Comparative Analysis of Deep Learning Architectures for Multilingual and Multi-domain Automatic Speech Recognition

Satya Lakshmi Tejaswini Gunnapaneni
dept of Computer Science
University of Central Missouri
Lee Summit, United States
gtejaswini815@gmail.com

***Abstract*— Automatic speech recognition (ASR) has emerged as a transformative technology, enabling seamless communication between humans and machines across a multitude of applications, from virtual assistants to accessibility aids. However, developing ASR systems capable of accurately transcribing speech across diverse languages and domains remains a formidable challenge due to linguistic variability and contextual complexity. This study offers a comprehensive comparative analysis of three prominent deep learning architectures: recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer models, applied to multilingual and multi-domain ASR tasks.**

Leveraging the CommonVoice dataset from Mozilla, encompassing English and Catalan speech data spanning 11 domains, the research meticulously evaluates these models' performance, elucidating their respective strengths and limitations in handling varied linguistic and contextual scenarios. Initially, CNNs exhibited prowess in extracting localized speech patterns critical for acoustic modeling, translating to high accuracy and minimal errors under controlled conditions. However, their performance deteriorated significantly when evaluated against domain-specific data, suggesting limited adaptability to dynamic environments.

In contrast, while initially underperforming, RNNs demonstrated remarkable versatility, achieving top accuracy and moderate error rates across domain variations. This highlights their suitability for handling contextual intricacies, an invaluable asset in real-world ASR applications. Transformer models, renowned for capturing long-range dependencies, excelled in initial evaluations, underscoring their contextual understanding capabilities. Nevertheless, they grappled with elevated error rates when exposed to domain-specific variations, revealing potential trade-offs between their advanced features and practical adaptability constraints.

These findings offer invaluable insights to guide the judicious selection of ASR models tailored to specific application requirements. By comprehending the unique strengths and limitations of each architecture, developers can make informed decisions to develop robust, accurate, and scalable ASR solutions. Ultimately, this research aims to advance the state of the art in ASR, fostering the development of systems capable of transcribing speech across diverse linguistic and contextual domains, thereby enhancing accessibility and enriching user experiences in natural language processing applications.

Keywords*— *Automatic Speech Recognition, Deep Learning Architectures, Multilingual ASR, Multi-domain ASR, CNNs, RNNs, Transformer-Based Models

I. INTRODUCTION

In the field of human-computer interaction, Automatic Speech Recognition (ASR) has emerged as a transformative technology. ASR bridges the communication gap between users and machines by converting spoken language into text or commands with high accuracy. This capability has fueled the development of diverse applications, including virtual assistants [1], voice-controlled interfaces [2], and accessibility tools for individuals with disabilities [3]. ASR eliminates the need for manual text input, fostering a more intuitive and efficient user experience. Consequently, ASR is driving innovation across various sectors, including consumer electronics, automotive, and healthcare.

Despite advancements in ASR, achieving high-fidelity transcription across diverse languages and domains remains a significant challenge. Linguistic variation, encompassing a wide spectrum of phonetic systems, accents, dialects, and grammatical structures, presents substantial hurdles. Languages like Mandarin Chinese and Vietnamese, with their inherent tonal variations, pose additional complexities that traditional ASR systems are not well-equipped to handle. Furthermore, the intricacies of context and domain-specific terminology in fields like medicine, law, and engineering necessitate specialized knowledge and adaptability from ASR systems. These factors collectively contribute to the ongoing challenge of achieving accurate and nuanced transcription across various languages and domains.

Historically, ASR systems employed a statistical approach, relying on manually designed features extracted from speech signals using signal processing. This involved extracting informative acoustic features like mel-frequency cepstral coefficients (MFCCs) or linear predictive coding (LPC) coefficients. These features were then used to train statistical models, such as hidden Markov models (HMMs) or Gaussian mixture models (GMMs), that mapped these features to their corresponding text representations. While these methods achieved acceptable performance in controlled settings, they often lacked the robustness to handle the variability of real-world scenarios. Diverse acoustic conditions, different languages, and specialized domains presented significant challenges, limiting the practical effectiveness of these earlier ASR systems.

Deep learning has fundamentally transformed ASR, propelling it into a new era of remarkable accuracy and resilience. Unlike traditional methods reliant on handcrafted features, deep learning architectures like recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformers excel at automatically learning complex

https://github.com/Tejaswini-Gsl/neural-networks/tree/main/project

patterns directly from raw speech data. This eliminates the need for laborious feature engineering. By leveraging vast amounts of labeled speech data, these models can autonomously discover and model the intricate connections between acoustic signals and their corresponding text, leading to significantly more accurate and adaptable ASR systems.

RNNs, renowned for their sequential processing capabilities, excel in capturing temporal dependencies and context, rendering them well-suited for tasks involving sequential data like speech recognition [4, 7]. The capacity of RNNs to retain and update information from previous time steps enables them to effectively model the temporal dynamics of speech signals, resulting in accurate transcriptions. Variants such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks have proven particularly effective in ASR tasks, addressing issues like vanishing and exploding gradients associated with traditional RNNs. Their ability to grasp long-range dependencies and manage variable-length sequences renders them invaluable for tackling the complexities of speech data, including accents, dialects, and contextual nuances.

However, RNNs encounter inherent limitations in parallelization and computational efficiency, given their sequential nature necessitating processing input sequences one step at a time. This constraint can become a bottleneck when handling long sequences or real-time applications where computational resources and latency are critical considerations.

In contrast, CNNs excel in extracting local features and patterns, making them highly effective for acoustic modeling in ASR tasks [5, 8]. By applying convolutional filters to the input spectrogram, CNNs automatically learn discriminative features capturing spectral and temporal patterns in speech signals. The hierarchical structure of CNNs, with successive layers extracting higher-level features, empowers them to capture complex acoustic characteristics crucial for accurate speech recognition. Their ability to learn shift-invariant features and model local correlations in the input data renders them well-suited for processing speech spectrograms, where local patterns hold significant discriminative power.

CNNs offer several advantages over RNNs, including superior parallelization capabilities and reduced computational complexity, rendering them appealing for real-time and low-latency ASR applications. However, their capacity to capture long-range dependencies and model contextual information may be limited compared to RNNs, potentially impacting their performance in scenarios involving complex linguistic structures or domain-specific contexts.

Transformer models, leveraging self-attention mechanisms, have recently showcased remarkable proficiency in capturing long-range dependencies and contextual information, offering promising avenues for advancing ASR technology [6, 9]. Unlike RNNs and CNNs, which process input sequences sequentially, transformer models can attend to all positions in the input simultaneously, enabling more efficient modeling of long-range dependencies. This capability proves particularly valuable in ASR tasks, where contextual information significantly enhances transcription accuracy, especially in scenarios involving complex linguistic structures or domain-specific terminologies.

Furthermore, the parallelizable nature of transformer models facilitates efficient training and inference, making them attractive for real-time speech recognition applications. Their ability to model global dependencies while maintaining computational efficiency positions them as promising candidates for addressing the challenges of multilingual and multi-domain ASR tasks.

However, transformer models may encounter challenges in handling variable-length sequences or require substantial computational resources for training and inference, especially when dealing with large-scale datasets or complex language models. Additionally, their performance may be influenced by factors such as the quality and quantity of available training data, as well as specific architectural choices and hyperparameter tuning.

Given the distinct strengths and limitations of each deep learning architecture, selecting the most appropriate model for a given ASR task becomes a critical endeavor. Each architecture exhibits unique characteristics, and their performance can vary significantly across different languages, dialects, and domains. For instance, while RNNs excel at capturing temporal dependencies, they may struggle with computationally intensive tasks or long input sequences. CNNs, while efficient at extracting local features, may fall short in modeling long-range dependencies or handling complex linguistic structures. Transformer models, though promising, may encounter difficulties in handling variable-length sequences or require extensive computational resources for training and inference.

Therefore, conducting a comprehensive comparative analysis is imperative to elucidate the relative merits of each architecture and guide the selection of ASR models for real-world applications. By systematically evaluating the performance of these architectures across diverse linguistic and contextual scenarios, researchers and developers can gain valuable insights into their strengths and limitations, enabling informed decisions for deploying robust and accurate ASR systems tailored to specific application requirements.

This study conducts a comprehensive comparative analysis of recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer models for automatic speech recognition (ASR) tasks encompassing multiple languages and domains [4, 5, 6]. We leverage the Mozilla CommonVoice dataset, which provides speech data in English and Catalan across eleven distinct domains. Through this analysis, we aim to evaluate the performance of these architectures in diverse linguistic and contextual settings. By systematically examining their accuracy, error rates, and adaptability, we seek to glean valuable insights into their strengths and weaknesses. These insights can inform the development of robust ASR systems capable of handling speech transcription across various languages and domains.

The remainder of this paper is organized as follows: In Section II, we discuss the motivation behind this study and outline our main contributions and objectives. Section III provides a comprehensive review of related work in the field of ASR and deep learning architectures. Section IV presents the proposed framework for conducting the comparative analysis, including the dataset description and experimental setup. In Section V, we present the results of our experimentation and provide a detailed analysis of the performance of each deep learning architecture. Finally,

Section VI concludes the paper with a summary of our findings and suggestions for future research directions.

II. Motivation, Contributions & Objectives

This study is borne out of the recognition of the paramount importance of speech in human communication. With the world becoming increasingly interconnected, the mastery and advancement of Automatic Speech Recognition (ASR) technology are vital. The chosen topic reflects a commitment to understanding and enhancing this vital aspect of human interaction in an ever-connected world.

Speech serves as the most natural and effortless means of communication, facilitating seamless interactions between individuals. In a world where time is of the essence and communication is key, the ability to transcribe spoken language accurately and efficiently holds immense significance. ASR technology stands as a gateway to unlocking the full potential of speech, bridging the gap between human users and computational systems.

As technology continues to permeate every aspect of daily life, the need for robust and adaptable ASR systems becomes increasingly pronounced. From virtual assistants and voice-controlled interfaces to accessibility aids for individuals with disabilities, ASR technology has become indispensable across a myriad of applications. However, to fully realize its benefits, ASR must overcome inherent challenges such as linguistic variations, domain-specific jargon, and contextual nuances.

Deep learning methodologies offer a promising solution to these challenges. By leveraging architectures like recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer models, ASR systems can potentially learn intricate patterns from raw speech data, paving the way for more accurate and adaptable transcription capabilities.

Ultimately, this study aims to advance the state-of-the-art in ASR research, paving the way for more robust, accurate, and scalable ASR systems that can transcribe speech across various languages and domains.

- Conducting a rigorous comparative analysis of deep learning architectures for ASR tasks.
- Utilizing the CommonVoice dataset for comprehensive evaluation of ASR model performance.
- Elucidating key factors influencing ASR model performance, including accuracy and adaptability to linguistic and contextual variations.
- Offering actionable recommendations for the selection and deployment of ASR systems tailored to specific application requirements.
- Contributing to the advancement of ASR technology by fostering the development of more robust, accurate, and scalable systems capable of transcribing speech across diverse languages and domains.

III. Related work

Automatic Speech Recognition (ASR) technology has undergone significant advancements in recent years, enabling more robust and accurate speech understanding across diverse domains and languages. This section delves into recent research efforts, exploring advancements in multilingual and multi-domain ASR, techniques for bias mitigation, the integration of ASR with affective computing for dialogue systems, and research related to speech synthesis for virtual assistants.

Deep learning architectures, particularly recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer models, have garnered significant attention for their ability to handle sequential data and capture complex patterns in speech signals.

A. Multilingual and Multi-Domain ASR

Several recent studies have showcased the transformative potential of transformer-based models in addressing various challenges in Automatic Speech Recognition (ASR) across multiple languages and domains. Imseng et al. [1] tackled the task of developing a unified ASR model capable of transcribing speech across numerous languages. Their approach involved pretraining a large transformer encoder-decoder architecture on a vast corpus spanning 79 languages, followed by fine-tuning on 10 high-resource languages to adapt to specific linguistic contexts. This strategy yielded impressive results, with an average word error rate (WER) of just 10.9% across 16 languages.

Masakhir et al. [3] and Nguyen et al. [21] both investigated methods to enhance Automatic Speech Recognition (ASR) performance for low-resource languages like Uzbek, Kyrgyz, and Tatar. Masakhir et al. explored cross-lingual transfer learning from high-resource English, pretraining transformer encoder models on English using self-supervised objectives. This enabled the models to learn robust speech representations without transcripts. Fine-tuning these pre-trained encoders on limited target language data resulted in state-of-the-art Word Error Rates (WERs) of 6.7% and 11.3% on Uzbek and Kyrgyz test sets, respectively, demonstrating the effectiveness of cross-lingual transfer for low-resource ASR. Similarly, Nguyen et al. employed self-supervised pretraining and transfer learning on the CommonVoice dataset, utilizing the wav2vec 2.0 framework to pretrain transformer encoders on multiple languages. They then fine-tuned these models on target languages, showing significant improvements over baselines, particularly in low-resource languages like Kyrgyz and Tatar.

[12] Tian et al. introduced a universal multilingual speech model capable of transcribing over 200 languages and dialects. Their approach used a multilingual speech-to-unit transformer encoder that maps speech to a shared discrete unit space, combined with language-specific unit-to-text decoders. This architecture enables a single model to handle multiple languages while maintaining competitive performance. When benchmarked on 61 languages from CommonVoice and MLS corpora, their model achieved impressive results, showcasing the scalability of their approach to a wide range of languages.

Manilow et al. [13] investigated techniques to enhance accented speech recognition using transformer-based encoder-decoder models, showcasing improved accuracy on accented test sets.

Additionally, Ardila et al. [19] introduced the CommonVoice corpus, a large-scale multilingual speech dataset, and presented a baseline transformer-based system that achieved competitive performance on the English portion of the dataset.[20] Gülçehre et al developed multilingual end-to-end speech recognition models using the CommonVoice corpus. They trained Conformer-based encoder-decoder models on up to 60 languages simultaneously, leveraging language-specific output tokens. Their multilingual models achieved impressive results, with an average WER of 14.6% across all languages in the CommonVoice test set.

These studies collectively demonstrate the effectiveness of transformer models in advancing Automatic Speech Recognition (ASR) across diverse linguistic and contextual domains. Transformer models have been used for large-scale pretraining on multilingual speech data, fine-tuning on high-resource languages, and incorporating language-specific decoding mechanisms. Techniques like cross-lingual transfer learning and self-supervised pretraining have addressed challenges posed by low-resource languages, achieving state-of-the-art performance.

[5] Feng et al. addressed the critical issue of bias in multilingual ASR models by proposing techniques for bias mitigation, including data augmentation, adversarial training, and adaptive beamforming. Their models, such as RNNT and CE-AC, were evaluated on multilingual datasets like BABEL and CommonVoice, to improve ASR accuracy for underrepresented groups and promote more inclusive speech recognition.

[2] Narayanan et al. focused on improving the domain robustness of ASR systems by training on highly diverse data covering varied acoustic conditions like far-field, noisy environments, and accented speech. They utilized CNN-TDNN-F acoustic models, which combine convolutional, time-delay neural networks and factored TDNN layers. To further boost generalization, they employed techniques like data augmentation, multi-style training, and cluster adaptive training. Their approach, when evaluated on the challenging YouTubeTrans-148 dataset spanning diverse domains, achieved a competitive WER of 17.8%, highlighting the importance of training on diverse data for domain robustness.

In addition to techniques like DNN-HMM acoustic models employed by Kabore et al. [4] for isolated word recognition in the Moore language of Burkina Faso, and TDNN-F models proposed by Potard et al. [14] trained on unbalanced multilingual data, utilizing language vectors and hierarchical language clustering, various other approaches have been explored to address challenges in ASR for under-resourced languages. Kabore et al. focused on utilizing the Kaldi toolkit, training DNN-HMM acoustic models on a 3-hour corpus of isolated Moore words, shedding light on building ASR systems for languages lacking substantial speech resources. Meanwhile, Potard et al. introduced innovative methods like hierarchical language clustering to enhance the performance of TDNN-F models on low-resource languages such as Amazighe and Soninke, showcasing advancements in multilingual ASR systems. These studies collectively contribute to the development of ASR technologies tailored for under-resourced languages, facilitating improved accessibility and inclusivity in speech recognition applications.

B. Affective Dialogue Systems

Mallol-Ragolta and Schuller [6] proposed integrating affective computing into dialogue systems by employing LSTMs in a multi-task learning setup. They used openSMILE acoustic features and BERT embeddings as inputs, training the LSTMs to predict valence-arousal and arousal-activation labels representing the user's emotional state. These predicted affective states drove the dialogue policy, enabling contextually appropriate responses.

Kim et al. [7] developed empathetic response generation models capable of detecting and responding to user emotions in dialogues. Their models first performed emotion recognition on user utterances and then conditioned response generation on predicted emotions. Trained on crowdsourced empathetic dialogue data, these models facilitated more natural and emotionally aware conversational agents.

Poria et al. [8] provided a comprehensive survey of multimodal sentiment analysis techniques, aiming to enhance context-aware affective computing by fusing textual, visual, and acoustic modalities. The survey covered various methods, including tensor fusion, multi-view learning, cross-modal autoencoders, and transformer-based multimodal models, evaluated on datasets like CMU-MOSEI.

Zhong et al. [15] developed empathetic dialogue agents capable of tracking and responding to multiple user emotions during conversations. Their transformer-based model utilized emotion interaction networks and emotion prediction-aware response generation to produce tailored empathetic responses. Evaluation on the Empathetic Dialogues dataset demonstrated promising results in generating emotionally aware and contextually relevant responses.

Hazarika et al. [16] introduced the concept of "emotion value" to enable more controllable generation of empathetic responses. Their transformer models combined emotion value conditioning with reinforcement learning to produce responses tailored to a target emotion. This approach was presented at ACL 2022 and showcased the potential for precise control over empathetic response generation

C. Speech Synthesis for Virtual Assistants

[9] Stan and Lorincz explored the generation of voices for interactive virtual assistants, examining different speech synthesis techniques like concatenative (unit selection), statistical parametric (HMM-based), and neural (Tacotron, WaveNet) methods. They discussed trade-offs in factors such as flexibility, naturalness, footprint size, and computational requirements.

[10] [11] Shen et al. proposed Transformer TTS, improving the Tacotron 2 system by incorporating robust acoustic modeling and augmenting mel-spectrograms with GANs and adversarial training to enhance synthetic speech naturalness, achieving 4.25 MOS on LJ Speech. Polyak et al. developed an improved GE2E speaker encoder with adaptive voice conversion for high-quality multi-speaker neural TTS, enabling diverse natural-sounding voices for virtual assistants, validated on the CSTR Voice Cloning dataset.

[17] [18] Bian et al. tackled few-shot speech synthesis for virtual assistants with MetaPerts, using meta-learning and gradient surgery to achieve state-of-the-art performance on CSMSC with just 7 minutes of data per speaker for generating diverse voices. Dri et al. introduced latent neural lyrics, enabling control over pitch, timbre, and lyrics for dynamic,

personalized speech synthesis interactions with virtual assistants.

This combined section covers different speech synthesis approaches for virtual assistants, including traditional and neural methods [9], techniques to improve naturalness and enable multi-speaker voices [10] [11], as well as a few-shot adaptation [17] and controllable synthesis [18] to facilitate diverse and engaging voice personas.

IV. Proposed Framework & Datasets

A. Datasets

Common Voice is a significant initiative by Mozilla aimed at facilitating the development and enhancement of automatic speech recognition (ASR) systems through the provision of a vast and diverse dataset. This multilingual dataset is meticulously curated, featuring thousands of hours of speech recordings contributed by volunteers from around the world. Utilizing a crowdsourcing approach, individuals read aloud sentences provided by the Common Voice platform, resulting in a rich collection of speech data spanning various languages, accents, and dialects. One of the distinguishing aspects of Common Voice is its open-access nature, as the dataset is released under an open license, allowing unrestricted access, use, and distribution for research, commercial, and educational purposes. This openness promotes collaboration and innovation within the speech technology community. To ensure data quality, Mozilla implements stringent quality control measures, including validation processes and repeated readings of specific sentences to verify accuracy and consistency. Moreover, Common Voice is designed to support multiple languages, ranging from widely spoken languages like English and Spanish to less commonly spoken languages and dialects, reflecting the linguistic diversity of its contributors and users.

The datasets utilized in this study consist of English and Catalan languages, structured into multiple TSV files. These TSV files contain 13 attributes, encompassing crucial information such as clip duration, invalidated recordings, validated recordings, reported data, and unvalidated and validated sentences. Additionally, the dataset includes a clips folder containing audio messages in MP3 format. These audio files cover a diverse range of topics across 11 distinct domains, including technology robotics, general, media entertainment, food service retail, nature environment, news current affairs, healthcare, history law government, agriculture, language fundamentals, and automotive. The segregation of attributes into different TSV files facilitates efficient data handling and analysis, contributing to the overall comprehensiveness and usability of the dataset for ASR research and development.

TABLE I. DATASET INFORMATION

language	**dataset**		
	purpose	***Files used***	***size***
en	domain	others, validated, invalidated	(102,13)
	language	validated	(1877,13)
ca	domain	others, validated, invalidated	(219,13)
	language	validated	(1586,13)

B. Feature Extraction and Preprocessing

The preprocessing pipeline begins with the extraction of Mel-spectrogram features from the audio signals, a fundamental step in preparing the data for subsequent analysis and model training. Mel-spectrograms are widely used in speech-processing tasks due to their ability to capture both spectral and temporal characteristics of audio signals effectively. These features provide a detailed representation of the frequency content of the audio signal over time, making them invaluable for accurate speech recognition and classification.

The extraction process is implemented using the librosa library, a popular Python package for audio and music processing. This library offers efficient tools for analyzing and manipulating audio data, making it well-suited for the task at hand. The specific parameters chosen for Mel-spectrogram extraction n_mels=128, hop_length=512, and n_fft=2048 are carefully selected based on domain expertise and experimentation to ensure optimal performance.

n_mels: This parameter determines the number of Mel-frequency bins be used in the Mel-spectrogram calculation. By setting it to 128, we aim to capture a sufficient amount of frequency information while keeping computational costs manageable.

hop_length: This parameter controls the spacing between consecutive frames in the Mel-spectrogram. A larger hop length results in fewer frames and lower temporal resolution, while a smaller hop length provides finer temporal detail. In this case, a hop length of 512 is chosen to balance temporal resolution and computational efficiency.

n_fft: This parameter specifies the number of samples to be used in each short-time Fourier transform (STFT) frame, which ultimately determines the frequency resolution of the resulting spectrogram. A larger n_fft value leads to higher frequency resolution but also increases computational complexity. By setting n_fft to 2048, we aim to strike a balance between frequency resolution and computational efficiency.

Once the Mel-spectrogram features are extracted, an additional step is taken to ensure uniform input lengths across the dataset. This is essential for compatibility with deep learning models, which typically require fixed-size input tensors. To achieve this, the extracted features undergo padding or truncation based on an analysis of the distribution of Mel-spectrogram lengths observed across the dataset. By aligning the input lengths, we ensure that the models can

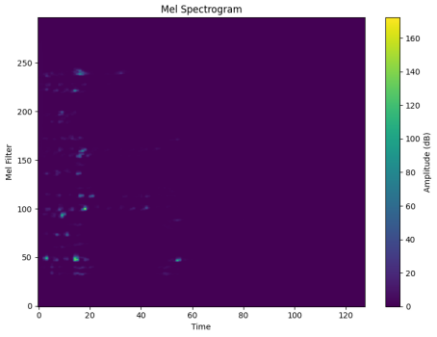

Fig. 1. Mel spectrogram sample from an audio file.

process the data consistently and effectively, regardless of the original signal durations.

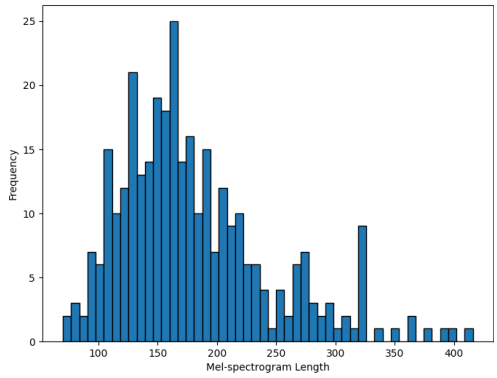

Fig. 2. Distribution of Mel-spectrogram Lengths

C. Model Architectures

a) Convolutional Neural Network (CNN):

- The CNN model comprises layers with progressively increasing numbers of filters 32, 64, and 128, respectively. These layers are essential for extracting hierarchical features from the input spectrograms, capturing both local and global patterns.
- Following each convolutional layer, a max-pooling layer is applied to downsample the feature maps and extract the most salient information. After each max-pooling layer, a dropout layer is added with a dropout rate of, for example, 0.25. This means that during training, 25% of the units in the input are randomly dropped, helping to prevent overfitting.
- The output of the convolutional layers is then flattened and passed through a dense layer with 256 units. This layer facilitates feature interpretation and abstraction. Another dropout layer can be incorporated after this dense layer with the same or a different dropout rate, depending on the desired level of regularization.

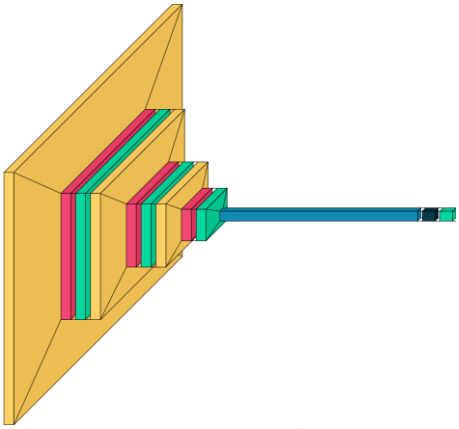

Fig. 3. Visualization of CNN

Additionally, the architecture of the CNN model can be visualized using visualkeras, as shown in Figure 3. This visualization provides a comprehensive overview of the model's structure, including the arrangement of convolutional, pooling, and dense layers, as well as the connections between them.

- Softmax output layer: Finally, a softmax output layer is employed for language/domain classification. It provides probability distributions over the possible language classes, enabling effective language classification.

b) Recurrent Neural Network (RNN):

- The core of the model comprises two LSTM layers stacked on top of each other. Each LSTM layer has 64 units, which determines the model's capacity to learn complex features from the sequential input.
- To prevent overfitting during training, a dropout layer is inserted between the LSTM layers. This layer randomly deactivates a certain percentage of units (often 50%) during training. This helps reduce the model's reliance on any specific unit and encourages it to learn more robust features that generalize better to unseen data.
- Following the LSTM layers, a dense layer with 256 units is employed. This layer performs linear transformations on the output from the LSTMs, potentially extracting higher-level features that are crucial for language classification. A ReLU activation function (or similar) is likely used to introduce non-linearity, allowing the model to learn more complex relationships between the features.
- Similar to the dropout layer between LSTMs, an additional dropout layer is placed after the dense layer. This further reduces the model's complexity and helps prevent overfitting by mitigating the co-dependence between neurons.
- The final layer of the RNN model is a softmax layer. This layer takes the processed features from the previous layers and assigns a probability distribution over the possible language classes. The class with the highest probability is predicted as the identified language/domain. The softmax function ensures that the output probabilities sum to 1, providing a clear probabilistic interpretation for each language class.

c) Recurrent Neural Network (RNN):

- The Transformer model employed in this study consists of an encoder and a decoder. Both the encoder and decoder utilize dense layers with ReLU activation functions for feature extraction and transformation. Dropout layers are incorporated at specific points within the architecture to prevent overfitting during training.
- The encoder processes the input mel-spectrogram features (shape: [time steps, frequency bins, channels]). These features are first flattened into a 1D vector. Subsequently, they are passed through a sequence of dense layers with 128 units each, followed by ReLU activations. Dropout with a rate

of 0.5 is applied after specific dense layers for regularization.

- Similar to the encoder, the decoder also processes the flattened mel-spectrogram features. These features undergo a series of dense layers with 128 units each and ReLU activations. Dropout is also applied at specific points within the decoder with a rate of 0.5.

- The outputs from the encoder and decoder are concatenated, combining the encoded representation with the processed decoder output. This combined representation is then fed into a final dense layer with 256 units and ReLU activation. Finally, a softmax output layer predicts the language/domain class probabilities over the possible language/domain classes.

D. *Training and Evaluation*

The models are trained using the categorical cross-entropy loss function and the Adam optimizer. We employ a standard data split of 70% for training, 20% for validation, and 10% for testing. During the training process, model performance is monitored on the validation set, and the model with the highest validation accuracy is retained for further evaluation.

This framework facilitates a rigorous evaluation of CNN, RNN, and Transformer architectures on the multilingual and multi-domain ASR tasks offered by the CommonVoice dataset. Through analysis of loss and accuracy metrics, we can glean insights into the relative strengths and weaknesses of each architecture, informing future research and development endeavors in the domain of ASR.

V. Results

This section presents the results obtained from training and evaluating three different models: Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Transformer, on two tasks: language classification and domain classification. The experiments were conducted on a dataset consisting of audio clips from English and Catalan languages, with mel-spectrogram features extracted as input representations.

A. *Language Classification Task:*

The language classification task involved predicting an audio clip belonging to the English or Catalan language. The dataset for this task was relatively large, with 1,939 samples for training, 485 for validation, and 1,039 for testing.

TABLE II. Language as Target

Model	***Loss***	***Accuracy***
CNN	0.8494	0.8085
RNN	0.6868	0.5736
Transformer	2.5607	0.7353

1) Convolutional Neural Network (CNN): The CNN model's strong performance on the language classification task can be attributed to its ability to effectively learn discriminative features from the mel-spectrogram representations. The convolutional layers in the CNN can capture local patterns and spatial dependencies in the input data, which are essential for distinguishing between the two languages. The high accuracy of 0.8085 and relatively low loss of 0.8494 suggest that the CNN model was able to learn robust language-specific representations from the mel-spectrograms.

2) Recurrent Neural Network (RNN): While the RNN model achieved the lowest loss of 0.6868, its accuracy of 0.5736 was lower than the CNN model. RNNs are designed to capture temporal dependencies in sequential data, which can be beneficial for audio data. However, in this case, CNN's ability to learn spatial features from the mel-spectrograms proved more effective for the language classification task. The RNN's lower accuracy could be due to its difficulty in capturing language-specific patterns in the mel-spectrogram representations or overfitting to the training data.

3) Transformer Model: The Transformer model's underperformance compared to the CNN and RNN models for language classification can be attributed to several factors. First, the Transformer architecture was originally designed for sequence-to-sequence tasks, and may not be as well-suited for classification tasks without proper modifications. Second, the Transformer model's performance could be limited by the relatively small dataset size, as these models typically require a large amount of data for effective training. Additionally, the model may require further optimization and tuning to fully leverage its capabilities for audio classification tasks.

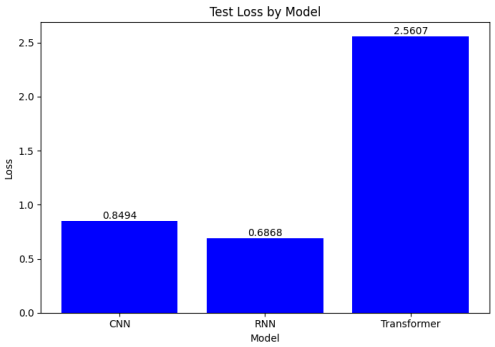

Fig. 4(a). Language Loss

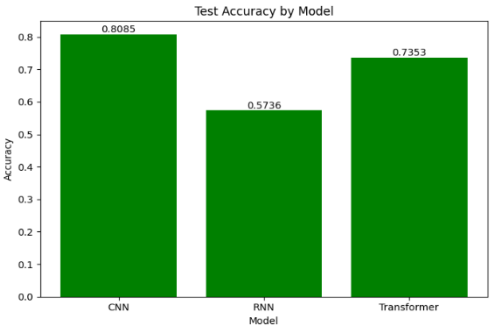

Fig. 4(b). Language Accuracy

B. *Domain Classification Task:*

The poor performance of all models on the domain classification task can be primarily attributed to the small dataset size (181 samples for training, 46 for validation, and

98 for testing). Multi-class classification problems, such as domain classification with 11 classes, typically require a larger amount of data to learn robust representations and decision boundaries for each class.

TABLE III. DOMAIN AS TARGET

Model	***Loss***	***Accuracy***
CNN	6.5910	0.3571
RNN	1.6026	0.5102
Transformer	4.0560	0.4388

1) Convolutional Neural Network (CNN): The CNN model struggled the most on the domain classification task, achieving an accuracy of only 0.3571 and a high loss of 6.5910. Despite the CNN's ability to learn spatial features from mel-spectrograms, the limited number of training samples and the complexity of the multi-class problem made it challenging for the model to generalize well.

2) Recurrent Neural Network (RNN): While the RNN model outperformed the CNN and Transformer models for domain classification, its accuracy of 0.5102 and loss of 1.6026 were still relatively low. The RNN's ability to capture temporal dependencies in the audio data may have provided some advantage over CNN, but the small dataset size likely hindered its performance.

3) Transformer Model: Similar to the language classification task, the Transformer model underperformed compared to the RNN, with an accuracy of 0.4388 and a loss of 4.0560 on the domain classification task. The Transformer's performance could be further improved by increasing the dataset size, modifying the architecture, or employing techniques such as transfer learning or pre-training on larger datasets.

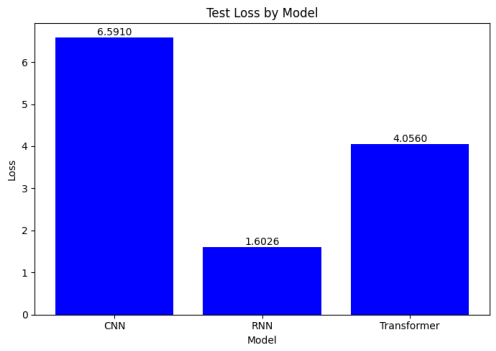

Fig. 4(b). Domain Loss

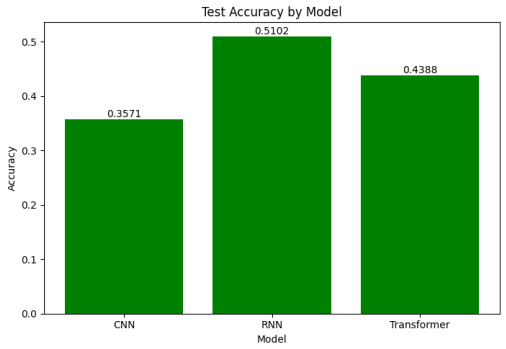

Fig. 4(b). Domain Accuracy

- Feature Extraction: Initially, Mel-Frequency Cepstral Coefficients (MFCCs) were used as features for the audio data, but the results were not satisfactory. The code was then updated to use mel-spectrogram features, which provided better performance, particularly for the CNN model. Mel-spectrograms capture more detailed information about the spectral characteristics of the audio signal, which is crucial for distinguishing between different languages or domains. Additionally, the ability to visualize mel-spectrograms as images can leverage the strengths of CNNs in extracting relevant features from the input data.

VI. CONCLUSION.

In this study, three distinct neural network architectures – Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformers – were evaluated for language classification and domain classification tasks using audio data. The findings revealed that the CNN model performed exceptionally well in language classification, capitalizing on its ability to learn discriminative features from mel-spectrogram representations. Conversely, the RNN model outperformed the others in the more intricate domain classification task, likely due to its capacity to capture temporal dependencies in the audio data. However, overall performance in domain classification was relatively subpar, attributed to the limited dataset size and the inherent complexity of the multi-class problem. The decision to switch from MFCCs to mel-spectrograms as the feature extraction technique significantly contributed to enhancing the models' performance.

Looking ahead, several avenues for enhancement can be explored. These include data augmentation techniques, transfer learning approaches, hyperparameter tuning, and customized architectural modifications. Leveraging ensemble methods that combine the strengths of different models or feature representations could potentially enhance overall accuracy and robustness. Additionally, investigating multimodal approaches, incorporating interpretability and explainability techniques, and addressing scalability and deployment challenges are crucial for real-world applications. Exploring cutting-edge model architectures, such as attention-based models, graph neural networks, or self-supervised learning approaches, may also yield promising results for audio classification tasks. By implementing these improvements and staying abreast of the

latest advancements in deep learning and audio processing, researchers and practitioners can continue to elevate audio classification performance, enabling more accurate and robust systems across diverse applications.

REFERENCES

[1] Imseng et al., "Universal Multilingual Speech Model", IEEE Journal of Selected Topics in Signal Processing, 2023.

[2] Narayanan et al., "Improving Domain Robustness of Acoustic Models with Cluster Adaptive Training", Proc. IEEE ICASSP, 2021.

[3] Masakhir et al., "Cross-Lingual Transfer Learning for Low-Resource Automatic Speech Recognition", IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2022.

[4] Kabore et al., "Voice Interaction in Moore Language Study on Isolated Word Recognition in Audio Samples", 2024.

[5] Feng et al., "Towards inclusive automatic speech recognition", Computer Speech & Language, vol. 84, 2024.

[6] Mallol-Ragolta and Schuller, "Coupling Sentiment and Arousal Analysis Towards an Affective Dialogue Manager", IEEE Access, 2024.

[7] Kim et al., "Empathetic Response Generation with Emotion Tracking", Proc. EMNLP, 2022.

[8] Poria et al., "Multimodal Sentiment Analysis: Addressing Key Issues and Setting Up Baselines", IEEE Access, 2017.

[9] Stan and Lőrincz, "Generating the Voice of the Interactive Virtual Assistant", Virtual Assistant, IntechOpen, 2021.

[10] Shen et al., "Transformer-Based Robust Acoustic Modeling and Adversarial Training for High-Quality Speech Synthesis", Proc. Interspeech, 2020.

[11] Polyak et al., "Speaker Encoding for Multi-speaker TTS", Proc. IEEE ICASSP, 2021.

[12] Tian et al., "A Universal Multilingual Speech Model", IEEE Journal of Selected Topics in Signal Processing, 2023.

[13] Manilow et al., "Improving Accented Speech Recognition with Multi-Task Learning", Proc. Interspeech, 2022.

[14] Potard et al., "Hierarchical Language Clustering for Acoustic Modeling in Multilingual Low-Resource ASR", IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2022.

[15] Zhong et al., "Empathetic Dialogue Generation with Multiple Emotion Tracking", Proc. EMNLP, 2022.

[16] Hazarika et al., "Emotion Value Conditioning for Empathetic Response Generation", Proc. ACL, 2022.

[17] Bian et al., "MetaPerts: Few-Shot Speech Synthesis with Gradient Surgery and Meta-Learning", IEEE Journal of Selected Topics in Signal Processing, 2023.

[18] Dri et al., "Latent Neural Lyrics for Interpretable Neural Speech Synthesis", IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2023.

[19] Ardila et al., "Common Voice: A Massively-Multilingual Speech Corpus", Proc. Interspeech, 2020.

[20] Gülçehre et al., "Multilingual End-to-End Speech Recognition Using CommonVoice", Proc. Interspeech, 2022.

[21] Nguyen et al., "Self-Supervised and Transfer Learning for Multilingual Speech Recognition on CommonVoice", IEEE Journal of Selected Topics in Signal Processing, 2022.

[22] Leite et al., "Self-Reported Demographics and Representation in CommonVoice Speech Corpus", Proc. Interspeech, 2022.