Scene4M: A Multimodal Scene Model

Lara Marinov (403901), Christian Egeland (404170), Anthonin Duval (355584), Kenza Driss (344206)
COM-304 Final Project Report, Spring 2025

***Abstract*—For individuals with impaired vision, navigating their surroundings independently can be a daily struggle. Traditional assistive technologies, which rely on isolated sensory cues, may not fully capture the complexity of real-time environments. This paper presents Scene4M, an extended version of the 4M multimodal model, that combines the current 4M modalities with new video and audio data to provide environment descriptions. We developed custom VQ-VAE tokenizers for both audio and video, showing moderate reconstruction quality, though results were not consistently accurate. The full multimodal training pipeline demonstrated good loss convergence, but generated scene descriptions were often noisy and lacked coherence. A transfer evaluation study using frame-wise captions summarized into video-level descriptions produced weak outputs, highlighting the limitations of untrained modalities. Overall, while the system components show promise, results suggest that stronger tokenizers, more training time, and deeper integration of new modalities are necessary for robust performance. Our website is at: https://www.andrew.cmu.edu/user/lmarinov/scene4m**

I. Introduction

Navigating the physical world presents daily challenges for individuals with visual impairments, limiting independent mobility and access to public spaces. Traditional assistive technologies—such as tactile canes, GPS-based audio navigation, or wearable sensors—typically provide isolated and low-resolution sensory feedback. While useful, these systems often lack the contextual awareness and adaptability required to interpret complex, dynamic real-world environments, such as busy streets or unfamiliar urban settings.

Advances in multimodal learning have introduced promising new approaches to enhance navigation systems by fusing audio and video inputs to capture a richer representation of the surrounding environment. By integrating visual and auditory cues, such systems have the potential to recognize contextual elements (e.g., approaching vehicles, people, or background sounds), enabling more accurate and adaptive guidance. However, constructing such a system introduces multiple technical challenges including aligning video, audio, and text data streams, producing discrete token representations, and extending on top of an existing multimodal system.

Scene4M addresses these challenges by developing a prototype multimodal system that jointly processes audio and video data to generate textual scene descriptions. By building a system that fuses multimodal input into interpretable scene descriptions, this work lays foundational steps toward future assistive technologies capable of real-time, context-aware navigation for visually impaired users. The method emphasizes modularity, scalability, and the ability to extend to additional modalities or downstream tasks such as directional instructions. The focus on aligning rich audiovisual input with natural language and the other existing 4M [1] image modalities provides a flexible interface for human-understandable guidance and opens pathways for further research into accessible AI-driven navigation systems.

II. Recent Related Work

Recent developments in multimodal learning have advanced the capabilities of models to understand complex visual scenes. VideoLLaMA 3, introduced earlier this year, significantly improves video understanding by focusing on vision-centric processing of dynamic environments [2]. Although it offers strong performance in visual tasks, its architecture remains primarily focused on visual modality. The authors note: "the core design philosophy of VideoLLaMA 3 is vision-centric." In contrast, Scene4M integrates both video and audio modalities, enabling richer and more comprehensive scene understanding that better reflects the multisensory nature of real-world environments.

Other work has explored the integration of audio, vision, and language for improving performance across a range of multimodal tasks. The VALOR model jointly models vision, audio, and text data to achieve competitive results in tasks such as captioning, retrieval, and question answering [3]. However, while VALOR emphasizes general multimodal capabilities, Scene4M targets a more specific objective: understanding environmental scenes through the combined use of video, audio, and textual inputs. This targeted focus provides a clearer path toward real-world applications such as navigation assistance, where environmental awareness across multiple sensory streams is critical.

In the 3D domain, 3DMIT (3D Multi-modal Instruction Tuning) has recently introduced an effective framework for enriching large language models with 3D spatial awareness [4]. Recognizing the scarcity of 3D scene-language data, the authors constructed a dataset of 75,000 instruction-response pairs tailored to 3D scene tasks, including 3D visual question answering, grounding, and dialogue. Their novel prompt tuning paradigm

bypasses the need for explicit alignment between 3D scenes and textual descriptions, instead integrating segmented object and scene-level spatial information directly into the instruction prompt. Although 3DMIT operates in a different modality space than Scene4M, it highlights the growing interest in extending LLMs with richer spatial and perceptual context—an aim shared by Scene4M through its integration of audio and video for 2D environmental understanding.

III. Methodology

Our multimodal model has three key components: an audio tokenizer, a video tokenizer, and a training pipeline that integrates all modalities. We designed each tokenizer to convert raw data into a discrete representation compatible with a transformer-based architecture.

A. Dataset Selection

Training a multimodal model for navigation assistance requires a dataset containing both audio and visual information from real-world environments. VGG-Sound [5] is a large-scale, audio-visual dataset with 200,000+ video clips and 550+ hours of content across 300+ categories. The dataset offers a diverse distribution of scenes, including people, animals, vehicles, and environmental sounds, making it well-suited for developing models that support real-life navigation tasks.

We used 14 categories, chosen to reflect common entities encountered in street-level navigation—such as vehicles, humans, and pets. 13,000 clips were extracted, evenly split between audio and video, with each clip having an audio and video portion to preserve the multimodal nature of the data. To maintain a manageable scope and optimize training efficiency, we used a limited subset in size and category diversity, though a broader selection could be incorporated with additional computational resources.

The breakdown of our dataset is detailed in VII-A.

B. Audio Tokenizer

The audio tokenizer is responsible for converting raw waveform inputs into a sequence of discrete tokens suitable for multimodal processing. The pipeline begins with raw audio clips from the VGG-Sound dataset, which are first resampled to a lower sampling rate. Resampling serves a dual purpose: it reduces computational overhead and retains the most perceptually relevant frequency range for human listeners.

Following resampling, we transform the audio waveform into a time-frequency representation using a mel spectrogram. The mel scale is specifically chosen for its ability to mimic the nonlinear frequency sensitivity of the human auditory system, placing greater emphasis on lower frequencies where most informative content resides. The resulting mel spectrogram captures the power distribution of the signal across both time and frequency dimensions.

To discretize this continuous representation, we employ a Vector Quantized Variational Autoencoder (VQ-VAE) [6]. The encoder network within the VQ-VAE processes the mel spectrogram and maps it to a lower-dimensional latent space, where it is quantized into a fixed vocabulary of discrete tokens. These tokens serve as a compact, information-rich representation of the audio signal, enabling alignment with other modalities and facilitating efficient multimodal training.

The configuration and training of the audio tokenizer are detailed in VII-B.

C. Video Tokenizer

The video tokenizer processes raw video clips from the VGG-Sound dataset and converts them into sequences of discrete tokens. We first preprocess each video clip by removing black padding bars to eliminate irrelevant visual information. The frame rate and resolution are then reduced to predefined values (1.6 fps and 32×32 resolution) to improve computational efficiency while preserving essential spatio-temporal features. The preprocessed video is then passed into a VQ-VAE, which transforms the continuous video data into a discrete tokenized form.

The VQ-VAE consists of an encoder-decoder architecture, where the encoder comprises three layers of 3D convolution with batch normalization and ReLU activations in the first two layers. The encoder maps the video into a grid of latent vectors in the codebook space. Each vector in this grid is replaced by the nearest entry from a fixed-size codebook using nearest-neighbor search, and the resulting indices are used as the discrete video tokens. The decoder mirrors the encoder architecture but uses a tanh activation in the final layer to produce outputs within $[-1, 1]$, matching the normalized input video format.

The codebook is initialized using a k-means clustering procedure: multiple input videos are passed through the encoder, and the resulting latent vectors are clustered using k-means (with $k = codebook\_size = 256$). The cluster centroids are used as the initial codebook vectors. During training, these codebook vectors are updated using an exponential moving average (EMA) rather than direct gradient descent. This EMA mechanism stabilizes training by moving each codebook vector toward the centroid of all encoder vectors currently mapped to it, based on moving averages of the cluster centers and cluster sizes.

The configuration and training of the video tokenizer are detailed in VII-C.

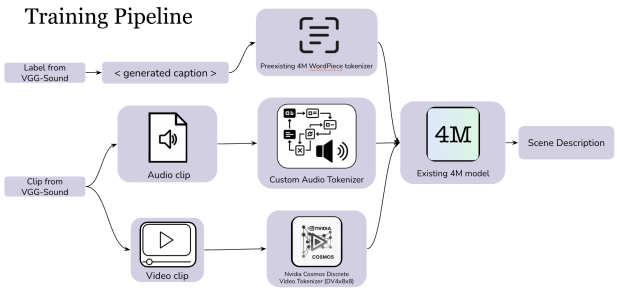

Fig. 1. Full training pipeline; each modality is passed to a specialized tokenizer, then all three modalities are fed together to the 4M model

D. Full Training Pipeline

The complete training pipeline integrates audio, video, and textual modalities to enable multimodal scene understanding. Each data point begins with a label from one of the 14 selected VGG-Sound categories. As these labels are only one or two words long, we created a curated dictionary that contains three to four extended natural language captions for each category. For every data point, a caption is randomly sampled from the corresponding set to introduce textual variability and provide a richer supervision signal. The captions are then passed into the WordPiece Tokenizer used by 4M.

The raw audiovisual clip from VGG-Sound is then split into separate video and audio components. The video is tokenized using Nvidia's COSMOS discrete video tokenizer (DV4x8x8), which generates compact spatio-temporal token sequences. We extended the initial timeline for the self-developed video tokenizer, so the pre-trained COSMOS model was used to ensure stability and efficiency in the final training. The audio component is processed through the custom audio tokenizer, which transforms the waveform into discrete tokens as described earlier.

Once tokenized, all three modalities—video tokens, audio tokens, and the sampled text caption—are fed into the 4M model (specifically, the 4M-7_B_CC12M).

The 4M model was initialized by loading the EPFL-VILAB/4M-7_B_CC12M pre-trained checkpoint. Initially, all parameters of this pre-trained 4M model were frozen to ensure they remained unchanged during the early stages of fine-tuning. The AdamW optimizer was chosen for training, with an initial learning rate set at 1×10^{-4}. The cross entropy loss function used was the same modality-specific loss computation performed within the 4M model. The overarching training objective was to minimize this calculated loss, which was then backpropagated to facilitate the updating of the model's trainable parameters.

The full configuration of the training pipeline is detailed in VII-D.

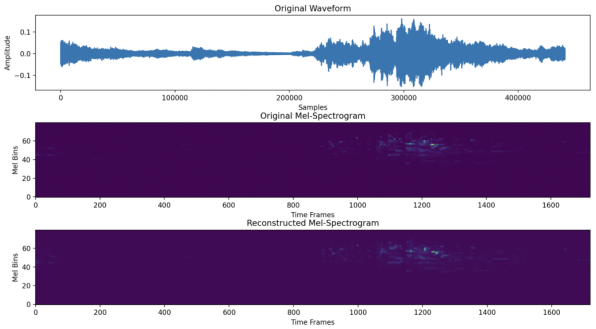

Fig. 2. Audio tokenizer results; top image is the original waveform of the audio clip; middle image is the original mel spectrogram; bottom is the reconstructed mel spectrogram

IV. Experiments

The experimental evaluation covers the three core system components (audio tokenizer, video tokenizer, full training pipeline), along with a transfer evaluation study assessing the extensibility of the 4M model.

For the audio tokenizer, qualitative results demonstrate the successful conversion of raw waveforms into discrete token representations. The tokenizer had a final loss of 0.2041. Figure 2 shows an example mel spectrogram input and its corresponding reconstruction.

Fig. 3. Video tokenizer results: the top row contains the first five frames of an input video after preprocessing; the bottom row contains the same five frames reconstructed

The video tokenizer also shows promising results with a perceptible retention of spatial and temporal structure. Figure 3 includes original and reconstructed frames for an example video, illustrating the encoder's ability to preserve semantic content despite compression.

For the full multimodal pipeline, evaluation was performed using a subset of test examples where audio, video, and sampled caption prompts were input to the 4M model. Figure 4 shows a representative output, including tokenized video/audio inputs and the corresponding generated scene description. The final average Loss was 0.3705. Figure 6 in the Appendix shows the WandB charts of the final training.

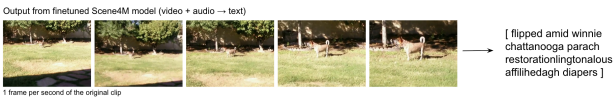

Fig. 4. Results of the full training; input modalities are video + audio, output modality generated is a (nonideal) text description of the scene

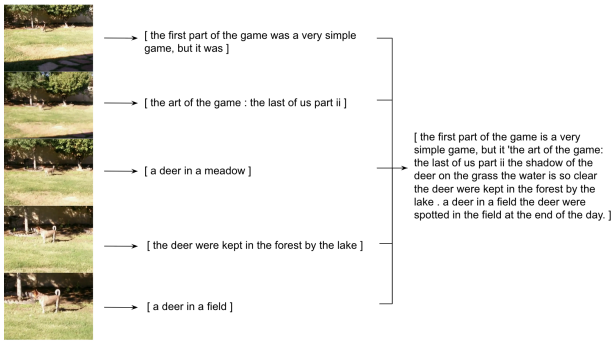

Fig. 5. Transfer evaluation; the text is generated by the original EPFL-VILAB/4M-7_B_CC12M model without any additional finetuning

To evaluate the model's flexibility, a transfer evaluation was conducted to explore whether video understanding could emerge in 4M when it was not explicitly trained for it. 16 video frames were passed as individual image inputs to the base 4M model, generating frame-wise captions that were then aggregated into a single video-level scene summary using the Google T5 summarization model. Figure 5 presents an example of a full-scene description generated via this approach. The resulting summary was often disjointed, repetitive, or lacking coherence, highlighting the difficulty of the model in integrating temporal context between frames. This suggests that explicitly training 4M on a dedicated video modality could significantly improve its capacity to understand dynamic scenes and better support complex, real-world tasks such as navigation assistance.

V. Conclusion and Limitations

This project demonstrates the first steps in building a multimodal system for scene understanding by integrating audio, video, and textual modalities using discrete tokenizers and a unified training pipeline. While the results show promising alignment between modalities, several limitations remain. Training on video data is computationally intensive and time-consuming, which constrained the scale and depth of experimentation. Additionally, although VGG-Sound provides a large and diverse dataset, the selected subset may not fully capture the variability and complexity of real-world environments, limiting the generalizability of the model. Furthermore, the generated captions used during training are limited in both diversity and richness; captions within the same category often lack variation and are short in length, reducing the model's exposure to nuanced language. These factors highlight the need for more scalable training infrastructure, richer and more diverse datasets, and enhanced textual supervision to further improve the system's robustness and real-world applicability.

Several extensions could improve the performance and generalizability of the current system. Increasing the amount of training time and expanding the volume of training data used could lead to more stable and robust model, capable of generating better captions. Additionally, experimenting with alternative audio and video tokenizers would provide insight into how different discretization strategies affect downstream performance, potentially revealing more efficient or semantically rich tokenization methods.

VI. Individual Contributions

Two team members carried out the majority of the work in this project. L. Marinov was responsible for the training pipeline code, authored all three project reports, created the final presentation slides, helped with the audio tokenizer, and helped with the dataset preprocessing. C. Egeland implemented the video tokenizer, preprocessed the dataset, conducted the transfer evaluation, generated the outputs of the finetuned model, and assisted with the full training setup.

A. Duval contributed to the implementation of the audio tokenizer, but the configuration was not updated to work within the training pipeline. Other tasks assigned to him were either not completed or had to be substantially rewritten due to critical issues in the code. K. Driss did not contribute to the implementation or documentation of the project but helped prepare the presentation slides.

References

[1] R. Bachmann, O. F. Kar, D. Mizrahi, A. Garjani, M. Gao, D. Griffiths, J. Hu, A. Dehghan, and A. Zamir, "4M-21: An Any-to-Any Vision Model for Tens of Tasks and Modalities," 2024. [Online]. Available: https://arxiv.org/abs/2406.09406

[2] B. Zhang, K. Li, Z. Cheng, Z. Hu, Y. Yuan, G. Chen, S. Leng, Y. Jiang, H. Zhang, X. Li, P. Jin, W. Zhang, F. Wang, L. Bing, and D. Zhao, "VideoLLaMA 3: Frontier Multimodal Foundation Models for Image and Video Understanding," 2025. [Online]. Available: https://arxiv.org/abs/2501.13106

[3] J. Liu, S. Chen, X. He, L. Guo, X. Zhu, W. Wang, and J. Tang, "VALOR: Vision-Audio-Language Omni-Perception Pretraining Model and Dataset," 2025. [Online]. Available: https://arxiv.org/abs/2304.08345

[4] Z. Li, C. Zhang, X. Wang, R. Ren, Y. Xu, R. Ma, and X. Liu, "3DMIT: 3D Multi-modal Instruction Tuning for Scene Understanding," 2024. [Online]. Available: https://arxiv.org/abs/2401.03201

[5] H. Chen, W. Xie, A. Vedaldi, and A. Zisserman, "VGGSound: A Large-scale Audio-Visual Dataset," 2020. [Online]. Available: https://arxiv.org/abs/2004.14368

[6] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural Discrete Representation Learning," 2018. [Online]. Available: https://arxiv.org/abs/1711.00937

VII. Appendix

A. Dataset Breakdown

14 of the available 300+ categories were chosen from VGG-Sound. The breakdown of number of clips obtained from each category and their percentage weight in the full training dataset are detailed in Table VII-A.

Category	Count	Percentage
driving buses	794	10.39%
engine accelerating, revving, vroom	788	10.31%
driving motorcycle	772	10.10%
people crowd	757	9.90%
police car, siren	753	9.85%
female speech, woman speaking	699	9.15%
male speech, man speaking	698	9.13%
skateboarding	606	7.93%
dog barking	510	6.67%
car passing by	455	5.95%
hammering nails	401	5.25%
wind chime	270	3.53%
people eating	140	1.83%
Total	7643	100%

TABLE I
Breakdown of dataset categories

B. Audio Tokenizer Configuration and Training

The audio tokenizer is configured with a latent dimension of 64 and uses 512 embeddings. Its architecture comprises an Encoder with 1 input channel, 128 hidden channels, and 64 output z_channels, and a Decoder with 64 input z_channels, 128 hidden channels, and 1 output channel. The VQ-VAE component has an embedding dimension of 64. Training uses a batch size of 32, and the model is optimized with AdamW, applying a learning rate of 1×10^{-3} for the encoder/decoder and 1×10^{-4} for the VQ embeddings.

We train for 20 epochs, and the total loss is a combination of reconstruction loss (Mean Squared Error between reconstructed and original audio) and VQ loss, which itself includes a commitment loss and an embedding loss. A linear warmup schedule is applied to the commitment cost (β) over the first 5 epochs, increasing from 0.01 to 0.25. Audio preprocessing involves resampling waveforms to 16000 Hz and generating mel spectrograms with an n_fft of 1024, a hop_length of 256, and 80 mel bins. These mel spectrograms are then normalized by their median and standard deviation, and log1p is applied. A custom padding function ensures consistent audio sequence lengths within batches.

C. Video Tokenizer Configuration and Training

Since nearest-neighbor selection is non-differentiable, we employ a straight-through estimator to enable backpropagation. During the forward pass, the quantized codebook vectors are passed to the decoder, while during the backward pass, gradients from the decoder are redirected through the unquantized encoder outputs to update encoder weights. This combination of techniques allows the video tokenizer to produce stable, discrete token sequences that preserve the essential spatio-temporal structure of the original video data.

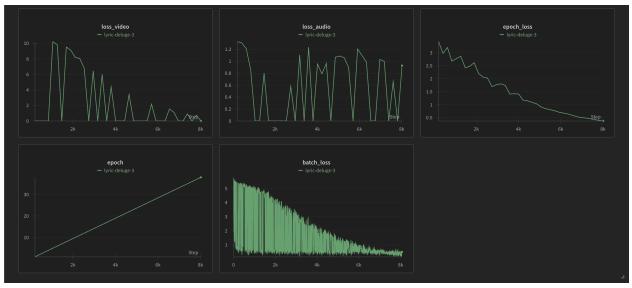

Fig. 6. WandB charts of full training; from top left to bottom right as follows: video loss, audio loss, epoch loss, epoch, batch loss

D. Training Pipeline Configuration

During training, we used a batch size of 32, and the model was trained for a total of 40 epochs. We consistently apply a token_budget of 128 for both the encoder and decoder during the fm model's forward pass. New `BaseModalityEncoderEmbeddings` and `BaseModalityDecoderEmbeddings` were instantiated specifically for the audio, video, and caption modalities. These newly introduced embedding layers were initialized using the *init_std* value of 0.02, inherited from the original 4M model, and were subsequently configured to be trainable. Finally, the 4M modality information dictionary, along with the 4M encoder and decoder modality sets, were updated to accurately reflect the incorporation of these new audio, video, and caption modalities into the model's architecture.

E. Full Training WandB Results

The training progress visualized in the charts in Figure 6 demonstrates somewhat encouraging trends, particularly in overall model convergence. The epoch loss (top right of Figure 6 and enlarged in Figure 7) shows a clear and consistent downward trajectory, steadily decreasing from over 3.0 to below 0.5 across approximately 8,000 steps. This loss reduction indicates that the model is learning effectively over time and that the training process is stable.

The video loss (top left of Figure 6) follows a similarly positive pattern, with initial volatility becoming a more sustained reduction, eventually reaching near-zero levels. This mild improvement suggests that the video component of the model is somewhat successfully reconstructing and encoding visual data as training progresses.

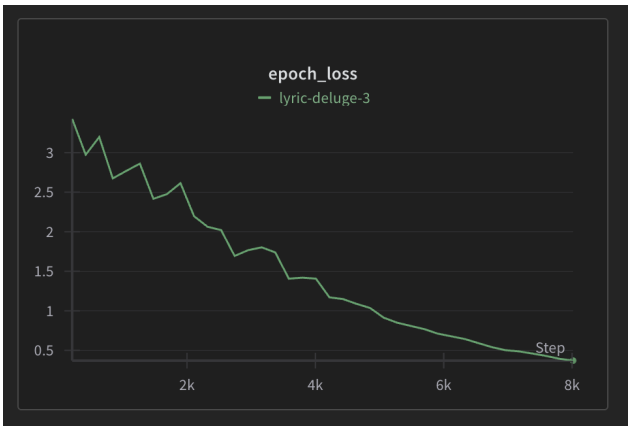

Fig. 7. Enlarged epoch loss chart from full training

The audio loss (top middle of Figure 6), does not trend downward overall. The observed fluctuations likely reflect the increased difficulty of audio tokenization or possible noise in the data, but the loss generally remains within a manageable range and decreases over time.

The batch loss (bottom right of Figure 6) aligns with the epoch loss trend, showing a rapid early decline and gradually tapering off as the model improves.

Overall, these results indicate some effective training behavior, particularly in terms of video reconstruction and overall convergence.