WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

HEAR2SEE: An End-to-End Audio-to-Video System via ASR-Driven Prompting and Diffusion

Dr. Swathi K[a], Aditya Sharma[b], Aman H. P[c], *

[a]Guide & Associate Professor, Dept. of CSE, Jyothy Institute Of Technology, Bangalore, India
[b]Dept. of CSE, Jyothy Institute Of Technology, Bangalore, India
[c]Dept. of CSE, Jyothy Institute Of Technology, Bangalore, India
*Corresponding author: k.swathi980@gmail.com
Email: adityapa2004@gmail.com (Aditya Sharma), amanphaller@gmail.com (Aman H. P)

A B S T R A C T :

HEAR2SEE is a lightweight, end-to-end system that combines three steps to transform raw audio into brief, illustrative videos: on-device speech recognition to generate a transcript, a prompt builder to refine the transcript into a succinct visual scene description, and a diffusion renderer, optimised with LoRA on Stable Diffusion 1.5, to create image frames that are stitched into a video. With a focus on practicality, the entire pipeline operates on consumer hardware and scales smoothly on a single GPU, aiming for general visual storytelling that follows narration rather than specific tasks like talking-head reenactment or motion prediction. Our implementation consists of a Gradio "Audio → Prompt" wizard and LoRA adaptation on a domain set of about 150 images. In addition to Drive-based I/O and JSONL logging for reproducibility and experiment tracking, we measure semantic and temporal alignment using CLIPSIM and an AV-Align heuristic that links audio onsets to visual changes and evaluate video quality using FVD and IS. With prompt edits and generation settings (steps, CFG, frames, fps, and seed), the system produces narrative-aligned clips while preserving user control. We outline an upgrade path: stronger ASR (Kaldi/Whisper), video-native diffusion backbones for improved temporal consistency, and expanded fine-tuning datasets for higher fidelity.

Keywords: Audio-to-video generation, Automatic Speech Recognition (ASR), Vosk, Stable Diffu- sion, LoRA fine-tuning, Diffusion models, Prompt engineering, temporal consistency, CLIP similarity (CLIPSIM), Fréchet Video Distance (FVD), Inception Score (IS), Gradio UI.

Introduction

Three skills are required to transform narration into moving images: comprehending speech, determining what is visually significant, and producing consistent imagery over time. To solve this, HEAR2SEE provides a practical "transcribe → prompt → diffuse" pipeline that turns raw audio into short, narration- aligned videos. After obtaining a transcript via on-device ASR, the system converts the text into a succinct, *editable* visual prompt and then creates frames that are stitched into a video using a Stable Diffusion 1.5 renderer adapted with LoRA. With Drive-based I/O and per-run metadata logging, this design supports reproducible experiments, operates on consumer hardware (Colab or a single GPU), and maintains the human in the loop.

HEAR2SEE targets general visual storytelling rather than narrow tasks, such as talking-face reenact- ment or motion prediction. By anchoring prompts to the spoken narration, the system produces scenes that reflect the audio content and rhythm while allowing users to correct entities, add context, or specify style before generation. To mitigate common issues in frame-wise diffusion, such as drift and artifacts, we used fixed seed schedules, conservative guidance scales, negative prompts, and optional slow-mode playback. The framework remains modular: stronger ASR (Kaldi/Whisper), larger and more diverse fine-tuning sets, and video-native diffusion backbones with temporal attention can all slot in without altering user flow.

Literature Survey

Before deep learning became dominant, researchers relied on handcrafted features tuned to narrow problem families. Modern methods exploit learnable representations and multimodal fusion to map audio cues to temporally coherent visuals. We review six core strands that informed HEAR2SEE.

1.1 1) Sound2Sight: Generating Visual Dynamics from Sound and Context (ECCV 2020)

Authors: Moitreya Chatterjee, Anoop Cherian.

Venue: ECCV 2020, Springer.

Brief: An early system treating audio as a driver for *future visual frames*. By framing audio as both a semantic and temporal signal,

Sound2Sight models plausible visual dynamics from sound with limited context, anticipating the principle HEAR2SEE adopts: use audio to steer what should appear and when [1].

1.2 2) Sound-Guided Semantic Video Generation (ECCV)

Authors: Seung Hyun Lee, Gyojin Oh, Wonmin Byeon, Chan-Hyun Kim, Woo Jin Ryoo, Seung Hwan Yoon, Hyeonseung Cho, Jaejin Bae, Jihwan Kim, Seunghoon Hong.

Venue: ECCV, Springer.

Brief: Conditions video generation directly on *sound* to guide *what* appears in frames, validating the sound-to-semantic control. This complements HEAR2SEE' transcript-to-prompt stage by showing that audio can reliably guide content selection beyond motion-only cues [2].

1.3 3) Robust One-Shot Audio-to-Video Generation (CVPR Workshops 2020)

Authors: Nishant Kumar, Shobhit Goel, Anant Narang, Md. Hasan.

Venue: CVPR Workshops 2020.

Brief: Addressing low-data regimes for audio-conditioned generation and demonstrating that *limited supervision* can still produce usable audio→visual mappings. This supports HEAR2SEE' emphasis on lightweight components and small fine-tuning sets [3].

1.4 4) Music2Video: Automatic Generation of Music Video from Music Audio (arXiv)

Authors: Yoonjeon Kim, Joel Jang, Sumin Shin.

Venue: arXiv preprint (CC BY 4.0).

Brief: Targets music-conditioned video, aligning the *audio structure* with visual rhythm/semantics and automating mapping from sound patterns to scene evolution. This motivates staged design (ASR → prompt → diffusion) as a generalisation beyond music-only pipelines [4].

1.5 5) The Power of Sound (TPoS): Audio-Reactive Video Generation with Stable Diffusion (arXiv 2023)

Authors: Yujin Jeong, Wonjeong Ryoo, Seunghyun Lee, Dabin Seo, Wonmin Byeon, Sangpil Kim, Jinkyu Kim (corr.).

Venue: arXiv:2309.04509 (Sept 2023).

Brief: Builds on Stable Diffusion: first frame from text and subsequent frames conditioned by *temporally encoded audio embeddings*. Introduces LSTM-based temporal semantics, a Temporal Attention Module (TAM), and audio semantic guidance with identity regularisation to maintain coherence—directly relevant to reducing frame-wise drift in HEAR2SEE [5].

1.6 6) MM-Diffusion: Learning Multi-Modal Diffusion Models for Joint Audio and Video Generation (CVPR)

Authors: L. Ruan, Y. Ma, H. Yang, H. He, B. Liu, J. Fu, N. J. Yuan, Q. Jin, B. Guo.

Venue: CVPR.

Brief: A diffusion-based framework for *joint* audio-video generation shows that a single model can align sound and visual signals during synthesis. Positioned alongside Latent/Video Diffusion, Imagen Video, and CogVideo, it charts the path from image diffusion to temporally coherent video that HEAR2SEE can adopt. Related work like AudioToken and Wav2CLIP, plus optical-flow literature, informs our evaluation plan (e.g., AV-Align via flow) [6, 10, 9].

Objectives

- **(O1) Practical end-to-end pipeline on consumer hardware:** A small ASR → prompt → diffusion stack that runs on Colab or a single GPU with modest defaults (20–30 steps, 24–48 frames, 6–12 fps) and attention/VAE slicing, with CPU fallback for brief clips.
- **(O2) Human-in-the-loop control:** The editable prompt and transparent parameters (steps, CFG, frames, fps, seed, and slow mode) are presented in a three-step Gradio wizard (Audio → Prompt → Video).
- **(O3) Fast domain adaptation via LoRA:** Compact checkpoints, parameter-efficient SD-1.5 fine-tuning on small, carefully selected datasets, and simple model swapping as domains change.
- **(O4) Simple, repeatable evaluation:** CLIPSIM and AV-Align for temporal and semantic alignment, optional MOS to supplement the objective metrics, and FVD/IS for visuals.
- **(O5) Reproducibility:** Drive-based I/O with per-run JSONL metadata (prompt, negatives, steps, CFG, frames, fps, seed, model/LoRA version, and paths).

Proposed Methodology

The proposed system presents a modular and effective pipeline that transforms audio narration into brief videos that are visually coherent and semantically relevant. The three main stages of this methodology are diffusion-based video synthesis, prompt generation and refinement, and automatic speech recognition (ASR). Scalability, interpretability, and computational efficiency were balanced in the design of each step.

1.7 A. Audio-to-Text Conversion (ASR)

The Vosk ASR model, which works well on devices without external cloud dependency, is used in the first stage to convert the input audio into textual content. In this step, the unprocessed speech is converted into a transcript that preserves the semantic structure of the narration. In subsequent implementations, the ASR module can be swapped out for Kaldi or Whisper, which provides multilingual transcription, enhanced noise robustness, and reduced word error rates (WER) to improve accuracy and flexibility. The linguistic foundation for creating context-aware visual cues is the output transcript.

1.8 B. Prompt Construction and Enhancement

By extracting important objects, scenes, moods, and styles from a transcript, an NLP heuristic pipeline generates a descriptive visual prompt. To maintain creative control, users edit and improve prompts through a Gradio user interface. Final visuals can be customized by adding more modifiers, such as lighting, composition cues, and visual styles (realistic, anime, and cinematic).

1.9 C. Diffusion-Based Image Generation and Frame Rendering

A Stable Diffusion 1.5 model that has been improved with Low-Rank Adaptation (LoRA) receives the refined prompt. Even with a small dataset (∼150 images), LoRA improves domain alignment while consuming less memory and computing power. In order to preserve quality, the diffusion model creates frames in a sequential manner using regular random seeds and common negative prompts (such as "blurry," "low-resolution," and "watermark"). FFmpeg stitches frames together to create a continuous, high-quality video.

1.10 D. Temporal Consistency and Frame Smoothing

We used an incremental seeding scheme, where each frame used a slightly perturbed seed derived from the previous one to minimise flicker and maintain motion smoothness. To further improve the temporal coherence, future extensions will incorporate temporal diffusion modules and optical-flow-guided interpolation.

1.11 E. Evaluation and Quality Metrics

We employ both objective and subjective metrics: MOS from user studies for perceptual realism, CLIPSIM for semantic alignment, AV-Align for temporal synchronisation, and FVD and IS for fidelity/diversity.

Implementation

The goal of the HEAR2SEE system is to create a modular, reproducible, end-to-end pipeline that can transform unprocessed audio input into visually coherent and semantically meaningful video outputs. Using Gradio for the interactive user interface and PyTorch for the deep-learning components, the entire
framework was implemented in Python. While retaining scalability and reproducibility, the system runs effectively on consumer-grade hardware, such as GPUs found in Google Colab or mid-range laptops.

1.12 A. Development Environment and Frameworks

Python 3.12 was used to implement the system, which incorporated the Diffusers library for inferring diffusion models, Transformers for text processing, and PEFT (Parameter-Efficient Fine-Tuning) for fine-tuning based on LoRA. The Vosk ASR model was chosen for speech-to-text processing owing to its lightweight design and offline capability. Google Drive was utilised for experiment logging and data persistence, while the FFmpeg library was utilised to encode generated image frames into high-quality MP4 videos.

Key dependencies include:

- PyTorch 2.x (CUDA 12.x) – core deep learning framework.
- Diffusers 0.30+ – Stable Diffusion pipeline for image generation.
- PEFT 0.11+ – Low-Rank Adaptation (LoRA) fine-tuning library.
- Vosk ASR – offline automatic speech recognition engine.
- Gradio 4.x – user interface for interactive workflow.

1.13 B. Data Flow and System Pipeline

Using the ASR model, the system first converts audio files into text when they are uploaded. The Prompt Builder receives this transcript and turns it into a thorough visual description. Then, the Diffusion Renderer generates frames one after the other using the optimized Stable Diffusion 1.5 model with LoRA. FFmpeg was used to stitch these frames into a video clip. For traceability and reproducibility, all intermediate and final artifacts, such as audio, transcripts, prompts, image frames, and produced videos, are kept in Google Drive under organized directories.

1.14 C. Audio-to-Text Conversion (ASR)

The Vosk model was selected because of its offline compatibility and efficiency, which make it appropriate for consumer electronics. Prior to transcription, the input audio was standardized to a mono WAV format at 16 kHz. Both the temporal and semantic elements of the narrative are captured in the final transcript. Upcoming versions of HEAR2SEE can incorporate the Kaldi or Whisper ASR models to facilitate transcription that is both multilingual and noise-resistant.

1.15 D. Prompt Builder and NLP Processing

After the transcription is finished, the text is normalized by adding punctuation, fixing the casing, and eliminating fillers. To find pertinent objects, actions, and scene descriptors, a set of heuristic rules was applied in conjunction with fundamental NLP functions (such as keyword extraction and part-of-speech tagging). Adding optional modifiers, such as lighting, mood, or artistic style (e.g., realistic, anime, or cinematic), improves the final visual prompt. The Gradio interface allows this prompt to be edited.

1.16 E. Diffusion Model Fine-Tuning and Frame Generation

Stable Diffusion 1.5, optimised with LoRA on a custom dataset of approximately 150 domain-specific images, is the foundation of the visual generation component of HEAR2SEE. The LoRA preserves high-quality adaptation to new visual domains while reducing training time and memory usage. The model created each frame during inference using a fine-tuned visual cue. A constant random seed scheduling technique was applied across the frames to reduce flickering and preserve visual coherence. For compatibility and quality

preservation, the produced images were subsequently encoded into a video using FFmpeg with H.264 compression and the YUV420p pixel format.

1.17 F. Temporal Stability and Video Assembly

An incremental seed perturbation technique was used to achieve temporal smoothness, using a slightly modified random seed from the previous frame for each frame. Visual continuity and seamless transitions between frames are guaranteed using this technique. Future developments include the use of native video diffusion models for increased realism and optical flow-based frame interpolation.

1.17.1 Inference Strategy

To minimise flicker, we used a fixed seed schedule (seed + i per frame) to generate N frames per clip. The default parameters were 24–48 frames, 6–12 FPS, 20–30 sampling steps, and CFG=6–8. "Slow mode" reduces frame rates to improve on-screen readability. Common artifacts are suppressed by using negative prompts (such as "text," "watermark," "lowres," "blurry").

1.17.2 Evaluation Metrics

FVD/IS for visual fidelity and diversity. **CLIPSIM** computes cosine similarity between CLIP embeddings of frames and reference text derived from the transcript/prompt:

$$\text{CLIPSIM} = \frac{1}{T}\sum_{t=1}^{T} \frac{\langle f_{\text{CLIP}}(I_t), f_{\text{CLIP}}(x)\rangle}{\|f_{\text{CLIP}}(I_t)\| \, \|f_{\text{CLIP}}(x)\|}. \quad (1)$$

AV-Align correlates audio onset peaks with frame-to-frame optical-flow magnitude changes within a $\pm k$ frame window:

$$\text{AVAlign} = \frac{1}{|P|}\sum_{p\in P} \mathbb{1}\{\exists\, t \text{ s.t. } |t-p| \le k \wedge \Delta\phi(I_t) \ge \tau\}. \quad (2)$$

1.18 H. User Interface and Experience

To make things easier for non-technical users, a three-step Gradio interface was created:

- **Audio Input:** Audio can be recorded or uploaded by users directly within the interface.
- **Prompt Refinement:** The generated transcript and an automatically created visual prompt for user editing are shown by the system.
- **Video Generation:** Users create the final video after setting rendering parameters (frames, FPS, seed, and CFG scale).

1.19 I. Reproducibility and Experiment Tracking

Essential metadata such as timestamps, prompts, model parameters, random seeds, and output file paths are automatically logged in JSON Lines (.jsonl) files for each system run. This allows for a dependable comparison of model configurations and deterministic re-execution of earlier experiments. Both the final video outputs and intermediate data (ASR transcripts, image frames) are preserved across sessions via Google Drive integration.

1.20 J. Performance and Hardware Requirements

HEAR2SEE is designed to operate smoothly on mid-range hardware and requires approximately 8–16 GB of VRAM. On a Tesla T4 GPU, the system can produce a 10-second video (24 frames, 20 sampling steps, 6 FPS) in less than two minutes. For short clips, lowering the resolution, frame count, and denoising steps allows CPU-only operation.

System Architecture

Overview

HEAR2SEE comprises three modular services: (1) ASR for speech-to-text; (2) prompt builder for text-to-visual description; (3) diffusion renderer for prompt-to-frames and MP4 assembly.

ASR Module

We default to Vosk for offline transcription with 16 kHz mono WAV inputs. The ASR slot is pluggable; Kaldi enables graph-based decoding, while Whisper improves WER and multilingual coverage at higher compute.

Prompt Builder

The transcript is normalised (punctuation, casing, filler removal) and enriched with style tags (*cinematic, realistic, anime*). The prompt remains user-editable to correct entities, add constraints (objects, locations), and specify camera/lighting cues.

Diffusion Renderer

We use SD-1.5 with LoRA adapters. Inference exposes steps, CFG, frames, fps, and seed. Negative prompts (e.g., "text, watermark, lowres, blurry") suppress artefacts. Frames are encoded via ffmpeg (H.264, yuv420p).

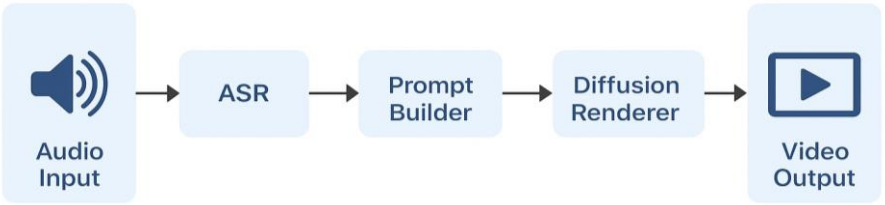

Figure 1: HEAR2SEE high-level architecture: Audio Input → ASR → Prompt Builder → Diffusion Renderer → Video Output.

Conclusion

HEAR2SEE uses a straightforward, controllable pipeline that can be implemented on common hardware to convert narration into visual information. Stronger ASR (Kaldi/Whisper), scaling fine-tuning datasets, switching to video-native diffusion with temporal attention, and incorporating light post-processing (deflicker/interpolation) are examples of future work that will further enhance temporal coherence.

REFERENCES

[1] A. Cherian and M. Chatterjee, "Sound2Sight: Generating Visual Dynamics from Sound and Context," in *ECCV*, 2020.

[2] S. H. Lee, G. Oh, W. Byeon, C.-H. Kim, W. J. Ryoo, S. H. Yoon, H. Cho, J. Bae, J. Kim, and S. Hong, "Sound-Guided Semantic Video Generation," in *ECCV*, Springer.

[3] N. Kumar, S. Goel, A. Narang, and M. Hasan, "Robust One-Shot Audio-to-Video Generation," in *CVPR Workshops*, 2020.

[4] Y. Kim, J. Jang, and S. Shin, "Music2Video: Automatic Generation of Music Video from Music Audio," *arXiv preprint*, CC BY 4.0.

[5] Y. Jeong, W. Ryoo, S. Lee, D. Seo, W. Byeon, S. Kim, and J. Kim, "The Power of Sound: Audio- Reactive Video Generation with Stable Diffusion," *arXiv:2309.04509*, 2023.

[6] L. Ruan, Y. Ma, H. Yang, H. He, B. Liu, J. Fu, N. J. Yuan, Q. Jin, and B. Guo, "MM-Diffusion: Learning Multi-Modal Diffusion Models for Joint Audio and Video Generation," in *CVPR*.

[7] H. Chen, W. Xie, A. Vedaldi, and A. Zisserman, "VGG-Sound: A Large-Scale Audio-Visual Dataset," in *ICASSP*, 2020.

[8] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," in *CVPR*, 2022.

[9] J. Ho, A. Saharia, C. Chan, et al., "Video Diffusion Models," 2022. (arXiv preprint)

[10] J. Ho, C. Saharia, et al., "Imagen Video: High Definition Video Generation with Diffusion Models," 2022. (arXiv preprint)

[11] M. Brand, "Voice Puppetry," in *SIGGRAPH*, 1999.