BME FACULTY OF ELECTRICAL ENGINEERING AND INFORMATICS

MŰEGYETEM 1782

Architectural Enhancements and Feature Optimization of AutoVocoder for High Quality Speech Synthesis

Author: Riad Larbi
Supervisor: Dr. Mohammed Salah Al-Radhi

SmartLab Intelligent Interactions

February 3rd, 2025

TMiT

Table of contents

1

Introduction

2

Methodology

3

Results

4

Summary

1. Introduction

1. Background

2. Problem Definition

1. Background:

The implementation of neural network architectures in speech synthesis is one of the most researched tasks in signal processing.

In most state of the art models, synthesizing speech is approached by representing the audio as a mel-spectrogram, which allows speech synthesis to be treated similarly to image generation tasks leveraging CNNs.

The fundamental problem in solely relying on mel-spectrograms, is the loss of important features, notably the *phase* information.

1. Background:

The AutoVocoder addressed this problem by allowing the model to learn its own representation of the speech.

By Implementing an AutoEncoder architecture, the encoder uses phase, magnitude, real and imaginary spectrums as features to learn how to accurately represent the audio.

The decoder uses that representation to reproduce the waveform.

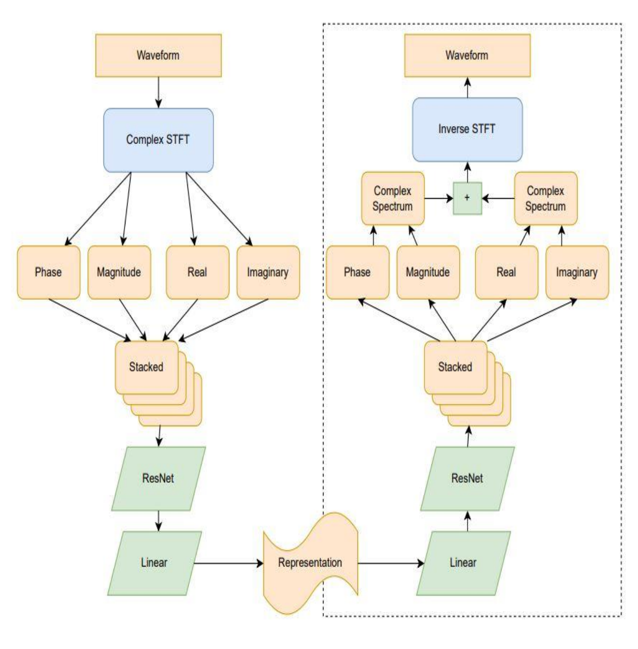


Figure 1 **Autovocoder architecture. Dashed box shows decoder**

1. Background:

The decoder follows a mirrored structure of the encoder, both utilizing ResBlocks as the core of the encoding-decoding process.

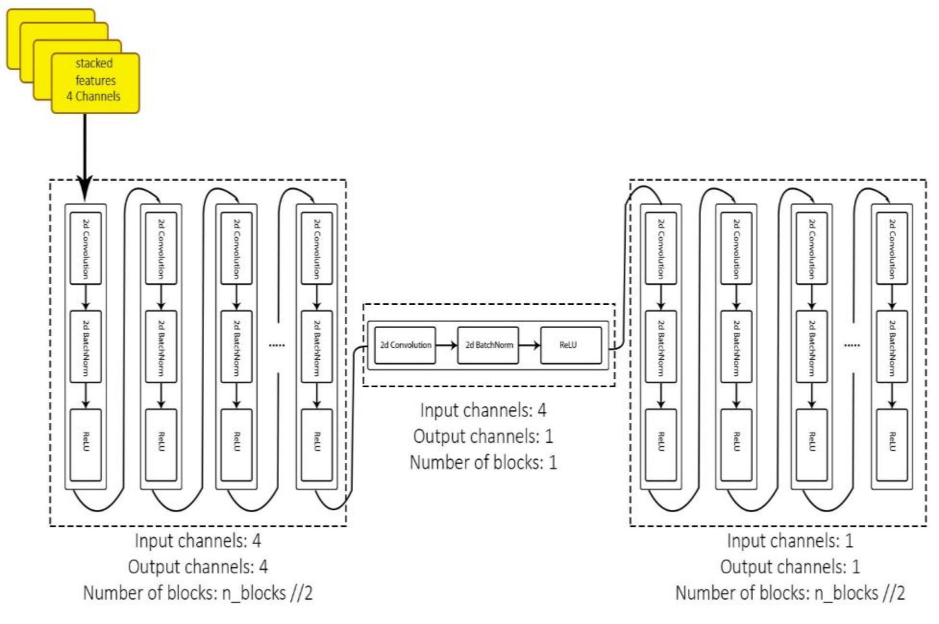


Figure 2 **Autovocoder's Encoder ResNet architecture**

2. Problem Definition:

The AutoVocoder's novelty consists in the learning of better waveform representations using CNNs encoding.

There are still some details that could be leveraged to produce more accurate representations.

Capturing these details necessitates adjustments in the data pre-processing and conditioning, architecture modifications, and better refinement in the post-processing.

2. Problem Definition:

The proposed model will focus on key improvements in three different stages:

Pre-Processing: conditioning the data to better suit the new architecture.

Architecture: designing a configuration that allows capturing more details.

Post-Processing: refining the synthesized audio for better quality in the results.

Each stage is designed to address existing limitations in the AutoVocoder.

2. Methodology

1. Preprocessing
2. Architecture
3. Postprocessing
4. Training &Dataset
5. Evaluation Metrics

1. Preprocessing:

Log-Magnitude: instead of the linear magnitude in the baseline model, the *log* of the magnitude is used, compressing the dynamic range of the feature.

Sine-Cosine Phase: this representation is useful to capture periodic patterns, while still preserving the phase information which is critical for determining the timing and tonal quality of the speech.

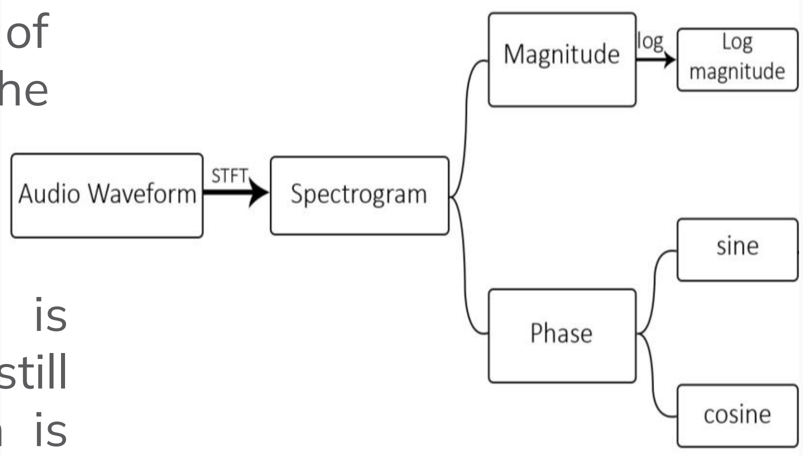


Figure 3 **Waveform representation and processing pipeline.**

1. Preprocessing:

The features are then stacked, and ready to be forwarded to the encoder.

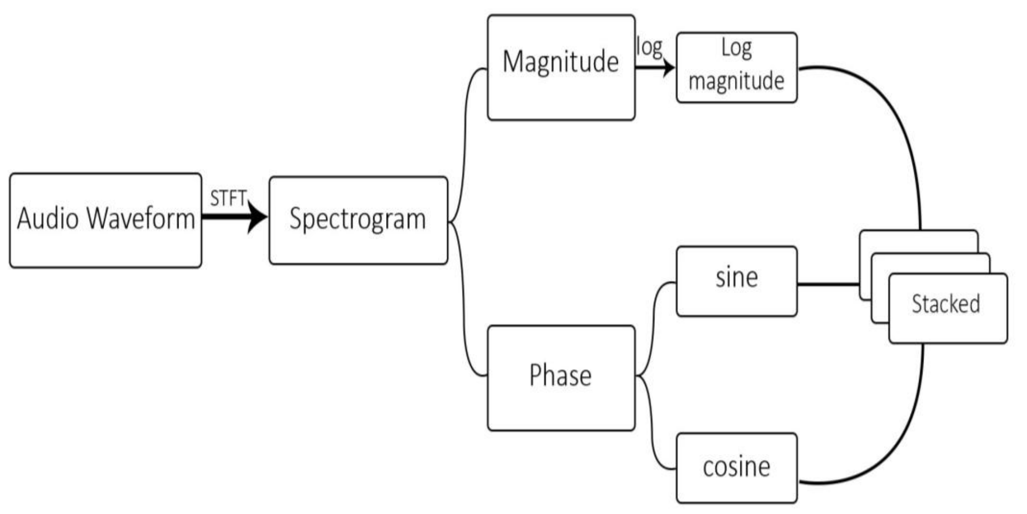


Figure 4 **Waveform representation and processing pipeline.**

2. Architecture:

Replacing ResNet with ConvNeXtV2:

ConvNeXtV2 has demonstrated superior performance in image classification tasks, offering higher accuracy and efficient feature extraction.

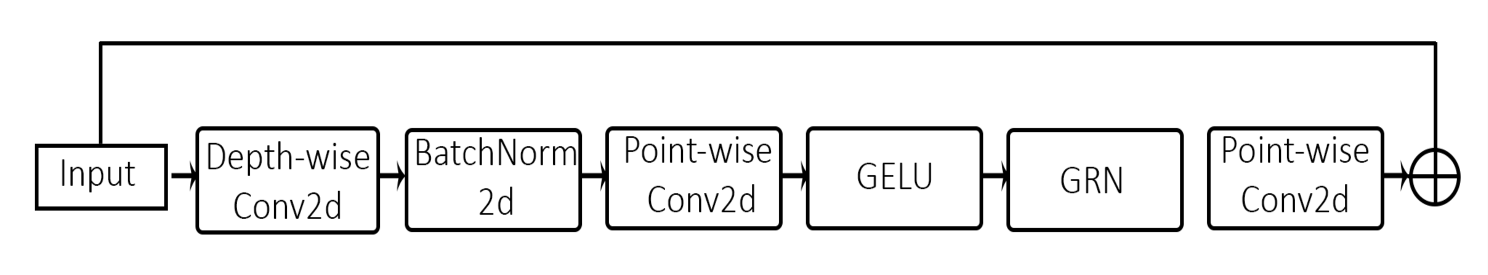


Figure 5 **ConvNeXtV2 block**

2. Architecture:

Replacing ResNet with ConvNeXtV2:

ConvNeXtV2 Blocks will improve upon ResBlocks by implementing:

Depth-wise Convolution: An efficient convolution method that applies a separate filter to each input channel independently

Point-wise Convolution: A 1x1 kernel convolution that enables channel information mixing and dimensionality reduction.

GELU (Gaussian Error Linear Unit): Compared to ReLU, it offers smoother non-linearity and improved gradient flow.

GRN (Global Response Normalization): A layer that normalizes preceding layer outputs by computing global response normalization with learnable scaling and shifting parameters.

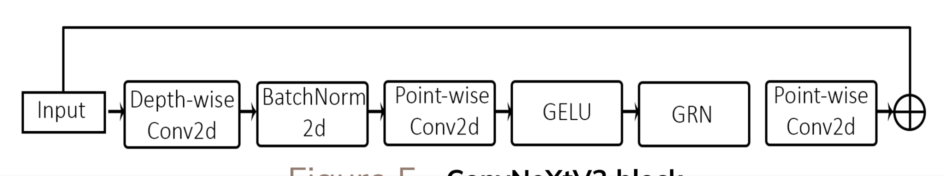


Figure 5 **ConvNeXtV2 block**

2. Architecture:

Separate Phase and Magnitude Encoders:

A key architecture change in the proposed model is a separate encoder for phase and magnitude.

This approach recognizes both features as distinct and fundamentally different, hence allowing the model to learn unique patterns for each separately.

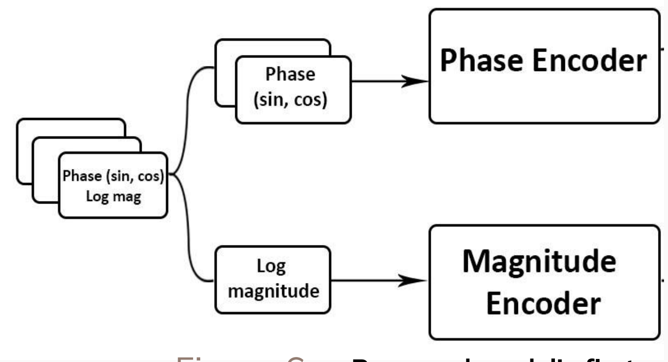


Figure 6 **Proposed model's first encoding steps architecture**

2. Architecture:

Separate Phase and Magnitude Encoders:

The Phase Encoder takes sine-cosine channels as input.

Forwards both channels through a series for ConvNextV2 blocks.

Outputs a single channel representing the encoded phase.

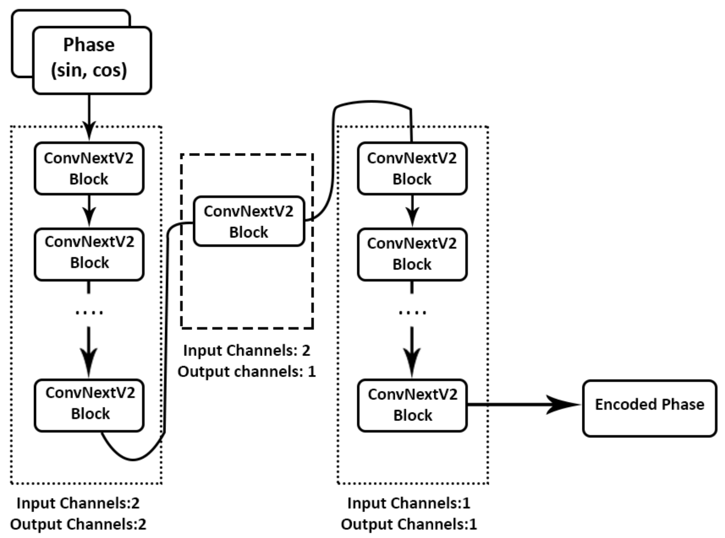


Figure 7 **Phase Encoder architecture**

2. Architecture:

Separate Phase and Magnitude Encoders:

The Mangnitude Encoder takes log-magnitude as input

Forwards the single channel through a series for ConvNextV2 blocks.

Outputs a single channel representing the encoded magnitude.

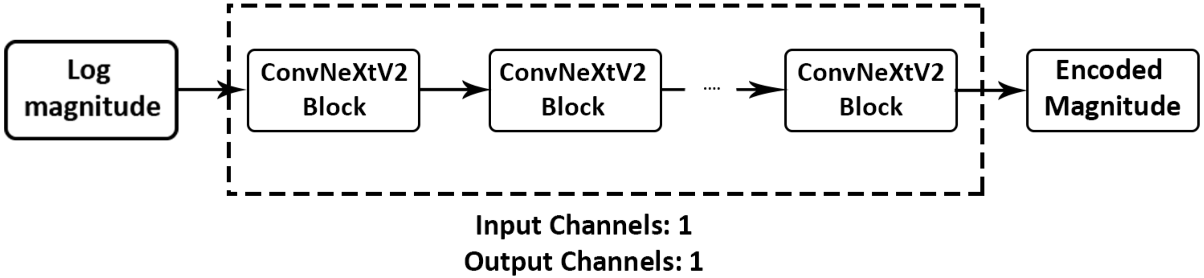


Figure 8 Magnitude Encoder’s architecture

2. Architecture:

Unified Encoder:

The Encoded magnitude and phase are concatenated, and forwarded to a U.E

The Unified Encoder takes Real & Imaginary spectrums as extra features, these features are only used for the encoding process.

The U.E outputs a single channel tensor after a series of ConvNextV2 Blocks.

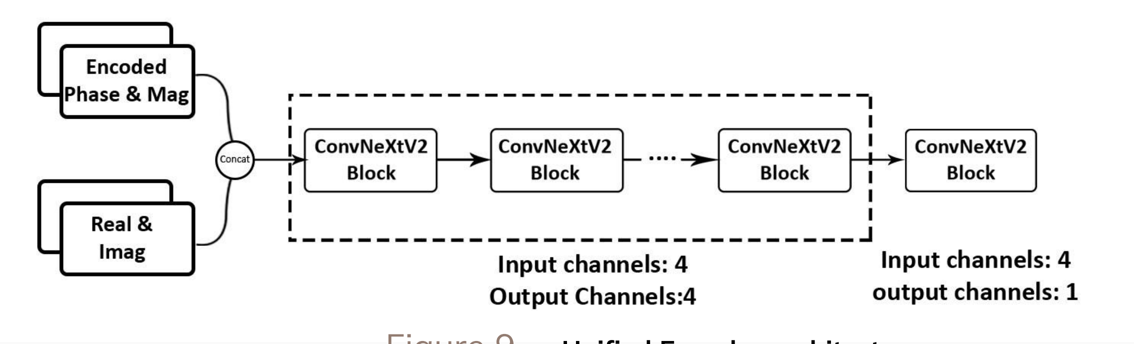


Figure 9 **Unified Encoder architecture**

2. Architecture:

Finally, the single channel tensor that the U.E outputs is passed through a linear layer, obtaining our representation:

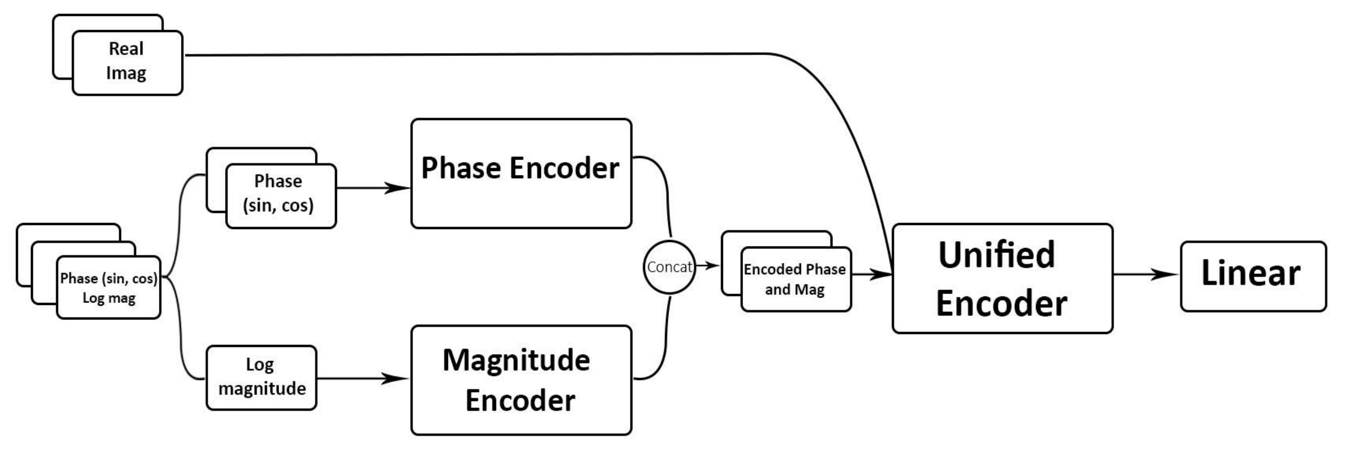


Figure 10 **Proposed model's Encoder architecture**

2. Architecture:

Decoder:

The decoder consists of mirrored steps of the encoder, without reproducing the real and imaginary spectrums and using phase and magnitude to reconstruct the waveform.

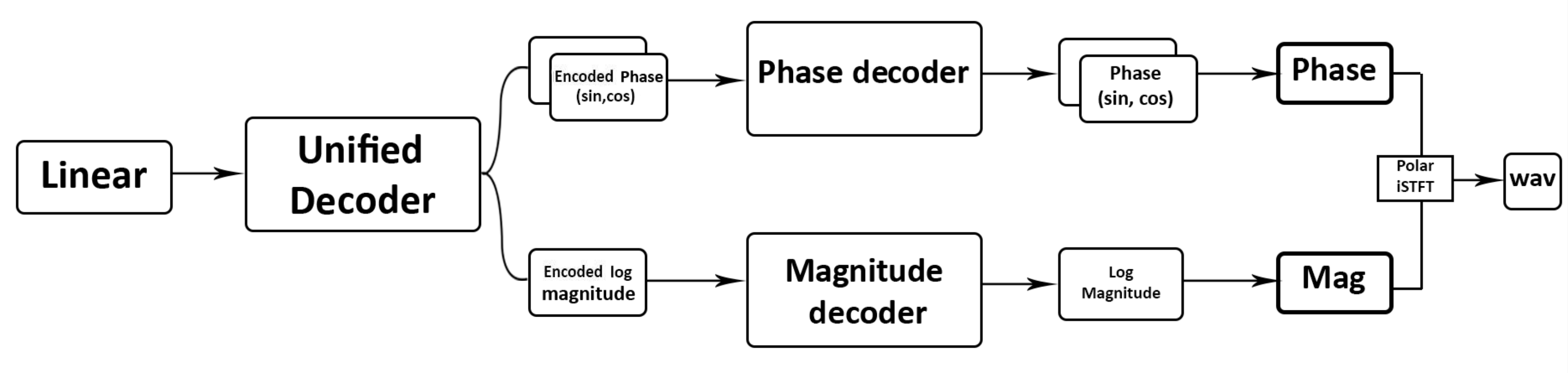


Figure 11 **Proposed Model's decoder**

3. Post-Processing:

Spectral Gating :

This serves to remove unwanted frequencies while preserving the quality, and is done through :

Noise Profile Estimation: Identify noise characteristics from non-speech segments.

Threshold Determination: Calculate frequency-based thresholds to separate noise from speech.

Applying Spectral Gate: Transforming the audio to the frequency domain (STFT), and attenuating frequencies below the threshold.

4. Training & Dataset:

LJSpeech 1.1 Dataset:

A Dataset consisting of 13100 short audio clips of a native female English speaker.
The total audio length is over 24 hours, and each clip is accompanied by its corresponding text transcription.

Training:

Conducted training on SmartLab provided server, taking approximately 100 hours to complete

Speech clips were randomly cropped to 8000 samples.

5. Evaluation Metrics

Objective Evaluation metrics:

Root Mean Square Error of Logarithmic Amplitude Spectra (LAS-RMSE):
evaluates the difference in the logarithmic amplitude spectra between the synthesized and reference speech.

Root Mean Square Error of F0 (F0-RMSE):
measures the accuracy of fundamental frequency (pitch) synthesis by calculating the error between the reference and synthesized F0 values.

Voiced/Unvoiced (V/UV) Error:
measures the proportion of frames incorrectly classified as voiced or unvoiced in the synthesized speech compared to the reference speech.

5. Evaluation Metrics

Subjective Evaluation metric:

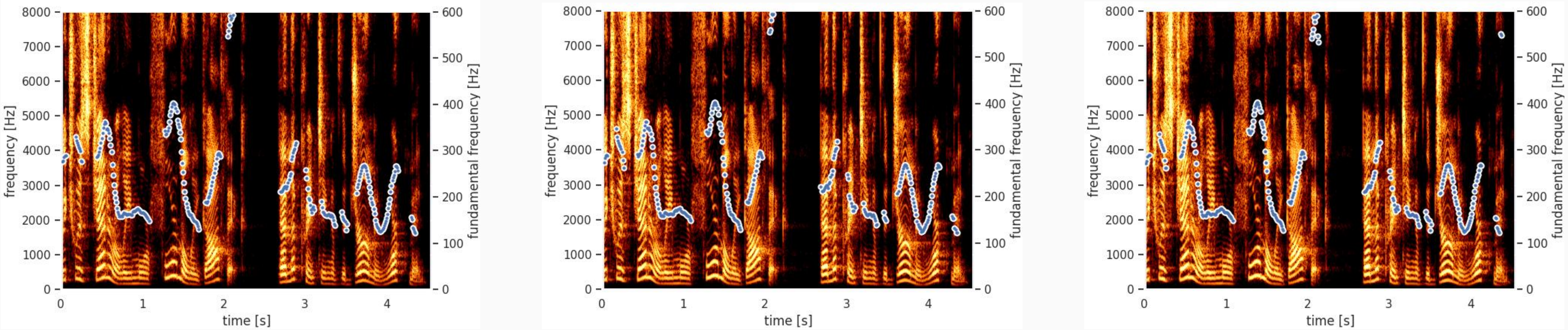
MosNet:
To evaluate the perceptual quality of the synthesized speech of our model, MOSNet was implemented.
This model gives a mean opinion score (MOS) on the perceptual quality of the speech.

Noisy data:
The proposed model was also evaluated with synthesizing noisy speech, performing a subjective evaluation of the synthesized speech.

3. Results

1. Comparison
2. Objective Evaluations
3. Subjective Evaluations
4. Robustness Evaluations

1. Comparison:



PROPOSED

ORIGINAL

BASELINE

which was generally more formally Gothic, than the printing of the German workmen

Figure 12 **The comparison of spectrograms and F0 values between the proposed, baseline autovocoder, and original speech. The text represents the transcript of the speech.**

2. Objective Evaluations:

	LAS-RMSE (dB) (lower is better)	F0-RMSE (lower is better)	V/UV Error (%) (lower is better)
Baseline	7.34	0.170	3.48
Proposed	**7.10**	**0.109**	**2.54**

Table 1 **Objective Evaluation results for the Baseline and the proposed Autovocoder**

The proposed model achieved lower errors in the objective evaluations stated in the table above. With the F0-RMSE being the biggest improvement, where the error is 34% lower for the proposed model, followed by a 27% improvement in V/UV Error and 5% improvement in LAS-RMSE.

3. Subjective Evaluations:

	Score
Baseline	3.01
Proposed	3.07
Original	**3.11**

Table 2 **MOSNet Subjective Evaluation results for the original, baseline and the proposed Autovocoder**

The proposed model achieved better scores in the evaluation with MOSNet, offering a more natural sounding synthesized speech.

4. Robustness Evaluation:

	Score
Baseline	2.93
Proposed	**3.01**
Original	2.99

Table 3 **MOSNet Subjective Evaluation results of noisy speech for the original, baseline and proposed Autovocoder**

The proposed model performed very well in synthesizing noisy audio, exceeding the score of the original speech and producing better perceptual quality.

4. Summary

1. Conclusion
2. Future Work

1. Conclusion:

this study presents significant advancements in the AutoVocoder for Text-to-Speech systems by addressing critical areas of data preprocessing, architectural design, and post-processing.

These enhancements collectively improve the model’s ability to generate clearer, more natural, and high-quality speech, as evidenced by objective and subjective evaluations.

The robustness evaluation showed potential for better handling of noisy data, which could be enhanced with proper fine tuning.

2. Future Work:

Implementing attention mechanisms and transformer based encoding for achieving better representations.

Leveraging other speech features such as F0 and MFCCs.

Optimizing the model to reduce the inference time and increase efficiency, for real-life application and resource limited environment like mobile devices.

Thank You!