Quantum speech recognition and synthesis: a comparative study

Gabriele Morello
Department of Computer Science
Royal Institute of Technology KTH
Stockholm, SE
morello@kth.se

Ennio Rampello
Department of Computer Science
Royal Institute of Technology KTH
Stockholm, SE
ennio@kth.se

Stefano Cubeddu
Department of Computer Science
Royal Institute of Technology KTH
Stockholm, SE
cubeddu@kth.se

Ioannis Theodosiou
Department of Computer Science
Royal Institute of Technology KTH
Stockholm, SE
ioathe@kth.se

Abstract

[TODO]

1 Introduction

The project explores two different approaches for speech recognition: Convolutional Neural Networks (CNN) and Quantum CNN. These approaches are chosen to compare their respective performance and determine the effectiveness of quantum-inspired techniques in speech recognition tasks. By evaluating and contrasting the results obtained from both approaches, we can gain insights into the potential advantages and limitations of each method.

In addition to speech recognition, the project focuses on speech generation using Variational Autoencoders (VAEs). VAEs are powerful generative models that can learn latent representations of data and generate new samples based on these learned representations. By employing a VAE, we aim to generate speech features that can subsequently be transformed into realistic speech waveforms.

The utilization of probabilistic models and the VAE approach enables us to capture the underlying distribution of speech data, allowing for both recognition and generation tasks. This project's methodology involves training the VAE on the dataset of isolated words, learning the latent representations, and generating new speech features based on these representations.

The report aims to provide a comprehensive analysis of the performance of both the CNN and Quantum CNN approaches for speech recognition. Additionally, it delves into the capabilities of VAEs in generating speech features. By comparing the results and evaluating various metrics, such as accuracy, efficiency, and quality, we can draw conclusions regarding the strengths and weaknesses of each approach.

1.1 Quantum Introduction

To give context to our work we briefly overview Quantum Computing, this summary will not be an exhaustive explanation but it should provide enough information to understand the rest of the report.

The fundamental elements of quantum computation are the Qubits (from Quantum Bits), which are the basic unit of information as the bits are in classical systems, in the classical world the information

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

in a bit can be 0 or 1. In the quantum world, we can have multiple states at once, this phenomenon is known as superposition, and we associate a probability to the possible states.[8]

Another important property of qubits is entanglement, it is a quantum mechanical phenomenon in which two or more particles (Qubits) become correlated in a way that their properties become dependent on each other. This means that if one of the particles is observed or measured, it will affect the state of the other particles, this has important implications for quantum computing, as entangled qubits can be used to create quantum circuits that perform operations in parallel.

The building blocks of computations are the Quantum Gates, which are like logical gates but they operate on qubits instead of bits. They can manipulate the state of qubits by changing the probabilities of measuring them in different states. Some examples of quantum gates are the Hadamard gate which puts a qubit into a superposition state, and the phase gate, which introduces a relative phase shift between the 0 and 1 states of a qubit.

With quantum gates, we can create quantum circuits, where each gate performs a specific operation on one ore qubit changing its state and in the end, each qubit is measured. A notable example of quantum circuits to solve real-world problems is for example the Variational Quantum Eigensolver (VQE) algorithm [7] which is a classical-quantum circuit to calculate the ground state energy of a molecule. It is particularly useful for molecules that are too complex to be solved exactly with classical computers, therefore it provides a powerful tool for quantum chemistry research and drug discovery.

To develop programs there are several solutions, Microsoft proposed its own language called Q#[1] based on C#. Another option is OpenQASM, which is comparable to Verilog as a level of abstraction. IBM instead developed a framework called Qiskit[2], where developers can write quantum circuits in Python. A rather large area of research focuses on compilers that allow developers and programmers to write, build, and execute software for quantum computers.

1.2 Quantum Machine Learning

The advent of quantum information opened a new field: quantum machine learning (QML), combining principles from quantum physics and machine learning to develop algorithms and techniques for data analysis and pattern recognition. The two main quantum properties that bring advantages to QML are superposition and entanglement, they enable algorithms such as Quantum Support Vector Machine (QSVM)[4] and Quantum Neural Networks (QNNs)[5], they enhance feature representation, they enable quantum systems to perform parallel computations on multiple states simultaneously.

Quantum machine learning algorithms can also exploit interference effects to find optimal solutions efficiently: Quantum algorithms like Quantum Annealing and the Quantum Approximate Optimization Algorithm (QAOA)[**?**] leverage interference to search through the solution space and find the global minimum or maximum with higher probability.

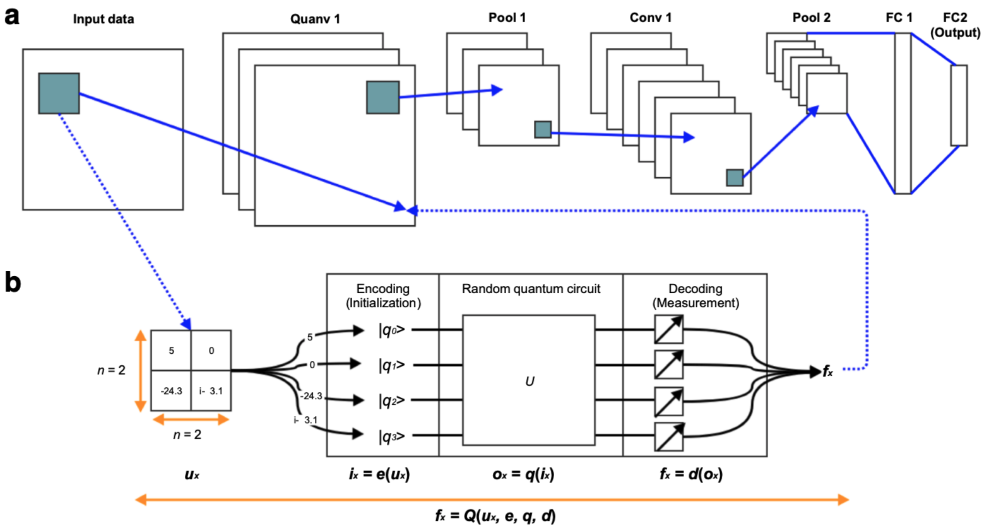
1.3 Hybrid quantum-classical Neural Networks and state of the art

In this project, we will use quantum computing together with CNNs to make a QNN,

A Convolutional Neural Network (CNN) is a deep learning model designed for processing grid-like data, such as images or spectrograms. It utilizes convolutional layers to extract local patterns and features, pooling layers for downsampling and preserving spatial relationships, and fully connected layers for capturing global dependencies. CNNs excel at analyzing structured data by automatically learning hierarchical representations and making predictions based on the learned features. They are commonly used in speech synthesis and other domains.

To focus a bit more on what we are going to do, we have to introduce QNNs, they were originally defined as Quanvolutional neural networks by Henderson et al. [5]. They introduced a new layer that they have called the quanvolutional layer it is a type of transformational layer that is made up of a group of quantum filters, in their paper they explore different numbers of filters, this layer operates like its classical counterpart making feature maps by transforming input data. The difference is how the feature maps are created, in this case using random quantum circuits. The networks they present consist of a quanvolutional layer followed by a normal CNN.

The first layer operates taking spatially local subsections of the tensor as inputs but instead of a normal matrix multiplication like CNN, they use these inputs as initial states for quantum circuits, they present the case with random quantum circuits, and the measured output of the circuit is the local result of the filter.



Quanvolutional layer in a network stack [5]

The advantages they present compared to other QML algorithms are the absence of a QRAM requirement and resiliency to consistent errors due to noise in quantum systems. Some drawbacks are not knowing how to encode and decode classical information in a quantum and since we don't need memory we could have to execute a large number of quantum circuits, another major drawback is that at the current state of hardware, we can't demonstrate a clear quantum advantage. This study however only uses QNNs for image classification. To get closer to our goal we present part of the work of Chao-Han Huck Yang et al. in Decentralizing feature extraction with quantum convolutional neural networks [9] They use QNNs (defined by them as QCNN) for feature extraction for speech recognition, they also explore decentralized feature extraction for privacy preservation but that's out of the scope of this project. The system they propose is composed of a quantum convolutional layer made up of variational quantum circuits, similar to what we have seen before, and a deep neural network. A variational quantum circuit is a circuit whose gates can be parameterized by some variables making its design very accessible and thus resistant to noise and without any requirements of error correction. In particular, they feed Mel spectrograms to the Quantum layer, and its output is given to a recurrent neural network with a self-attention encoder, which has been reported as the best model so far among deep neural networks.

A Recurrent Neural Network (RNN) with attention represents an advanced deep learning architecture that extends the capabilities of conventional RNNs by integrating an attention mechanism. Unlike the sequential processing of traditional RNNs, an RNN with attention selectively focuses on pertinent segments of the input sequence during the output generation phase. By dynamically allocating attention weights, the model effectively captures long-range dependencies and adeptly handles sequences of varying lengths. This attention mechanism substantially enhances the model's capacity to prioritize pertinent information and yield precise predictions, making it particularly advantageous in tasks involving speech recognition.

Their results show an improvement in terms of accuracy when using the quantum layer but only for a certain kernel size (the kernel size has the same meaning as in classical convolutional layers).

2 Related work

2.1 CNN for speech recognition

Several studies have explored the use of speech recognition models and their applications in various domains. In the field of speech control, the design and implementation of accurate word-tracking models have garnered significant attention. A notable contribution in this area is the work presented by Ayad Alsobhani et al. in [3], which utilized deep convolutional neural network techniques to develop a robust word-tracking model. Their study focused on six control words (start, stop, forward, backward, right, left) and incorporated speech recognition features to enhance the performance of the model.

By training and testing their proposed models on the collected dataset, they achieved an impressive word classification accuracy of 97.06% even when presented with completely unknown speech samples. A key differentiating factor of their work is the utilization of a diverse and realistic dataset, as opposed to relying on pre-existing isolated word datasets commonly used in other studies.

2.2 VAEs for speech synthesis

Variational autoencoders (VAEs) have brought about a paradigm shift in the domain of speech synthesis by enabling the generation of speech that exhibits high quality and naturalness. VAEs, constituting generative models comprising an encoder and a decoder network, facilitate the mapping of input speech features, such as MFCCs or Mel spectrograms, to a lower-dimensional latent space representation. Through the subsequent decoding process, the original features are reconstructed from the latent representation, effectively synthesizing speech.

The scholarly contribution of Lu et al. in [6] has exerted a significant impact on the field of VAE-based speech synthesis. This research endeavor introduces a pioneering approach that merges VAEs with non-autoregressive modeling for the synthesis of text-to-speech.

Of particular note is the innovative training scheme proposed in the aforementioned study, which has spurred advancements in both the efficiency and fidelity of speech synthesis. By harnessing the capabilities of VAEs in conjunction with non-autoregressive modeling, the authors succeed in enhancing synthesis performance while upholding the integrity of the resulting speech quality.

3 Methods

3.1 Data pre processing

In the data preprocessing phase, the "speech_command" dataset was utilized as the primary data source. Ten distinct groups of words were randomly selected from this dataset, with each group consisting of approximately 1,000 samples in the form of .wav files. To facilitate the processing of audio data, the librosa library was employed, leveraging its loadAudio method to import the audio waveform and the associated sampling rate for each sample. The subsequent computation of Mel-frequency cepstral coefficients (MFCC) and Mel spectrogram (MSpec) features for each sample was accomplished using the corresponding methods provided by the librosa library. To organize the processed data, an array of dictionaries was created, where each dictionary represented an individual sample and contained essential information such as the audio waveform, sampling rate, computed MFCC features, MSpec features, and the corresponding word label. This meticulous preprocessing procedure ensured the appropriate representation and readiness of the audio data for subsequent stages of the speech synthesis process, facilitating further analysis and modeling.

3.2 Speech recognition method

We designed our CNN architecture specifically for speech recognition tasks. The architecture consisted of several key components:

- Convolutional layers: We experimented with different numbers of convolutional layers to capture different levels of abstraction in the input spectrograms. The number of filters for each convolutional layer was a parameter in our grid search, allowing us to explore the impact of varying filter sizes on the model's performance.

- Activation functions: We utilized rectified linear units (ReLU) as the activation function after each convolutional layer to introduce non-linearity into the model.
- Pooling layers: To reduce the spatial dimensions and extract the most salient features, we included max pooling layers after certain convolutional layers.
- Fully connected layers: After the convolutional layers, we added fully connected layers to perform classification based on the extracted features. The number of fully connected layers and their dimensions were parameters we tuned during the grid search process.
- Regularization techniques: We employed regularization techniques such as dropout and L2 regularization to prevent overfitting and improve generalization of the model.

To find the optimal combination of hyperparameters, we performed a grid search. We systematically varied the parameters mentioned above and evaluated the model's performance on a separate validation set. The grid search allowed us to explore different combinations of hyperparameters efficiently and identify the best configuration that yielded the highest accuracy on the validation set.

During training, we employed various optimization techniques to update the model's weights and biases. The type of optimizer was one of the parameters included in the grid search. We experimented with popular optimization algorithms such as Stochastic Gradient Descent (SGD), Adam, and RMSprop, comparing their performance in terms of convergence speed and accuracy.

3.3 Quantum recognition method

We trained two different models to explore Quantum Neural Networks, the first one is an Attention Recurrent Neural Network, the second is still the same Attention Recurrent Neural Network but we added a Quanvolutional layer at the beginning.

The Attention RNN is composed of two layers of bi-directional long short-term memory and a self-attention encoder, this RNN model has been reported as the best among other Deep Neural Network solutions [9]. Compared to traditional RNN, using attention improves performances for long sequences.

We added the Quantum Layer as described in the papers explained in paragraph 1.3: it's similar to a convolutional layer but it uses variational quantum circuits instead of matrix multiplications. The quantum computing part has been executed on a simulator (Pennylane) that emulates a Qiskit device, we ran random circuits on the simulator, as described in the papers explained before.

We trained both models and compared the accuracy and the loss, the results are reported in paragraph 4.2.

3.4 Speech synthesis method

This project leverages the Variational Autoencoder (VAE) as the underlying methodology for speech synthesis. The VAE architecture comprises an encoder and a decoder network. The encoder consists of three convolutional layers followed by two linear layers, enabling the extraction of hierarchical features from the input Mel spectrograms and their transformation into a lower-dimensional latent space representation. The convolutional layers capture relevant information from the spectrograms, while the subsequent linear layers further process the extracted features and map them to the latent space. The encoder outputs two crucial parameters, namely the latent mean and the log-variance, encoding the distribution of the latent variables.

To compute the latent space representation, the VAE employs the reparameterization trick, facilitating the generation of samples from the latent variables while maintaining differentiability. This technique involves sampling from a standard Gaussian distribution and then transforming the samples using the learned mean and standard deviation parameters obtained from the encoder.

The decoder network in the VAE is responsible for reconstructing the synthesized speech from the latent space representation. It comprises two linear layers followed by three transpose convolutional layers. The linear layers take the latent space representation as input and transform it into a higher-dimensional space. The transpose convolutional layers subsequently upsample the features to reconstruct the speech spectrograms accurately. The objective of the decoder network is to generate a faithful representation of the original input features based on the information encoded in the latent space.

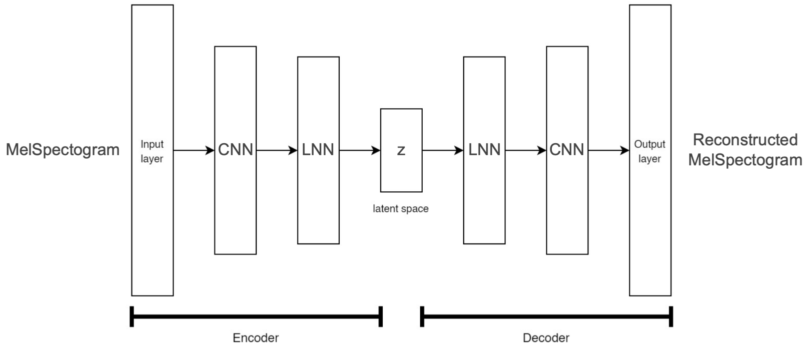


Figure 1: VAE architecture

During training, the VAE aims to minimize the reconstruction loss, typically quantified using the mean squared error (MSE) metric. This loss ensures that the synthesized speech features closely resemble the original input features, promoting accurate speech synthesis. Additionally, a regularization loss is incorporated using the Kullback-Leibler (KL) divergence. This loss encourages the latent variables to approximate a standard Gaussian distribution, promoting a regularized latent space that enables meaningful manipulation and interpolation of speech features during the synthesis process.

The training of the VAE model utilizes backpropagation and Adam optimization to iteratively update the model parameters and minimize the composite loss function, which includes both the reconstruction loss and the regularization loss. The VAE is trained on a dataset consisting of Mel spectrograms for each sample. The training process involves presenting the input features to the model, computing the reconstruction and regularization losses, and updating the model parameters accordingly. This iterative training procedure continues until the VAE achieves the desired level of performance in speech synthesis.

4 Experiments and Results

4.1 CNN for speech recognition

Table 1 presents the validation accuracies achieved by different model configurations in the speech recognition task. All the models have 3 layers and they vary in terms of the number of filters, the optimizer used, the number of nodes, the batch size, and the presence of regularization.

In terms of architecture, the best performing model is model 5, with 3 layers, 128 filters, SGD optimizer, 64 nodes, a batch size of 32, and dropout regularization. It achieved an impressive accuracy of 0.969 on the validation set. This suggests that a deeper network with a larger number of filters can capture more complex patterns in the data, leading to better performance. Additionally, the use of dropout regularization helps in reducing overfitting and improving generalization.

4.2 Quantum for speech recognition

In the following picture (Figure 2) we report the accuracy and the loss for both the Attention RNN model and the Quantum Attention RNN model. We are very satisfied with these results because, even if we didn’t manage to overcome the baseline accuracy adding a quantum layer, we still got very good results close to 0.95 of accuracy and below 0.3 of loss, we also noticed how the quantum model converged faster than the classical one.

num_filters	optimizer	num_nodes	batch_size	regularization	Accuracy
64	SGD	64	32	Dropout	0.964
64	SGD	128	32	Dropout	0.957
64	Adam	64	32	Dropout	0.966
64	Adam	128	32	Dropout	0.944
128	SGD	64	32	Dropout	0.969
128	SGD	64	32	None	0.803
128	SGD	64	32	None	0.952
128	SGD	128	32	Dropout	0.965
128	Adam	64	32	Dropout	0.942
128	Adam	64	128	Dropout	0.952
128	Adam	128	32	Dropout	0.957
128	Adam	128	64	Dropout	0.947
128	Adam	128	128	Dropout	0.958

Table 1: Grid Search results with Validation Accuracies for Different Model Configurations.

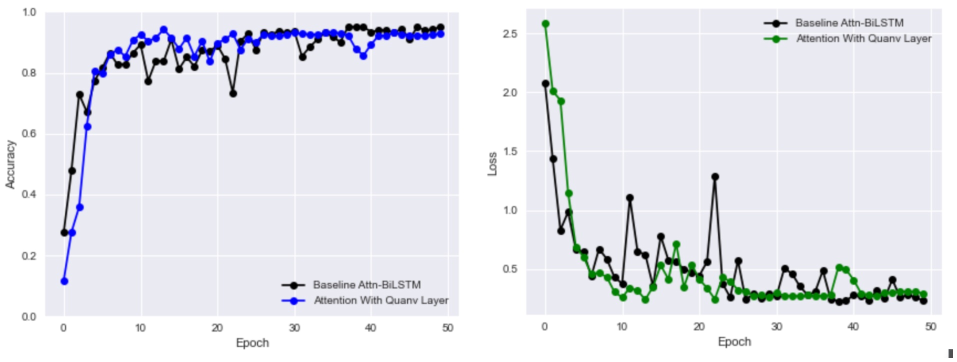


Figure 2: Accuracy and loss for Attention RNN and Quantum Attention RNN

4.3 VAE for speech synthesis

In this section, we present the experiments conducted to evaluate the effectiveness of the Variational Autoencoder (VAE) model for speech synthesis.
The model was trained for 10 epochs using an Adam optimizer with a learning rate of 0.001. The training dataset was divided into mini-batches of size 32 for efficient computation. The loss function employed a combination of reconstruction loss and KL divergence loss, balancing the fidelity of reconstructions and the regularization of the latent space.

The following results were obtained:

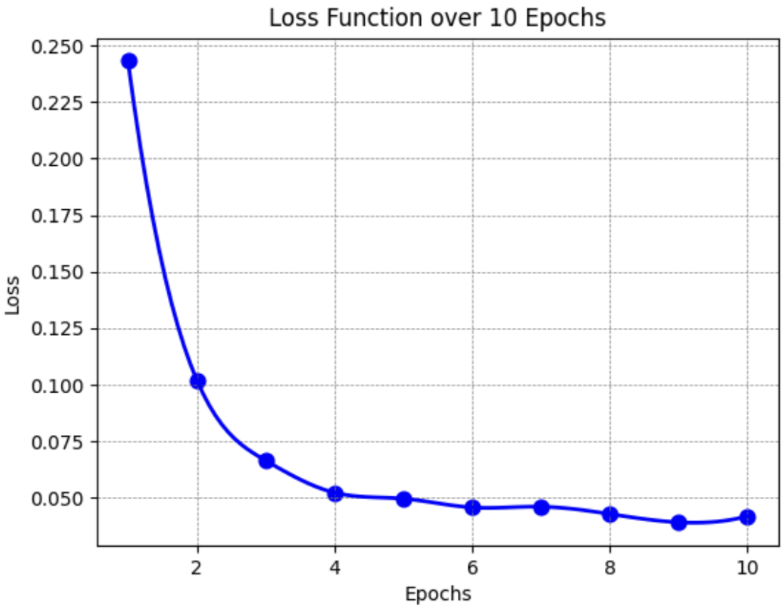


Figure 3: VAE loss trend

The trained VAE model demonstrated the potential for generalization by generating plausible reconstructions for unseen speech samples. This suggests that the model successfully learned a useful representation of speech and can synthesize speech-like outputs. The following results highlight the efficacy of the VAE model in speech synthesis tasks:

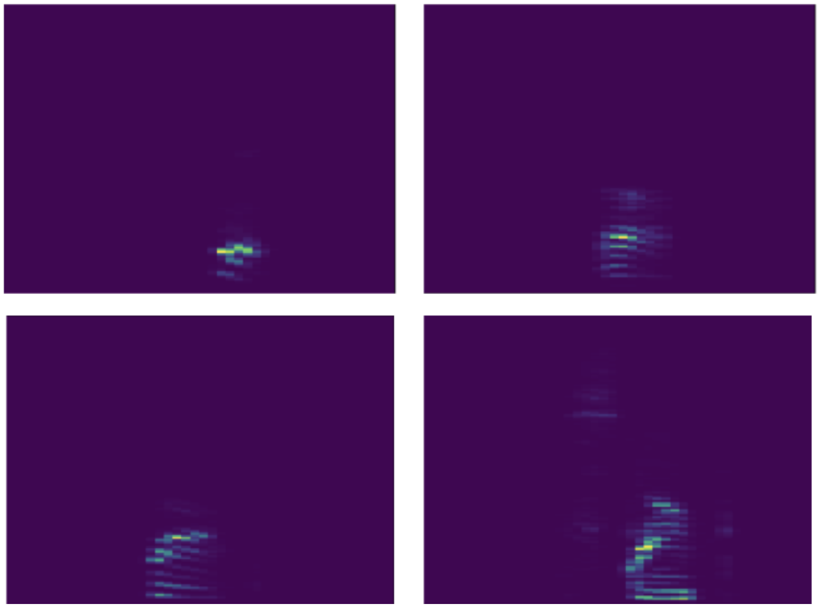


Figure 4: generated spectograms

5 Discussion and Conclusions

The experiments we did on quantum convolutional neural networks show that hybrid quantum-classical solutions should be explored more and we expect more research on this field in the coming years as quantum computers will be more reliable and accessible to researchers. We didn't get a clear quantum advantage but our results leave a door open to future developments.

Our study demonstrates the efficacy of the Variational Autoencoder (VAE) model for speech synthesis. The VAE successfully reconstructs speech spectrograms with high fidelity and learns a meaningful latent space representation. These findings highlight the potential of the VAE in generating high-quality speech-like outputs and its ability to generalize to unseen samples. Further research can focus on refining the model architecture and exploring alternative loss functions to enhance its performance. Overall, the VAE presents a promising approach to speech synthesis with implications for various applications in the field.

Additionally, compelling evidence has been presented regarding the efficacy of employing a Convolutional Neural Network (CNN) architecture for the purpose of speech recognition. The performance of our model yielded highly favorable outcomes, exhibiting an accuracy rate of 96.9%. Remarkably, these results were achieved with minimal fine-tuning, thereby suggesting the potential for further enhancements through an expanded grid search.

References

[1] The q# quantum programming language user guide, https://learn.microsoft.com/en-us/azure/quantum/user-guide/?view=qsharp-preview.

[2] Gadi Aleksandrowicz, Thomas Alexander, and Panagiotis Barkoutsos. Qiskit: An open-source framework for quantum computing, jan 2019.

[3] Ayad Alsobhani, Hanaa M A ALabboodi, and Haider Mahdi. Speech recognition using convolution deep neural networks. *Journal of Physics: Conference Series*, 1973(1):012166, aug 2021.

[4] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, mar 2019.

[5] M. Henderson, S. Shakya, and S. Pradhan. Quanvolutional neural networks: powering image recognition with quantum circuits. *Quantum Machine Intelligence*, 2020.

[6] Hui Lu, Zhiyong Wu, Xixin Wu, Xu Li, Shiyin Kang, Xunying Liu, and Helen Meng. Vaenar-tts: Variational auto-encoder based non-autoregressive text-to-speech synthesis, 2021.

[7] Alberto Peruzzo et al. A variational eigenvalue solver on a photonic quantum processor. 2014.

[8] Benjamin Schumacher. Quantum coding. *Phys. Rev. A*, 51:2738–2747, Apr 1995.

[9] Chao-Han Huck Yang, Jun Qi, Samuel Yen-Chi Chen, Pin-Yu Chen, Sabato Marco Siniscalchi, Xiaoli Ma, and Chin-Hui Lee. Decentralizing feature extraction with quantum convolutional neural network for automatic speech recognition. *CoRR*, abs/2010.13309, 2020.

6 Appendix

In response to the feedback received during the peer review process, we have made an addition to enhance the clarity and comprehensibility of the methods section. Specifically, we have included an image that illustrates the architecture of the model used in our study. This visual representation serves to provide a more intuitive understanding of the model's components and their interconnections.