Exploring Whisper Embeddings for Stutter Detection: A Layer-Wise Study

Ashita Batra, Brajesh Kar, Pradip K Das
Department of Computer Science and Engineering
Indian Institute of Technology Guwahati, Guwahati-781039, India
Guwahati, India
{b.ashita, b.kar, pkdas}@iitg.ac.in

***Abstract*—As voice assistants become increasingly integrated into our daily lives, the need for disfluency detection continues to grow. This paper highlights the importance of encoder representations in Whisper model, particularly in the middle layers, for capturing stutter characteristics. Our findings reveal that sound repetitions, blocks, and prolongations are best captured by the middle layers, whereas word repetition and interjections are predominantly recognized in the final layers. This is likely because Whisper processes them as filler words or repeated speech within normal conversation. We evaluate performance across various data splits and datasets to determine whether Whisper primarily learns language representations or stutter characteristics. However, results from cross-corpora testing suggest that Whisper does not inherently learn stutter characteristics, highlighting the need for further refinement in stutter detection methodologies.**

***Index Terms*—Stuttering, Whisper, disfluency detection, layer embeddings**

I. Introduction

Speech disfluencies refer to obstructions in the normal flow of speech occurring 4 - 6% of times while speaking normally [1], [2]. Stuttering is one such type of speech disfluency which affects around 1% of the world's population approximating to 70 million people [3]. It significantly shapes an individual's personality, distinguishing them from others by affecting their confidence. This, in turn, can influence social interactions, sometimes leading to challenges such as fear, stress, and anxiety [4]. It is recognized in one's speech by taking the form of repeating syllables, words, phrases or taking a long pause in between words or adding filler words like 'umm', 'uhh', etc as explained with example in Table I.

Intrinsically, automatic speech recognition (ASR) systems are trained on natural fluent speech. When it comes to detecting disfluent speech using ASRs, they are unable to identify the presence of disfluency. The disfluency in the speech is treated as noise and it may aim to remove that part of speech. Due to this reason, people who stutter are unable to use voice assistants like Alexa, Siri, or Google Assistant because it becomes difficult for these devices to identify the disfluency and thus act accordingly [5], [6]. Recently, in 2022 around 24% of people [6] bought smart speakers to help with their disfluency, yet it does not work accurately for people who stutter [7]. The challenges associated with ASRs highlight the importance of training them on disfluent speech. By doing so, we can ensure that a larger segment of the global population—often overlooked—benefits from improved accessibility. This approach can support individuals in their everyday communication, aid in speech therapy, and foster inclusivity in voice-based technologies. There are several approaches to addressing these challenges, including collecting and publicly sharing datasets, generating synthetic data, or fine-tuning ASRs to recognize stuttering and disfluencies. Additionally, layers in ASR models such as wav2vec2.0 [8] and Whisper [9] play a crucial role in detecting speech irregularities. This work is inspired by recent research efforts to adapt transformers for identifying speech disfluencies through encoded representations. Our primary focus is to analyze the layer-wise characteristics captured by the Whisper Large v3 [9] model with an aim to explore how Whisper's middle layers interpret the characteristics of disfluent speech, providing deeper insights into its processing capabilities. In conclusion, our contribution can be summarized as follows:

1) Our work helps uncover whether stuttering patterns are captured earlier in the model or only in later decision-making layers by Whisper?
2) Can we propose a lightweight efficient model using Whisper for stutter detection?
3) Whether fine-tuning the Whisper architecture learns language representation or speaker's stutter characteristics?

Stuttering Label	**Definition and Examples**
Prolongation (PR)	Elongated syllables e.g. *"Nnnnot now'*
Block (BLK)	Stoppages in speech. e.g. *"Call ...me"*
Sound Repetition (SR)	Repeated syllables. e.g. *"I **ca-ca-**can."*
Word Repetition (WR)	Repeated words. e.g. *"He **[he]** knows."*
Interjection (INJ)	Insertion of fillers. e.g. *"**[Uh]** Wait"*

TABLE I: Types of Stuttering and Examples

II. Related Work

Disfluency detection has generally been done using acoustic features, as it is evidently visible in the waveforms and their respective spectrograms. The majority of studies have exercised spectral features like Mel-frequency

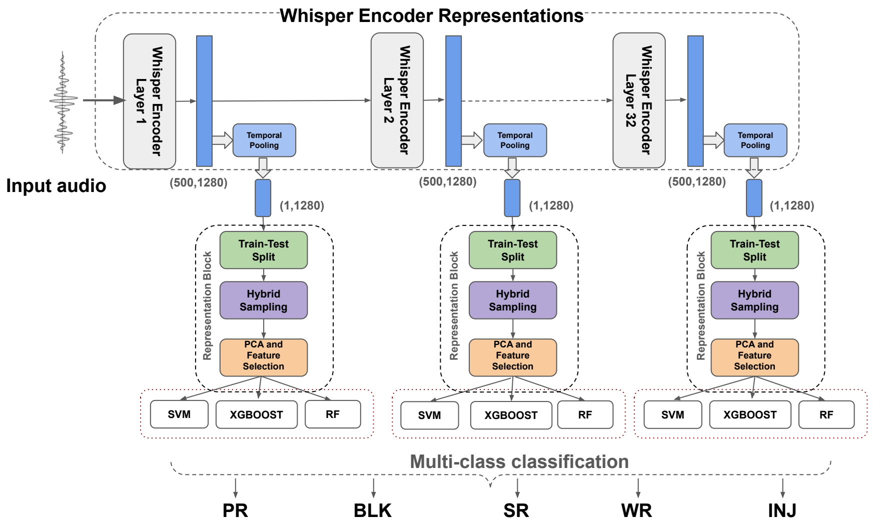

Fig. 1: Architecture for stutter detection using Whisper encoder representations

cepstral coefficients (MFCCs) or Linear predictive coefficients (LPCs) and their respective derivatives to identify different stutter types [10], [11]. Several research studies [12], [13] have explored representations obtained from neural networks for stutter detection (SD) of acoustic encoder models particularly focusing on wav2vec2.0 [8] and Whisper [9]. These transformer-based models have shown remarkable performance in detecting disfluencies [13]–[15]. A work done by [16] exploiting the use of ECAPA-TDNN and wav2vec2.0 embeddings which focused on layers performance and finding whether layer fusion helps in better SD or not. A similar study by [17], where wav2vec2.0 layers were analysed for capturing stutter characteristics. It demonstrated that middle layers capture better and comparable stutter characteristics than those at last layers. Another study [18] demonstrated the use of encoder-decoder models for disfluency detection using wav2vec2.0 and Whisper. They focused on extracting features from layer 12 and 24. In [19], they also compared ASR models like Google ASR [20] and WhisperX for disfluency detection with one of the objective that whether choice of ASR systems impacts model's performance. Building on the successful adaptation of large pre-trained models for stutter classification, Ameer et al. [15] used frozen encoded representations, further which were used for disfluency detection.

III. Data

In our study, we use the Sep-28k [21], FluencyBank [22], KSoF [23] and Boli [10] datasets. Among these, Sep-28k is the largest publicly available dataset, derived from 265 podcasts with 28,177 clips ($\approx 3 seconds$). It includes sound and word repetition, prolongation, blocks, and interjections. FluencyBank, with 4,144 clips from 33 podcasts, mirrors these disfluencies. KSoF, a clinical German dataset, consists of 5,597 clips ($\approx 3 seconds$) from 214 clinical recordings, containing the same disfluency types. Boli, the first publicly available Indian stutter dataset, spans multiple languages, is word-level annotated ($\approx 5 seconds$), and includes 2.8 hours of audio from 28 speakers with both labels and speaker information.

A. Data Curation & Preparation

The Sep-28k [21], FluencyBank [22], and KSoF [23] datasets use labels based on agreement among three annotators. To ensure consistency, we keep files where at least two annotators agreed on a label. Clips with Poor Audio Quality, Difficult to Understand, Music, No Speech, Unsure, and Natural Pause are discarded for better reliability. Also, to distinguish repetition types, we maintain separate labels for sound and word repetition across all datasets. In this paper, we use cross-corpora testing, training Whisper on Sep-28k and testing on Boli, KSoF, and FluencyBank. This helps assess whether the model learns stuttering patterns or language characteristics. For speaker-exclusive testing, we apply five-fold cross-validation and Leave-One-Podcast-Out (LOPO), ensuring rigorous data partitioning for reliable evaluation.

B. Datasets' Limitations

All datasets show class imbalance, especially in Block and Prolongation categories. Most data is in English, which may affect performance in other languages. In the *HeStutter podcast* from Sep-28k, the female host and male guest cause misclassification in speaker identification, as the model incorrectly attributes both voices to the host. The dataset also lacks guest demographics, making it harder to analyze speaker differentiation challenges. These issues make Leave-One-Speaker-Out (LOSO) partitioning unsuitable for Sep-28k. Meanwhile, although Boli has only 280 clips, expanding it could improve model training rather than just testing.

Layer	**Classifier**	**Sep-28k (5CV)**					**Sep-28k (LOPO)**					**FluencyBank (5CV)**					**KSoF (5CV)**				
		PR	BLK	SR	WR	INJ	PR	BLK	SR	WR	INJ	PR	BLK	SR	WR	INJ	PR	BLK	SR	WR	INJ
15	RF	0.21	0.15	0.20	0.08	0.26	0.35	0.14	0.22	0.13	0.37	0.07	0.14	0.13	0.05	0.19	0.44	0.65	0.24	0.11	0.51
	SVM	0.38	0.28	0.31	0.27	0.39	0.28	0.21	0.15	0.15	0.24	0.20	0.22	0.07	0.21	0.15	**0.65**	**0.67**	**0.39**	0.16	0.53
	XGBoost	0.27	0.22	0.24	0.14	0.33	0.17	0.12	0.11	0.06	0.19	0.20	0.19	0.13	0.16	0.27	0.44	**0.71**	**0.37**	0.11	0.53
16	RF	0.20	0.18	0.24	0.09	0.26	0.36	0.14	0.22	0.14	0.35	0.27	0.05	0.07	0.05	0.19	0.48	**0.70**	0.23	0.16	0.50
	SVM	0.41	0.28	0.30	0.29	0.39	0.31	0.21	0.16	0.15	0.25	0.33	0.32	0.07	0.21	0.31	0.58	0.65	**0.41**	0.16	0.54
	XGBoost	0.26	0.26	0.27	0.17	0.32	0.19	0.14	0.11	0.08	0.18	0.27	0.16	0.20	0.05	0.31	0.45	**0.71**	0.28	0.11	0.59
19	RF	0.27	0.21	0.23	0.13	0.32	**0.45**	0.13	0.30	0.22	0.41	0.07	0.16	0.07	0.00	0.19	0.41	0.61	0.23	0.11	0.57
	SVM	0.43	**0.31**	0.32	0.32	0.45	0.37	**0.25**	0.20	0.20	0.33	**0.40**	0.30	0.07	0.26	0.38	0.57	0.65	0.35	0.16	**0.68**
	XGBoost	0.31	0.25	0.29	0.20	0.34	0.31	0.19	0.17	0.12	0.23	0.07	0.19	0.07	0.16	0.19	0.51	**0.67**	0.31	0.05	0.59
20	RF	0.27	0.16	0.21	0.13	0.39	**0.48**	0.10	0.30	0.18	0.42	0.13	0.05	0.13	0.11	0.19	0.38	**0.67**	0.31	0.11	0.51
	SVM	0.41	0.28	0.32	0.32	0.45	0.36	**0.25**	0.19	0.19	0.34	0.33	0.30	0.07	0.16	0.23	0.59	0.65	**0.39**	0.21	**0.66**
	XGBoost	0.34	0.21	0.31	0.24	0.36	0.29	0.18	0.17	0.13	0.27	0.33	0.08	0.27	0.16	0.15	0.48	0.65	0.25	0.05	0.58
25	RF	0.21	0.21	0.24	0.15	0.32	0.42	0.11	**0.32**	0.22	**0.46**	0.20	0.16	0.13	0.05	0.23	0.35	**0.67**	0.20	0.05	0.51
	SVM	**0.47**	0.30	0.34	0.41	0.46	0.38	0.24	0.29	0.26	0.41	0.20	0.35	0.20	0.16	0.23	0.59	0.65	**0.37**	**0.47**	0.64
	XGBoost	0.32	0.28	**0.35**	0.23	0.38	0.30	0.20	0.24	0.16	0.30	0.40	0.22	0.20	**0.32**	0.31	0.40	**0.67**	**0.37**	0.05	**0.68**
27	RF	0.25	0.22	0.27	0.22	0.36	0.39	0.12	**0.34**	0.25	**0.48**	0.13	0.24	0.27	0.21	0.19	0.38	**0.67**	0.28	0.05	0.65
	SVM	**0.49**	**0.36**	**0.35**	0.43	**0.55**	0.38	**0.27**	0.28	0.36	0.42	0.20	0.41	0.27	0.21	0.27	0.56	0.65	**0.39**	0.32	**0.69**
	XGBoost	0.37	0.28	0.32	0.31	0.43	0.27	0.23	0.23	0.23	0.28	0.27	0.30	0.20	0.21	0.27	0.47	**0.66**	0.28	0.05	**0.68**
28	RF	0.23	0.22	0.25	0.21	0.37	0.36	0.12	**0.36**	0.30	**0.47**	0.20	0.30	0.20	0.11	0.27	0.41	**0.67**	0.25	0.11	0.64
	SVM	**0.48**	**0.35**	**0.37**	0.52	**0.53**	0.39	**0.25**	0.29	0.37	0.43	0.20	0.38	0.13	0.21	0.38	0.56	0.63	**0.41**	0.32	**0.68**
	XGBoost	0.36	**0.32**	**0.37**	0.37	0.41	0.28	0.22	0.25	0.28	0.29	**0.40**	0.35	0.13	0.11	0.35	0.44	**0.68**	0.24	0.05	0.63
29	RF	0.24	0.23	0.28	0.32	0.39	0.40	0.11	**0.33**	0.29	**0.49**	0.27	0.22	0.13	0.16	0.31	0.26	**0.66**	0.32	0.05	**0.71**
	SVM	**0.49**	**0.35**	**0.36**	0.54	**0.57**	0.39	**0.28**	**0.31**	0.40	0.45	0.27	**0.49**	0.33	0.11	0.31	0.56	**0.67**	0.35	**0.47**	**0.68**
	XGBoost	0.35	**0.30**	**0.36**	0.40	0.46	0.29	**0.25**	0.28	0.28	0.32	0.33	0.30	0.20	**0.37**	0.35	0.45	**0.67**	0.31	0.05	0.63
30	RF	0.27	0.24	0.30	0.33	0.39	**0.45**	0.12	0.33	0.35	**0.50**	0.13	0.16	0.07	0.05	0.19	0.32	0.60	0.21	0.11	0.58
	SVM	**0.47**	**0.33**	**0.40**	**0.58**	**0.55**	0.40	**0.29**	0.31	**0.46**	0.43	0.20	0.38	**0.40**	0.16	0.35	0.49	0.63	**0.37**	**0.47**	0.65
	XGBoost	0.36	0.30	**0.35**	0.44	0.41	0.32	0.23	0.28	0.33	0.31	0.33	0.24	0.13	**0.32**	0.35	0.43	**0.66**	0.28	0.21	**0.67**
31	RF	0.24	0.26	0.24	0.31	0.38	0.40	0.15	**0.33**	0.37	**0.51**	0.13	0.30	0.13	0.16	0.31	0.35	0.65	0.25	0.21	0.52
	SVM	**0.50**	**0.35**	**0.37**	**0.59**	**0.55**	0.39	**0.30**	**0.32**	**0.46**	0.44	0.20	0.35	0.27	0.11	0.31	0.47	0.64	**0.37**	**0.42**	0.64
	XGBoost	0.36	**0.31**	**0.36**	0.43	0.39	0.30	0.22	0.26	0.34	0.36	0.20	0.22	0.07	0.21	0.35	0.41	0.65	0.30	0.21	0.60
32	RF	0.22	0.25	0.25	0.39	0.37	0.41	0.15	**0.34**	**0.49**	**0.48**	0.07	0.30	0.07	0.21	0.19	0.24	0.63	0.14	0.05	0.54
	SVM	0.44	**0.33**	**0.39**	**0.61**	**0.52**	0.41	**0.29**	**0.33**	**0.51**	0.43	0.27	0.35	0.27	0.26	**0.46**	0.40	**0.67**	0.27	**0.42**	0.64
	XGBoost	0.31	0.27	0.31	0.52	0.39	0.28	**0.25**	0.28	0.39	0.35	0.20	0.27	0.13	0.21	**0.42**	0.26	**0.67**	0.14	0.05	0.64

TABLE II: Layer-wise accuracies of Whisper Large v3 features evaluated on Sep-28k, FluencyBank, and KSoF datasets using different classifiers for stutter detection. Here, PR stands for prolongation, BLK for blocks, SR for sound repetition, WR for word repetition, INJ for interjections, 5CV for 5-cross fold validation, LOPO for leave-one-podcast-out, RF for random forest and SVM for support vector machine.

IV. Method

A. Whisper

Whisper [24] is a robust ASR consisting of a family of sequence-to-sequence transformer [25] based encoder-decoder architecture. Whisper's uniqueness lies not in its architecture, but in its training data and training approach. Its training data consists of 1 million hours weakly labeled data and 4 million hours of pseudolabeled data which is far more larger and more diverse. By default, we use the Whisper Large v3 model unless mentioned otherwise.

B. Understanding the selection of Whisper

There is a significant gap in the availability of a comprehensive stutter dataset that encompasses multilingual stuttered speech, real-world audio samples, and a balanced distribution of various stutter types. Additionally, the consistency in Sep-28k, KSoF and Boli dataset is lacking, making it challenging to combine different datasets into a large, unified corpus. Addressing these gaps would enable better utilization of publicly available stutter data. Our work aims to address some of the aforementioned limitations by utilizing the Whisper model. By leveraging the hidden representations of Whisper, which has been trained on 680,000 hours of multilingual and real-world data, we seek to enhance the effectiveness of stutter detection and analysis. Additionally, the extensive training on 96 languages increases the likelihood of reducing language bias, making the model more robust and adaptable across diverse linguistic contexts [24]. Gong et al. [26] utilized Whisper's representations for audio event classification, achieving promising results across multiple audio-tagging datasets. Building on this, Changawala et al. [27] applied Whisper representations for stutter detection. Inspired by these works, we explore Whisper-encoder representations to investigate whether the middle layers effectively capture stutter characteristics, particularly for stutter detection.

C. Model Architecture

We propose an architecture that utilizes only the encoder component of the transformer, leveraging hidden representations from each Whisper encoder layer for multi-class classification. Specifically, we extract stuttered speech characteristics from the audio encoder of Whisper, which consists of 32 encoded layers, each with a representation of 500 × 1280 per audio file. This architecture captures layer-wise information while employing various train-test split techniques based on the dataset, which includes five-fold cross-validation and LOPO. Additionally, we use Principal Component Analysis (PCA) to reduce dimensionality, preserving 95% variance and selecting upto

Layer	Classifier	PR	BLK	SR	WR	INJ	AVG F1
Test set: FluencyBank							
22	RF	0.31	0.06	0.30	0.12	**0.56**	0.25
	SVM	0.31	0.15	0.21	0.20	0.36	0.33
26	RF	**0.45**	0.04	0.22	0.04	0.37	0.26
	SVM	0.23	0.20	0.16	0.18	0.19	0.33
27	RF	0.36	0.04	**0.41**	0.14	0.41	0.30
	SVM	0.27	0.23	0.21	0.28	0.26	0.38
28	RF	0.40	0.07	0.21	0.13	0.39	0.30
	SVM	0.35	**0.25**	0.24	0.28	0.30	0.40
29	RF	0.27	0.07	0.25	0.15	0.29	0.32
	SVM	0.29	**0.25**	0.22	0.32	0.34	0.41
32	RF	0.39	0.07	0.30	**0.43**	0.36	0.36
	SVM	0.35	0.23	0.30	0.36	0.37	0.44
Test set: KSoF							
19	RF	**0.45**	0.08	0.33	0.03	0.21	0.29
	SVM	0.02	0.05	0.12	0.00	0.06	0.16
20	RF	0.21	0.05	0.26	0.00	**0.33**	0.26
	SVM	0.02	0.06	0.19	0.00	0.08	0.18
26	RF	0.10	0.14	**0.54**	0.00	0.15	0.22
	SVM	0.01	0.05	0.12	0.00	0.03	0.14
31	RF	0.08	**0.22**	0.34	0.00	0.04	0.22
	SVM	0.00	0.05	0.15	0.00	0.03	0.15
32	RF	0.06	0.08	0.25	**0.09**	0.03	0.20
	SVM	0.02	0.08	0.16	0.00	0.03	0.17
Test set: Boli							
15	RF	**0.25**	0.54	0.16	0.04	**0.50**	0.22
	SVM	0.08	0.62	0.22	0.17	0.25	0.24
16	RF	0.18	0.46	0.22	0.22	0.50	0.25
	SVM	0.08	0.63	0.18	**0.39**	0.25	0.25
21	RF	0.18	0.82	0.06	0.13	0.25	0.23
	SVM	0.03	**0.96**	0.02	0.00	0.00	0.11
26	RF	0.10	0.73	**0.24**	0.00	0.13	0.20
	SVM	0.05	0.89	0.07	0.00	0.00	0.13
32	RF	0.15	0.82	0.10	0.13	0.13	0.23
	SVM	0.03	0.94	0.02	0.22	0.00	0.18

TABLE III: Comparison of accuracies for stuttering event detection using RF, SVM, trained on Sep-28k and testing on FluencyBank, KSoF and Boli dataset using layer encoded representations

top 200 features. The overall architecture is illustrated in Figure 1.

Corresponding to every encoder layer, each audio clip has a shape of $C \times R$ (C=500, R=1280), where C represents the temporal dimension and R denotes the hidden feature dimension. To enable efficient classification, we apply temporal pooling to down-sample the temporal representations to 1×1280. This process is performed for every layer, where the representations are then passed through a representation block and subsequently used for classification.

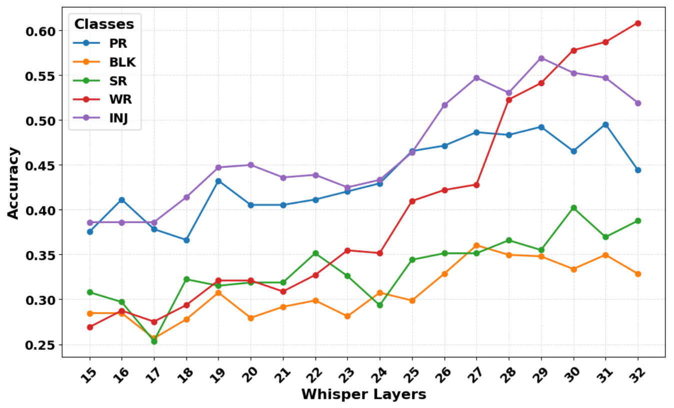

Fig. 2: Layer wise performance of Whisper using SVM on Sep-28k dataset

V. Experiments

This section outlines our experiments in evaluating the Whisper model across different layers. We use FluencyBank, KSoF and Boli dataset for cross-corpus evaluation. LOPO evaluation is conducted on Sep-28k dataset, as it includes information about which clips belong to specific podcasts. Finally, we present our results in a layer-wise analysis, highlighting which stutter types are best captured at different layers. Notably, some stutter types are more effectively represented in the middle layers rather than the final layers, providing insights into the model's internal feature extraction. For experiments, we use the precomputed hidden representations of Whisper model and apply temporal pooling (mean-pooling in temporal dimension) resulting in a tensor of size 1×1280. Additionally, we perform data partitioning in two ways namely 5-cross-fold validation and LOPO. All experiments are designed as multi-class classification tasks. To handle data imbalance effectively, we employ the hybrid sampling to tackle data imbalancing problem similar to [10].

VI. Results & Discussion

The primary objective of this work is not to surpass state-of-the-art performance but rather to explore whether a computationally expensive model like Whisper can be optimized for efficiency. Specifically, we aim to determine whether, among its 32 encoder blocks, certain layers are more effective in capturing different stutter characteristics, allowing us to focus computational resources accordingly. Additionally, we employ machine learning classifiers like random forest (RF), support vector machine (SVM) and XGBoost to further reduce computational overhead, ensuring a more lightweight and efficient model. The results in Table II indicate that the model begins learning stutter characteristics from the middle layers (starting at layer 15). Certain stutter types, such as sound repetition, prolongation and blocks, are best captured in these middle layers. In contrast, word repetition and interjections are primarily recognized in the final layers (30, 31), as Whisper interprets them as filler words or repeated speech segments within the broader sense of treating them like fluent speech. This pattern is consistently observed across all datasets, highlighting how different stutter types are processed at different layers. Notably, in the KSoF dataset, layers 15–29 effectively capture all stutter types, except for prolongation, which is best identified at layer 15 alone. The above observations can be visually interpreted from Figure 2. The cross-corpora testing as shown in Table III, the FluencyBank dataset captures stutter characteristics most effectively from layer 22 onward until the final layer. For KSoF, these characteristics are best represented between layers 19 and 32, while for Boli, they are optimally captured from layers 15 to 32. However, in the KSoF and Boli datasets, certain stutter classes remain undetected by one of the classifiers. This is likely due to multilingual testing in KSoF and variations in speaking styles in Boli, which, as an Indian dataset, reflects diverse linguistic influences.

VII. Conclusion & Future work

We investigated the effectiveness of Whisper's layer representations for classifying different types of stuttered speech. Whisper's exceptional performance in ASR and audio tagging inspired us to explore its layers' potential for stutter detection and classification. Our primary objective was to find out whether Whisper focuses on learning stutter characteristics rather than language representation, particularly in the middle layers. If these layers inherently capture stuttering patterns, they can be leveraged for early stutter detection, reducing the reliance on the final output. This approach allows middle layers to be repurposed for specific stutter classification tasks, eliminating the need for a fully trained ASR system. For future work, we propose a pipeline architecture that focuses solely on the best-performing layers for training. This approach aims to enhance efficiency while improving detection and classification accuracy.

References

[1] E. E. Shriberg, "Preliminaries to a theory of speech disfluencies," Ph.D. dissertation, Citeseer, 1994.

[2] H. Branigan, R. Lickley, and D. McKelvie, "Non-linguistic influences on rates of disfluency in spontaneous speech," in *Proceedings of the 14th International Conference of Phonetic Sciences*. Citeseer, 1999, pp. 387–390.

[3] E. Yairi and N. Ambrose, "Epidemiology of stuttering: 21st century advances," *Journal of fluency disorders*, vol. 38, no. 2, pp. 66–87, 2013.

[4] NHS, "Stammering," 2023, available: https://www.nhs.uk/conditions/stammering/.

[5] V. Mitra *et al.*, "Analysis and tuning of a voice assistant system for dysfluent speech," *arXiv preprint arXiv:2106.11759*, 2021.

[6] N. P. Radio and E. Research, "The smart audio report," 2022, available: https://www.nationalpublicmedia.com/insights/reports/smart-audio-report/.

[7] C. Lea, Z. Huang, J. Narain, L. Tooley, D. Yee, D. T. Tran, P. Georgiou, J. P. Bigham, and L. Findlater, "From user perceptions to technical improvement: Enabling people who stutter to better use speech recognition," in *Proceedings of the 2023 CHI conference on human factors in computing systems*, 2023, pp. 1–16.

[8] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[9] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.

[10] A. Batra, N. K. Sharma, P. K. Das *et al.*, "Boli: A dataset for understanding stuttering experience and analyzing stuttered speech," *arXiv preprint arXiv:2501.15877*, 2025.

[11] A. Batra, P. Shrivastava, and P. K. Das, "Does data balancing impact stutter detection and classification?" in *International Conference on Distributed Computing and Intelligent Technology*. Springer, 2024, pp. 209–224.

[12] S. P. Bayerl, D. Wagner, E. Nöth, T. Bocklet, and K. Riedhammer, "The influence of dataset partitioning on dysfluency detection systems," in *Text, Speech, and Dialogue*, P. Sojka, A. Horák, I. Kopeček, and K. Pala, Eds. Springer International Publishing, 2022, pp. 423–436.

[13] S. P. Bayerl, D. Wagner, E. Nöth, and K. Riedhammer, "Detecting dysfluencies in stuttering therapy using wav2vec 2.0," *arXiv preprint arXiv:2204.03417*, 2022.

[14] S. A. Sheikh, M. Sahidullah, F. Hirsch, and S. Ouni, "Advancing stuttering detection via data augmentation, class-balanced loss and multi-contextual deep learning," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 5, pp. 2553–2564, 2023.

[15] H. Ameer, S. Latif, R. Latif, and S. Mukhtar, "Whisper in focus: Enhancing stuttered speech classification with encoder layer optimization," *arXiv preprint arXiv:2311.05203*, 2023.

[16] S. A. Sheikh, M. Sahidullah, F. Hirsch, and S. Ouni, "Introducing ecapa-tdnn and wav2vec2. 0 embeddings to stuttering detection," *arXiv preprint arXiv:2204.01564*, 2022.

[17] M. Sen, A. Batra, and P. K. Das, "Comparative analysis of classifiers using wav2vec2.0 layer embeddings for imbalanced stuttering datasets," in *2024 First International Conference on Electronics, Communication and Signal Processing (ICECSP)*, 2024, pp. 1–6.

[18] D. Wagner, S. P. Bayerl, I. Baumann, K. Riedhammer, E. Nöth, and T. Bocklet, "Large language models for dysfluency detection in stuttered speech," *arXiv preprint arXiv:2406.11025*, 2024.

[19] M. Teleki, X. Dong, S. Kim, and J. Caverlee, "Comparing asr systems in the context of speech disfluencies," in *Proc. Interspeech 2024*, 2024, pp. 4548–4552.

[20] G. Cloud, "Speech-to-text: Automatic speech recognition," 2024, available: https://cloud.google.com/speech-to-text.

[21] C. Lea *et al.*, "Sep-28k: A dataset for stuttering event detection from podcasts with people who stutter," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6798–6802.

[22] N. B. Ratner and B. MacWhinney, "Fluency bank: A new resource for fluency research and practice," *Journal of Fluency Disorders*, vol. 56, pp. 69–80, 2018.

[23] S. P. Bayerl *et al.*, "Ksof: The kassel state of fluency dataset–a therapy centered dataset of stuttering," *arXiv preprint arXiv:2203.05383*, 2022.

[24] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. Intl. Conf. Machine Learning (ICML)*, 2023, pp. 28 492–28 518.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[26] Y. Gong, S. Khurana, L. Karlinsky, and J. Glass, "Whisper-at: Noise-robust automatic speech recognizers are also strong general audio event taggers," *arXiv preprint arXiv:2307.03183*, 2023.

[27] V. Changawala and F. Rudzicz, "Whister: Using whisper's representations for stuttering detection," in *Interspeech*, vol. 2024, 2024, pp. 897–901.