big data and cognitive computing

MDPI

Article

The Development of a Kazakh Speech Recognition Model Using a Convolutional Neural Network with Fixed Character Level Filters

Nurgali Kadyrbek [1], Madina Mansurova [1,*], Adai Shomanov [2] and Gaukhar Makharova [3]

[1] Department of AI & Big Data, Faculty of Information Technologies, Al-Farabi Kazakh National University, Al-Farabi Ave., 71, Almaty 050040, Kazakhstan; kadyrbek.nurgali@kaznu.kz
[2] School of Engineering and Digital Sciences, Nazarbayev University, Kabanbai Batyr Ave., 53, Astana 010000, Kazakhstan; adai.shomanov@nu.edu.kz
[3] Department of Foreign Language, Faculty of Philology, Al-Farabi Kazakh National University, Almaty 050040, Kazakhstan; maharova.gauhar@kaznu.kz
* Correspondence: madina.mansurova@kaznu.edu.kz; Tel.: +8-(727)-221-1587

Abstract: This study is devoted to the transcription of human speech in the Kazakh language in dynamically changing conditions. It discusses key aspects related to the phonetic structure of the Kazakh language, technical considerations in collecting the transcribed audio corpus, and the use of deep neural networks for speech modeling. A high-quality decoded audio corpus was collected, containing 554 h of data, giving an idea of the frequencies of letters and syllables, as well as demographic parameters such as the gender, age, and region of residence of native speakers. The corpus contains a universal vocabulary and serves as a valuable resource for the development of modules related to speech. Machine learning experiments were conducted using the DeepSpeech2 model, which includes a sequence-to-sequence architecture with an encoder, decoder, and attention mechanism. To increase the reliability of the model, filters initialized with symbol-level embeddings were introduced to reduce the dependence on accurate positioning on object maps. The training process included simultaneous preparation of convolutional filters for spectrograms and symbolic objects. The proposed approach, using a combination of supervised and unsupervised learning methods, resulted in a 66.7% reduction in the weight of the model while maintaining relative accuracy. The evaluation on the test sample showed a 7.6% lower character error rate (CER) compared to existing models, demonstrating its most modern characteristics. The proposed architecture provides deployment on platforms with limited resources. Overall, this study presents a high-quality audio corpus, an improved speech recognition model, and promising results applicable to speech-related applications and languages beyond Kazakh.

Keywords: automatic speech recognition; deep learning; low-resource; Kazakh; speech corpus; character embedding

Citation: Kadyrbek, N.; Mansurova, M.; Shomanov, A.; Makharova, G. The Development of a Kazakh Speech Recognition Model Using a Convolutional Neural Network with Fixed Character Level Filters. *Big Data Cogn. Comput.* **2023**, *7*, 132. https://doi.org/10.3390/bdcc7030132

Academic Editors: Zuchao Li, Min Peng and Carson K. Leung

Received: 10 April 2023
Revised: 22 June 2023
Accepted: 5 July 2023
Published: 20 July 2023

1. Introduction

Automatic speech recognition (ASR) systems convert incoming audio speech into text [1] and allow further analysis of the speech content of the source. The quality of automatic transcription tools is progressively improving with the introduction of hybrid modeling based on a deep neural network (DNN) [2].

Research and the development of universal acoustic systems are becoming increasingly complex due to the lack of publicly available datasets that are relatively normally distributed by age, gender, and regional affiliation.

Kazakh is a Turkic language spoken mainly in Kazakhstan, and it is spoken by about 13 million people worldwide. To date, some major studies have been conducted in the field of recognition of Kazakh speech, but the language itself cannot afford enough digitized

resources. And one more fundamental problem lies at the level of vocabulary, that is, dialectics, as usually happens with languages that have a limited number of speakers, and the situation is aggravated by the fact that many speakers synthesize their speech using Kazakh and Russian lexicons in a mix [3].

Another fundamental problem of language is related to the study of sound systems in the language. And there are different opinions about the classification of sounds into vowels and consonants [4,5].

The main objectives of this study:

- To develop a speech recognition model for the Kazakh language based on deep learning;
- To investigate the relationship between Embedding vectors of Kazakh characters and spectrograms at the encoder level;
- To develop a relatively lightweight model that allows us to deploy it on a device with limited computing resources.

Developing a lightweight speech recognition model that can be deployed on a device with limited computing resources such as a Raspberry Pi is a challenging task but is certainly feasible. This is due to the following facts:

1. Increased accessibility: Speech recognition technology has the potential to revolutionize the way we interact with devices and computers. However, many people do not have access to high-end computing devices or high-speed Internet connections. Developing a lightweight speech recognition model that can run on a low-cost device like a Raspberry Pi means that this technology can be made more accessible to people with limited resources [6];
2. Increased privacy: Many speech recognition systems rely on cloud-based processing, which means that users' data are transmitted over the Internet and stored on remote servers. This can raise privacy concerns, particularly when dealing with sensitive information. By deploying a speech recognition model locally on a device like a Raspberry Pi, users can maintain greater control over their data and ensure that they are not being transmitted over the Internet [7];
3. Increased reliability: Cloud-based speech recognition systems require a reliable Internet connection to function. In areas with poor internet connectivity or during periods of network congestion, these systems may not work effectively. By deploying a speech recognition model locally on a device like a Raspberry Pi, users can ensure that the system will continue to function even if Internet connectivity is lost [8];
4. Increased speed: Cloud-based speech recognition systems require data to be transmitted over the Internet, which can introduce latency and slow down the system. By deploying a speech recognition model locally on a device like a Raspberry Pi, users can benefit from faster response times and a more seamless user experience [9].

There are several techniques that we can use to optimize our model and make it suitable for deployment on a low-power device like a Raspberry Pi:

1. Choose an appropriate speech recognition algorithm: There are several algorithms available for speech recognition, each with its strengths and weaknesses. For a device with limited computing resources, we will want to choose an algorithm that is efficient in terms of memory and processing power, such as hidden Markov models (HMMs) [10] or convolutional neural networks (CNNs). We could also consider using a hybrid approach that combines multiple algorithms to improve accuracy while keeping the computational requirements low;
2. Select and preprocess the dataset: The quality and size of the dataset used for training our speech recognition model can significantly impact its performance. Choose a dataset that is relevant to our use case and has sufficient diversity in terms of speaker gender, accents, and environmental noise. Preprocessing the dataset to remove noise and normalize audio levels can also help improve accuracy and reduce the computational requirements;

3. Optimize the model architecture: Once we have selected an algorithm and dataset, we can optimize the architecture of the model to make it as efficient as possible. This could involve reducing the number of layers or neurons in the network, using smaller filter sizes in CNNs, or using more compact representations of the audio signal such as MFCCs (Mel frequency cepstral coefficients) [11];
4. Train and test the model: Train the model using the preprocessed dataset and evaluate its performance using a separate testing dataset. It may take several iterations of tweaking the model architecture and parameters to achieve the desired level of accuracy while keeping the computational requirements low;
5. Deploy the model on a Raspberry Pi (device): Once we have trained and tested our model, we can deploy it on a device. We will need to ensure that the device has the necessary dependencies and libraries installed and that it has sufficient memory and processing power to run the model. We can then integrate the model with our application and test its performance in real-world scenarios [12].

By using these techniques, we can optimize our complex PyTorch model and make it suitable for deployment on devices. However, it is essential to test and evaluate the performance of our model on the devices before deploying it in a production environment.

In fact, the representation of spoken phonemes by the shape of the vocal tract is displayed as the envelope of the power spectrum of the short-term Fourier transform. The Mel filter bank is used to accurately represent this envelope because it is characterized by a set of filters that cover the frequency range that a person is able to perceive. However, manually created Mel filter bank functions are limited by their fixed set of filters, and finding suitable hyperparameters can be challenging to preserve important information for related purposes [13]. And this, in turn, led us to the hypothesis that if we create (select) a layer with a fixed filter consisting of (character) embedding vectors and separately correlate them with incoming spectrograms, it can help extract more useful information.

This study presents a significant contribution to the field of speech recognition for the Kazakh language. By addressing the challenges specific to Kazakh speech recognition and proposing an improved architecture, we provide state-of-the-art performance with reduced model complexity. The implications of this research reach beyond Kazakh, offering insights and methodologies that can enhance speech recognition for other languages and facilitate communication, transcription, and language preservation in diverse linguistic communities.

In this article, we make the following significant scientific contributions:

- *CNN Hybrid Architecture Development:* We propose a new CNN hybrid architecture that combines spectrogram-based analysis with character-level information for more complete speech recognition. This architecture allows the model to capture both the acoustic and linguistic features of the speech signal, which leads to an increase in the accuracy of transcription.
- *Integration of language filters:* We introduced an additional fixed filter into the architecture based on embedding character vectors in the Kazakh language. This language-specific layer increases the model's ability to recognize the unique phonetic characteristics of the Kazakh language, contributing to increased transcription accuracy and improved recognition capabilities.
- *Combination of supervised and unsupervised learning:* To effectively train the model, we use a combination of supervised and unsupervised learning methods. By using a labeled set of speech data to train a speech recognition model and applying unsupervised learning to an unlabeled set of data to train filters at the character level, we maximize the available data and increase the overall accuracy of the model.
- *Reducing the weight and complexity of the model:* Our proposed architecture provides comparable or superior accuracy with a significant reduction in the weight of the trained model by 66.7%. This reduction in the complexity of the model makes it suitable for deployment on platforms with limited computing resources, expanding its practical application.

- *Modern Performance:* Thanks to a thorough evaluation using a test sample, our model demonstrates modern performance in Kazakh speech recognition. The symbolic error rate (CER) achieved by our model is 7.6% lower than that of existing models, which confirms the effectiveness of our architectural solution.

2. Related Works

Despite the fact that the Kazakh language is classified as a low-resource language, many studies have been conducted on the development of Kazakh speech recognition systems that can be used for various applications, such as voice-controlled devices, transcription of speech into text, and automatic translation.

Weijing Meng et al. [14] discuss the challenges of building good speech recognition systems for low-resource languages like Kazakh and how unsupervised pre-training can be used to improve performance. The authors present a model called wav2vec-F, which uses unsupervised pre-training to learn potential speech representations from large amounts of unlabeled audio data and integrates a factorized TDNN layer to better preserve the relationship between the voice and the time step before and after quantization. The authors also use speech synthesis to enhance the performance of speech recognition. The experiments showed that wav2vec-F can effectively utilize unlabeled data from non-target languages, and multi-language pre-training is better than single-language pre-training. The proposed model achieved comparable results to previous end-to-end models with only a small amount of labeled Kazakh speech data synthesized by multi-language combined with TTS [14].

Regarding the work of researchers in their experiments, the best performing models were found to be the E2E-Transformer, followed by the E2E-RNN and then the DNN-HMM. Language-specific challenges were also discussed, including data sparsity due to the agglutinative nature of the language, code-switching between Kazakh and Russian, and data efficiency. The most challenging characters and words for the ASR system were also identified. The results demonstrate the utility of the Kazakh Speech Corpus (KSC) database for the speech recognition task [15].

The authors, led by Mamyrbayev Orken [16], achieved the following results: an experiment on building end-to-end (E2E) speech recognition systems for the Kazakh language using a combination of methods, such as CTC and encoder-decoder based on the attention mechanism. The results showed that the constructed model works well with the use of language models for Kazakh and surpassed models based on DNN-HMM and CTC models. However, there are some limitations, such as the need for a large amount of training data, the system's delay, and the fact that the model is not adapted for real-time speech recognition. The paper also suggests using the transformer network with multiple heads to improve the system's accuracy with limited data. Further research is planned on combining other models of E2E systems, as hybrid E2E models show better results than using them separately.

In the field of Kazakh speech recognition, Mamyrbayev et al. [17] investigated the implementation of an end-to-end model based on RNN-T. They focused on streaming speech recognition, in which the audio stream is directly converted to text in real time. The study compared their RNN-T-based model with other approaches, in particular with the CTS model commonly used for the recognition of Kazakh speech. The results showed that the RNN model worked well without additional components, such as the language model, and achieved the best result in their dataset. It is noteworthy that the system achieved a 10.6% character error rate (CER), surpassing other end-to-end Kazakh speech recognition systems.

Musakhodzhaeva et al. (2022) [18] expanded the Kazakh text-to-speech synthesis corpus (KazakhTTS), known as KazakhTTS2, to solve the problems of creating high-quality TTS systems for the Kazakh language. The volume of the corpus has increased from 93 h to 271 h, and additional speakers and diverse coverage of topics have appeared. The authors highlight the linguistic problems of the Kazakh language and the agglutinative language

of the Turkic family and provide detailed information about the corpus creation process, training, and evaluation procedures. The corpus proved to be sufficient to build reliable TTS models, obtaining a subjective average score above 3.6 for all five native speakers. The availability of the corpus and related resources on GitHub contributes to speech and language research of Kazakh and other low-resource Turkic languages [18].

However, there is still a need for further research and development in this area, especially to improve the accuracy and performance of Kazakh speech recognition systems. In addition, there is a need for more data and resources to support research in this area, such as large datasets of transcribed Kazakh speech and the development of standardized evaluation criteria. And another aspect that needs to be expanded in the current study is in terms of how these models will behave depending on the regional dialect of the speaker.

In the course of research, we had a hypothesis that exploring a separate layer with fixed weights to extract information from incoming spectrograms would accumulate more information. It may be beneficial to use a fixed embedding layer that maps the input data to the same representation used by the GRU.

There are several advantages to using a fixed embedding layer:

1. Faster training: The fixed embedding layer can be used to preprocess the input data before feeding it into the GRU. This can speed up training by reducing the number of parameters that need to be learned during training;
2. Better generalization: Pre-trained embeddings can help the model generalize better to new data, as they capture meaningful patterns in the input data;
3. Reduced overfitting: Pre-trained embeddings can help reduce overfitting by providing a regularization effect.

However, there are also some potential drawbacks to using a fixed embedding layer:

1. Limited flexibility: If the pre-trained embedding layer does not capture all of the relevant patterns in the input data, the model may not be able to learn them during training;
2. Task-specificity: The pre-trained embedding layer may be optimized for a specific task and may not generalize well to other tasks.

Overall, using a fixed embedding layer with a pre-trained GRU can be a useful technique in some situations. However, it is important to consider the trade-offs and experiment with different approaches to determine what works best for a specific task.

Available Datasets

There are several existing datasets for Kazakh speech recognition that researchers can use to train and evaluate their models. Here are a few examples:

1. Kazakh Speech Corpus (KSC): This is a large corpus of Kazakh speech data that was developed by researchers at Nazarbayev University. It contains around 330 h of speech data from 1000 speakers, covering a wide range of topics and dialects [15].
2. Dataset of the Institute of Information and Computing Technologies.

The researchers used a sound-insulated cabin to record the audio data of 200 speakers reading a prepared text of 100 sentences each. The recorded data were saved in separate .wav files with specific identifiers [19].

Researchers can use these datasets to train and evaluate their Kazakh speech recognition models. However, it is worth noting that these datasets may not be large enough to train deep learning models from scratch, so researchers may need to use transfer learning or unsupervised pre-training techniques to achieve good performance on these low-resource datasets.

3. Data Collection and Cleaning

3.1. Text Collection

Text data in Kazakh were extracted from various sources, including electronic literary books, news articles on websites, and scientific dissertations. This approach provided a

wide range of topics and vocabulary variability, covering vocabulary used in various subject areas. For example, news texts reflected vocabulary related to various aspects of society, while interviews with people from different walks of life demonstrated dialect variations and the occasional inclusion of words in Russian in a sentence in Kazakh. In order to exclude any inappropriate content related to sensitive political issues, user privacy, or violence, careful manual filtering was carried out. Although, some sentences may contain a couple Russian words due to the presence of borrowings in the Kazakh language. The use of mixed vocabulary from both languages is common among native speakers of the Kazakh language [20]. It is important to note that the volume of text used for teaching word embedding models significantly exceeded the volume of text used for audio recording purposes.

In our corpus, we presented all texts using the Cyrillic alphabet consisting of 42 letters, except that we replaced the letter "ё" with "е" since the practical corpus was not found by weight and is extremely rare for the Kazakh language. The distribution of these letters is shown in Figure 1 and the similarity of the results with the distribution in the KSC of Yerbolat et al. can be seen [15]; the similarity is visible to the naked eye, and this allows us to talk about the relative preservation of the frequency of letters at the language level.

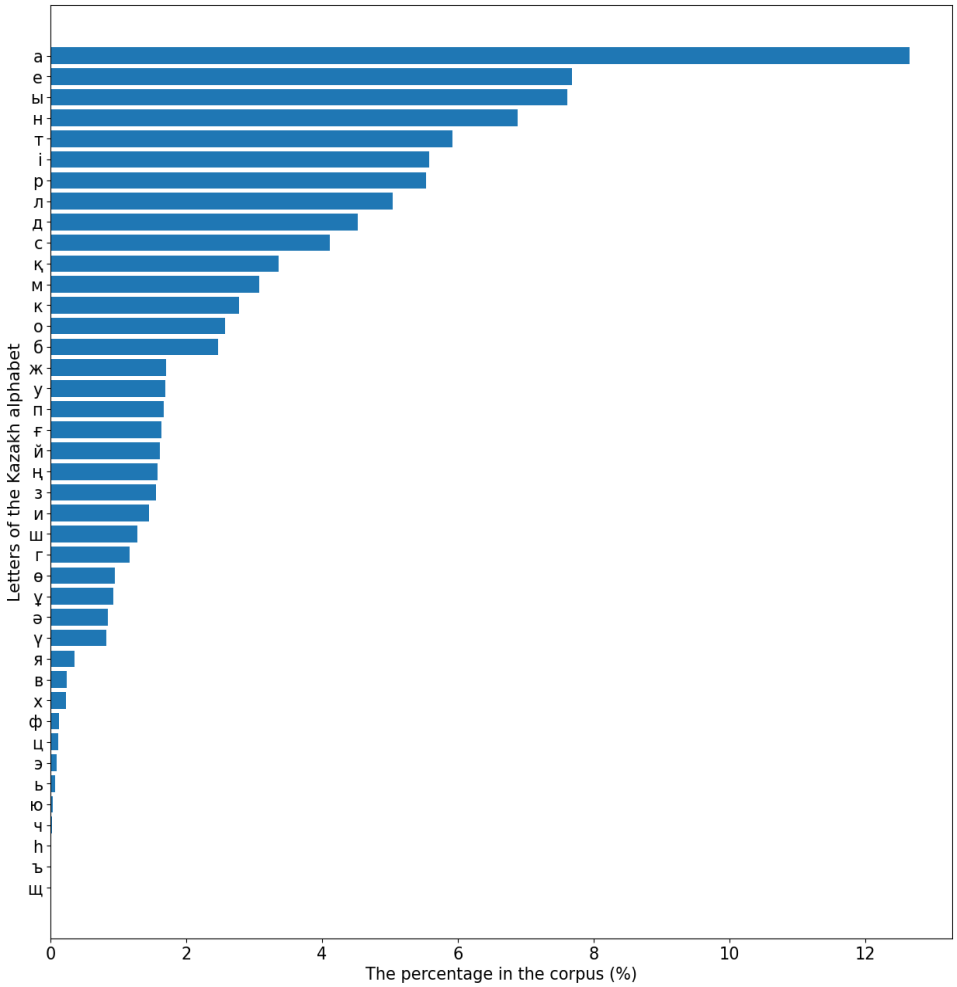

Figure 1. The distribution of letters in our corpus.

Here's the map list showcasing the corresponding letters of the Kazakh Cyrillic alphabet and their Latin equivalents:

Aa-Aa, Әә-Áá, Бб-Bb, Вв-Vv, Гг-Gg, Ғғ-Ğğ, Дд-Dd, Ее-Ee, Ёё-Yo yo, Жж-Jj, Зз-Zz, Ии-Ïï, Йй-Yy, Кк-Kk, Ққ-Qq, Лл-Ll, Мм-Mm, Нн-Nn, Ңң-Ŋŋ, Оо-Oo, Өө-Öö, Пп-Pp,

Pp-Rr, Cc-Ss, Тт-Tt, Уу-Úú, Ұұ-Uu, Үү-Üü, фф-Ff, Хх-Hh, Һһ-Hh, Цц-Ts ts, Чч-Ch ch, Шш-Sh sh, Щщ-Shch shch, Ъъ-', Ыы-Yy, Іі-Ii, Ьь-', Ээ-Ee, Юю-Yu yu, Яя-Ya ya.

In fact, the process of transition of the Kazakh alphabet from Cyrillic to Latin was still going on, but it was not yet completed.

It is also worth noting that each utterance is not always a continuous sentence (a complete context). Because when preparing the text, too long sentences were cut off, and if the sentences were too short, they were combined. And when evaluating the length of a sentence, we proceeded from the number of characters in it. The acceptable utterance length ranges from 60 to 200 letters (Figure 2). The graph shows the percentage of a sentence(utterance) with a certain number of letters in the dataset. On the graph, it may seem that the percentage of sentences with a length of 200 characters suddenly increased; this is due to the fact that all sentences of longer length were simply cut off and fell into this "category".

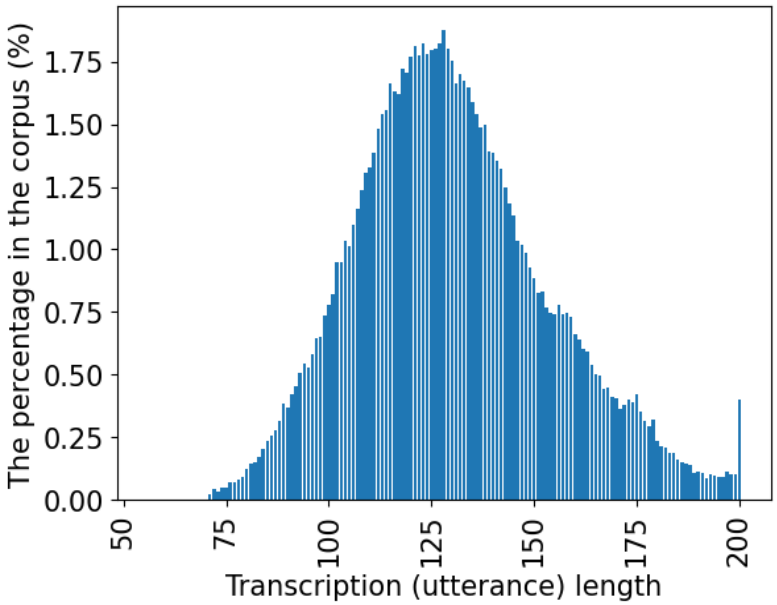

Figure 2. Distribution of utterance lengths in our corpus.

3.2. Audio Collection

Several native Kazakh speakers were involved to check the quality of the recordings. They were provided with an audio fragment and a corresponding sentence, which the speaker read. The task was to check whether the reader had read the sentences according to the instructions and to correct any deviations or other acoustic events based on a set of transcription instructions. As an additional measure of quality, we engaged a linguist, with whom we talked, observing the quality control process and selectively checking the tasks they performed. To coordinate the transcriptions, the linguist also conducted "error analysis" sessions with native speakers who helped with the validation. Table 1 below shows the main characteristics of the corpus and the division into training and test samples.

Table 1. The KSC database specifications.

Category	Train	Test	Total
Duration (hours)	470.9	83.1	554
Utterances Words (tokens)	3,109,970	571,237	3,681,207
Unique Words (tokens)	219,462	95,046	220,618
Speakers	-	-	873
Males (%)	61.17	59.42	61.18
Females (%)	38.82	40.58	38.82

It should be noted that when dividing the dataset, this was based not on the speaker but on each audio file. Thus, audio files recorded by one speaker can be included both in the training and in the test dataset.

Based on the distribution of speakers by the regions of Kazakhstan (Figure 3), most of them come from the southern region of the republic because this region is the most densely populated.

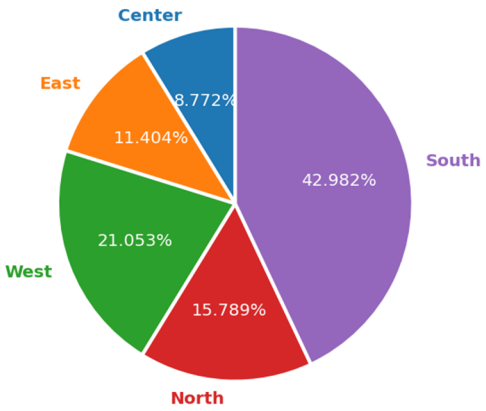

Figure 3. Distribution of speakers by the regions of Kazakhstan.

Also, the age distribution shows that most of the speakers were young people since most of the volunteers were students (Figure 4).

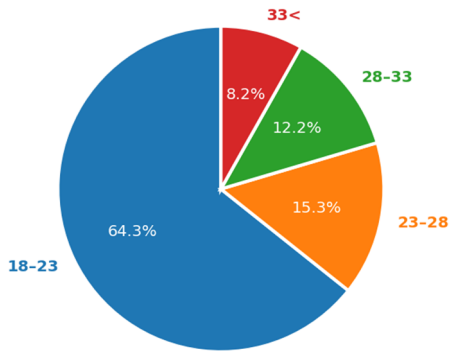

Figure 4. Distribution of speakers by age.

A transcribed database consisting of audio signals in the Kazakh language was accumulated. In further steps, this database was used to train a model of speech recognition in the Kazakh language.:

- Total number of speakers: 873;
- Each speaker was given: 250 sentences(utterances);

- Technical characteristics: .wav format, 16 kHz or 22 kHz, 16 bit, and Mono.

In total, about 218,000 utterances were obtained, which gave 554 h of transcribed speech data. And all the audio files were recorded using mobile devices (Android and iOS), in various dynamically changing environments (university, home, and office), which increases the attractiveness of our dataset. The sampling rate for Android devices is 16 kHz, and for iOS devices, it can be 22 kHz or 44 kHz, and this is due to the complexity of existing mobile audio recording applications for the operating system. In the case of Android, no problems arose thanks to the mobile application Splend Apps [21]. During the data collection process, due to technical problems, we had to use two different iOS apps; because of this, there are two different sampling rates in the dataset.

The whole process of creating and verifying the database took about 9 months, and the database size is about 57 GB. The collected data have been posted on OpenSLR and can be found through the link [22]. OpenSLR is a website dedicated to hosting speech and language resources, such as speech recognition tutorials.

The short-time Fourier transform function from the Librosa Python package was used during the data loading process to preprocess audio signals by transforming them into a spectrogram representation. This approach has proven to be effective in extracting relevant features from raw audio signals, resulting in improved performance for various machine learning tasks [23]. Typically, convolutional layers in an encoder can be used to extract spectrogram-like objects from raw audio signals. Filters in convolutional layers are trained to recognize patterns and features at various time and frequency scales that can be used to represent the basic structure of the audio signal, so no additional signal processing tools were used during this research.

3.3. Character Embedding in the Corpus

One of the ways to improve the performance of the speech recognition model is to include additional functions that capture information that goes beyond what is presented on the spectrogram. One of these functions is character-level information that can be extracted using CNN filters prepared specifically for this purpose. To incorporate character-level information into a speech recognition model, one approach is to use a hybrid CNN architecture that combines filters for both spectrogram characteristics and character characteristics. This architecture would take a spectrogram as input, and then apply convolutional filters to extract both spectrogram objects and symbolic objects. These elements can then be combined and passed through one or more fully connected layers to produce the final result.

In our approach, we generated a special vector for each character using the Wod2Vec algorithm [24]. Using Word2Vec for character embedding can be an interesting approach, but it has some potential drawbacks.

Word2Vec is a popular technique used to generate word embeddings, which are vector representations of words that capture their meaning and relationships with other words in a corpus. However, Word2Vec is typically used with words rather than characters, and using it to generate character embeddings would require some modifications.

One option would be to treat each character as a "word" and train a Word2Vec model on a corpus of characters. However, this approach has some limitations. First, the vocabulary of characters can be very large, which can make training the model computationally expensive. Second, character embeddings generated using this method may not capture some of the nuances of character-level information, such as prefixes or suffixes.

We used the method of PCA to reduce the dimension of the vectors to two-dimensional in order to compactly visualize them (Figure 5). It can be noted by the example of the letters of the area highlighted by the circle that their vectors were generated appropriately since these letters are combined with each other in accordance with the laws of synharmonism [5].

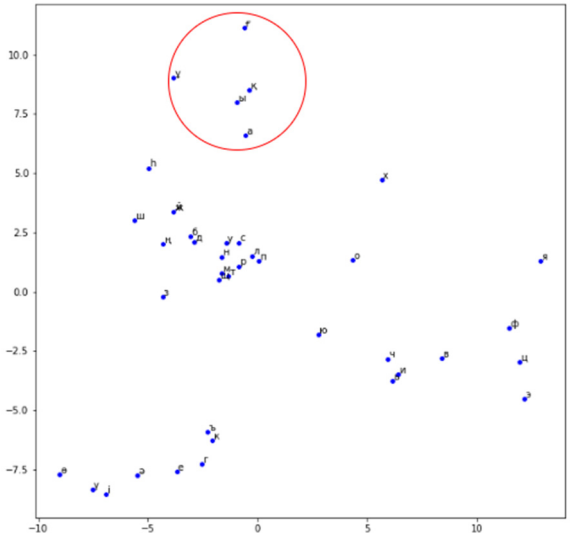

Figure 5. Two-dimensional character embeddings visualization.

Another option would be to use pre-trained Word2Vec embeddings for words, and then use these embeddings to generate character embeddings. This approach can be computationally efficient and can leverage the rich semantic information captured by Word2Vec embeddings. However, it may not capture some of the specific character-level information that is relevant to the task at hand.

3.4. Constructing a Filter from Embedding Vectors for a Convolutional Layer

In the Kazakh language, we use 15 letters to represent vowel sounds and 25 for consonant sounds [5]. But for the optimality of the experiments, the letter "ё" was excluded from the list of vowels and the letter "у" was also duplicated in the list of consonants, since it can be both a vowel and a consonant, depending on the context of the sounds and extended with separating letters "ъ" and "ь", although they are found only inside loanwords. So at the end, we obtained vowels as V = [а, ә, о, ө, ы, і, у, ұ, ү, э, и, ю, я, and е] and consonants as C = ['й', 'ц', 'к', 'н', 'г', 'ш', 'щ', 'з', 'х', 'ъ', 'ф', 'в', 'п', 'р', 'л', 'д', 'ж', 'ч', 'с', 'м', 'т', 'ь', 'б', 'ң', 'ғ', 'қ', 'һ', and 'у'].

To generate a filter, we first need to build a matrix of adjacency of vowels and consonants in one syllable with the dimensions 28 × 14 (Figure 6).

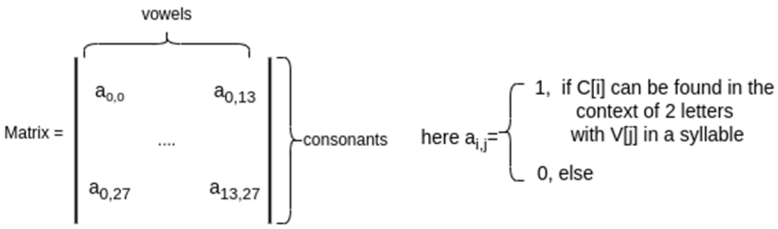

Figure 6. The construction of the adjacency matrix.

Once we initialize the Matrix, we can create the filter by adding two embedding vectors of the characters C[i] and V[j] in case a[i,j] is 1; otherwise, we initialize it with a zero vector and generate the array with the dimensions 32 × 1 × 28 × 14.

Overall, using Word2Vec for character embeddings can be an interesting approach, but it is important to carefully consider the potential benefits and drawbacks for a given task and dataset. Other methods, such as training character embeddings directly using a neural network, may be more appropriate in some cases. And here there is a place for other transformers like BERT et al.

3.5. Model

We used the standard DS2 model [25] with architectural and methodological modifications. DeepSpeech2 uses a deep neural network architecture consisting of an acoustic model and a language model. The acoustic model is responsible for converting input audio signals into a sequence of phonetic representations or symbol-level representations. It usually consists of several levels of bidirectional RNNs to fix time dependencies in the input characteristics of the spectrogram. The input data for the DeepSpeech2 model are a spectrogram representing an audio signal. The audio signal is divided into small time windows, and a spectrogram is calculated for each window using methods such as short-term Fourier transform (STFT). The spectrogram provides a representation of the audio signal in the frequency domain, which is then used as input data for the acoustic model. Convolutional layers: The encoder of this model has convolutional layers which are effective at extracting features from the raw audio signals. These layers train to identify patterns in the speech signal that are important for speech recognition, such as phonemes, syllables, and words. Convolutional neural networks tend to increase the density of information per neuron from the lower layers to the higher ones. The change concerns the convolutional subsampling layer, for this, we use two different convolutional networks with the same dimensions; only with the difference of the activation function and a fixed filter in one of them. And these fixed filters were initiated using character embedding vectors as described in the section above.

Therefore, it is useful to reduce the core size when reaching deeper layers. Thus, we gradually reduced the size of the cores, and in the first layer, the size is 28×14, which is due to the number of consonants and vowels in the Kazakh language (Figure 7).

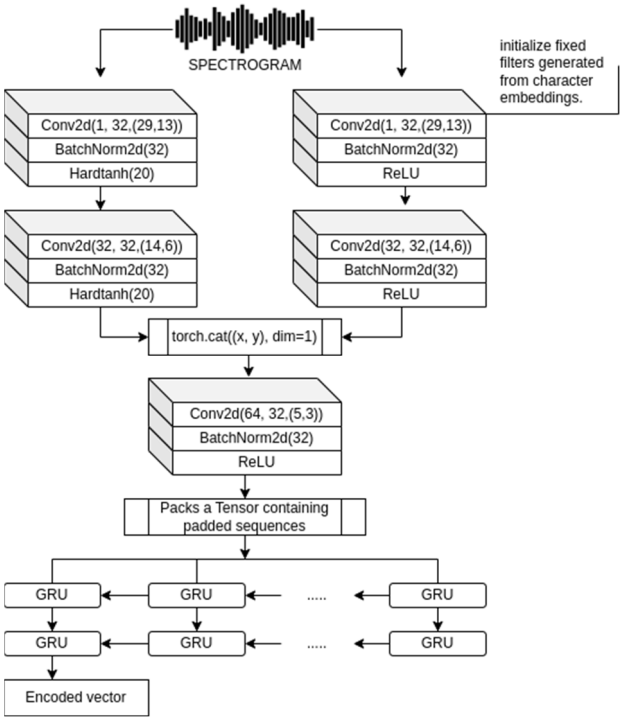

Figure 7. Encoder.

The LSTM in the encoder of this model is a type of neural network that is good at processing sequential data. In speech recognition, RNNs can learn to capture the temporal dependencies between phonemes and words, which is important for accurately transcribing speech. The decoder of this model has an attention mechanism that helps it focus on the most relevant parts of the input sequence. This is important because, in speech recognition, different parts of the audio signal may be more informative for different parts of the

transcription. Dropout is a regularization technique that is used to prevent overfitting. It randomly drops out some of the neurons during training, which forces the model to learn more robust features. This can help prevent the model from memorizing the training data and can improve its ability to generalize to new data.

The combination of convolutional layers, RNNs, attention, and dropout make the Seq2Seq model a powerful and effective approach for speech recognition.

As we know, if Conv2D filters are untrainable, it means that the weights of these filters are fixed and will not be updated during training. In this case, the input distribution to the Conv2D layer will remain constant throughout training, and there may not be a significant benefit to using BatchNorm. However, we have other layers in our model that are trainable, such as RNN layers and other convolutional layers with trainable filters; using BatchNorm after the untrainable Conv2D layer could still be beneficial. This is because the input distribution to these layers will change during training, and BatchNorm can help to normalize the input and improve performance.

Also in the decoder, after GRU layers in the embedding layer, we initiated weights with actual embedding vectors of the Kazakh characters (Figure 8).

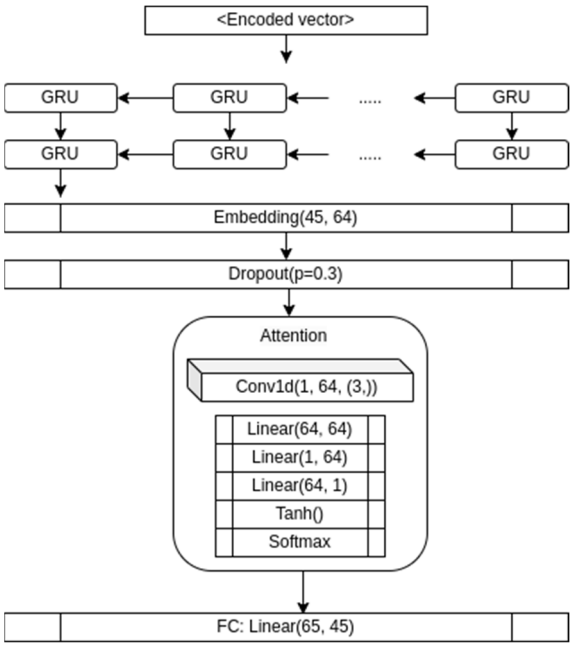

Figure 8. Decoder.

It is worth noting that the advantage of using fixed filters in subsampling layers is that they can effectively reduce the spatial dimensionality of the input, which helps to reduce the computational complexity of the model and can prevent overfitting. Additionally, using fixed filters means that the model has fewer parameters to learn, which can help to improve the model's generalization performance.

However, there are some disadvantages to using fixed filters in subsampling layers. For example, fixed filters may not be able to capture all of the relevant features in the input, particularly if the input contains complex patterns or textures. Additionally, fixed filters may not be able to adapt to different input distributions, which can limit the model's ability to generalize to new data.

We also tried using character embedding vectors at the embedding layer in the decoder, and when initializing and remaining unchanged, the CER was 6% worse than when randomly initializing weights, but when we made these weights unfixed, the CER accelerated its downward movement.

3.6. Comparison Results on the Datasets

All experiments were conducted in 20 epochs. We first used a standard DeepSpeech2 model in our dataset. The model has an encoder and decoder with a three-layer RNN 164 hidden size. The training process can be observed in Figure 9.

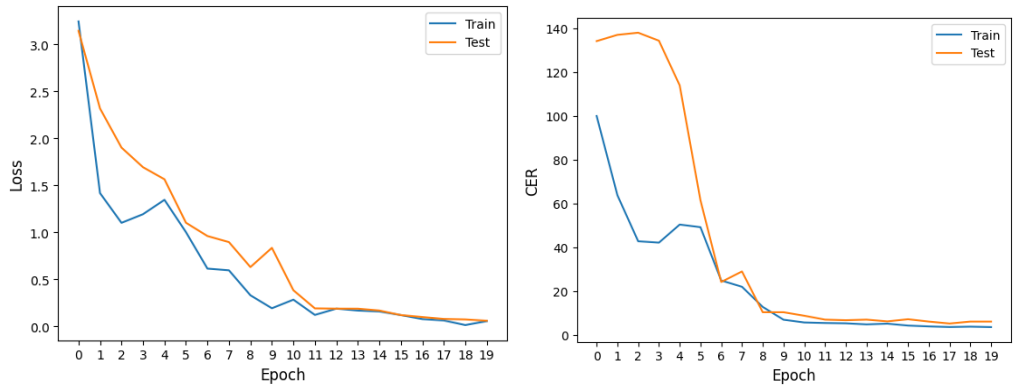

Figure 9. Original DeepSpeech2 model.

To concretize the comparisons, we selected a model with a simplified architecture: an encoder and decoder with a two-layer RNN 96 hidden size. The training process can be observed in Figure 10.

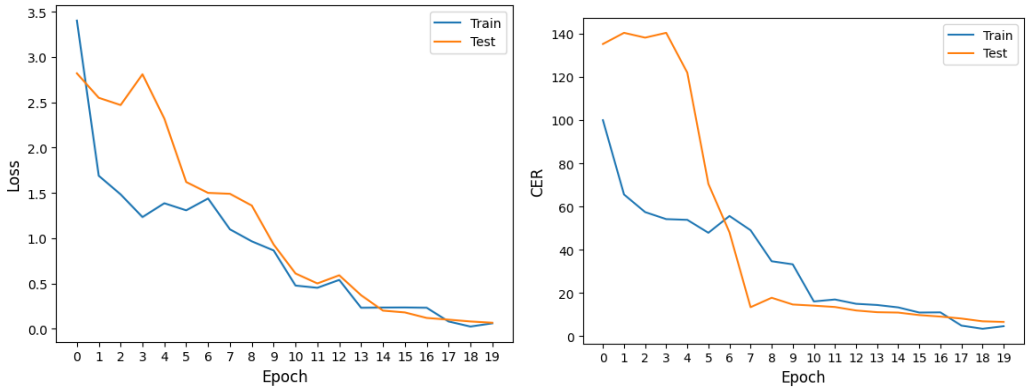

Figure 10. Simplified model.

And as a result, the model we chose showed a good result, comparable with other experiments, despite the fact that the model has a more simplified architecture and fewer trainable parameters. The training process can be observed in Figure 11.

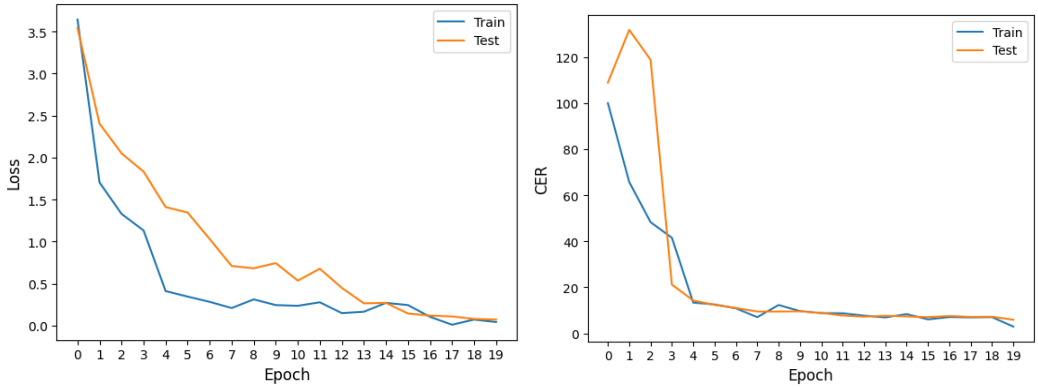

Figure 11. Simplified model with fixed filters.

Table 2 presents the results of the experiments conducted using different architectures. In this comparison, we assessed the models based on their Character Error Rate (CER), Loss, Number of Parameters, and Trained Model Physical Size.

Table 2. The results of the experiments.

Model	Number of Parameters	LOSS		CER		Trained Model Physical Size (kB)
		Train	Test	Train	Test	
DeepSpeech2	5,979,558	0.0619	0.062	3.8314	6.144	52,346
Simplified model without fixed filters	2,645,870	0.066	0.07	6.35	6.63	21,428
Simplified model with fixed filters	905,550	0.065	0.072	5.48	6.61	17,428

The *simplified model with fixed filters*, which includes both a spectrogram and symbol-level information using a hybrid CNN architecture, has an advantage for several reasons.

Firstly, the use of both a spectrogram and information at the symbol level allows for a more comprehensive analysis of audio data, which leads to an increase in the accuracy of speech recognition. Using convolutional filters to extract both spectrogram objects and symbolic objects, the model is able to capture both the acoustic and linguistic features of the speech signal, which leads to more accurate transcription.

Secondly, an additional filter, which was installed in accordance with the embedding vectors of the symbols of the Kazakh language, helps the model to better recognize the specific phonetic characteristics of the language. This filter acts as a kind of language layer that increases the ability of the model to recognize the unique features of the Kazakh language.

Finally, using a combination of supervised and unsupervised learning methods allows us to train the model more effectively. By using a labeled speech set to train a speech recognition model and an unlabeled dataset to train character-level filters, the model can learn from more data, which increases its overall accuracy.

These factors contribute to the superiority of the proposed methodology over the alternative architecture without these changes or it has the same accuracy with fewer parameters compared to larger neural networks. The approach resulted in a reduction in the model's weight by 66.7% while maintaining its relative accuracy. The performance of the model was evaluated using a test sample, which revealed a CER that was 7.6% lower than that of existing models. These results demonstrate state-of-the-art performance and validate the effectiveness of the proposed architectural solution, which enables deployment of the model on platforms with limited computing resources.

4. Conclusions

In the course of the study, we collected a transcribed audio corpus in the Kazakh language, with an overall duration of about 554 h. And the corpus is distinguished by its quality and validation and is suitable for the development of various modules related to speech tasks. The current audio corpus was used to develop a speech recognition model in the Kazakh language in order to enable deployment on devices with limited computing resources.

A model, DeepSpeech2, was chosen as the base module, and we applied architectural changes. According to our approach, incoming spectrograms pass through an additional filter (a CNN with a fixed filter). To incorporate character-level information into the speech recognition model, we used a hybrid CNN architecture that combines filters for both spectrogram characteristics and character-level information. This architecture takes a spectrogram as input and then applies convolutional filters to extract both spectrogram objects and symbolic objects. The additional filter was set in accordance with the embedding

vectors of symbols of the Kazakh language. This language-specific layer improved the recognition capabilities of the model, especially considering the subtleties of Kazakh phonetics. We used a combination of supervised and unsupervised learning methods, such as using a labeled speech set to train a speech recognition model and using unsupervised learning on an unmarked set of texts to train filters at the character level. And this approach reduced the weight of the trained model by 66.7 percent while maintaining the relative accuracy of the model (in the test sample, the CER is lower by 7.6 percent). We have shown that our approach provides state-of-the-art performance on the dataset significantly superior to existing models. The results demonstrate the effectiveness of the architectural solution proposed by us and provide the possibility of deploying the model on platforms with limited computing resources. And this approach can be applied to other languages as well.

Our approach, which combines spectrogram- and symbol-level information, as well as symbol-level filters, has demonstrated the most up-to-date performance in our dataset. Moving forward, future research may focus on the following steps:

1. *Real-time systems and embedded systems:* Expanding the deployment capabilities of our model, it is extremely important to optimize the architecture for real-time data processing and implement it on platforms with limited resources. Such optimization would allow for smooth integration with applications such as voice assistants, mobile devices, and Internet of Things (IoT) devices.
2. *Low-resource scenarios:* Expanding the applicability of the model developed by us to low-resource scenarios is a valuable direction for further research. To make effective use of limited annotated data and to prepare accurate speech recognition models, methods such as unsupervised or partially supervised learning, active learning, and multi-frame learning should be explored.

Author Contributions: Conceptualization, G.M.; methodology, N.K.; software, N.K.; validation, G.M.; formal analysis, N.K.; investigation, M.M. and A.S.; resources, N.K.; data curation, N.K.; writing—original draft preparation, N.K.; writing—review and editing, G.M.; visualization, G.M.; supervision, M.M.; project administration, A.S.; funding acquisition, M.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Committee of Science of the Republic of Kazakhstan AR09261344 "Development of methods for automatic extraction of geospatial objects from heterogeneous sources for information support of geographic information systems" (2021–2023).

Data Availability Statement: The data supporting the reported results in this study have been made publicly available on OpenSLR, a platform dedicated to hosting speech and language resources. The Kazakh Speech Dataset (KSD) can be accessed online at: http://www.openslr.org/140/ (accessed on 9 May 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Malik, M.; Malik, M.K.; Mehmood, K.; Makhdoom, I. Automatic speech recognition: A survey. *Multimed. Tools Appl.* **2021**, *80*, 9411–9457. [CrossRef]
2. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.-R.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97. [CrossRef]
3. Ryssaldy, K. Problems of the Kazakh Language as the State Language in Modern Kazakhstan. In *Kazakh in Post-Soviet Kazakhstan*; Harrassowitz Verlag: Wiesbaden, Germany, 2015; pp. 27–34.
4. Badanbekkyzy, Z.; Yeshimbetova, Z. Inventory of Phonemes in Kazakh Language. *Int. J. Res. Humanit. Arts Lit. (IMPACT:IJRHAL)* **2014**, *2*, 95–102.
5. McCollum, A.G.; Chen, S. Kazakh. *J. Int. Phon. Assoc.* **2020**, *51*, 276–298. [CrossRef]
6. Abdullah, H.; Warren, K.; Bindschaedler, V.; Papernot, N.; Traynor, P. SoK: The Faults in Our ASRs: An Overview of Attacks against Automatic Speech Recognition and Speaker Identification Systems. In *2021 IEEE Symposium on Security and Privacy (SP)*; IEEE: Piscataway, NJ, USA, 2021.

7. Wang, J.; Pan, C.; Jin, H.; Singh, V.; Jain, Y.; Hong, J.I.; Majidi, C.; Kumar, S. RFID Tattoo. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2019**, *3*, 155. [CrossRef]
8. Gondi, S.; Pratap, V. Performance Evaluation of Offline Speech Recognition on Edge Devices. *Electronics* **2021**, *10*, 2697. [CrossRef]
9. Oh, Y.R.; Park, K.; Park, J.G. Fast Offline Transformer-based End-to-end Automatic Speech Recognition for Real-world Applications. *ETRI J.* **2021**, *44*, 476–490. [CrossRef]
10. Gales, M.; Young, S. The Application of Hidden Markov Models in Speech Recognition. *Found. Trends®Signal Process.* **2007**, *1*, 195–304. [CrossRef]
11. Mahmood, A.; Utku, K. Speech recognition based on convolutional neural networks and MFCC algorithm. *Adv. Artif. Intell. Res.* **2021**, *1*, 6–12.
12. Gondi, S.; Pratap, V. Performance and Efficiency Evaluation of ASR Inference on the Edge. *Sustainability* **2021**, *13*, 12392. [CrossRef]
13. Wongpatikaseree, K.; Singkul, S.; Hnoohom, N.; Yuenyong, S. Real-Time End-to-End Speech Emotion Recognition with Cross-Domain Adaptation. *Big Data Cogn. Comput.* **2022**, *6*, 79. [CrossRef]
14. Meng, W.; Yolwas, N. A Study of Speech Recognition for Kazakh Based on Unsupervised Pre-Training. *Sensors* **2023**, *23*, 870. [CrossRef] [PubMed]
15. Khassanov, Y.; Mussakhojayeva, S.; Mirzakhmetov, A.; Adiyev, A.; Nurpeiissov, M.; Varol, H.A. A Crowdsourced Open-Source Kazakh Speech Corpus and Initial Speech Recognition Baseline. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021.
16. Mamyrbayev, O.Z.; Oralbekova, D.O.; Alimhan, K.; Nuranbayeva, B.M. Hybrid End-to-End Model for Kazakh Speech Recognition. *Int. J. Speech Technol.* **2022**, *10*, 6. [CrossRef]
17. Mamyrbayev, O.; Oralbekova, D.; Kydyrbekova, A.; Turdalykyzy, T.; Bekarystankyzy, A. End-to-end model based on RNN-T for Kazakh speech recognition. In Proceedings of the 2021 3rd International Conference on Computer Communication and the Internet (ICCCI), Nagoya, Japan, 25–27 June 2021.
18. Mussakhojayeva, S.; Khassanov, Y.; Varol, H.A. KazakhTTS2: Extending the Open-Source Kazakh TTS Corpus With More Data, Speakers, and Topics. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, Marseille, France, 6–10 June 2022; European Language Resources Association: Marseille, France, 2022; pp. 5404–5411.
19. Mamyrbayev, O.; Turdalyuly, M.; Mekebayev, N.; Alimhan, K.; Kydyrbekova, A.; Turdalykyzy, T. Automatic recognition of kazakh speech using deep neural networks. In Proceedings of the 11th Asian Conference on Intelligent Information and Database Systems (ACIIDS), Yogyakarta, Indonesia, 8–11 April 2019; pp. 465–474. [CrossRef]
20. Khomitsevich, O.; Mendelev, V.; Tomashenko, N.; Rybin, S.; Medennikov, I.; Kudubayeva, S. A Bilingual Kazakh-Russian System for Automatic Speech Recognition and Synthesis. In *Speech and Computer*; Springer International Publishing: Cham, Switzerland, 2015; pp. 25–33. [CrossRef]
21. Splend Apps. Voice Recorder Pro. Apps on Google Play. Available online: https://play.google.com/store/apps/details?id=com.splendapps.voicerec&pli=1 (accessed on 30 March 2023).
22. Kazakh Speech Dataset (KSD). Available online: http://www.openslr.org/140/ (accessed on 9 May 2023).
23. Lee, S.; Yu, H.; Yang, H.; Song, I.; Choi, J.; Yang, J.; Lim, G.; Kim, K.-S.; Choi, B.; Kwon, J. A Study on Deep Learning Application of Vibration Data and Visualization of Defects for Predictive Maintenance of Gravity Acceleration Equipment. *Appl. Sci.* **2021**, *11*, 1564. [CrossRef]
24. Naseem, U.; Razzak, I.; Khan, S.K.; Prasad, M. A Comprehensive Survey on Word Representation Models: From Classical to State-Of-The-Art Word Representation Language Models. *arXiv* **2020**, arXiv:2010.15036. Available online: https://arxiv.org/abs/2010.15036 (accessed on 12 February 2023). [CrossRef]
25. Amodei, D.; Anubhai, R.; Battenberg, E.; Case, C.; Casper, J.; Catanzaro, B.; Chen, J.; Chrzanowski, M.; Coates, A.; Diamos, G.; et al. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. *arXiv* **2015**, arXiv:1512.02595. Available online: https://arxiv.org/abs/1512.02595 (accessed on 25 January 2023).