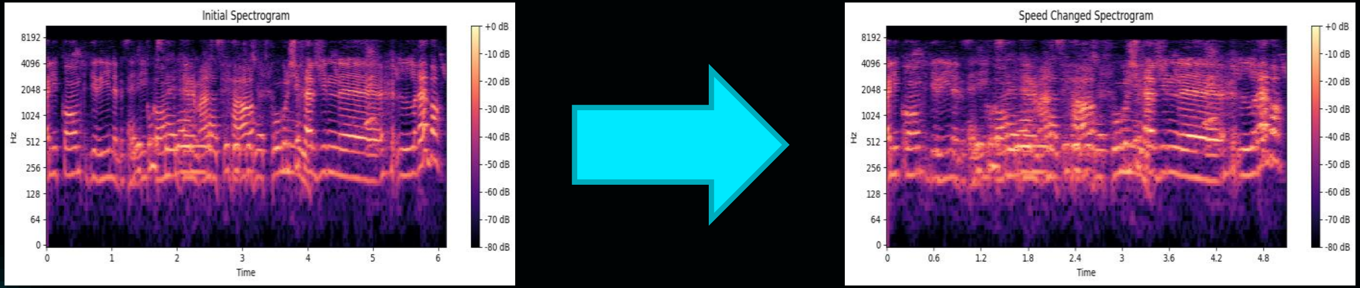
Initial Spectrogram



Noise Injected Spectrogram

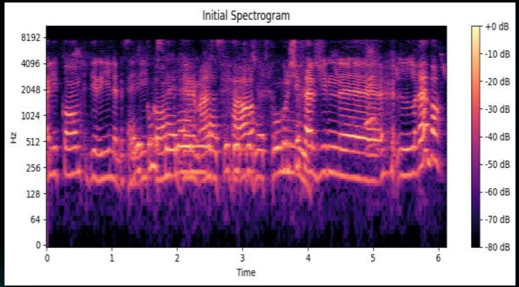
Which ones did we use?

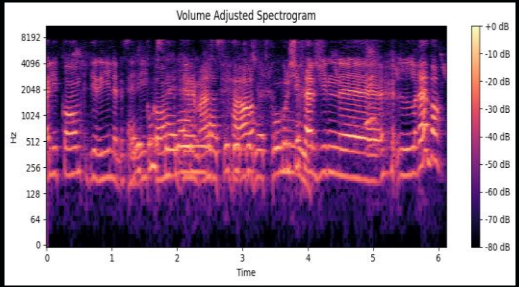
Speed Change: Altering the speed of the audio signal, also known as time stretching or time compression, without affecting its pitch. This can simulate variations in the speaking rate of different individuals. Slowing down or speeding up the audio can introduce different temporal patterns for the model to learn from.



Which ones did we use?

Volume Adjustment: Adjusting the loudness of the audio data. This can make your model invariant to the volume of the speech.





03

Results and observations

Showing what we got so far using our current approaches

What did we get?

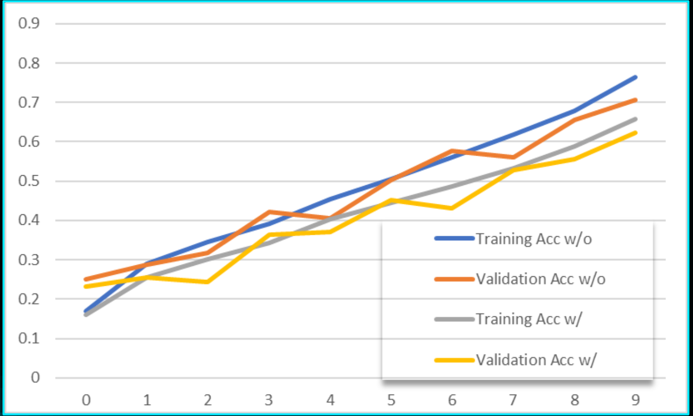
Pitch shift:

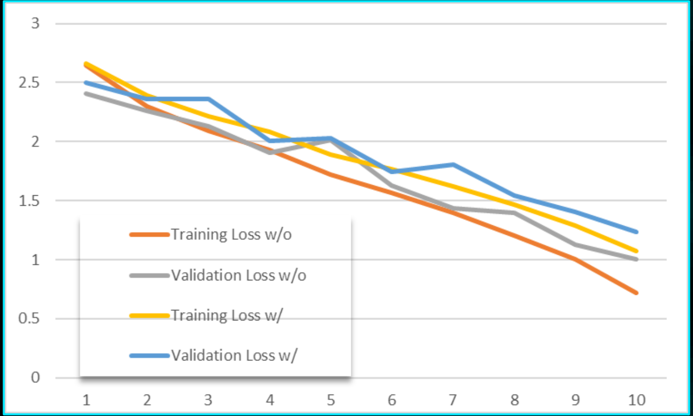
Pitch shift augmentation generally results in higher loss values during training, indicating increased difficulty in predicting target labels. However, it has the potential to improve the model's ability to generalize, as seen in comparable or slightly higher validation accuracy.

Speed change:

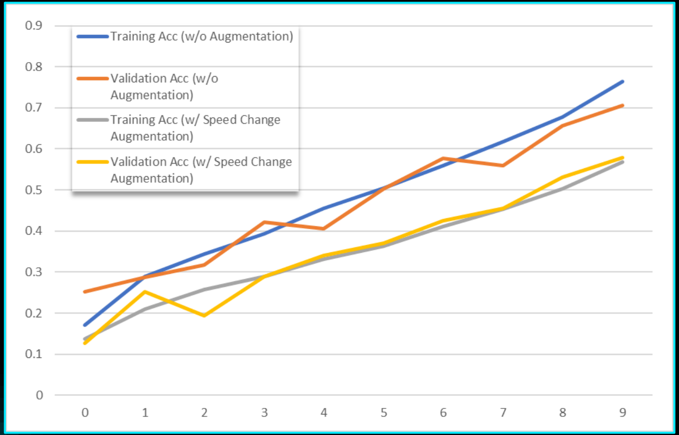
The speed change augmentation introduces variability in the data, resulting in higher losses and mixed impact on accuracy. Overall, it does not consistently improve the model's performance compared to the model without augmentation.

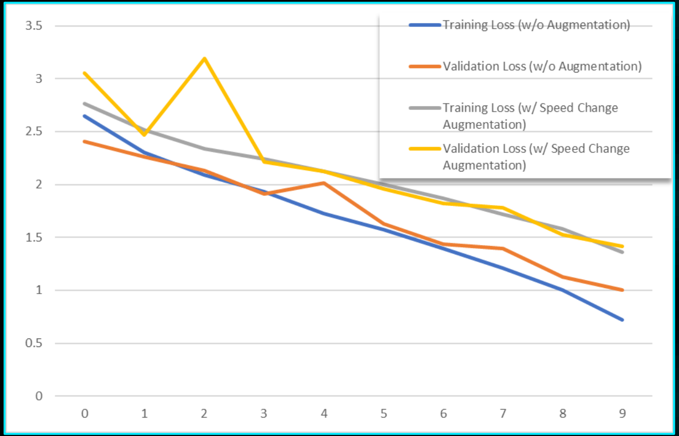
Pitch shift results:





Speed change results:





What did we get?

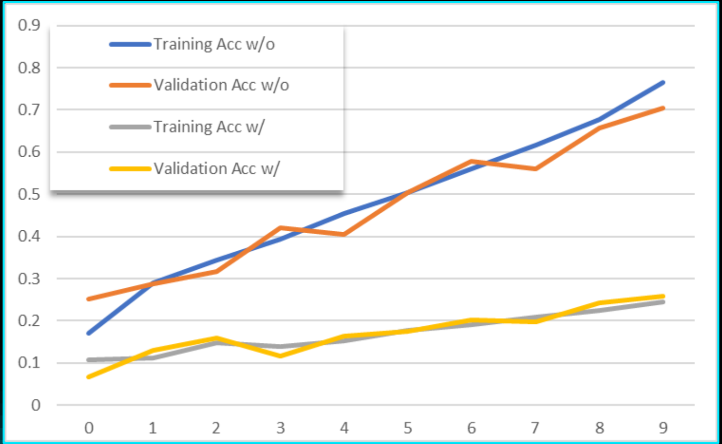
Noise injection:

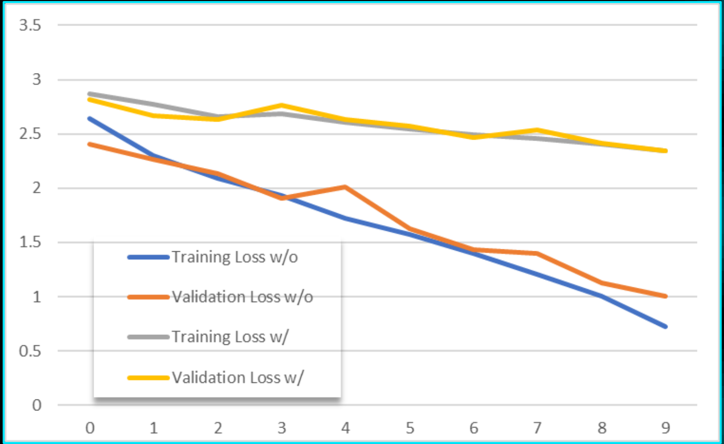
The background noise augmentation leads to higher losses and consistently lower accuracy in both training and validation compared to the model without augmentation. This suggests that the introduction of background noise negatively affects the model's performance, indicating that it does not improve accuracy.

Volume adjustment:

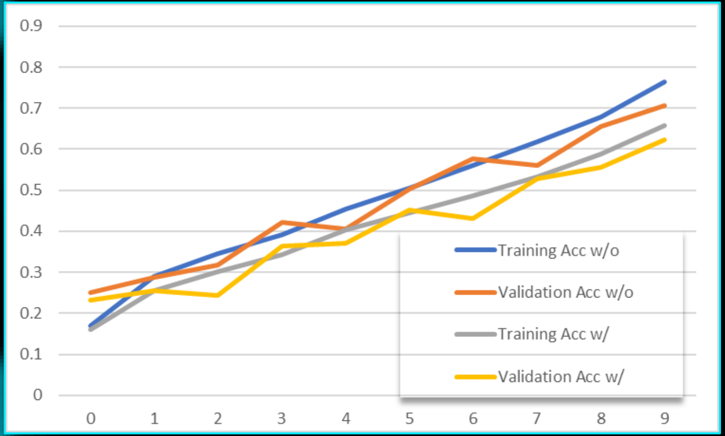
The volume control augmentation leads to comparable or slightly better loss values and generally higher accuracy in both training and validation. This suggests that it effectively controls volume levels, resulting in improved performance and accurate predictions. Overall, volume control augmentation positively impacts the model's accuracy and generalization.

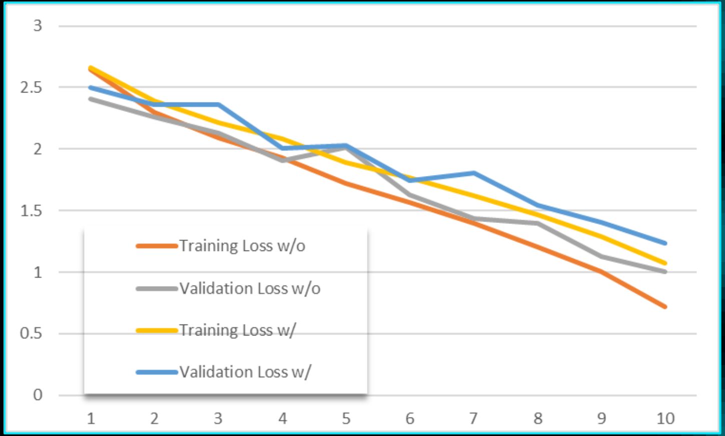
Noise injection results:





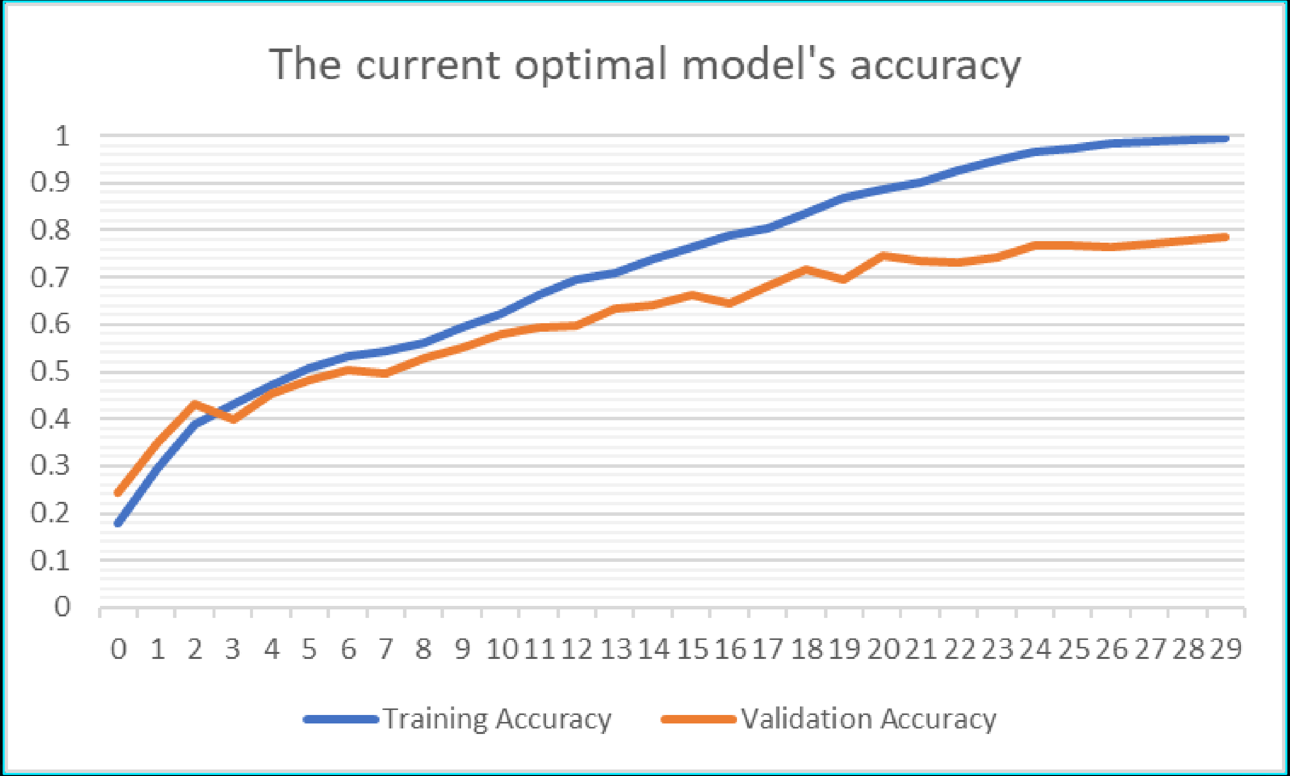
Volume adjustment results:

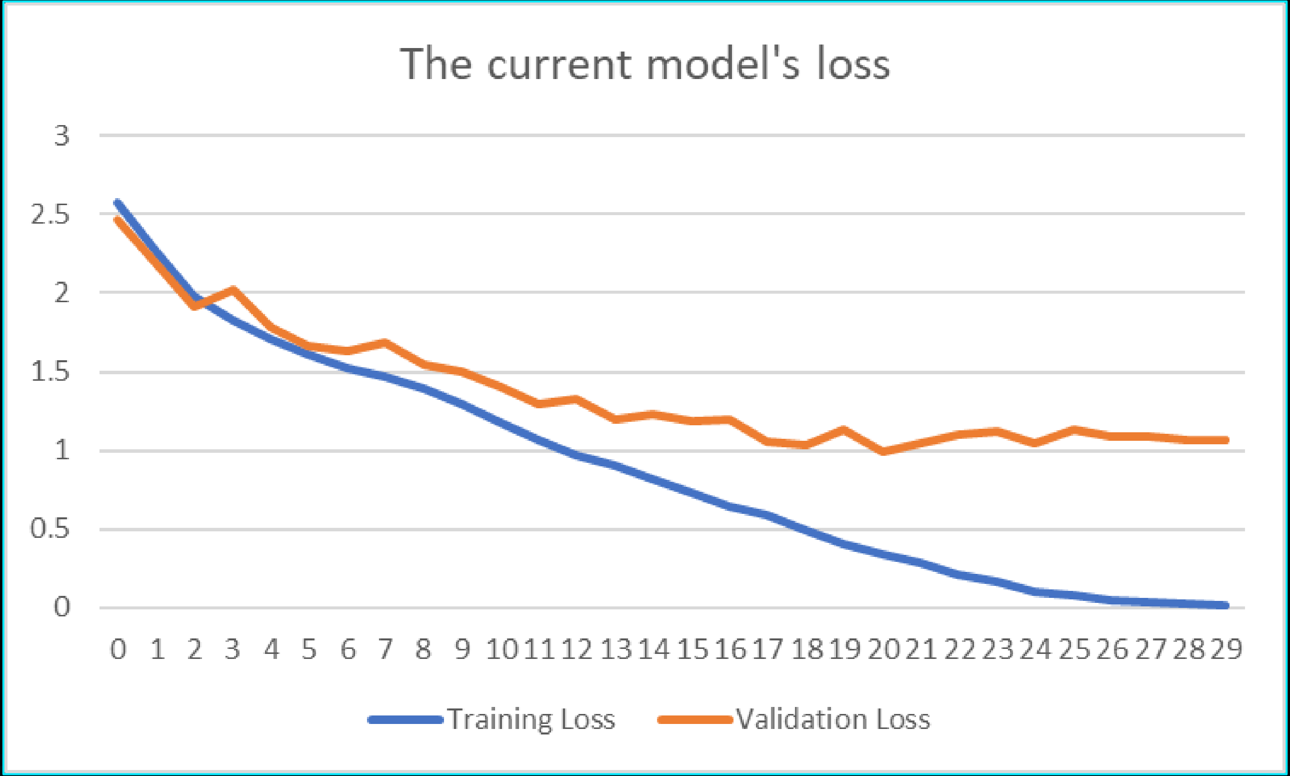




Best result:

The conducted evaluations involved adjusting hyperparameters (LR=0.001, batch size=17, 30 epochs, 50 layers, weight decay=0.0001) and implementing suitable augmentation techniques. The results showed that the model consistently improved over time, with decreasing training and validation loss. Training accuracy steadily increased, and validation accuracy showed improvement, indicating the model's ability to learn patterns and perform well on both training and validation sets. Overfitting was not a major concern, as the model's training loss decreased without a significant drop in validation metrics. The model exhibited reasonable performance on unseen data, as indicated by lower validation loss and higher validation accuracy.


The current optimal model's accuracy
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
Training Accuracy
Validation Accuracy


The current model's loss
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
Training Loss
Validation Loss

04

Lessons learned and conclusion

Discussing the main takeaways from our project so far and the possibilities to improve

Lessons learned and conclusion

Our study compared two models, the custom CNN and ResNet-50, for Arabic dialect detection. ResNet-50 outperformed the custom CNN, achieving higher accuracy and lower loss values. Data augmentation techniques and the complexity of ResNet-50 contributed to these improvements.

In conclusion, the comparative study of custom and ResNet-50 models for Arabic dialect detection has provided valuable insights. The ResNet-50 model outperformed the custom CNN model in detecting Arabic dialects. Additionally, the impact of different data augmentation techniques was evaluated. While pitch shift and speed change augmentations showed mixed results, background noise augmentation had a negative impact on model performance. On the other hand, volume control augmentation had a positive impact, improving accuracy and generalization.

Future work

For future work, there are several potential areas to explore in Arabic dialect detection. These include investigating transfer learning with pretrained models, employing ensemble methods for improved accuracy and reliability, exploring advanced audio processing techniques, utilizing adaptive learning methods, addressing domain adaptation and few-shot learning challenges, integrating multi-modal approaches, leveraging visual information, and exploring alternative architectures designed for audio processing

REFERENCES

- Monigatti, L. (Mar 28). Data Augmentation Techniques for Audio Data in Python: How to augmentaudio in waveform (time domain) and as spectrograms (frequency domain) with librosa, numpy, andPyTorch. Towards Data Science. Retrieved from here.
- Hartquist, J. (Year, Month Day). Fine-Tuning ResNet-18 for Audio Classification. Retrieved from here.

THANK YOU!

Speech recognition using class-based neural networks (Arabic dialect classification)

Advisors: Pásztor Adél, Lukács András

Ferfar Mohamed Chakib

01/06/2023

ELTE
EÖTVÖS LORÁND
TUDOMÁNYEGYETEM

1. Introduction
2. Our Approach
3. Results and Observations
4. Lessons Learned and Conclusion

1. Introduction
2. Our Approach
3. Results and Observations
4. Lessons Learned and Conclusion

Introduction

The first part of the project explored the use of Convolutional Neural Networks (CNNs) for recognizing and classifying Arabic dialects. The goal was to develop an accurate method by preprocessing the data, training CNNs, and evaluating different architectures and hyperparameters. Arabic dialects were diverse and complex, making classification challenging. The results showed less than 50% accuracy, suggesting the need for a larger and more diverse dataset and more sophisticated models.

1 Introduction

2 Our Approach

3 Results and Observations

4 Lessons Learned and Conclusion

Our Approach

In this section we will be elaborating on the changes we will implement this time and the rationale behind them.

What changes?

During the concluding phase of the initial part of our project, we deliberated on several potential methods to enhance the outcomes and draw nearer to our objective. Among these, two approaches stood out: adjusting hyperparameters and implementing data augmentation. Now, as we proceed, we will address these strategies.

What are hyperparameters?

In the context of machine learning and deep learning, hyperparameters are parameters whose values are set before the learning process begins, as opposed to the parameters of the model, which are learned during the training process.

Some examples of hyperparameters include:

- **Learning Rate**: This is one of the most critical hyperparameters. It determines the step size taken in the gradient descent process when updating the weights. A large learning rate might make the learning process faster but can overshoot the minimum point. Conversely, a small learning rate might require more time to converge and might get stuck in local minima.

What are hyperparameters?

In the context of machine learning and deep learning, hyperparameters are parameters whose values are set before the learning process begins, as opposed to the parameters of the model, which are learned during the training process.

Some examples of hyperparameters include:

- **Batch Size**: This refers to the number of training samples used in one iteration. Larger batch sizes use more memory but can lead to more accurate estimates of the gradient, while smaller batch sizes can make the training process faster.

What are hyperparameters?

In the context of machine learning and deep learning, hyperparameters are parameters whose values are set before the learning process begins, as opposed to the parameters of the model, which are learned during the training process.

Some examples of hyperparameters include:

- **Number of Epochs**: An epoch is a single pass through the entire training dataset during the training process. The number of epochs is the number of times the learning algorithm will work through the entire training dataset.
- **Number of Layers and Neurons**: These hyperparameters define the structure of the neural network. They greatly influence the complexity of the model and its capacity to learn intricate patterns. However, too many layers or neurons can lead to overfitting.

What are hyperparameters?

In the context of machine learning and deep learning, hyperparameters are parameters whose values are set before the learning process begins, as opposed to the parameters of the model, which are learned during the training process.

Some examples of hyperparameters include:

- **Activation Function**: The activation function determines the output of a neuron given a set of inputs. Popular activation functions include ReLU, sigmoid, and tanh.
- **Regularization Parameters**: These include dropout rate, weight decay coefficients, etc., that are used to prevent overfitting.

Data Augmentation Techniques

Which ones did we use?
We will describe the purpose and effects of each data augmentation technique and provide a visual example by showcasing a spectrogram of one of our audio files.

Pitch Shifting: Changing the pitch without changing the speed. This can mimic variations in the speaking pitch among different speakers.

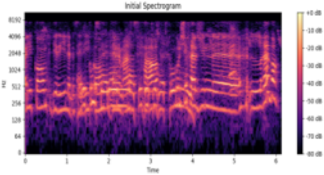


Figure 1: Before Pitch Shift

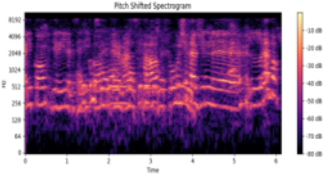


Figure 2: After Pitch Shift

Data Augmentation Techniques (Cont.)

Noise Injection: Adding random noise to the audio data. This can help the model become more robust to noise, which is especially useful if the test data might include noisy samples.

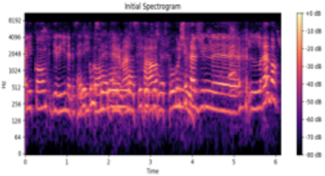


Figure 3: Before Noise Injection

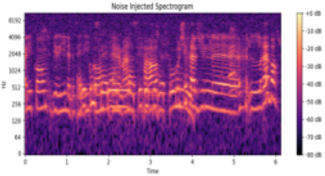


Figure 4: After Noise Injection

Data Augmentation Techniques (Cont.)

Speed Change: Altering the speed of the audio signal, also known as time stretching or time compression, without affecting its pitch. This can simulate variations in the speaking rate of different individuals. Slowing down or speeding up the audio can introduce different temporal patterns for the model to learn from.

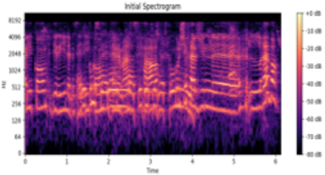


Figure 5: Before Speed Change

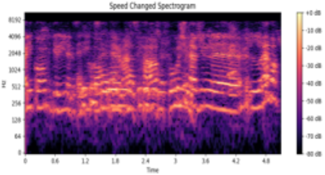


Figure 6: After Speed Change

Data Augmentation Techniques (Cont.)

Volume Adjustment: Adjusting the loudness of the audio data. This can make your model invariant to the volume of the speech.

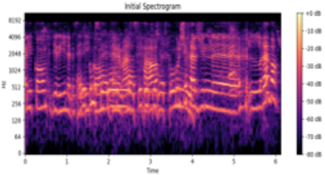


Figure 7: Before Volume Adjustment

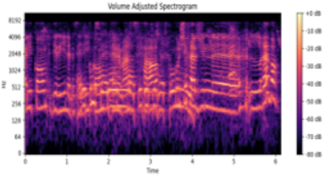


Figure 8: After Volume Adjustment

1 Introduction

2 Our Approach

3 Results and Observations

4 Lessons Learned and Conclusion

Results and Observations

Now we will showcase the outcomes obtained thus far using our current approaches.

Pitch Shift:
The pitch shift augmentation generally leads to higher loss values during training, indicating increased difficulty in predicting target labels. However, it has the potential to improve the model's ability to generalize, as evidenced by comparable or slightly higher validation accuracy.

Speed Change:
The speed change augmentation introduces variability in the data, resulting in higher losses and yielding mixed impact on accuracy. Overall, it does not consistently improve the model's performance compared to the model without augmentation.

Results and Observations

Pitch Shift Results:

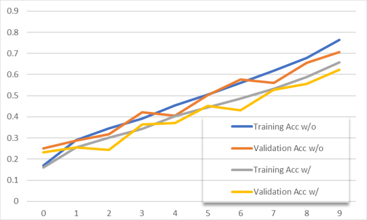


Figure 9: Accuracy with Pitch Shift

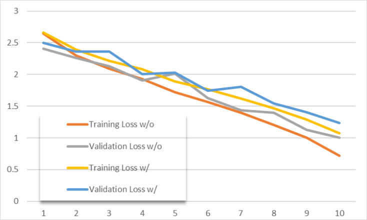


Figure 10: Loss with Pitch Shift

Results and Observations

Speed Change Results:

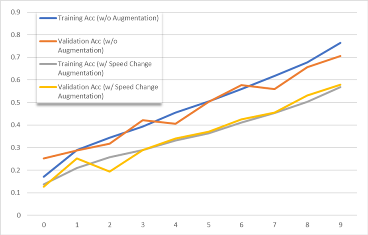


Figure 11: Accuracy with Speed Change

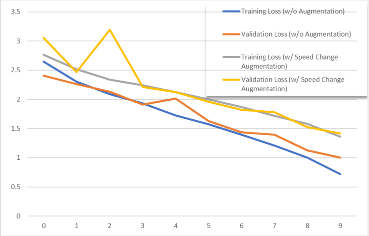


Figure 12: Loss with Speed Change

Results and Observations

Noise injection:

The background noise augmentation leads to higher losses and consistently lower accuracy in both training and validation compared to the model without augmentation. This suggests that the introduction of background noise negatively affects the model's performance, indicating that it does not improve accuracy.

Volume adjustment:

The volume control augmentation leads to comparable or slightly better loss values and generally higher accuracy in both training and validation. This suggests that it effectively controls volume levels, resulting in improved performance and accurate predictions. Overall, volume control augmentation positively impacts the model's accuracy and generalization.

Results and Observations

Noise Injection Results:

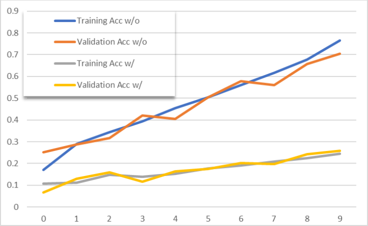


Figure 13: Accuracy with Noise Injection

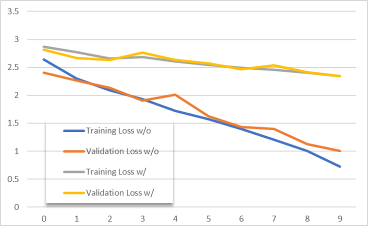


Figure 14: Loss with Noise Injection

Results and Observations

Volume Adjustment Results:

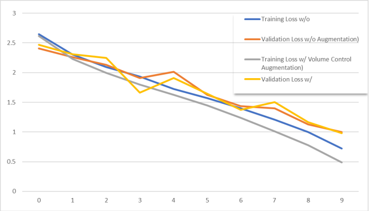


Figure 15: Accuracy with Volume Adjustment

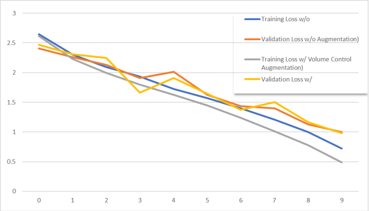


Figure 16: Loss with Volume Adjustment

Results and Observations

Best Result:
The conducted evaluations involved adjusting hyperparameters (LR=0.001, batch size=17, 30 epochs, 50 layers, weight decay=0.0001) and implementing suitable augmentation techniques. The results showed that the model consistently improved over time, with decreasing training and validation loss. Training accuracy steadily increased, and validation accuracy showed improvement, indicating the model's ability to learn patterns and perform well on both training and validation sets. Overfitting was not a major concern, as the model's training loss decreased without a significant drop in validation metrics. The model exhibited reasonable performance on unseen data, as indicated by lower validation loss and higher validation accuracy.

Results and Observations

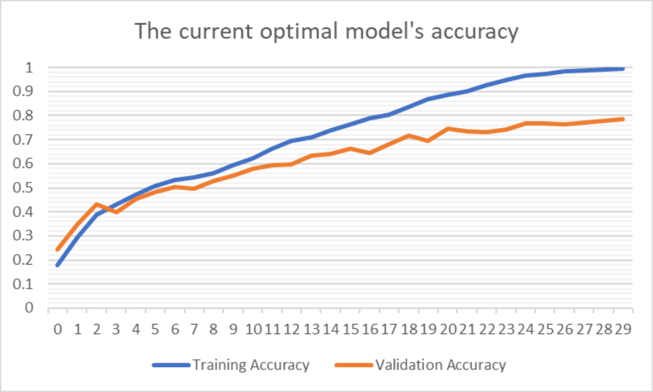


Figure 17: Best Accuracy

Results and Observations

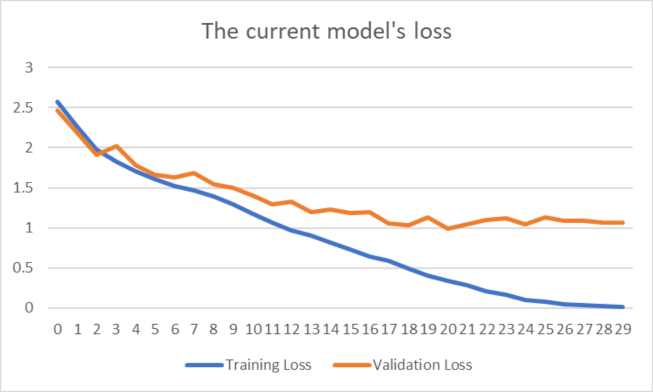


Figure 18: Best Loss

1. Introduction
2. Our Approach
3. Results and Observations
4. Lessons Learned and Conclusion

What Did We Learn?

In our project, we compared two models, the custom CNN and ResNet-50, for Arabic dialect detection. The ResNet-50 model outperformed the custom CNN, achieving higher accuracy and lower loss values. These improvements were attributed to the complexity of ResNet-50 and the use of data augmentation techniques.

In conclusion, our comparative study of custom and ResNet-50 models for Arabic dialect detection has provided valuable insights. The ResNet-50 model proved to be more effective in detecting Arabic dialects. We also evaluated the impact of different data augmentation techniques. While pitch shift and speed change augmentations showed mixed results, background noise augmentation negatively affected the model's performance. On the other hand, volume control augmentation had a positive impact, improving accuracy and generalization.

Future Work

Moving forward, there are several potential areas to explore in Arabic dialect detection. These include:

- Investigating transfer learning with pretrained models
- Employing ensemble methods for improved accuracy and reliability
- Exploring advanced audio processing techniques
- Utilizing adaptive learning methods
- Addressing domain adaptation and few-shot learning challenges
- Integrating multi-modal approaches and leveraging visual information
- Exploring alternative architectures designed for audio processing

[1] J. Hartquist. *Fine-Tuning ResNet-18 for Audio Classification*. Nov 10, 2022. URL: https://wandb.ai/jhartquist/fastaudio-esc-50/reports/Fine-Tuning-ResNet-18-for-Audio-Classification--VmlldzoyNjU3OTQ.

[2] L. Monigatti. *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*. Mar 28,2023. URL: https://towardsdatascience.com/data-augmentation-techniques-for-audio-data-in-python-15505483c63c.

Thanks!