



#AI 스마트 프롬프터

2025 KBS 미디어창의기술전 수상작 소개

AI 스마트 프롬프터 개발

글. 김도형 KBS 제작기술국 제작기술1



그림 1. KBS 미디어창의기술전 본심사 현장

KBS 방송인프라본부 내 인프라기획부 주관으로 매년 개최하는 ‘KBS 미디어창의기술전’ 본심사가 지난 2025년 11월 27일 본관 6층 대회의실에서 진행되었습니다. KBS는 미디어 환경변화를 선도할 방송기술 기반을 확대하고 미디어 서비스의 경쟁력을 강화하기 위해 매년 미디어창의기술전을 개최해오고 있습니다. ‘미디어창의기술전’은 제작, 송출, 송신 운용방식 개선을 위한 AI 등 신기술 도입 및 시스템/장비/플랫폼/서비스 개발품과 기술, 미디어 환경변화에 능동적으로 대처할 수 있는 창의적인 기획/의견 등을 논문으로 응모할 수 있도록 영역을 확대해 왔습니다.

‘미디어창의기술전’은 KBS 전 직원을 대상으로 크게 연구개발형식 부문과 자유형식 부문에서 기획안 신청을 받았으며, 예비 심사를 통과한 기획안 중 본심사를 거쳐 수상작이 결정됩니다. 본심사는 방송인프라본부장(심사위원장)을 비롯한 심사위원들의 공정한 심사를 통해 연구개발형식 부문에서는 최우수상(대상) 1팀, 우수상 1팀, 노력상 2팀이 선정됩니다. 또한, 자유형식 부문에서 노력상 1팀이 선정됩니다. 표는 수상의 영광을 안게 된 작품들입니다.

수상 내역	출품명	출품자
대상 (연구개발형식)	AI 스마트 프롬프터 개발	제작기술국 김도형, 원효민, 김수정, 박경길
우수상 (연구개발형식)	FM 송-중계기 특성 측정 디바이스 (경량화/경제형) 연구 개발	송신플랫폼국 한창진, 박정준, 김상안, 여지현
노력상 (연구개발형식)	WebRTC를 활용한 실시간 중계 리턴 시스템	제작기술국 임준하, 박소영, 락규재, 강나인
노력상 (자유형식)	폐쇄망 대응형 AI를 활용한 원격제어 종합지원시스템 개발	송신플랫폼국 이하주, 김영신, 송보석, 창원총국 기술국 장세현
노력상 (자유형식)	지능형 UHD-SFN 방송물관리를 위한 전파측정 센서 개발	송신인프라부 이재권, 기종수

AI 스마트 프롬프터 개발

개요

인공지능(AI)은 지난 몇 년간 방송기술 분야에서 뜨거운 화두였습니다. 특히 2025년을 기점으로 AI는 단순히 콘텐츠를 생성하는 생성형 AI(Generative AI) 단계를 넘어, 업무를 보조하고 자동화하는 ‘에이전틱 AI(Agentic AI)’ 시대로의 전환이 가속화되고 있습니다.

이러한 기술적 흐름 속에서 우리는 끊임없이 고민했습니다. “과연 AI를 활용해 현재의 방송 제작 워크플로우에 실질적인 변화와 혁신을 가져올 수 있을까?” 비슷한 고민을 가지고 있던 동료들이 모여 제작기술국 내에 ‘AI T/F’가 결성되었고, 수차례의 치열한 아이디어 회의를 거쳐 방송 현장의 문제를 해결할 다양한 방안을 모색했습니다.

제작기술국 AI T/F의 결과물로서 ‘AI 스마트 프롬프터’는 2025년 ‘KBS 미디어창의기술전’에서 대상을 받으며, AI 기술을 현장에 가장 적절하게 접목한 사례로 평가받았습니다. 이 시스템은 STT(Speech-To-Text) 기술을 활용해 진행자의 음성을 인식하고 프롬프터를 자동 제어함으로써, 인력 의존적인 제작 환경에서 AI를 활용해 ‘무인 운용’이 가능하다는 이정표를 제시할 수 있었습니다.

특히, 본 시스템은 테스트 단계에 머무르지 않고 실제 2026년 시무식 당시, 사장님의 신년사 발표에 ‘AI 스마트 프롬프터’가 사용되었으며, 즉흥 멘트가 있는 생방송 환경에서도 발화 속도에 맞춰 원고를 넘겨주었고, 안정적으로 동작하는 것을 확인할 수 있었습니다. 본 원고에서는 AI 스마트 프롬프터를 개발하기까지의 여러 개발 과정과 기술적 성취를 상세히 소개하고자 합니다.

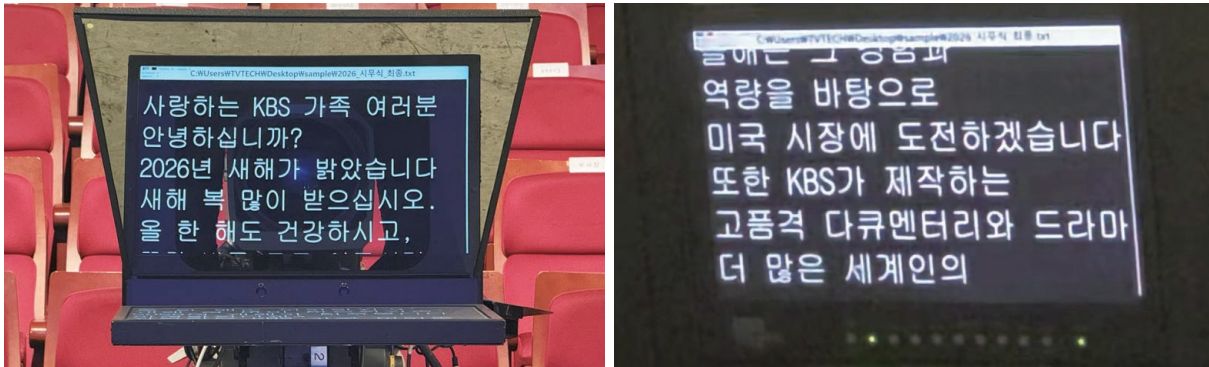


그림 2. 2026년 시무식에 사용된 ‘AI 스마트 프롬프터’

개발 배경

현실적 제약과 ‘음성 AI’로의 선택과 집중

수차례의 브레인스토밍 회의를 거듭한 결과, 다양한 아이디어가 쏟아졌습니다. AI 영상을 활용한 뷰티 메이크업, 스포츠 자동 컷팅, 관객 모니터링 등 다양한 기술들이 논의되었습니다. 그러나, 곧 현실적인 벽에 부딪혔습니다. 당시 제작기술국에는 고성능 GPU 서버가 단 한 대도 없는 열악한 환경이었기 때문에, 무거운 연산이 필요한 AI 기반의 영상 처리는 도저히 결과물로 내놓을 수 없는 상황이었습니다.

하지만 우리는 포기하지 않고 가용 가능한 자원 내에서 답을 찾고자 했습니다. 테스트 결과, AI 기반의 음성 처리는 고가의 GPU 서버 없이 일반적인 GPU 노트북만으로도 실시간에 가까운 연산이 가능하다는 것을 확인했습니다. 이에 우리

는 영상 대신 음성(Voice)에 집중하기로 방향을 선회했고, STT(Speech-To-Text) 기술을 활용해 제작 워크플로우를 개선하는 데 집중하기로 했습니다. 그렇게 탄생한 프로젝트가 'AI 스마트 프롬프터'입니다.

🔍 기존 프롬프터 시스템의 한계

우리가 프롬프터에 주목한 이유는 가장 기초적인 장비이면서도, 동시에 가장 아날로그적인 방식으로 운용되며 여러 가지 문제점을 가지고 있기 때문입니다.

첫째, 사람에 의존한 수동 조작의 불안정성입니다.

현재는 FD, 작가와 같은 운용자가 발화자의 읽는 속도를 수동으로 따라가는 형태로 프롬프터가 운용되고 있습니다. 이러한 형태의 운용방식은 개인의 경험이나 운용자가 누구인지에 따라 운용 품질의 편차가 발생하는 구조이며, 이 과정에서 조작 실수에 따른 인적 사고의 위험을 상존하게 합니다. 운용 품질의 저하 및 방송 사고의 위험은 진행자에게 부담으로 다가올 수 있으며 특히 생방송이나 관객이 있는 녹화에서는 더더욱 치명적일 수 있습니다.

둘째, 라이선스 비용입니다.

기존 상용 프롬프터 소프트웨어는 USB 타입의 라이선스 키가 별도로 있어서, USB 라이선스 키가 없으면 프롬프터 소프트웨어가 동작하지 않습니다. 작가 등 제작진이 스튜디오에서 프롬프터 카메라 출력 대신 트롤리 화면에 프롬프터 소프트웨어를 출력해서 사용하고 싶어도 라이선스 이슈로 인해 아래아한글을 사용하고 있었습니다.

셋째, 기능적 제약입니다.

기존 소프트웨어는 글자 크기가 최대 120으로 제한되어서 발화자가 먼 거리에서 프롬프터 화면을 바라볼 때 글자가 작아서 잘 보이지 않아 가독성이 떨어지는 불편함이 있었습니다.

시스템 구성

🔍 개발 환경(Development Environment)

우리의 개발 철학은 '경량화'와 '최적화'였습니다. 제작기술국 내 GPU 서버가 부재한 현실적인 제약을 극복하기 위해, 고가의 전용 장비 없이도 운용 가능하도록 효율성을 극대화했습니다.

하드웨어(H/W) : 별도의 GPU 기반 서버 도입 없이, 일반적인 GPU 기반 노트북만으로도 실시간 AI 연산이 가능하도록 최적화하였습니다.

소프트웨어(S/W) 아키텍처

- **Backend(AI 엔진)** : Python을 기반으로 구축된 AI 엔진은 실시간성을 보장하기 위해 다음과 같은 핵심 기술을 적용하였습니다.
 - **고속 추론 엔진** : OpenAI의 Whisper 모델을 C++로 최적화한 WhisperX(Medium 모델)를 도입하여 추론 속도를 높였습니다.
 - **화자 분리** : 뉴스 대담이나 토론 프로그램처럼 여러 발화자가 섞여서 말하는 상황에 대비해 Pyannote.audio 3.1 라이브러리를 통해 화자 분리(Speaker Diarization) 기능을 구현했습니다.
 - **TF32 가속 최적화** : PyTorch의 TF32(TensorFloat-32) 및 FP16(Half Precision) 연산을 활성화하여, 일반 소비자

용 GPU에서도 높은 연산 속도를 확보했습니다.

- **Frontend(사용자 인터페이스)** : 안정적인 .NET Framework 4.7.2 환경에서 C# 언어로 자체 개발. STT가 음성을 텍스트로 변환해주면, C# 프로그램이 이를 받아 스크롤을 제어하는 구조입니다. 특히 Python 엔진과는 비동기 IPC(Inter-Process Communication) 파이프라인을 구축하여, 데이터 병목 없이 실시간 인식 데이터를 수신합니다.

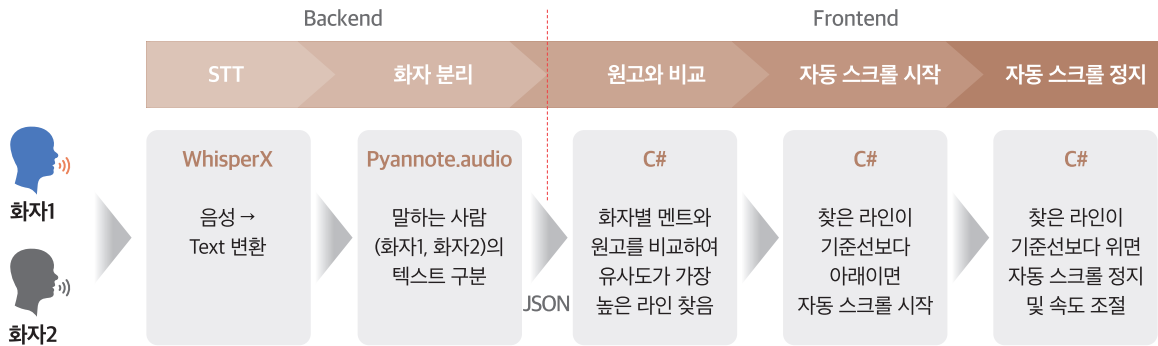


그림 3. 동작 단계별 Flow Chart

개발 목표(Development Goals)

본 시스템은 단순한 자동화를 넘어, 실제 방송 현장의 니즈를 반영한 ‘사용자 중심의 AI 시스템’ 구축을 목표로 했습니다.

실시간 동기화 및 핸드프리 제어

- 검증된 STT를 활용하여 높은 정확도의 음성 인식을 확보하고, STT를 통해 변환된 텍스트를 원고와 즉시 비교하여 읽기 진행 상황을 줄 단위로 즉시 추적, 말하는 사람의 속도에 따라 스크롤 속도를 자동으로 조절합니다.

스마트 오토 스크롤링

- AI 모델에 의존하지 않고, C# 기반의 정교한 알고리즘을 통해 제어 로직을 짰습니다. 이 알고리즘은 진행자가 말을 멈추면 스크롤을 즉시 정지하고, 말하는 속도에 따라서 스크롤 속도를 감속 또는 가속합니다.

비용 절감

- 오픈소스를 활용하고 자체 개발을 통해 라이선스 비용을 전액 절감합니다. 또한, 제작기술국이 보유하고 있던 범용 장비를 활용하여 하드웨어 구축 비용 또한 최소화했습니다.

시각적 편의성 및 기능 개선

- 기존 상용 제품의 한계였던 최대 글자 크기(120)를 190으로 대폭 확대하여 프롬프트가 설치된 카메라와 발화자가 멀어지는 환경에서도 가독성에 문제가 없도록 하였습니다.
- 진행자별 맞춤 설정(폰트, 색상, 속도 등)을 저장하는 ‘프리셋(Preset)’ 기능과 무제한 파일 로딩 기능을 탑재하여 운용 편의성을 극대화하였습니다.

운용 안정성 확보

- AI 자동 제어와 수동 제어를 즉시 전환할 수 있는 하이브리드 시스템을 목표로 하여, 생방송 중에 발생할 수 있는 돌발 상황에 유연하게 대처하도록 설계했습니다.

Backend 핵심 알고리즘 및 기술적 차별성

단순히 '음성을 인식하는 것'을 넘어, 끊김 없고 정확한 프롬프트 진행을 위해 정교한 로직을 구현했습니다.

슬라이딩 윈도우 기반 컨텍스트 버퍼링(Context Buffering)

- 실시간 음성 인식의 가장 큰 문제는 문장 중간에서 녹음이 끊길 경우 인식률이 급격히 떨어진다는 점입니다. 이를 해결하기 위해 우리는 '1초 문맥 버퍼(Context Buffer)' 기술을 도입했습니다. 현재 녹음된 2초 분량의 오디오 앞에 직전 1초 분량의 오디오를 결합(Concatenate)하여 AI에 총 3초의 데이터를 전달함으로써, 문맥의 연결성을 유지하고 인식 정확도를 대폭 개선했습니다.

RMS 에너지 기반 묵음 감지(Silence Detection)

- GPU 자원을 효율적으로 사용하기 위해 RMS(Root Mean Square) 에너지 기반의 묵음 감지 알고리즘을 사용했습니다. sounddevice 라이브러리로 수신된 오디오 파형의 에너지를 분석하여, 진행자가 말을 하지 않는 구간(Silence)에서는 AI 추론을 건너뛰도록 설계했습니다. 이를 통해 불필요한 연산 부하를 줄이고 시스템의 반응 속도를 최적화했습니다.

I/O 최적화 및 안정적 파이프라인

- logging 레벨을 CRITICAL 레벨로 조정하고 불필요한 출력을 억제하여 프로세스 간 통신(IPC) 시 발생할 수 있는 데이터 노이즈를 줄였습니다. 분석된 데이터는 JSON 포맷으로 직렬화되어 C# 프론트엔드로 전송됩니다.

Frontend 핵심 알고리즘 및 스크롤 제어 기술

단순히 '들리는 단어를 찾는 것'으로는 생방송에서 사용하기가 어렵습니다. 발화자가 조사를 틀리거나 단어를 바꿔 읽더라도 시스템은 정확한 위치를 찾아야 합니다. 그리고, 원고에 없는 내용을 말하는 이른바 에드리브 상황에도 프롬프트는 안정적으로 운용되어야 합니다. 이를 위해 우리는 '편집 거리 알고리즘(Levenshtein Distance)'을 기반으로 한 독자적인 유사도 판단 로직을 C#으로 구현했습니다. 그리고, 방송용 프롬프트의 생명은 '부드러움'입니다. 일반적인 텍스트 박스의 스크롤 기능은 푹푹 끊기는 느낌을 주어 방송용 프롬프트에 사용하기 부적합합니다. 이를 해결하기 위해 운영체제 레벨의 API를 직접 제어했습니다.

유사도 계산(Similarity Computation)

- ComputeSimilarity 함수는 STT가 인식한 문장과 원고의 각 라인을 비교하여 유사도 점수를 산출합니다. 오타자가 발생해도 가장 높은 점수를 가진 라인을 '현재 읽는 위치'로 판단하는 퍼지(Fuzzy) 매칭 기술을 적용했습니다.

진행 흐름 보정

- 단순히 점수만 높다고 이동하지 않습니다. 현재 스크롤 위치를 기준으로 역방향 이동 시 페널티를 부여하고, 정방향 이동 시 가산점을 주는 로직을 통해 스크롤이 튀는 현상(Jittering)을 방지했습니다.

Win32 API 픽셀 스크롤링

- user32.dll의 SendMessage 함수와 EM_SETSCROLLPOS 메시지를 직접 호출하여, 라인 단위가 아닌 픽셀(Pixel) 단위의 정밀 스크롤을 구현했습니다. 이를 통해 부드러운 텍스트 이동이 가능해졌습니다.

동적 속도 조절(Dynamic Speed Regulation)

- 발화자의 발화 속도에 맞춰 스크롤 속도는 실시간으로 연동됩니다. 발화자가 원고보다 앞서가면 가속하고, 뒤처지면 감속하며, 지정된 '뷰포트(Viewport)' 범위를 벗어나면 SafetyTimer가 작동하여 즉시 스크롤을 멈추고 진행자를 기다리도록 설계했습니다.

AI 스마트 프롬프트 UI

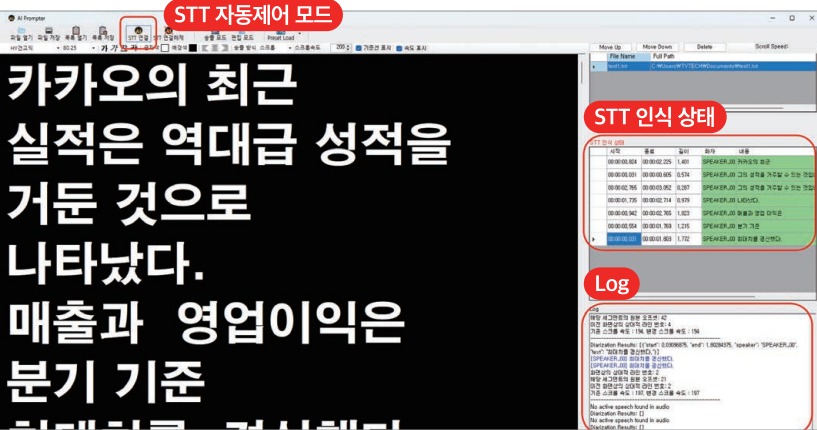


그림 4. 편집 모드

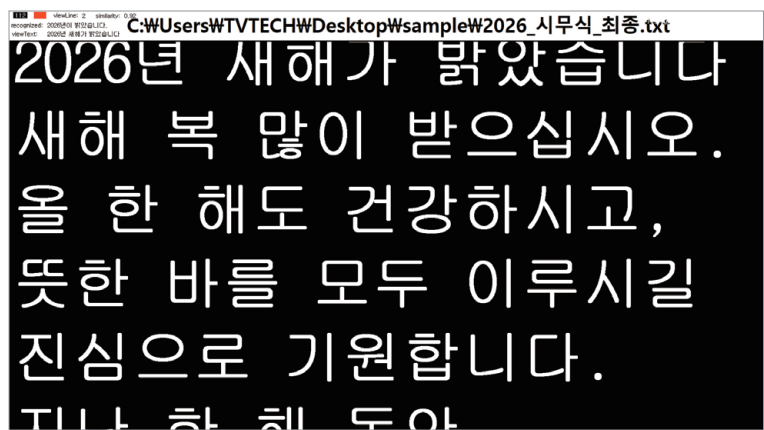


그림 5. 송출 모드

오토스크롤 상태 모니터링 기능

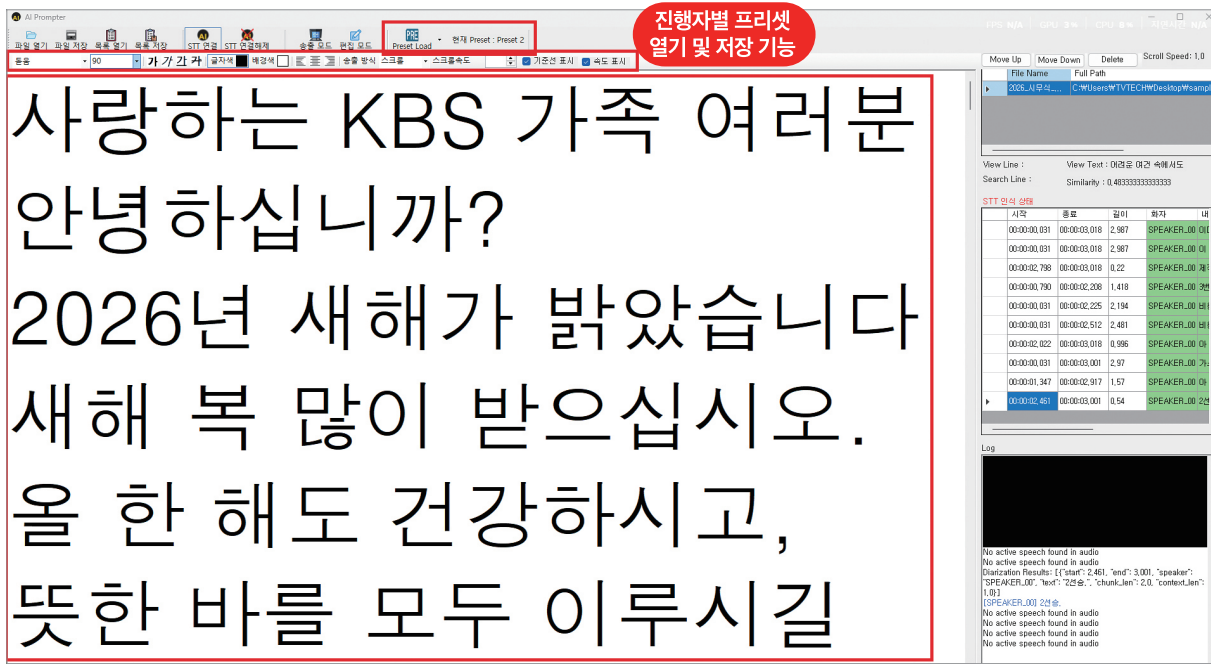


그림 6. 진행자별 프리셋 열기 및 저장 기능

결론

‘AI 스마트 프롬프트’의 개발은 단순히 기술적인 성취를 넘어, 방송 제작 워크플로우에 ‘AI’를 성공적으로 안착시킨 실증적 사례라는 점에서 의미를 가질 수 있습니다.

첫째, 제약이 혁신의 도구가 될 수 있음을 증명했습니다.

제작기술국 내 GPU 서버가 부재한 열악한 인프라 환경은 역설적으로 효율적이고 가벼운 시스템을 고안하게 만든 원동력이 되었습니다. 우리는 무거운 영상 처리 대신 ‘음성’에 집중함으로써, 고가의 GPU 서버 없이 일반 GPU 노트북만으로도 기술을 완성하여, 대규모 예산 투입 없이도 현업에 AI 기술을 적용할 가능성을 보였습니다.

둘째, 경제적 가치와 운용의 안전성을 동시에 확보했습니다.

자체 개발을 통해 라이선스 비용을 ‘0원’으로 만들었으며, 예산 절감 효과를 기대할 수 있게 되었습니다. 무엇보다 사람의 개입을 최소화한 ‘핸즈프리(Hands-free)’ 자동 제어시스템은 인적 오류로 인한 방송 사고의 위험성을 줄이고, 예측 가능한 운용 안정성을 확보했습니다.

셋째, 사용자 중심의 지속 가능한 생태계를 구축했습니다.

기존 상용 제품이 제조사의 업데이트에 의존해야 했던 것과 달리, 우리는 현업 사용자의 피드백을 즉각 반영할 수 있는 자체 기술력을 확보했습니다. 최대 글자 크기 확대, 개인화된 프리셋 기능 등은 실제 현업의 불만 사항을 개선한 대표적인 사례입니다.

최근 구축이 완료된 TS-14 AI 스튜디오에 고성능 GPU 서버가 도입되었습니다. 앞으로 고성능 GPU 서버에서 보다 안정적으로 동작하도록 시스템을 고도화하고 실제 녹화에 본격 적용할 계획이며, 다양한 출연자와 돌발 상황에 대한 데이터를 축적하여 알고리즘을 계속 고도화할 예정입니다. 또한, 이번 GPU 서버 도입으로 개발 초기에 우리를 좌절하게 했던 하드웨어의 제약이 사라지게 됩니다. 우리는 AI 스마트 프롬프트 개발 성공 경험을 바탕으로, 그동안 시도하지 못했던 AI 기반 영상처리 기술에도 과감히 도전할 예정입니다.



그림 7. 2025 미디어창의기술전 시상식

‘AI 스마트 프롬프트’는 시작일 뿐입니다. 이번 대상 수상작이 방송기술의 AI 전환을 가속하는 마중물이 되어, 앞으로 더 많은 창의적인 기술들이 현장을 누비기를 기대합니다. 📺