음성 및 영상 인식 인공지능 모델을 활용한 실시간 의성어 증강현실 시스템

조예원, 심현화, 이다민, 정의현 ¦ 코리아IT아카데미 강남

기존 AI는 사람의 발소리와 간단한 소리를 영상 하단의 자막으로 표현할 수 있으나, 영상 내 어느 위치에서 어떤 소리가 발생하는지 시각적으로 보여주지는 못한다.
본 프로젝트는 음성 및 영상 인식 모델을 구축하여 실제로 소리가 발생하는 곳에서 생동감 있는 의성어를 생성하는 것을 목표로 하였다. 이는 청각 장애인의 콘텐츠 소비와 생활을 보조할 수 있다.

I. 서론

청각 장애인의 영상 콘텐츠 소비는 자막을 이용하거나 VR 등의 체험형 매체를 활용하는 수준에 머물러 있으며 그중 대다수가 자막을 보고 콘텐츠를 소비한다[1,2]. 따라서 청각 장애인의 영상 콘텐츠 소비에 있어 자막의 중요성이 높으나, 현재 제공되는 자동 생성 자막의 낮은 정확도는 소비의 질에 영향을 주는 주요 한계점으로 지적된다[1,2,3]. 본 프로젝트는 이러한 자막의 질적 개선을 목표로 하며, 이를 위해 객체에서 발생하는 소리를 의성어로 표현하고 그 위치에 적절한 크기의 애니메이션을 동반한 직관적인 자막을 생성하고자 한다.

프로젝트 진행에 앞서 2가지의 선행 사례로 HoloSound[3]와 BeyondHearing[4]을 소개한다. 첫째, HoloSound는 증강현실 전용 하드웨어와 결합하여, 인식한 안면의 하관 바로 아래에 음성 자막을 고정하거나 다양한 소리(물 내리는 소리, 걷는 소리)에 대한 설명을 뷰포트 하단에 표시한다[3]. 이를 통해 소리를 시각적으로 ‘설명’한다. 그러나 이 사례는 소리에 대한 직관적, 공간적 정보가 부족하며, 소리의 분위기나 느낌을 ‘표현’하지 못하는 한계가 존재한다. 둘째, BeyondHearing은 Respeaker V2.0 센서를 활용해 소리의 방향과 위치를 감지하고 구분한 뒤, 소리에 상응하는 그래픽을 AR글래스 위에 투영하는 방식으로 소리를 직관적으로 표현한다[4]. 위 사례는 소리의 주체에 따라 그래픽의 형태를 달리 표현하지만, 사전 제작된 그래픽을 물체 표면에 따라 약간만 변형시킬 뿐 실제 소리를 듣고 이에 적절하게 그래픽을 변형하는 방식은 아니므로 소리 자체의 특성을 정확하게 보여주지 못한다는 한계가 있다.

본 프로젝트는 두 선행 사례에서 부족하다고 판단되는 소리의 위치, 생동감, 실제 소리의 표현을 개선하고자 하며, 이를 위해 자막에 애니메이션 효과를 추가하였다. 다만, 본 프로젝트의 핵심은 의성어를 올바른 위치에 생성하는 것이므로, 애니메이션은 차후 연구를 위한 방향 제시 차원에서 간단하게 구현하였다.

본 프로젝트는 다음과 같은 의의가 있다. 첫째, 상대적으로 발전이 더딘 분야에 관심을 유도한다. 둘째, 필요한 데이터셋의 탐색을 통해 향후 연구를 위한 발전 방향을 제시한다. 마지막으로 청각이 불편한 이들을 위한 새로운 방향의 AI 활용 가능성을 제시한다.

II. 본론

2-1. 데이터셋

본 프로젝트에 적합한 데이터는 이미지와 소리를 포함하며, 각 데이터에는 레이블과 객체 탐지를 위한 최소한의 좌표, 그리고 의성어가 포함되어야 한다. 그러나 의성어가 포함된 소리-이미지 데이터셋은 기존에 존재하지 않았다. 이에 따라, 프로젝트를 수행하기 위해 기존 소리-이미지 데이터셋에 의성어 레이블을 추가하기로 정하였다.

기반 데이터셋으로 이미지 사운드 매칭 데이터[5]를 활용하였으며, 해당 데이터 중 '국 끓이기, 면 끓이기, 고구마 튀기기, 닭고기 굽기, 냉장고 작동하기, 당근 썰기, 헤어드라이어 작동하기, 사과 썰기, 전기 압력밥솥 사용하기, 믹서기 사용 야채 갈기, 전자레인지 작동하기' 총 11종의 데이터를 선별하였다. 이들 데이터는 3단계 범주 중 차상위 수준에서 분류하여 '굽기, 끓이기, 냉장고 사용, 드라이어 사용, 믹서기 사용, 썰기, 압력밥솥 사용, 전자레인지 사용, 튀기기' 총 9개 클래스로 분류하였다.

MODEL	ASUS Vivobook pro 15 M3500QC Laptop
OS	Window 11 Home
CPU	AMD Ryzen 5800H
GPU	NVIDIA GeForce RTX 3050 4GB
RAM	DDR4 16GB

Table 1. 프로젝트 학습 장비

오디오 데이터셋은 학습용 539개, 검증용 66개, 비디오 데이터셋은 학습용 6,093개, 검증용 771개로 구성하였으며, 시험 데이터는 각각 33개, 1,716개이다.

의성어 데이터를 구축하기 위해, 각 오디오 데이터에 적합한 의성어를 분류할 필요가 있었다. 이를 위해 keras의 사전 학습 모델인 OpenL3[6,7]로 음향적 특성을 추출한 뒤, k-평균 군집 분석을 적용해 생성된 각 군집에 부합하는 의성어를 할당하였다. 이 과정에서 기존 어노테이션이 포함된 JSON 파일에 onomatopoeia 태그로 'Sizzle, Woooong, ai_speaking, tack-, glug glug, Whirrrrl, chzzzzzz, Whuuuuu, drrrrrr' 총 9개의 의성어를 삽입하였다.

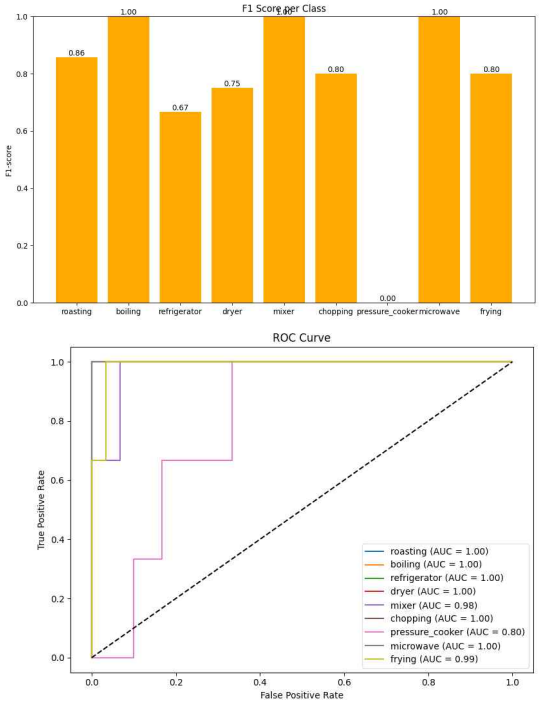

Fig 1. 소리 모델 지표

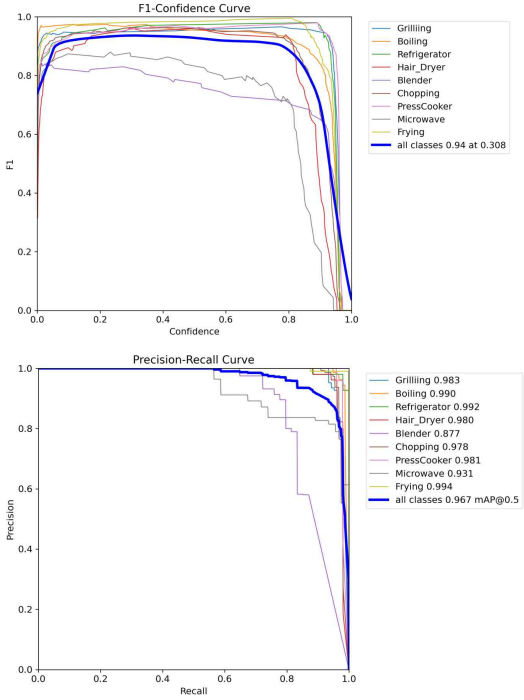

Fig 2. 영상 모델 지표

2-2. 모델 선정

본 프로젝트는 2개의 모델을 사용하였다. 하나는 소리 분류 및 의성어 라벨링을 위한 소리 모델이며, 다른 하나는 영상 객체 탐지 모델이다.

소리 모델은 ResNet 기반 커스텀 모델과 torch_vggish_yamnet 라이브러리로 구현되어있는 YAMNet[8,9]을 비교 및 평가하였다. 두 모델은 학습 및 평가에서 Precision 기준 YAMNet은 89%, 커스텀 모델은 94%로 후자가 더 높은 성능을 보였다. 두 모델의 성능은 유사하지만, 커스텀 모델은 의성어 라벨 출력을 위한 파인 튜닝이 쉽다는 장점이 있으므로, 이를 프로젝트의 소리 모델로 선정하였다.

객체 탐지 모델은 ResNet 기반 커스텀 모델과 PyTorch 라이브러리에 구현되어있는 SSD[10,11], 그리고 YOLOv8 라이브러리[12,13]를 비교 및 평가하였다. 세 모델을 mAP.5 기준으로 평가한 결과, 각각 17%, 42%, 94%의 성능을 기록하였다. 이 중 YOLOv8 모델이 월등히 높은 성능과 속도를 보여 객체 탐지 모델로 선정되었다.

각각의 모델 학습은 하드웨어의 한계로 인하여 4~8 배치 범위에서 가중치 누적 기법을 활용해 100에포크 동안 진행되었다. 커스텀 모델과 YAMNet은 CosineAnnealingLR을 사용하였으며, SSD는 MultiStepLR, YOLO는 라이브러리에 내장된 기존 방식을 사용하였다.

2-3. 텍스트 애니메이션

프로젝트 기획 단계에서 생동감 있는 의성어 표현을 위해 다양한 효과가 가미된 자막을 제안하였다. 그러나 본 프로젝트에서는 의성어를 올바른 위치에 생성하는 것에 집중하여 전체적으로 잘 어울리는 하나의 애니메이션만 적용하였다. 또한 프로젝트의 주요 라이브러리인 opencv 환경에서 해당 기능의 정상적인 작동을 위해 별도의 라이브러리를 구축하였다.

2-4. 프로젝트 스크립트

프로젝트 스크립트는 선정한 모델과 애니메이션 라이브러리를 기반으로 구성하였다. 소리 모델은 sounddevice 라이브러리를 통해 전달받은 데이터로 소리 분류와 의성어 라벨링을 수행하였다. 학습 시 무음에 대한 클래스를 별도로 할당하지 않았으므로, 성능 및 예측 결과를 향상시키기 위하여 torchaudio.funtional의 VAD(Voice Activity Detection)로 소리를 필터링한 후, MelSpectogram 형식으로 변환된 데이터를 모델에 전달하였다. 객체 탐지 절차는 opencv를 기반으로 구성하였으며, 객체 탐지 모델로 예측한 bbox의 위치를 애니메이션 라이브러리에 전달할 경우, bbox 영역 내 불특정 위치에서 시각적 효과가 적용된 자막이 생성되도록 구현하였다. 소리가 발생하는 물체를 정확히 판별하기 위해 소리 모델의 분류와 객체 탐지 모델의 레이블이 일치하는 순간에만 의성어를 출력하기로 기획하였으나, 모델 학습 전략의 오판과 소리 수집 장치의 한계로 인하여 해당 기능은 올바르게 작동하지 않았다. 이에 대해서는 결과에서 심층적으로 논의한다.

Fig 3. 프로젝트 실행. 압력밥솥

III. 결과

본 프로젝트의 실행에서는 노트북 내장 마이크 배열과 내장 카메라 및 별도의 외장 마이크 2대를 사용하였다.

실행 과정에서 bbox와 의성어를 올바르게 추론하였으나, 가정의 주방 환경에서 나오는 잡음들로 인하여 소리 모델의 분류가 정확하지 않았다. 또한, 대부분의 식별 대상 소리는 사람의 청각으로는 구별하기 어려운 유형이었으므로 테스트 결과와 전혀 다른 실행 결과를 얻었다. 따라서 본래 의도하였던 소리 모델과 객체 탐지 모델의 분류 일치에 따른 의성어 위치 결정 절차는 생략되었다. 또한, 소리 모델은 전기압력밥솥을 한 차례도 정확히 분류해내지 못하였다.

이는 데이터셋의 선택과 모델 학습 전략에 따른 한계에서 발생한 문제로 유추된

다. 이미지 사운드 매칭 데이터[6]는 이미지와 사운드 간의 상호연관성을 통해 이미지로 사운드를, 사운드로 이미지를 생성하는 과제에 적합한 데이터이다. 이는 본 프로젝트에 부합하지 않는 데이터였다. 특히, 하나의 긴 클립에서 여러 개의 이미지와 소리를 추출하는 방식은 데이터의 양은 많았지만 다양한 상황에서 일반화된 모델로 학습시키기에는 부족하였다. 전기압력밥솥의 경우, 데이터의 양은 다른 클래스와 유사하였으나 소리 데이터의 일관성이 낮았다. 밥솥을 여닫는 소리, 취사 후 김 빼는 소리, 각종 설정과 완료에 대한 기계 음성 등 여러 소리를 포함하지만, 이를 전처리하지 않아 소리 모델의 성능을 저해하는 요소로 작용하였다.

OpenL3와 k-평균 군집 분석을 통해 의성어를 정의하는 방식은 한국어 의성어 데이터셋 부재의 한계를 극복하기 위한 선택이었다. 그러나 이는 의성어의 주관성과 군집 분류의 한계로 인해 소리의 미묘한 차이를 의성어로 정확히 표현하는 것에 한계를 가졌다. 더욱 정교한 의성어 표현을 위해서는 사람이 직접 청취하여 작성한 라벨링과 자연어 처리 기술과의 연계를 통해 의성어 생성의 유연성을 확보하는 방안을 모색해볼 수 있다.

모델 학습 과정에서는 하드웨어의 제약으로 인해 데이터 증강을 수행하지 못하였다. 또한, 마이크를 바꾸어 실행할 때마다 각기 다른 결과를 출력하였다. 노트북 내장 마이크의 경우 소음 억제 기능이 작동하여 특정 분류의 소리를 인식하지 못하였고, 한편 purer 마이크는 심한 잡음으로 인해 90% 이상의 음성을 ‘chzzz’라는 의성어로 분류하는 현상이 관찰되었다. 이를 통하여 마이크의 음향적 특성에 따라 예측되는 의성어나 인식의 범위가 크게 달라질 수 있음을 확인할 수 있었다.

Ⅲ. 결론

본 프로젝트의 한계점을 극복하기 위해서는 과제에 특화된 구체적인 방법론을 설계해야 하며, 그중에서도 과제에 부합한 데이터셋을 개발, 구축, 검증하는 것은 연구를 위한 가장 중요한 절차이다. 또한, 더 많은 물체와 소리에 대해서도 모델을 일반화하기 위해서 자연어 처리 기술과의 연동이 필수적이다.

한편, 자막에 생동감을 주기 위해서는 영상 편집의 원본 파일(PRPROJ파일, FCPXML파일, AEP파일 등)의 학습과 같은 추가적인 연구가 요구된다. 이와 더불어, 다양한 의성어를 적절한 상황에 생성하기 위한 데이터셋 구축에 대해서도 지속적인 관심이 필요하다.

본 프로젝트는 소리가 발생하는 객체를 정확히 판별하지 못하는 문제를 해결하지 못한 채 마무리하였다. 이는 기획 초기 프로젝트의 목표를 달성하지 못하였음을 의미한다. 그러나, 의성어라는 새로운 방식으로 소리를 표현하려는 시도는 향후 연구를 위한 시행착오이자 중요한 방향성을 제시하는 의미 있는 결과라 할 수 있다. 이러한 시도를 바탕으로 향후 더욱 고도화된 자막 생성 기법이 구축되어, 청각 장애인이나 난청인을 위한 영상 콘텐츠 접근성과 삶의 질을 향상하는 제품 개발로 이어지

기를 기대한다.

참고문헌

[1] C.H. Ahn, “장애인방송 기술개발 현황”, ETRI, 2019, doi:10.22648/ETRI.2019.J.340301

[2] S. Kafle, M. Huenerfauth, “Evaluating the Usability of Automatically Generated Captions for People who are Deaf of Hard of Hearing.” in proc. the 19th Int. ACM SIGACCESS Conf. Computers and Accessibility, pp. 165-174, doi: 10.1145/3132525.3132542

[3] P. Millett, “Accuracy of speech-to-text captioning for students who are deaf or hard of hearing,“ J. Educ. Pediatr. Rehabil. Audiol., vol. 25, 2021.

[4] R. Guo, Y. Yang, J. Kuang, X. Bin, D. Jain, S. Goodman, et al, “HoloSound: Combining speech and sound identification for deaf or hard of hearing users on a head-mounted display,“ in Proc. 22nd Int. ACM SIGACCESS Conf. Computers and Accessibility (ASSETS ’20), Virtual Event, USA, Oct. 2020, pp. 1-4, doi: 10.1145/3373625.3418031.

[5] “Beyond Hearing: An AR interface for Deaf people to localise sound,“ DesignSpark, Jul. 4, 2022. [Online]. Available: https://www.rs-online.com/designspark/beyond-hearing-an-ar-interface-for-deaf-people-localise-sound

[6] 전주대학교 산학협력단 컨소시엄, “이미지 사운드 매칭 데이터,” AI Hub, Dec. 2023. [Online]. Available: https://www.aihub.or.kr/aihubdata/data/view.do?searchKeyword=%EC%9D%B4%EB%AF%B8%EC%A7%80+%EC%82%AC%EC%9A%B4%EB%93%9C&aihubDataSe=data&dataSetSn=71602

[7] marl, “OpenL3: Open-source deep audio and image embeddings,” GitHub, 2025. [Online]. Available: https://github.com/marl/openl3.

[8] Aurora Cramer, Ho-Hsiang Wu, Justin Salamon, and Juan Pablo Bello. “Look, Listen and Learn More: Design Choices for Deep Audio Embeddings,” IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), pages 3852-3856, Brighton, UK, May 2019.

[9] Google, “Sound classification with YAMNet,“ TensorFlow, 2024. [Online]. Available: https://www.tensorflow.org/hub/tutorials/yamnet

[10] S. Giacomelli, “torch-vggish-yamnet: PyTorch VGGish & YAMNet models,“ PyPI, 2024. [Online]. Available: https://pypi.org/project/torch-vggish-yamnet/

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, “SSD: Single Shot MultiBox Detector,“ arXiv preprint arXiv:1512.02325, Dec. 2015, doi:

10.48550/arXiv.1512.02325. [Online]. Available: https://arxiv.org/abs/1512.02325
[12] PyTorch, “pytorch/vision: references/detection,“ GitHub, 2025. [Online]. Available: https://github.com/pytorch/vision/tree/main/references/detection
[13] R. Varghese, M. Sambath , “YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness,” IEEE, Apr. 2024, doi: 10.1109/ADICS58448.2024.10533619. [Online]. Available: https://ieeexplore.ieee.org/document/10533619
[14] Ultralytics, “Ultralytics YOLO Documentation,“ [Online]. Available: https://docs.ultralytics.com/. Accessed: Jun. 24, 2025.