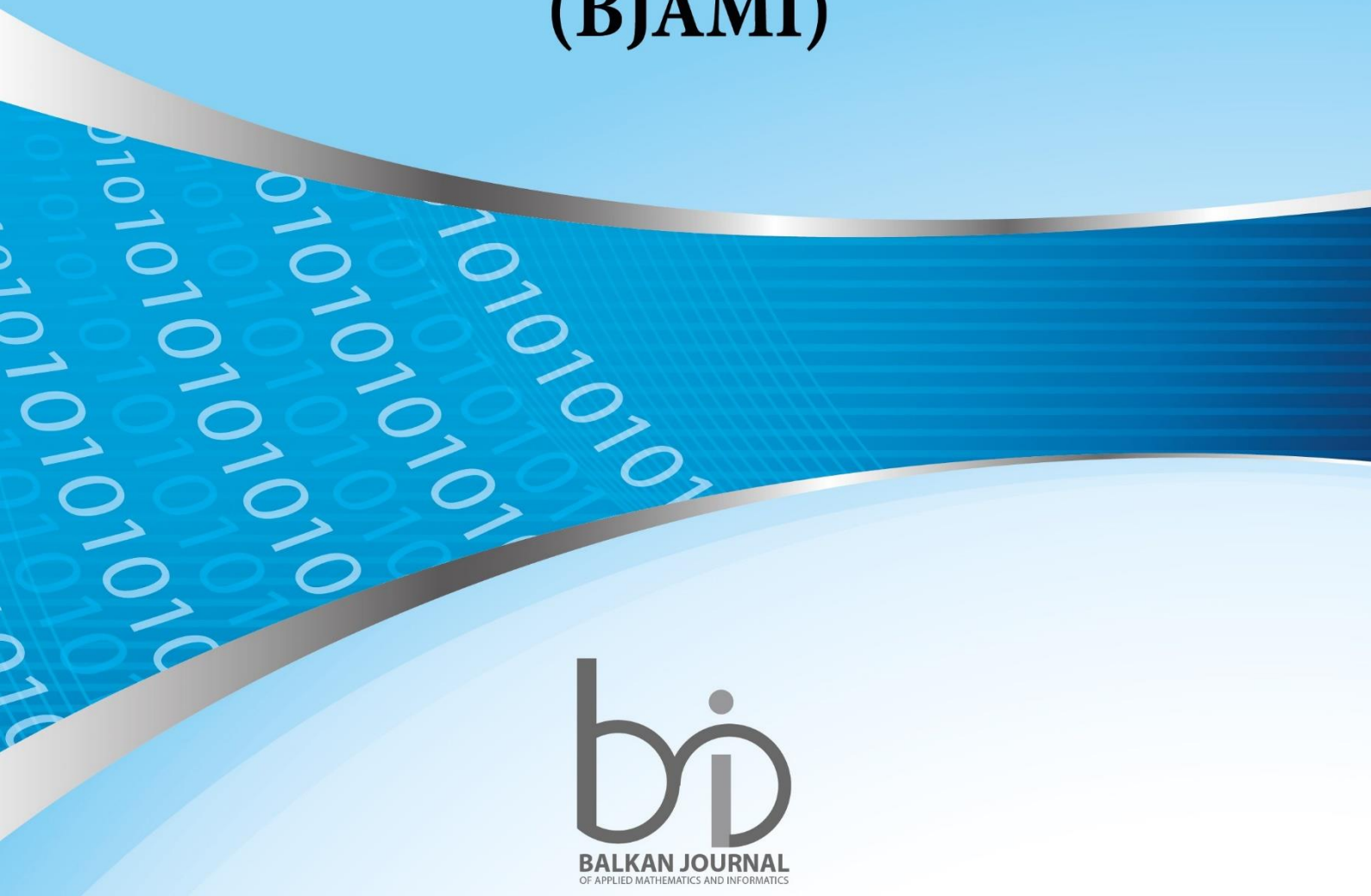
GOCE DELCEV UNIVERSITY - STIP
FACULTY OF COMPUTER SCIENCE

The journal is indexed in
EBSCO

ISSN 2545-4803 on line
DOI: 10.46763/BJAMI

BALKAN JOURNAL OF APPLIED MATHEMATICS AND INFORMATICS (BJAMI)



BALKAN JOURNAL
OF APPLIED MATHEMATICS AND INFORMATICS

YEAR 2025

VOLUME VIII, Number 1

AIMS AND SCOPE:
BJAMI publishes original research articles in the areas of applied mathematics and informatics.

Topics:
1. Computer science;
2. Computer and software engineering;
3. Information technology;
4. Computer security;
5. Electrical engineering;
6. Telecommunication;
7. Mathematics and its applications;
8. Articles of interdisciplinary of computer and information sciences with education, economics, environmental, health, and engineering.

BALKAN JOURNAL
OF APPLIED MATHEMATICS AND INFORMATICS
(BJAMI), Vol 8

ISSN 2545-4803 online
Vol. 8, No. 1, Year 2025

EDITORIAL BOARD

C O N T E N T

Balkan Journal of Applied Mathematics and Informatics Online ISSN 2545-4803
Volume VIII Number 1 Year 2025 UDC: 004.8:004.934'1]:004.77.031.4

INTEGRATING AI AND CLOUD COMPUTING FOR EFFICIENT AUDIO ANALYSIS

ALEKSANDAR KOTEVSKI

Abstract. This paper proposes a pipeline for sentiment analysis of audio inputs by using AI and techniques for speech recognition. A key aspect of our approach involves determining audio chunks based on silence detection and decibel height, allowing for more effective segmentation and analysis of the input data. These techniques will be instrumental in categorizing audio materials, enabling more organized and accessible content management. In the scope of this paper, we will compare and utilize several Python packages for audio detection, including Librosa, PyDub, and SoundFile, which offer various functionalities for audio analysis and manipulation. Additionally, we discuss the high computational demands associated with training and deploying of models, highlighting the potential cost barriers for many organizations. To address these challenges, we propose the use of cloud-based solutions, specifically AWS Spot Instances, Azure Spot VMs, and Google Cloud Preemptible VMs, which offer substantial cost savings for processing audio data. By leveraging these cloud resources, organizations can significantly reduce expenses while maintaining high performance in AI audio processing tasks.

1. Introduction

Audio processing plays a vital role in many AI applications, such as virtual assistants, automated transcription services and audio classification systems. Traditional speech recognition methods often depend on handcrafted features and rule-based systems, which can struggle with flexibility and precision. On the other hand, AI techniques have shown an impressive ability to learn and recognize patterns directly from raw audio data, resulting in substantial performance enhancements. In a world inundated with a lot of media content, from music and podcasts to audiobooks and environmental sounds, efficient categorization and analysis allows users to easily navigate and discover relevant materials. Accurate media categorization enhances user experiences in various applications, such as music streaming services, where users rely on personalized playlists and recommendations, or content moderation systems, which require effective filtering of inappropriate audio. By organizing audio content effectively, organizations can significantly improve search ability and relevance, ultimately driving better engagement and satisfaction among users. Furthermore, the advent of cloud computing, particularly the use of spot instances, offers a cost-effective solution for processing audio data. Spot instances provide access to unused cloud capacity at a fraction of the cost of standard instances, enabling users to take advantage of significant savings, often up to 90%. This economic model allows businesses to train complex models without incurring high operational costs, making it feasible to leverage advanced AI techniques for large-scale audio analysis. The flexibility and scalability of cloud resources enable

organizations to adapt quickly to changing workloads, ensuring they can meet demand without overspending.

Importantly, the methodologies employed in audio categorization can also be effectively extended to audio processing. The integration of audio analysis opens new avenues for applications in security surveillance, media archiving, and content creation, providing a comprehensive approach to multimedia data management. It is a critical component of numerous AI applications, including virtual assistants, automated transcription services, and audio classification systems. Conventional approaches to audio recognition typically depend on manually designed features and rule-based algorithms, which can restrict their flexibility and precision. In contrast, artificial neural networks have demonstrated remarkable proficiency in learning complex patterns from raw audio data, leading to significant improvements in performance.

2. Related papers

In recent years, audio processing, speech recognition and sentiment analysis have become increasingly popular research topics, leading to a lot of published papers focused on various methods and technologies aimed at improving these processes. A particularly exciting development is the role of cloud computing, which provides scalable resources and robust processing power, enhancing the effectiveness of audio recognition systems. By utilizing cloud infrastructure, researchers can tap into extensive datasets, foster collaboration, and implement advanced models that enable real-time transcription and recognition across a wide range of applications.

In [1], the authors analyze the concept of NLP (Natural Language Processing) with speech recognition and the most accurate technique which can achieve the best results. They did a comparative analysis to indicate the difference and demerit points of various speech recognitions. The authors in [2] investigate a method for extracting lexical knowledge from video transcriptions in the MuSe-CAR dataset. By utilizing SenticNet, the authors extract natural language concepts and fine-tune various features on a subset of the data. They analyze video content and predict emotional valence, arousal, and speaker topics. The paper [3] introduces three innovative language modeling techniques for spoken dialog systems that leverage semantic analysis: concept sequence modeling, two-level semantic-lexical modeling, and joint semantic-lexical modeling. These methods integrate lexical information with varying levels of semantic data, utilizing annotations from either a shallow or full hierarchical semantic parser. The proposed models show improved recognition accuracy compared to traditional word and class N-gram models across three task domains. The authors in [4] present a novel method of semantic modeling for speech recognition that shares similarities with latent semantic analysis but achieves better experimental results. The effectiveness of the method is measured by the percentage of correctly recognized sentences from a corpus. A key distinction is the selection of related topics that influence a matrix representing the probabilities of words appearing within those topics, leading to improved performance in recognition accuracy. The study [5] explores the effectiveness of an emotional chatbot, facial expressions, images, and social media texts in emotion detection, utilizing the PRISMA methodology for a comprehensive review of existing surveys. It

highlights various machine learning techniques commonly employed for emotion extraction, including Support Vector Machines (SVM), Naïve Bayes (NB), Random Forests (RF), Recurrent Neural Networks (RNN), and Logistic Regression (LR). In [6], the authors examined the semantic features of the simulated mini lectures in the listening sections of the International English Language Testing System (IELTS) and the Test of English as a Foreign Language (TOEFL) based on automatized semantic analysis to explore the content validity of the two tests. The paper [7] introduces a method called TeminAL to enhance multi-modal audio-language models (ALMs) with temporal understanding while preserving their existing capabilities. The approach involves a two-stage training process: TeminAL A, where the model learns to differentiate between various sounds, and TeminAL B, which instills a sense of time to improve temporal understanding. This method yields an average performance increase of 5.28% in temporal understanding on the ESC-50 dataset, while maintaining competitive performance in zero-shot retrieval and classification tasks on the AudioCap and Clotho datasets. In [8], the authors propose a baseline model called DocWhisper, which leverages the textual content from slides to enhance its transcription accuracy. The effectiveness of DocWhisper is validated using the SlideAVSR dataset, showcasing its ability to improve the AVSR performance in contexts rich in specialized language.

The paper [9] delves into several research challenges and future directions in the field of cyber-threat detection, highlighting key issues such as interpretability, scalability, and adaptability to evolving threats. The conclusion synthesizes the findings, emphasizing the pivotal role of Transformers and Large Language Models (LLMs) in enhancing cyber-threat detection capabilities, and identifies potential research and development paths to further improve these technologies, ensuring they remain effective against future challenges in cyber security. The research [10] introduces a novel concept for a remote audio processing network utilizing a small embedded system network. The implemented system aims to reduce hardware costs by employing low-grade 8-bit microcontroller recording units at the end nodes, which connect to a more powerful embedded system server capable of performing local signal processing. This approach balances cost efficiency with effective audio processing capabilities. In [11], the authors explore the theory that a key advantage of using speech recognition systems is their ability to allow users to engage in multiple tasks simultaneously. By utilizing voice input commands, users can direct their attention to observation and manual activities while still maintaining control of their devices. The study also reveals that voice recognition serves as a specific type of speech recognition. The paper [12] presents the development of an intelligent deep learning-based speech processing algorithm implemented on a quad copter, aimed at simplifying UAV control. As per the authors, the proposed algorithm has the potential for various applications beyond drone operation, such as automated data entry in ATMs and vending machines, home and office automation, speech-controlled vehicle navigation, and wheelchair operation. Another paper [13] offers a comprehensive overview of the literature on speaker recognition utilizing cloud computing. It examines various techniques, architectures, and challenges involved in implementing speaker recognition systems in the cloud environment. The objective of the study [14] was to assess students' educational and

training needs in audio processing as future journalists. The proposed model integrates audio capturing and intelligent content analysis tasks within a cloud environment. It emphasizes the adoption of well-designed strategies and a user-friendly graphical user interface (GUI) to minimize memory and computational demands, ensuring unrestricted access via smartphones, PDAs, tablets, laptops, or personal computers. The study [15] evaluates various automatic speech recognition (ASR) systems, including OpenAI's Whisper, to assess their performance with forensic-like audio. The findings indicate that transcription quality for high-quality audio files reached optimal levels, with some systems producing no errors. However, for the poor-quality forensic-like audio, Whisper emerged as the best-performing system but achieved only 50% accuracy in transcribing the entire speech material. The authors in paper [16] present a proof-of-concept exploration using three examples of automated transcription of audio recordings from diverse contexts: an interview, a public hearing, and a classroom setting. It compares the results against traditional manual transcription techniques for each case. The paper begins with an overview of the literature on automated captioning and the application of voice recognition tools in transcription. It then details the specific processes and tools utilized to generate automated captions, followed by a basic processing of these captions to produce automated transcripts. This exploration aims to highlight the effectiveness and limitations of automated transcription in various scenarios.

3. Challenges and limitations

Audio processing is a critical component of numerous AI applications, including virtual assistants, automated transcription services, and audio classification systems. In contrast, artificial neural networks have demonstrated remarkable proficiency in learning complex patterns from raw audio data, leading to significant improvements in performance.

During the implementation of the techniques for audio processing and speech recognition, several key challenges arise, particularly when dealing with large models, costs, and robustness. One of the primary concerns is the complexity and size of these models, which require significant computational resources for both training and inference. This often means relying on specialized hardware, like GPUs or TPUs, and can lead to high memory usage, making it difficult to deploy these models on smaller devices such as smartphones or IoT gadgets. Additionally, training these large models can be a time-consuming process, which slows down experimentation and development. Financial considerations also play a big role. The need for high-performance computing infrastructure can be quite costly, especially for organizations without established resources. Energy consumption is another factor, as large models typically use more power, leading to higher operational expenses and raising environmental concerns. Moreover, gathering and labeling the extensive datasets required for training can be both expensive and labor-intensive.

Robustness and generalization present further challenges. Large models can easily overfit the data they are trained on, resulting in poor performance on new and unseen data, especially in varied audio environments. They also need to be resilient against adversarial attacks, where subtle modifications to audio inputs can trick the model.

Additionally, these systems must handle variations in audio quality, background noise, and differences between speakers.

Very often, real-time processing of audio is mandatory. This is a challenging approach, particularly in low-latency situations like live speech recognition. Optimizing model architectures and using techniques like knowledge distillation could improve inference speed for these applications. Many models also struggle in noisy environments, where they fail to generalize well to variations in background noise. Incorporating noise-robust training methods, such as data augmentation and adversarial training, could enhance resilience in adverse conditions.

Furthermore, it is crucial to consider the ethical and social implications of using large models. They can inadvertently reinforce biases found in training data, leading to unfair outcomes. The complexity of these models can make it challenging to understand their decision-making processes, raising important questions about accountability and trust.

Addressing these limitations and focusing on suggested improvements could significantly enhance the capabilities of AI models in audio processing and recognition. Ongoing research and innovation in these areas are crucial for advancing the field and meeting the increasing demands of real-world applications.

4. Comparative analysis

There is a wide variety of Python packages developed for speech recognition and audio processing, providing powerful tools for developers to work with sound and voice data. These libraries offer functionalities for converting speech to text, analyzing audio signals, manipulating sound files, and performing advanced audio processing tasks. With such a rich ecosystem of resources, building applications that can understand and process audio has become more accessible than ever. After conducting a thorough investigation, a detailed comparison was compiled and it is presented in Table 1, featuring the most used packages to help users make informed decisions about which tools best suit their needs.

Table 1. *Python packages for audio processing*

Library	Accuracy	Language Support	Features
SpeechRecognition	0.75 - 0.90	Multiple languages	Supports Google, CMU Sphinx, etc.
Pydub	N/A	N/A	Easy audio file manipulation
Mozilla DeepSpeech	0.85 - 0.95	English	Pre-trained models available
Google Cloud Speech-to-Text	0.90 - 0.95	Multiple languages	Real-time transcription, word hints
Vosk	0.85 - 0.90	Multiple languages	Lightweight, low resource usage
Kaldi	0.85 - 0.95	Multiple languages	Highly customizable, powerful
PocketSphinx	0.70 - 0.80	English	Designed for embedded systems
Wit.ai	0.85 - 0.90	Multiple languages	Easy integration with apps

AssemblyAI	0.85 - 0.95	Multiple languages	Real-time and batch processing
SpeechBrain	0.85 - 0.95	Multiple languages	Supports multiple tasks (ASR, TTS)

Table 1 shows a comparative analysis of the top 10 Python libraries for speech recognition and provides a comprehensive overview of the strengths and weaknesses of each library, highlighting their suitability for various use cases. In this context, SpeechRecognition stands out for its ease of use and broad compatibility with various speech engines, making it an excellent choice for beginners. Mozilla DeepSpeech and Kaldi are favored for their advanced capabilities and higher accuracy, particularly in complex applications that require robust performance. The comparison also reveals the trade-offs between accuracy, processing speed, and resource requirements. Some libraries excel in accuracy but may demand more computational power, while others prioritize lightweight performance, making them ideal for real-time applications or environments with limited resources. Libraries like Kaldi and Vosk not only offer strong performance but also benefit from active communities that can provide support and resources for troubleshooting. Because our proposed pipeline contains dedicated service for semantic analysis, we investigated several of them. Table 2 shows a short overview and comparison of the most used models on HuggingFace for sentiment analysis [17] which highlights the features of the most used fine-trained models for natural language processing tasks

Table 2. *HugginFace model for semantic analysis*

Model	Precision	Recall	Adequate Text Size
BERT	85-92%	0.85-0.92	Short to medium texts
RoBERTa	86-93%	0.86-0.93	Short to medium texts
DistilBERT	80-90%	0.80-0.90	Short to medium texts
XLNet	85-92%	0.85-0.92	Short to medium texts
ALBERT	90%	0.90	Short to medium texts
T5	Varies	Varies	Short to long texts
ELECTRA	90-92%	0.90-0.92	Short to medium texts
BART	High	High	Short to long texts
CamemBERT	90%	0.90	Short to medium texts
DeBERTa	92%	0.92	Short to medium texts

Models like BERT and RoBERTa demonstrate high precision and recall, making them ideal for tasks requiring deep understanding of context and nuance. Models such as DistilBERT offer a lighter alternative, maintaining good performance while being more efficient in terms of resource consumption. This is particularly beneficial for

applications needing faster inference times or operating in resource-constrained environments.

Table 2 also illustrates the importance of input size limitations for each model. Additionally, the metrics of precision and recall provide a clearer picture of each model's reliability and effectiveness in real-world scenarios. While high precision indicates the accuracy of positive predictions, high recall reflects the model's ability to capture relevant instances. Balancing these metrics is essential for tasks where both false positives and false negatives can significantly impact outcomes.

5. Practical implementation

Sentiment analysis, powered by AI and advanced speech recognition techniques, is transforming various sectors by providing insights into emotional responses expressed through audio and video inputs. This innovative approach enables organizations to better understand user experiences, engagement levels, and emotional states, leading to enhanced decision-making and improved outcomes. There are a lot of areas where these techniques can be implemented. In that context, in e-learning platforms, sentiment analysis can monitor student engagement during video lectures. For instance, by analyzing students' facial expressions and vocal tones, educators can assess whether learners are confused, bored, or engaged. This data can help instructors adapt their teaching styles and materials in real time, improving overall student satisfaction and comprehension. In online media, sentiment analysis can enhance content recommendations. For example, streaming services can analyze user comments and reactions in video reviews or social media discussions. By understanding viewers' sentiments toward certain genres or shows, platforms can suggest similar content that aligns with viewers' preferences, thereby improving user engagement and retention. In e-commerce, sentiment analysis can be applied to customer feedback and product reviews. Analyzing audio and video reviews allows companies to understand customer sentiments about specific products, identifying strengths and weaknesses. This insight can inform marketing strategies, product development, and customer service improvements to enhance the overall shopping experience. In e-health, sentiment analysis can be used to monitor patients' emotional states during the health consultations. By analyzing audio and video inputs, healthcare providers can gain insights into patients' mental health and emotional well-being. This information could help doctors tailor their communication and treatment approaches, leading to more personalized care and improved patient outcomes.

6. Pipeline for audio sentiment analysis using cloud computing

In the scope of this paper, we propose a pipeline for performing sentiment analysis on audio materials, by using techniques for speech recognition, transcription and sentiment analysis, to categorize the audio input. To accomplish that, the proposed pipeline uses several processing units, separated into a few groups. As soon as the processing request comes, in the initial phase the pipeline is storing input data in the temp folder. This means that in this phase, the pipeline can save audio samples from

streams. Audio input will be split into smaller chunks to be able to handle more chunks in parallel by more processing instances at the same time. We are using Python packageSpeechRecognition for splitting the audio input stream to smaller chunks, based on silence detection techniques and decibel values. Dedicated background job for monitoring is running in the background and checks if there are new audio chunks in the temp folder which are ready for processing. If any, then the background job will send a request to the service to calculate and run the required number of spot instances. To determine the number of required Amazon Web Services (AWS) EC2 Spot Instances for transcription of the audio files, we need to estimate the average processing time per audio file based on its length and the model's complexity. This can be calculated using Formula 1.1:

$$processingTime = baseTime + (audioLenght \times extraTime) \quad (1.1)$$

The variable processingTime is the fixed processing time for each file, and extraTime represents the additional time needed based on the audio length. Next, the total processing time (time required for transcription) should be calculated by multiplying the average processing time per file by the total number of files you need to process.

In the next step, we should calculate the processing capacity of the Spot Instances by determining how many files can be processed simultaneously by one instance. This capacity will depend on the number of GPU cores of the instance and the number of files each core can handle concurrently. Finally, the number of required Spot Instances is calculated by dividing the total number of files by the number of files that can be processed simultaneously per instance.

$$numberInstance = \frac{numberOfFiles}{instanceProcessingCapacity} \quad (1.2)$$

In formula 1.2, the variable numberInstance is the number of required spot instances that should be used for processing the audio, numberOfFiles is the total number of files that should be processed and instanceProcessingCapacity indicates the number of files that can be processed per instance. It can be calculated as:

$$fileProcessedPerInstance = \frac{totalProcessingCapacity}{processingTimePerFile} \quad (1.3)$$

For example, if the processing time per file is 8 seconds, and there are 100 files that need to be processed, the total processing time would be 800 seconds. If each instance can process 8 files at a time (given that it has 8 GPU cores), the pipeline will activate approximately 13 Spot Instances to handle the workload efficiently. The main purpose of each spot instance is to get a single chunk from the temp folder, to call the service for speech recognition, perform transcription and store the transcription in the database. Then, a dedicated service for verifying the content will be called. If the transcript data is correct, then we remove the chunk from the temp folder.

After a comparative analysis, we decided to use Whisper because of its high transcription accuracy, multilingual support, and strong performance in noisy environments. To avoid concurrent access to the same chunk, the flow relocates the file in sub-working directory and reads the content from there. After processing the chunk, the active spot instance checks if there are more chunks for processing. If any, the instance starts with processing the next chunk, otherwise it will be in idle state for 5

minutes. A separate service for spot instance management will scan the instances to check its status. If some spot instance is idle for more than 5 minutes, the service will turn off the instance. When all chunks from the initial input are processed, the pipeline runs the next step - to do a semantic analysis of the transcription. In the final step, the semantic analysis result is stored in the database. The proposed pipeline is shown in Figure 1.

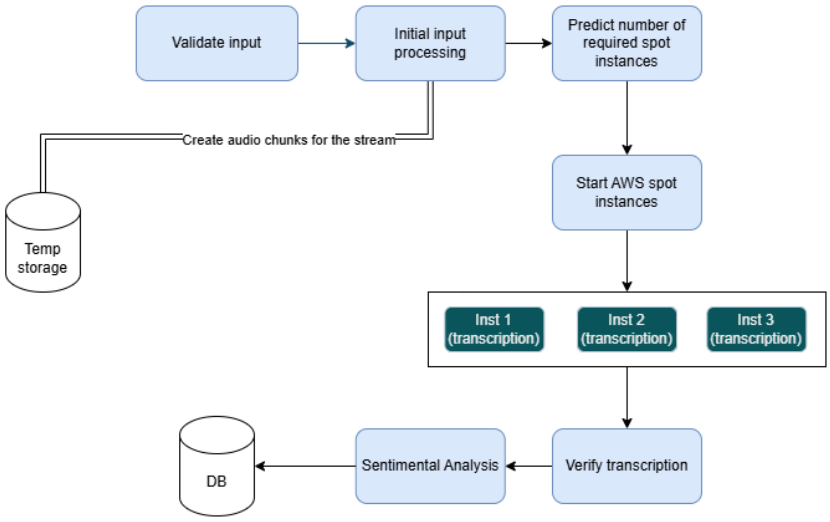


Figure 1. *Pipeline for audio analysis*

7. Results

In the scope of this paper, testing of the proposed pipeline was performed. We used a set of 50 audio files, with different lengths and different audio quality. Because the focus of this paper is to present how cloud can affect the audio recognition process, different AWS Spot Instance types were used. We chose Amazon EC2 Spot instances as processing instances for the transcription, mainly because they provide a great mechanism for scalability, allowing scaling of the computing capacity which is important for processing large numbers of chunks in a single batch job. Additionally, Spot Instances offer flexibility, enabling users to request a variety of instance types and sizes, optimizing their workload based on specific requirements such as memory and CPU.

The results shown in Table 3 reveal some interesting insights into how various configurations impact efficiency and cost-effectiveness.

Table 3. *Statistics of using AWS Spot instances for audio analysis*

Spot Instance Type	GPU Cores	Number of Instances	Average Length of Audio (min)	Processing Time per File (s)	Total Processing Time (s)	Files Processed Simultaneously	Required Instances

Type A	2	3	1	6	300	6	8
Type B	4	2	5	10	500	8	7
Type C	6	2	10	15	750	12	5
Type D	8	1	15	20	1000	8	7
Type E	8	1	20	25	1250	8	8
Type F	4	4	3	7	350	16	5
Type G	6	3	8	12	600	18	5
Type H	10	2	12	18	900	20	5

From the results in Table 3, using instances with fewer GPU cores, like Type A and Type F, can be a smart choice for processing shorter audio files. These setups allow for a higher number of instances, which help in managing costs while still getting the job done effectively. As the average length of audio files increases, it becomes clear that more powerful instances are needed. For instance, Type C and Type H, with their greater number of GPU cores, are well-suited for handling longer files while keeping processing times manageable. This highlights the importance of scalability in your computing choices. The processing capacity of each instance type also plays a crucial role. Instances with more GPU cores, such as Type G and Type H, can process more files at the same time, significantly reducing the total time needed to complete the workload. This is particularly beneficial when dealing with larger batches of longer files, where minimizing downtime is essential. Overall, the required number of Spot Instances varies greatly depending on the length of the audio files and the type of instance used. For example, Type C manages to keep the number of required instances low while effectively handling longer files, whereas Type F relies on a higher instance count to quickly process shorter files.

8. Possible pipeline improvements

There are several approaches which can be implemented in the proposed flows, to enhance the processing of audio chunks. In that context, RabbitMQ can be implemented - robust message broker that facilitates communication between different parts of the pipeline, allowing them to work together seamlessly. By using RabbitMQ, we can efficiently manage the flow of audio data as it gets processed in chunks. This means that as audio files are transcribed and analyzed, each chunk could be sent to different processing units without overwhelming any single component. This distributed approach helps to balance the load and ensures that no processing unit becomes a bottleneck. With other words, integrating RabbitMQ into the proposed pipeline can lead to a more efficient, stable, and resilient system, making it easier to handle large volumes of data while minimizing the risk of errors or delays. Because these services require a lot of hardware resources, improving computational efficiency is crucial. Research into model compression techniques, such as pruning, quantization, and knowledge

distillation, can reduce the computational demands of deep learning models. This would make the models more suitable for deployment in resource-constrained environments. In addition, strengthening real-time processing capabilities is important. Optimizing model architectures for faster inference times and lower latency is necessary for applications such as live speech recognition. Lightweight models and efficient architecture can ensure effective performance in real-time scenarios. When dealing with audio inputs, the pipeline should be able to process the audio materials with some noise. Incorporating robust training techniques that simulate various noise conditions during training can help models become more resilient to real-world acoustic variations. Adversarial training and the use of diverse datasets can further enhance the robustness. By focusing on these areas and implementing some of them, the proposed pipeline can be more efficient.

9. Future discussion and research

In the future, it would be interesting to explore how well transcription models work with audio text in different languages. By testing these models with various languages, we can learn how effective they are in understanding and transcribing speech. We can also use cloud platforms such as Microsoft Azure and Google Cloud Platform to process audio files and compare their transcription results. Looking at factors such as speed, accuracy, and cost will help us figure out which platform performs best for different languages. Additionally, we should consider comparing Amazon's CPU instances with its GPU instances. While GPU instances are often faster and more powerful for tasks such as audio transcription, it would be helpful to see how CPU instances stack up in terms of performance and cost. Overall, future research could focus on how transcription models handle different languages, comparing the processing abilities of Microsoft Azure and Google Cloud Platform and looking at the differences between CPU and GPU instances on AWS. These discussions could lead to important discoveries in audio processing and transcription. Future research about sentiment analysis in different fields can delve into a variety of interesting topics. One key area is the exciting advancements in technology. We could explore how new AI models and improved speech recognition tools make the sentiment analysis more accurate and efficient. There is also potential in combining different types of data, such as audio, video, and text, to gain deeper insights.

Ethical considerations will be another important discussion point. Because sentiment analysis can be implemented anywhere, especially in sensitive areas such as healthcare, we need to think about issues like data privacy, informed consent, and the risk of bias in AI systems. It is crucial to ensure that these technologies are used responsibly and transparently.

10. Conclusion

This paper aims to demonstrate the synergy between the technologies for audio processing, speech recognition, sentiment analysis and cost-effective cloud computing strategies, which produce scalable audio recognition solutions. The paper focuses on clarifying the role of AI in audio recognition, providing a detailed overview of methodologies and practical implementation. We suggested a pipeline which can

improve audio analysis, with speech recognition and semantic analysis techniques. By using advanced speech recognition technology, we can convert spoken content into text, making it easier to analyze and understand what is being said. This combination not only helps in categorizing audio materials accurately but also opens the possibility of suggesting relevant content in the future based on what has been analyzed. Additionally, leveraging AWS EC2 Spot Instances helps to tackle the challenges of high hardware costs and availability issues. With these instances, organizations can access powerful computing resources at a fraction of the price, eliminating the need for hefty upfront investments in infrastructure. This flexibility allows for quick scaling, enabling users to manage large volumes of audio data as needed while keeping expenses in check. In summary, the integration of speech recognition, semantic analysis, and Spot Instances forms a comprehensive framework for efficient audio materials. This approach not only streamlines operations but also empowers organizations to draw valuable insights from their audio content, all without the financial burden of expensive hardware.

References

[1] *R. Pahwa, H. Tanwar, S. Sharma*. (2023). Speech Recognition System: A review, International Journal of Future Generation Communication and Networking 13(3):2547-2559

[2] *S.Lukas, A. Cambria, E.Schuller*. (2021). Sentiment Analysis and Topic Recognition in Video Transcriptions. IEEE Intelligent Systems. 36. 10.1109/MIS.2021.3062200

[3] *H. Erdogan, R. Sarikaya, S. Chen, Y. Gao and M. Picheny*. (2022). Using semantic analysis to improve speech recognition performance. Computer Speech & Language Volume 19, Issue 3, July 2022, Pages 321-343

[4] *B. Ziolko, S.Wilson*. (2008). Semantic modelling for speech recognition.

[5] *T. Suryakant, S.Sandor.* (2023). Semantic speech analysis using machine learning and deep learning techniques: a comprehensive review. Multimedia Tools and Applications. 83. 1-30. 10.1007/s11042-023-17769-6.

[6] *Z. Yufan, V.Aryadoust.*(2024).Anautomatized semantic analysis of two large-scale listening tests: A corpus-based study. Language Testing, 0(0). https://doi.org/10.1177/02655322241288598

[7] *A. Sinha, C. Migozzi, A. Rey, C. Zhang*. (2024). Enhancing Audio-Language Models through Self-Supervised Post-Training with Text-Audio Pairs, arXiv:2408.092

[8] *H. Wang, S. Kurita, S. Shimizu, D. Kawahara.*(2024). SlideAVSR: A Datasetof Paper Explanation Videos for Audio-Visual Speech Recognition. Proceedings of the 3rd Workshop on Advances in Language and Vision Research (ALVR)

[9] *H. Kheddar*.(2024).TransformersandLargeLanguage Models for Efficient Intrusion Detection Systems: A Comprehensive Survey. https://doi.org/10.48550/arXiv.2405.04760

[10] *S. Bruno, A. Mohammad, Y.Yuuki.* (2017). Implementation of a Cloud Processing Based Voice Communication and Noise Reduction Embedded System Network.

[11] *B. Anusha, M. Mounika.* (2019). Voice Recognition Systems in the Cloud Networks: Has It Reached Its Full Potential. Asian Journal of Applied Science and Engineering. 8. 51-60. 10.18034/ajase.v8i1.12.

[12] *M.A.Ahmed*.2021.Design and Development of Audio Processing and Speech Recognition Algorithm. Seventh International Conference on Aerospace Science and Engineering (ICASE).

[13] *M. Hanafy.* (2024). Speaker Recognition using Cloud Computing: A Review. Journal of Communication Sciences and Information Technology (JCSIT)

[14] *S. Efstathios,K. Rigas, D. Charalampos, V.Andreas.* (2024). Audio Semantic and Intelligent Processing Concepts in the New Media Environment: A Cloud Computing Model.

[15] *L.Debbie.* (2024). Automatic speech recognition and the transcription of indistinct forensic audio: how do the new generation of systems fare. Frontiers in Communication. 9. 1-9. 10.3389/fcomm.2024.1281407/full

[16] *B.Christian, D.Christopher.* (2018). Automated generation of "good enough" transcripts as a first step to transcription of audio-recorded data. Methodological Innovations. 11. 10.1177/2059799118790743

[17] https://huggingface.co/models?other=sentiment-analysis

Aleksandar Kotevski, PhD
Republic of North Macedonia
E-mail address: kotevski.ace@gmail.com