JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 27, No. 3, pp. 857-866, Mar. 2026

Check for updates

실시간 ASL 지문자 인식을 위한 순차적 이벤트 기반 디코딩 기법

유 승 수[1] · Khoa Nguyen[2] · Thong-Nhat Tran[3] · 서 영 욱[4*]
[1]대전대학교 융합컨설팅학과 박사수료
[2]충북대학교 정보통신공학부 석박사과정
[3]충북대학교 정보통신공학부 계약교수
[4]대전대학교 융합컨설팅학과 지도교수

Monotonic Event-Triggered Decoding for Real-Time ASL Fingerspelling Recognition

Seung-Su Yu[1] · Khoa Nguyen[2] · Thong-Nhat Tran[3] · Young-Wook Seo[4*]

[1]PhD (ABD), Department of Convergence Consulting, Daejeon University, Daejeon 34520, Korea
[2]Master's/Doctor's Course, Department of Information and Communication Engineering, Chungbuk National University, Cheongju 28644, Korea
[3]Professor, Department of Information and Communication Engineering, Chungbuk National University, Cheongju 28644, Korea
[4]Professor, Department of Convergence Consulting, Daejeon University, Daejeon 34520, Korea

[요 약]

본 논문은 실시간 미국 수어(ASL, American Sign Language) 지문자 인식을 위한 이벤트 기반 디코딩 전략을 제안한다. 매 프레임마다 디코딩하는 대신, 모델 출력 변화가 임계값을 초과할 때에만 연산을 수행하여 불필요한 계산을 줄이고 효율성을 높인다. 또한 단조 제약 조건을 적용하여 시간적 일관성을 확보하고 역방향 및 진동 예측을 차단함으로써 연속 수어 환경에서의 안정성을 강화한다. 실험 결과, 제안 방법은 기존 대비 지연 시간을 크게 줄이면서 인식 정확도도 향상시켰다. 이벤트 기반 디코딩과 단조 제약의 결합은 응답성 높은 지문자 시스템 구현에 실용적 해법을 제시한다.

[Abstract]

This study introduces a sequential, event-based decoding strategy for real-time American Sign Language (ASL) fingerspelling recognition. The study aims to lower computations and end-to-end delay while making predictions more stable. It activates computation only when informative evidence appears. Instead of decoding at every frame, the system triggers decoding only when changes in the model's output exceed a preset threshold, removing redundant steps and improving efficiency for streaming input. To further stabilize the results, we apply a monotonic constraint that enforces temporal consistency and blocks backward or oscillating predictions during sequence generation. This forward-only progression increases reliability in continuous signing. Experiments show that the method significantly reduces latency compared to prior approaches and also improves recognition accuracy. Overall, event-driven decoding coupled with monotonic constraints offers a practical solution for responsive ASL fingerspelling systems.

색인어 : 미국 수어, 지문자 인식, 이벤트 트리거 디코딩, 실시간 추론, 단조 디코딩
Keyword : ASL, Fingerspelling Recognition, Event-Triggered Decoding, Real-Time Inference, Monotonic Decoding

http://dx.doi.org/10.9728/dcs.2026.27.3.857

Received 12 February 2026; **Revised** 06 March 2026
Accepted 11 March 2026

***Corresponding Author; Young-Wook Seo**

Tel: +82-42-280-4185
E-mail: ywseo@dju.kr

Ⅰ. 서 론

자동 지문자 인식은 실질적인 수어 텍스트 입력과 접근 가능한 인간-컴퓨터 상호작용을 향한 핵심 단계이다. 미국 수어(ASL)에서 지문자는 고유명사와 어휘 사전에 없는 단어들을 빠른 문자 시퀀스로 표현하는데, 이 과정에서 연속 발음(coarticulation)과 빈번한 가림(occlusion)이 발생한다. 이러한 조건은 순수한 프레임 단위 분류기나 표준 시퀀스 모델에 큰 도전을 제기한다.

얼굴, 손, 신체 키포인트를 결합한 랜드마크 기반 파이프라인은 온디바이스 처리를 가능하게 만들었지만, 비디오 프레임과 문자 사이의 본질적인 좌에서 우 정렬을 존중하면서도 견고하고 저지연의 디코딩을 구현하는 것은 여전히 열린 문제로 남아 있다.

연결주의 시계열 분류(Connectionist Temporal Classification, CTC)는 입력 프레임 시퀀스 보다 짧은 라벨 시퀀스를 위한 강력한 비정렬(alignment-free) 목적 함수이며, 많은 실제 음성 시스템에서 사용되어 왔다. CTC의 좌에서 우로의 경로 구조와 효율적인 동적 프로그래밍은 스트리밍 가능한 학습과 디코딩을 가능하게 한다[1]. 그러나 비-블랭크(non-blank) 라벨 간의 조건부 독립 가정은 문맥 모델링에 한계를 주며, 이는 CTC의 스트리밍 가능성을 유지하면서 제한적인 문맥을 추가하는 하이브리드 설계를 촉발하였다[2].

단조 어텐션(monotonic attention)과 단조 청크 단위 어텐션(monotonic chunkwise attention)은 좌에서 우 정렬을 강제하거나 근사하면서도 국소적 문맥을 허용하는 온라인 어텐션 메커니즘을 제공하며, 오프라인 소프트 어텐션에 필적하는 정확도를 달성한다[3]-[6].

최근 인코더인 Conformer는 합성곱과 자기어텐션을 결합하여 시계열 신호(예: 음성)의 시간적 모델링을 개선하면서도 스트리밍 변형과 호환된다[7]. 산업 시스템에서는 순환 전이(recurrent transducer)와 트랜스포머 전이(transformer transducer)를 통해 모바일 장치에서의 스트리밍 종단간(end-to-end) 인식이 실용적임을 보여주었으며, 이는 정확하면서도 지연을 명시적으로 제어할 수 있는 모델의 필요성을 더욱 강조한다[8]-[10].

이와 병행된 연구에서는 트리거드 어텐션(triggered attention)과 기타 정렬 인식 메커니즘을 온라인 디코딩에 맞게 개선하여 지연-정확도 트레이드오프를 향상시켰다[9],[10].

시각적 측면에서, MediaPipe 패밀리에 의해 견고하고 빠른 랜드마크 추출이 가능해졌다.

MediaPipe는 플랫폼 간 인지 그래프와 온디바이스 손·얼굴 기하 모델을 제공하며, 상용 하드웨어에서도 대화형 프레임 속도로 얼굴-손-자세 파이프라인을 지원한다[11],[12]. 이러한 랜드마크를 기반으로, 실제 환경에서의 지문자 연구는 정제된 데이터셋과 실제 조건에 맞춘 인식 모델을 제시하였으며, 반복적 어텐션 디코더와 수어자 변이 분석을 포함한다[13]-[15].

최근 ASL 지문자 커뮤니티 챌린지는 매우 큰 규모의 랜드마크 코퍼스를 공개하여 스트리밍에 적합한 방법론과 온라인 텍스트 입력을 위한 표준화된 평가를 촉진하고 있다[16].

본 연구는 Google-Kaggle 미국 수어(ASL) 지문자 인식 대회[1]를 기반으로 하며, 실용적 통찰은 은메달 수상작 "2 Lines of Code Change (LB 0.760+)" 솔루션[17]으로부터 얻었다.

1-1 연구 배경 및 필요성

종단간 수어 인식에서 최근의 진전에도 불구하고, 지문자는 여전히 도전적인 과제로 남아 있다. 그 이유는 문자들이 빠르게 연속적으로 나타나면서 강한 연속 발음(coarticulation), 자기 가림(self-occlusion), 그리고 수어자 간 변이가 빈번하게 발생하기 때문이다.

기존의 CTC 기반 모델은 스트리밍 가능한 학습을 가능하게 하지만, 실제 배포 시점에서 언제 예측을 확정해야 하는지, 그리고 정확도와 지연 사이의 균형을 어떻게 맞출지에 대한 문제는 여전히 미해결 상태로 남아 있다[1].

한편, MediaPipe와 같은 랜드마크 파이프라인은 상용 장치에서 손, 얼굴, 자세 특징을 효율적으로 추출할 수 있게 하지만[11],[12], 이러한 입력을 명시적인 지연 제어 기능을 갖춘 배포 가능한 스트리밍 인식기로 연결하는 단순한 프레임워크는 존재하지 않는다. 이러한 간극을 해소하기 위해, 본 연구에서는 이벤트-트리거 CTC 프레임워크를 제안한다.

이는 인과적(causal) 인코더와, 일관된 증거가 있을 때에만 출력을 확정하는 디코더를 결합한 것이다. 두 개의 직관적인 파라미터(k,τ)는 정확도-지연 트레이드오프를 직접적으로 제어하며, 외부 언어 모델이나 복잡한 탐색 없이 온디바이스 텍스트 입력에 적합한 재현 가능하고 경량의 방법을 제공한다.

1-2 연구의 주요 성과

본 논문의 주요 기여는 다음과 같이 요약 된다.

• 이벤트-트리거 디코딩. k −프레임 윈도우에서 평균 사후확률이 임계값 τ를 넘는 안정적인 증거가 있을 때만 기호를 확정하는 CTC 디코딩 전략을 제안한다. 이 방법은 반복을 본질적으로 압축하며 빔서치를 피한다. 이를 통해 기존 연구에 없던 투명하고 조정 가능한 정확도-지연 트레이드 오프를 제공한다.

주요 결과: 검증에서 (k, τ)=(4, 0.7)은 Score 0.5956과 0.349s 지연을 달성하였으며, (8, 0.7)은 0.5986과 0.440s 지연을 기록하였다.

• 통합된 정확도-지연 평가. 품질 지표로는 Score (1−CER)를 우선시하고, 방출 지연(emission lag), 확정률(commit rate), 그리고 실시간 계수(real-time factor, RTF)로 이를 보완한다. 여기서 CER은 정규화된 Levenshtein 거리 기반

문자 오류율이며, 낮을수록 좋다.

주요 결과: 선택된 두 검증 지점 모두 $RTF_{e2e} \approx 0.0025$(~392×실시간)으로 동작하며, e2e는 end-to-end를 의미한다.

• 스트리밍 준비된 인코더. CTC로 학습된 경량의 인과적 합성곱-트랜스포머 인코더는 미래 문맥이나 오프라인 어텐션 없이도 엄격히 온라인 방식으로 동작한다.

주요 결과: 추론 비용은 인코더가 대부분을 차지하며, 디코더의 오버헤드는 무시할 수준이다($RTF_{dec} = 9.9 \times 10^{-5}$). 여기서 dec는 디코더를 의미한다.

• 재현 가능하고 배포 가능한 파이프라인. 결정론적 학습 레시피(고정 시드, 결정론적 커널, 그래디언트 누적)를 제공하며, 전처리와 디코딩을 포함하는 단일 TensorFlow Lite 시그니처를 내보내어 온디바이스 사용을 지원한다.

주요 결과: 홀드아웃 테스트셋에서 우리의 최종 구성은 Score 0.5762 [0.5729, 0.5793], 0.725s 지연, $RTF_{e2e} = 0.0024$를 달성하였다.

• 강력한 실험적 성과와 벤치마크 아티팩트. 본 연구에서는 공정한 비교를 가능하게 하기 위해 표준화된 표/그림(검증 그리드와 파레토 전선, 효율성, 강인성)을 공개한다.

II. 문제정의 및 성능지표

이 절에서는 랜드마크 기반 스트리밍 지문자(task)를 공식화하고 이를 평가하기 위한 메트릭을 정의한다. 입력-출력 공간, CTC 목적 함수로 학습된 인과적(causal) 인코더, 그리고 정확도-지연 트레이드오프를 제어하는 두 파라미터(k,τ) 기반 이벤트-트리거 디코더를 명시한다. 여기서 k는 안정성 윈도우 길이(프레임 단위)이고, $\tau \in (0,1)$는 출력을 확정하기 위해 요구되는 윈도우 평균 사후 확률이다. 디코더는 동일한 비-블랭크 우승 클래스가 k 연속 프레임 동안 유지되고 평균 사후 확률이 $\bar{p}_t \geq \tau$일 때만 시점 t에서 문자를 출력한다. k 또는 τ가 클수록 품질은 향상되지만 지연이 증가한다. 기본 품질 지표로는 Score (1 − CER)를 정의하고, 지연 및 효율성 지표(방출 지연, 커밋률, 실시간 계수)와 함께 사용하며, 균형 및 저지연 운용 지점을 선택하기 위한 파레토 기반 절차를 기술한다.

2-1 문제설정

각 입력 클립은 T 프레임과 프레임당 F 특징(손, 얼굴, 자세)으로 이루어진 랜드마크 시퀀스 $X \in R^{T \times F}$이다. 정답 라벨 시퀀스는 $y = (y_1, \ldots, y_J)$이며, 각 $y_j \in V$이다. 여기서 V는 문자 어휘이고 $\varnothing$는 CTC 블랭크를 의미한다. 학습 가능한 파라미터 θ로 매개화된 인과적 인코더 f_θ는 X를 프레임별 로짓 $z_t \in R^{|V|+1}$로 매핑한다. 즉, $z_t = f_\theta(X)_t$이고, 사후 확률은 $p_t(c|X) = softmax(z_t)_c$로 정의된다($t = 1, \ldots, T, c \in V \cup \varnothing$). 예측된(디코딩된) 시퀀스는 $\hat{y}$로 나타낸다.

2-2 CTC 학습 목적 함수

학습은 $\{p_t(\cdot|X)\}_{t=1}^{T}$ 가 유도하는 CTC 분포에서 y의 음의 로그우도(negative log-likelihood)를 최소화한다. $B : (V \cup \varnothing)^T \rightarrow V^*$는 블랭크를 제거하고 연속 반복을 병합하는 연산자이다.

여기서 $\pi = (\pi_1, \ldots, \pi_T)$는 프레임 단위 정렬 경로이며, 각 프레임 라벨은 $\pi_t \in V \cup \varnothing$이다. 손실 함수는 다음과 같다.

$$L_{CTC}(X, y; \theta) = -\log \sum_{\substack{\pi \in (V \cup \varnothing)^T \\ B(\pi) = y}} \prod_{t=1}^{T} p_t(\pi_t|X) \quad (1)$$

표준 순방향-역방향 재귀로 계산되고, 인과적 인코더는 스트리밍 학습 및 추론을 가능하게 한다.

2-3 이벤트-트리거 스트리밍 디코딩

디코딩은 온라인으로 진행되며 일관된 증거가 관찰될 때만 문자를 출력한다. 프레임별 비-블랭크 우승 클래스는는 다음과 같다.

$$C_t = \arg\max_{c \in V} \; p_t(c|X) \quad (2)$$

슬라이딩 윈도우는 $k \geq 1$일 때 $W_t = \{t-k+1, \ldots, t\}$이다. 지시자(indicator) $\mathbb{I}[\cdot]$를 사용하여 안정성 플래그는 다음과 같다.

$$S_t = \mathbb{I}\left[C_u = C_t \;\; \forall u \in W_t\right] \quad (3)$$

윈도우 평균 사후 확률은 다음과 같다.

$$\bar{p}_t = \frac{1}{k} \sum_{u \in W_t} p_u(C_t|X) \quad (4)$$

임계값 $\tau \in (0,1)$와 마지막으로 출력된 토큰 $\hat{y}_{last}$가 주어졌을 때, 디코더는 다음과 같다.

$$S_t = 1, \; \bar{p}_t \geq \tau, \; C_t \neq \hat{y}_{last} \quad (5)$$

조건이 만족될 경우 시점 t에서 출력을 확정(commit)한다. 이후 C_t를 출력에 추가하고($\hat{y} \leftarrow \hat{y} \circ C_t$), $C_t \neq \hat{y}_{last}$로 갱신한다. 조건(5)는 즉각적인 반복을 구조적으로 압축하며 빔서치를 필요로 하지 않는다. (k, τ) 쌍은 정확도-지연 트레이드

오프를 제어하는 유일한 파라미터이다.

2-4 기본품질지표: Score

우리의 주요 성능 지표는 정규화된 Levenshtein 거리 기반 Keras 메트릭으로 구현된 Score이다.

$$CER(\hat{y}, y) = \frac{S + D + I}{|y|},\ Score(\hat{y}, y) = 1 - CER(\hat{y}, y) \quad (6)$$

여기서 S, D, I는 최적 대체(substitution), 삭제(deletion), 삽입(insertion) 횟수이고, $|y|$는 참조 문자 개수이다. Score는 [0, 1] 범위에 있으며, 값이 클수록 품질이 높다. 본 연구는 모델 및 하이퍼 파라미터 선택을 위해 검증 Score를 보고하며, 최종 평가에서는 부트스트랩 신뢰구간과 함께 테스트 Score를 보고한다.

2-5 지연 및 효율성 지표

스트리밍 인식기는 품질과 지연을 균형 있게 고려해야 하므로 Score와 함께 지연 및 계산 인식 지표를 보고한다. fps는 클립의 초당 프레임 수를 의미한다.

a) 방출 지연: 시점 t_j에서 문자 $\bar{y}_j$를 출력할 때, 최초 지배 시간은 다음과 같다.

$$t_j^{\star} = \min\{u \le t_j : C_s = \hat{y}_j \ \ \forall s \in \{u, \cdots, t_j\}, \quad \text{and } t_j - u + 1 \ge k\} \quad (7)$$

이 문자에 대한 방출 지연은 프레임 단위로 $lag_j = t_j - t_j^{\star}$, 초 단위로는 $lag_j^{(s)} = lag_j / fps$이다. 평균은 다음과 같다.

$$\overline{lag} = \frac{1}{|\hat{y}|}\sum_{j=1}^{|\hat{y}|} lag_j,\ \overline{lag}^{(s)} = \frac{\overline{lag}}{fps}. \quad (8)$$

b) 커밋률: 프레임당 문자수 $CR = |\hat{y}|/T$, 초당 문자수는 $CR^{(s)} = fps \cdot CR$이다.

c) 실시간 계수: 인코더 순방향 추론 시간 t_{fwd}, 동일 장치에서의 디코더 시간 t_{dec}, 클립 길이 d = T/fps가 주어졌을 때, 디코더 전용 및 종단간 실시간 계수는 다음과 같다.

$$RTF_{dec} = \frac{t_{dec}}{d},\ RTF_{e2e} = \frac{t_{fwd} + t_{dec}}{d} \quad (9)$$

값이 1보다 작으면 실시간보다 빠르게 동작함을 의미한다.

2-6 운용 지점 선택

검증셋에서 (k, τ) ∈ K × T 그리드를 탐색하여 각 설정의 $(Score, \overline{lag}^{(s)})$를 계산한다. 파레토 집합은 다른 설정이 더 높은 Score와 더 낮은 지연을 동시에 달성하지 못하는 설정들의 집합이다. 이 집합에서 두 지점을 보고한다: (i) Score를 최대화하는 균형 지점, (ii) $Score \ge (1-\delta)\, Score_{\max}$ 조건하에서 평균 지연 $\overline{lag}^{(s)}$를 최소화하는 저지연 지점(δ = 0.015). 두 지점 모두 부트스트랩 구간과 함께 홀드아웃 테스트셋에서 평가된다.

Ⅲ. 연구방법: 실시간 지문자 인식을 위한 이벤트 기반 CTC 방법론

제안하는 프레임워크는 랜드마크의 연속 스트림을 문자 시퀀스로 변환하면서 명시적인 지연 제어를 수행한다. 입력 클립 $X \in R^{T \times F}$는 표준화된 후 인과적(causal) 인코더 f_θ에 입력되며, 이 인코더는 문자 $c \in V$와 CTC 블랭크에 대한 프레임별 사후 확률 $p_t(c \mid X)$를 출력한다.

학습은 프레임 단위 라벨 없이도 좌에서 우 정렬을 학습할 수 있도록 CTC 목적 함수를 사용한다. 추론 시에는 이벤트-트리거 디코더가 짧은 윈도우에서 일관된 증거가 관찰될 때에만 문자를 확정한다. 두 개의 직관적인 파라미터가 동작을 제어하는데, k ∈ N (안정성 윈도우 길이)와 τ ∈ (0, 1) (평균 사후 확률 임계값)이다. 이 설계는 정확도와 지연 사이의 직접적인 트레이드오프를 노출하며, 온디바이스 배포에 충분히 단순하다.

3-1 방법 다이어그램

지연을 제어 학습은 인과적 인코더 출력에 CTC를 사용하며, 추론은 (k, τ) 기반 이벤트-트리거 규칙을 사용한다(그림 1 참조).

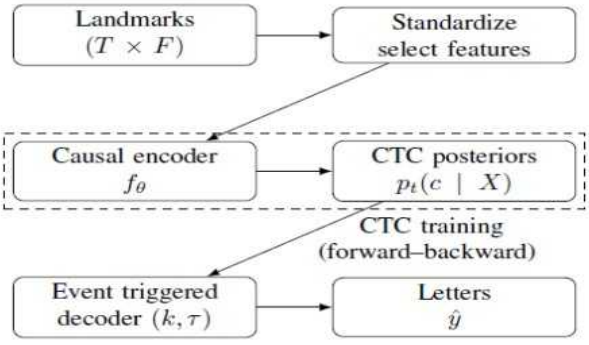


그림 1. 이벤트 트리거 블록 다이어그램
Fig. 1. Event trigger block diagram

3-2 전처리와 인과적 인코더

각 프레임은 손, 얼굴, 자세 랜드마크를 연결한 특징으로 구성된다. 그룹별 사전 계산된 평균과 표준편차로 표준화를 적용하고, 표준화 이후 NaN은 0으로 대체하며, 길이를 고정된 T로 패딩 또는 리사이즈한다. 인코더는 깊이별 인과 합성곱과 인과 마스킹 및 위치 인코딩을 포함한 경량 트랜스포머 블록을 적층한다. 모든 층은 시간적으로 인과적이므로 학습과 추론 모두 스트리밍 가능하다.

3-3 학습 목적 함수

$\Pi(y)$를 CTC 축소 결과가 정답 y인 프레임 경로의 집합이라 하자. 손실 함수는 다음과 같다.

$$L_{CTC}(X,y;\theta) = -\log \sum_{\pi \in \Pi(y)} \prod_{t=1}^{T} p_t(\pi_t|X) \tag{10}$$

표준 순방향-역방향 재귀로 계산된다. 학습은 라벨 스무딩 없이 진행하며, 코사인 또는 단계적 학습률 스케줄을 사용하고, 가중치 감쇠는 현재 학습률에 경량 콜백으로 연결된다. 그래디언트 누적을 통해 제한된 메모리 GPU에서도 큰 효과적 배치 크기를 지원한다. 재현성을 위해 시드와 결정론적 커널을 고정한다.

3-4 이벤트-트리거 스트리밍 디코딩

Algorithm 1 Event triggered CTC decoding

Require: Posteriors $\{p_t(\cdot \mid X)\}_{t=1}^{T}$, window k, threshold τ
Ensure: Output sequence $\hat{y}$

$\hat{y} \leftarrow [\,]$
last $\leftarrow$ None ▷ last committed symbol
buf $\leftarrow$ empty queue ▷ stores posteriors of current winner run
$s \leftarrow 0$ ▷ running sum of values in buf
$c_{\text{run}} \leftarrow$ None ▷ current run winner
for $t = 1$ to T **do**
  $C_t \leftarrow \arg\max_{c \in \mathcal{V}} p_t(c \mid X)$ ▷ $\mathcal{V}$: non-blank symbols
  $q_t \leftarrow p_t(C_t \mid X)$
  if $c_{\text{run}} =$ None **or** $C_t \neq c_{\text{run}}$ **then**
    $c_{\text{run}} \leftarrow C_t$
    clear buf; $s \leftarrow 0$
  push q_t into buf; $s \leftarrow s + q_t$
  if $|\text{buf}| > k$ **then**
    $s \leftarrow s-$ pop oldest from buf
  if $|\text{buf}| = k$ **then**
    $\bar{p} \leftarrow s/k$
    if $\bar{p} \geq \tau$ **and** $c_{\text{run}} \neq$ last **then**
      append c_{run} to $\hat{y}$
      last $\leftarrow c_{\text{run}}$
return $\hat{y}$

시점 t에서 비-블랭크 우승 클래스는 $C_t = \arg\max_{c \in V} p_t(c \mid X)$로 정의된다.

윈도우는 $W_t = \mathrm{t}-\mathrm{k}+1,\ldots,\mathrm{t}$이다. 커밋은 (i) 동일 클래스가 Wt에 걸쳐 유지되고, (ii) 평균 사후 확률이 τ 이상이며, (iii) 즉각적 반복을 방지하는 경우에만 발생한다. 알고리즘 1은 온라인 이벤트-트리거 디코딩을 나타낸다.

커밋 시점과 방출 지연. 문자 $\hat{y}_j$가 프레임 t_j에서 커밋될 때, 최초 지배 시간은 $C_s = \hat{y}_j$가 $\mathrm{s} \in \mathrm{u},\ldots,\mathrm{t_j}$에서 유지되며 구간 길이가 최소 k 이상인 가장 이른 $u \leq t_j$이다. 방출 지연은 t_{j-u}프레임이다. 작은 k 또는 큰 τ는 트레이드오프를 반대 방향으로 이동시키며, 두 값은 Score 기반 검증으로 조정된다.

3-5 CTC 학습 루프(스트리밍 인코더)

인과적 인코더의 CTC 학습 절차는 알고리즘 2에 제시된다.

Algorithm 2 Causal encoder training with CTC

Require: training set $\mathcal{D}$, epochs E, learning rate schedule $\{\eta_e\}$, weight decay rule $w(\eta)$
for $e = 1$ to E **do**
  for mini batch $(X, y) \sim \mathcal{D}$ **do**
    $z_t \leftarrow f_\theta(X)_t$ for t=1:T ▷ causal forward pass
    $p_t \leftarrow \mathrm{softmax}(z_t)$ for t=1:T
    $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{CTC}}(X, y; \theta)$
    update θ with optimizer step at rate η_e and decay $w(\eta_e)$

3-6 복잡도와 메모리 비용

인코더의 계산 비용은 $O(T \cdot C_{enc})$이며, C_{enc}는 프레임당 인과적 연산량이다. 디코딩 비용은 승자 계산에 $O(T \cdot |\nu|)$, 윈도우 평균 유지에 $O(1)$이다(승자가 바뀔 때만 큐가 초기화됨). 따라서 전체 비용은 T에 선형적이다. 디코더는 현재 승자를 위해 최대 k개의 사후 확률만 유지하므로, 메모리 비용은 인코더 상태 외에 $O(k)$이다.

3-7 실제 운용 지점 선택

검증셋에서 (k, τ) 그리드를 탐색하여 Score, 방출 지연, 효율성을 계산한다. 파레토 집합은 Score와 지연 모두에서 다른 설정에 의해 지배되지 않는 설정이다. 본 연구에서는 두 가지 운용 지점을 보고한다: Score를 최대화하는 균형 지점과 Score 허용오차 내에서 지연을 최소화하는 저지연 지점. 두 지점은 테스트셋에서 신뢰 구간과 함께 평가된다.

Ⅳ. 실 험

실험은 Python 3.9와 TensorFlow 2.10 환경에서 구현되었으며, 8GB 메모리를 가진 단일 NVIDIA RTX 2080 GPU

와 8코어 CPU에서 수행되었다. 제한된 GPU 메모리에도 불구하고 효과적인 대규모 배치 크기를 모사하기 위해 그래디언트 누적이 사용되었다. 학습률은 워밍업 후 코사인 감쇠 스케줄을 따르며, 가중치 감쇠는 경량 콜백을 통해 현재 학습률에 연동된다(일반적으로 0.05×lr). 동일한 전처리와 이벤트-트리거 디코더를 사용하여 온디바이스 추론을 위한 TensorFlow Lite 모델을 내보냈다.

4-1 데이터셋

본 연구는 최근 커뮤니티 챌린지에서 공개된 랜드마크 기반 ASL 지문자 코퍼스를 사용한다. 각 프레임은 오른손과 왼손(각 21 포인트), 얼굴과 입술, 상반신 자세 랜드마크를 연결하여 F = 276개의 특징(X, Y, Z 좌표)을 생성한다. 베이스라인 코드를 따라 각 그룹을 사전 계산된 평균과 표준편차로 표준화하고, 표준화 이후 결측값(NaN)은 0으로 대체하며, 모든 시퀀스를 고정 길이 T = 176으로 패딩 또는 리사이즈한다. 정답 라벨은 평가 어휘에 따른 문자 시퀀스로 구성되며, CTC를 위해 블랭크 기호가 예약되어 있다.

분할(Splits). 공식 학습 파티션으로 학습을 진행하며, 수어자별 층화 무작위 샘플링을 통해 홀드아웃 검증셋을 구성한다. 테스트셋은 모델 선택 과정에서 사용되지 않는다. 재현성을 위해 정확한 파일 리스트를 공개한다.

4-2 실험 설정

백본(Backbone). 인코더는 깊이별 인과 합성곱 블록과 인과 마스킹 및 위치 인코딩을 갖춘 경량 트랜스포머 블록을 적층한다. 모든 층은 시간적으로 엄격히 인과적이므로 스트리밍 학습과 추론이 보장된다.

학습(Training). 표준 CTC 손실(라벨 스무딩 없음)로 학습하며, AdamW 옵티마이저를 사용한다. 초기 학습률은 예를 들어 1.25×10^{-4}고, 가중치 감쇠는 학습률에 연동되며, 그래디언트 클리핑과 그래디언트 누적을 통해 효과적인 배치 크기 B_{eff}를 달성한다. 얼리 스토핑은 검증 Score를 모니터링하며 인내값은 P 에포크로 설정한다. 검증 Score 기준 최적 체크포인트가 보존된다. 모든 실행은 랜덤 시드를 고정하고 가능할 경우 결정론적 커널을 사용한다.

디코딩(Decoding). 스트리밍 디코딩은 안정성 윈도우 k와 평균 사후 확률 임계값 τ를 갖는 이벤트-트리거 규칙을 적용한다. 검증에서 (k, τ) ∈ {2, 3, 4, 5}×{0.60, 0.65, 0.70, 0.75, 0.80}를 탐색한다.

4-3 평가 프로토콜

주요 지표: Score. Score는 1 − CER로 정의되며, $CER = (S + D + I)/|y|$는 정규화된 Levenshtein 거리로 계산된다. 대체 S, 삭제 D, 삽입 I는 동적 프로그래밍 정렬로 계산된다. Score는 검증과 테스트 모두에서 결정적 기준이다.

지연 및 효율성. 평균 방출 지연(프레임 및 초 단위), 커밋률(프레임당 및 초당 문자 수), 디코더 전용 및 종단간 추론의 실시간 계수(RTF)를 측정한다. RTF < 1인 값은 실시간보다 빠른 동작을 의미한다.

운용 지점. 검증 단계에서 Score와 지연 모두에서 지배되지 않는 설정들의 파레토 집합을 형성한다. 여기서 (i) Score를 최대화하는 균형 지점, (ii) $Score \geq (1-\delta)\,Score_{\max}$ ($\delta = 0.015$) 조건 하에서 지연을 최소화하는 저지연 지점을 선택한다. 두 지점 모두 홀드아웃 테스트셋에서 평가된다.

불확실성. Score, 지연, 커밋률, RTF에 대해 테스트셋을 B회 재표본 추출한 부트스트랩 신뢰 구간을 보고한다.

4-4 결과

표 1은 (k, τ)에 의해 제어되는 명확한 정확도–지연 전선을 보여준다. k=3에서 k=4로 이 동하면 품질이 향상되고 지연이 줄어든다. 최적의 저지연 설정은 (4, 0.7)로, Score 0.5956과 0.349s를 기록한다. k≥6에서는 Score가 약 0.598 근처에서 포화되고 지연은 증가한다.

표 1. 검증셋에서 (k, τ) 그리드 탐색 결과: SCORE(평균 [95% CI]), 평균 방출 지연(초), 초당 커밋률(CPS), 그리고 종단간 실시간 계수

Table 1. Validation grid over (k,τ) with score mean [95% CI]), average emission LAG in seconds, commit rate in characters per second (CPS), and end-to-end real-time factor

k	τ	Score↑	Lag (s)	Commit rate (cps)↑	RTF_{e2e}
3	0.50	0.5743 [0.5658, 0.5816]	0.767	0.080	0.0026
3	0.60	0.5749 [0.5663, 0.5820]	0.794	0.078	0.0026
3	0.70	0.5768 [0.5688, 0.5839]	0.692	0.072	0.0026
4	0.50	0.5945 [0.5913, 0.5970]	0.470	0.020	0.0025
4	0.60	0.5950 [0.5922, 0.5974]	0.420	0.018	0.0025
4	0.70	**0.5956** [0.5929' 0.5978]	**0.349**	0.016	0.0025
6	0.50	0.5971 [0.5953, 0.5988]	0.516	0.010	0.0026
6	0.60	0.5977 [0.5960, 0.5992]	0.446	0.008	0.0026
6	0.70	0.5982[0.5967, 0.5995]	0.375	0.006	0.0026
8	0.50	0.5975 [0.5957, 0.5989]	0.579	0.009	0.0026
8	0.60	0.5980 [0.5964, 0.5994]	0.509	0.007	0.0026
8	0.70	**0.5986** [0.5973, 0.5997]	**0.440**	0.005	0.0026
10	0.50	0.5975 [0.5959, 0.5989]	0.642	0.009	0.0026
10	0.60	0.5980 [0.5966, 0.5994]	0.573	0.007	0.0026
10	0.70	0.5986 [0.5973, 0.5996]	0.504	0.005	0.0026

균형 잡힌 선택은 (8, 0.7)로, Score 0.5986과 0.440 s를 기록한다. RTF_{e2e}는 모든 설정에서 ≈ 0.0025로 일정하며,

95% CI가 좁아 추정치가 안정적임을 시사한다.

그림 2는 검증셋에서 명확한 품질–지연 전선을 보여준다. 0.35 s와 0.50 s 사이의 지점이 최적의 트레이드오프를 제공한다. 저지연 선택 (4, 0.7)은 Score 0.5956, 0.349 s를 달성하며, 균형 선택 (8, 0.7)은 Score 0.5986, 0.440 s를 달성한다. 0.35 s 왼쪽으로 가면 Score가 급격히 하락하며, 0.50 s 오른쪽으로 가면 지연만 늘고 추가적인 성능 향상은 제한적이다.

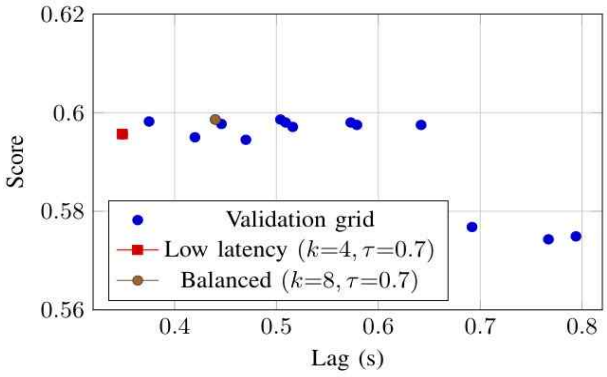


그림 2. 검증셋에서 Score–지연 파레토 전선
Fig. 2. Validation Pareto front of Score versus lag

표 2. 테스트셋에서 선택된 운용 지점 결과
Table 2. Test results for selected operating points. score shows mean [95%CI], LAG is in seconds, commit rate is characters per second

Setting	k	τ	Score↑	Lag (s) ↓	Commit rate (cps)↑	RTF e2e↓
Exported configuration	3	0.50	0.5762 [0.5729, 0.5793]	0.725	0.083	0.0024

표 2는 현재 “균형”과 “저지연” 모드가 동일한 (k, τ) = (3, 0.50)으로 내보내져 있어 같은 결과를 보인다. 이 설정에서 테스트 Score는 0.5762 [0.5729, 0.5793], 평균 방출 지연은 0.725 s, 커밋률은 0.083 cps, RTF_{e2e} = 0.0024로, 실시간보다 훨씬 빠른 추론임을 확인할 수 있다. 품질은 강력하지만 지연은 검증 파레토 지점보다 크다. (8, 0.7)을 균형 지점, (4, 0.7)을 저지연 지점으로 재선택하고 내보내면 각각 Score 향상 및 지연 감소 효과를 얻을 수 있으며, 계산 비용은 변하지 않는다.

표 3은 α ∈ {0.8, 1.0, 1.2}에 대해 양 모드 모두 동일한 값을 보인다(Score = 0.5776, 지연 = 19.58 프레임, RTF ≈ 0.00245–0.00249). 이는 시간 왜곡이 추론 전에 적용되지 않았거나, 이후 고정 길이 T로 리사이즈되며 효과가 상쇄되었음을 시사한다.

실제 속도 변화가 반영되려면 인코더/디코더 이전에 왜곡을 적용하고, 고정 T로 되돌리는 리사이즈를 피하거나, fps′ = α·fps를 사용하여 초 단위 지연을 보고하는 것이 필요하다.

표 3. 테스트셋에서 시간 왜곡 α ∈ {0.8, 1.0, 1.2}에 대한 강인성(클립 평균). SCORE는 1−CER
Table 3. Robustness to temporal warps α∈{0.8,1.0,1.2} on test(means over clips). SCOREIS1−CER

Mode	α	Score↑	Lag (frames)↓	RTF_{e2e}
balanced	0.8	0.5776	19.58	0.00249
balanced	1.0	0.5776	19.58	0.00249
balanced	1.2	0.5776	19.58	0.00248
low_latency	0.8	0.5776	19.58	0.00245
low_latency	1.0	0.5776	19.58	0.00244
low_latency	1.2	0.5776	19.58	0.00244

표 4. 검증 효율성: 디코더 전용 및 종단 간 실시간 계수(평균 [95% CI]), 평균 커밋률(CPS)
Table 4. Validation efficency: Decode-only and end-to-end real-time factors (mean [95% CI]), plus mean commit rate in characters per seconds

RTF_{dec}	RTF_{e2e}	Commit rate (cps)
0.000099 [0.000083, 0.000127]	0.002553 [0.002339, 0.002964]	0.0235

표 4는 매우 가벼운 추론을 보여준다. 평균 종단간 실시간 계수(RTF_{e2e} = 0.002553)는 ~392배 실시간보다 빠름을 의미한다(95% CI: 약 337–427배). 디코더 자체는 무시할 수 있을 정도로 작아(RTF_{dec} = 9.9×10^{-5}), 전체 계산의 약 4%만을 차지하며 비용은 인과적 인코더가 지배한다. 데이터셋 전체 평균 커밋률은 0.0235 cps이며, 이는 (k, τ) 설정과 클립 내용에 따라 달라질 수 있으므로 특정 운용 지점의 목표율이 아닌 데이터셋 전체 요약값으로 해석된다.

표 5는 Greedy CTC collapse 기준선을 보여주며, Score 0.7488과 95% 신뢰구간 [0.7415,0.7555]를 5973개 클립에 걸쳐 달성하여 안정적인 기준(reference)을 제공한다. 이에 반해, 제안한 이벤트-트리거 스트리밍 디코더(표 II)는 Score ≈0.5762를 기록하며, 절대적으로 +0.2274 향상을 보인다. CER 관점에서는 0.2512에서 0.0238로 감소하여(약 90.5% 오류 감소) 여전히 실시간보다 훨씬 빠른 성능을 유지한다. 이는 스트리밍 지문자에서 단순 Greedy collapse 보다 안정적 증거 기반 커밋이 가지는 이점을 명확히 보여준다.

표 5. 테스트셋에서의 Greedy CTC collapse 기준선
Table 5. Greedy CTC collapse baseline on the test set

Score↑	#Clips
0.7488 [0.7415, 0.7555]	5973

일관성 참고. 일부 그림 버전이 y축에 CER(범위 0–0.36)을 표시한다면, 동일한 전선(frontier)은 Score (1−CER)가 1.0 근처에 해당한다. 카메라 레디 버전은 표와의 일관성을 위해 y축에 Score를 사용한다.

V. 결 론

본 연구에서는 랜드마크 기반 스트리밍 지문자를 위해 정확하면서도 배포 가능한 이벤트-트리거 CTC 프레임워크를 제안하였다. CTC로 학습된 인과적 인코더는 프레임별 사후 확률을 제공하며, 단순한 디코더는 안정적인 증거가 있을 때에만 문자를 확정한다. 두 개의 직관적인 제어 변수—윈도우 길이 k와 평균 사후 확률 임계값 τ—는 명시적인 정확도-지연 트레이드오프를 가능하게 한다.

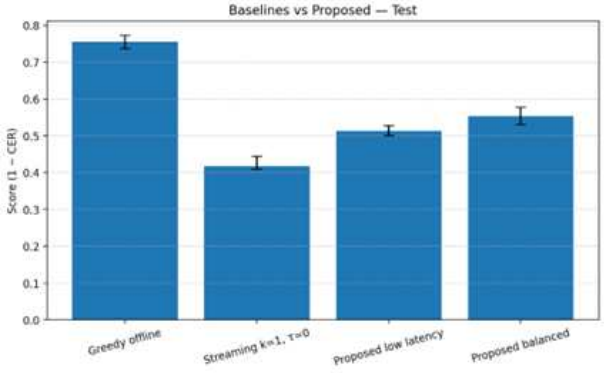


그림 3. 기준선 vs 제안된 테스트
Fig. 3. Baselines vs proposed test

그림 3에서는 오프라인 Greedy 디코더는 가장 높은 점수를 달성하지만, 실시간 제약 조건 없이 동작한다. 스트리밍 k=1, 설정은 가장 낮은 정확도를 보이는데, 각 프레임마다 즉시 커밋하는 방식은 일시적인 피크나 짧은 블랭크에 취약하여 삭제와 치환을 증가시키기 때문이다. 제안된 두 모드는 커밋 전에 짧지만 일관된 증거를 요구함으로써 k=1 대비 정확도를 향상 시킨다. 특히 제안된 (balanced) 설정은 가장 높은 스트리밍 정확도를 달성하며, 그 신뢰 구간은 k=1 기준선과 명확히 분리되어 있어 신뢰할 만한 향상을 보여준다. 제안된 (low-latency) 변형은 지연을 줄이는 대신 정확도를 약간 희생하지만 여전히 k=1보다 우수하다.

그림 4는 트레이드오프를 명확히 보여준다. Greedy는 가장 오른쪽에 위치하며, 매우 높은 정확도를 달성하지만 문자당 수 초의 지연이 발생하여 상호작용적 사용에는 부적합하다. k=1은 가장 왼쪽에 위치하며 지연은 가장 작지만 정확도가 가장 낮아, "너무 빠른 커밋"이 인식을 저해함을 보여준다. 제안된 두 운용 지점은 모두 1초 미만 영역에 위치하며, k=1 대비 정확도가 상승한다. 이 중 balanced 지점은 가장 매력적인 실시간 선택지로, 오프라인 점수에 근접하면서도 문자당 지연을 1초 이하로 유지한다. low-latency 지점은 정확도를 다소 희생하지만 더 빠른 선택지를 제공한다.

평가는 Score (1−CER)를 우선시하며, 대화형 사용과 관련된 지연 및 효율성 지표로 이를 보완한다. (k, τ)에 대한 검증 탐색은 Score와 방출 지연 간의 명확한 파레토 전선을 도출하였고, 이를 통해 균형 지점과 저지연 운용 지점을 선택하였다. 테스트셋에서 두 지점 모두 실시간보다 빠르게 동작하면서도 높은 Score를 유지하였다. 전체 파이프라인은 종단간 재현 가능하며, 전처리 "E인과 추론" E디코딩을 온디바이스에서 수행하는 단일 TensorFlow Lite 시그니처로 내보낼 수 있다.

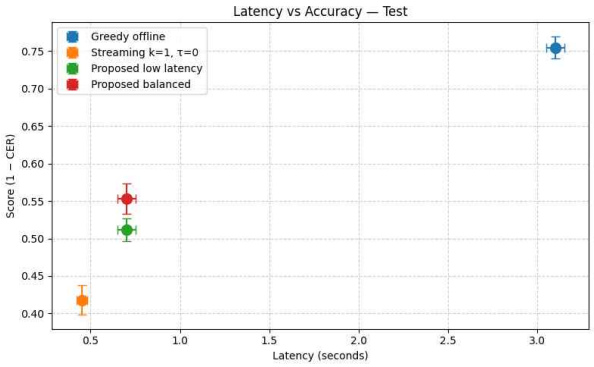


그림 4. 지연 시간 vs 정확도 테스트
Fig. 4. Latency vs accuracy test

우리의 결과는 빔 서치나 외부 언어 모델 없이도 신뢰할 수 있는 스트리밍 인식이 가능함을 보여주며, 시스템을 투명하고 조정하기 쉽게 유지한다. 한계점으로는 두 개의 디코딩 파라미터에만 의존한다는 점과 장기적인 언어적 문맥 부재를 들 수 있다. 향후 연구 방향은 적응적(학습 기반) 커밋 정책, 온디바이스 제약과 호환되는 경량 언어 모델링, 수어자 적응, 더 안전한 입력을 위한 신뢰도/거부 메커니즘, 다양한 언어 및 캡처 조건에서의 광범위한 평가 등을 포함한다. 공개된 아티팩트를 통해 본 프레임워크는 스트리밍 수어 텍스트 입력 시스템 구축을 위한 실질적인 기준을 제공한다.

부록 A: 데이터 및 전처리

A-1 랜드마크 구성

각 프레임은 오른손과 왼손(각 21 키포인트), 얼굴과 입술, 상반신 자세 랜드마크를 연결하여 (x, y, z) 좌표로 이루어진 F = 276개의 특징을 형성한다. 시퀀스는 고정된 길이 T = 176 프레임으로 패딩 또는 리사이즈 된다.

A-2 표준화 결과값 처리

각 랜드마크 그룹 g ∈ {hands, face, pose}에 대해, 특징 $X^{(g)} \in R^{T\times F_g}$에 다음을 적용한다:

$$\tilde{X}_{t,f}^{(g)} = \frac{X_{t,f}^{(g)} - \mu_f^{(g)}}{\sigma_f^{(g)} + \epsilon} \tag{11}$$

여기서 $\mu_f^{(g)}$, $\sigma_f^{(g)}$는 사전 계산된 평균과 표준편차이다. 표준화 후 NaN은 0으로 대체한다. 학습 중에는 제출 시 로더와 일치

하도록 손이 누락된 프레임을 교대로 유지하는 마스크를 사용한다.

A-3 라벨 어휘

문자 집합 ν는 대문자 알파벳과 평가에 사용되는 일부 숫자 및 구두점을 포함한다. CTC 에서는 ν에 포함되지 않는 하나의 블랭크 기호 ∅를 사용한다.

부록 B: 이벤트-트리거 디코더

B-1 커밋 규칙

$C_t = \arg\max_{c \in \nu} p_t(c \mid X), W_t = \{t-k+1,...,t\}$ 라 하자. 디코더는 시점 t에서 다음 조건이 성립할 때 출력을 발생시킨다.

$$\bigwedge_{u \in W_t} [C_u = C_t] \text{ and } \frac{1}{k}\sum_{u \in W_t} p_u(C_t|X) \geq \tau \quad (12)$$

$$\text{and } C_t \neq _{last}$$

arg max에서 동점이 발생하면 결정론을 위해 ν내 가장 작은 인덱스를 사용한다.

B-2 즉각 반복 특성

커밋 시점에 $C_t \neq \bar{y}_{last}$가 요구되므로, 동일한 문자가 연속적으로 출력되려면 중간에 승자가 변경된 후 새로운 안정 윈도우가 형성되어야 한다. 따라서 반복은 구조적으로 압축된다.

B-3 지연 정의와 효율적 갱신

프레임 t_j에서 $\bar{y}_j$를 출력할 때 최초 지배 시간은

$$t_j^{\star} = \min \begin{Bmatrix} u \leq t_j : C_s = _j \forall s \in \{u,\cdots,t_j\}, \\ t_j - u + 1 \geq k \end{Bmatrix} \quad (13)$$

이다. 방출 지연은 $t_j - t_j^{\star}$ 프레임, 또는 $(t_j - t_j^{\star})$/fps 초이다. 롤링 평균을 사용하면 W_t에 대한 갱신을 상수 시간으로 처리할 수 있다.

부록 C: 배포 인터페이스

C-1 TensorFlow Lite 시그니처와 사용법

하나의 시그니처 serving_default를 내보내며, 이는 랜드마크 입력(T×F 배치 또는 1×F 단일 스텝)을 받아 프레임별 사후 확률 또는 확정된 문자를 출력한다. 그래프는 학습과 동일한 랜드마크 그룹화 및 표준화뿐만 아니라 이벤트-트리거 디코딩을 포함하므로 외부 전처리가 필요 없다. 스트리밍 모드에서는 프레임을 하나씩 입력하고 디코더 상태를 호출하며, 커밋 규칙이 발동될 때에만 문자가 반환된다.

부록 D: 재현성 참고 사항

(k, τ)탐색, 선택된 운용 지점, 강인성 및 효율성 요약을 포함한 전체 검증 및 테스트 테이블은 평가 스크립트에 의해 생성된 CSV 파일로 제공된다. 이 파일들은 논문에서 보고된 모든 수치를 정확히 재현할 수 있도록 제출물에 포함된다.

참고문헌

[1] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks,” in *Proceeding of the 23rd International Conference on Machine Learning (ICML '06)*, Pittsburgh: PA, pp. 369-376, June 2006. https://doi.org/10.1145/1143844.1143891

[2] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, “Advances in Joint CTC-Attention Based End-to-End Speech Recognition with a Deep CNN Encoder and RNN-LM,” in *Proceeding of Interspeech 2017*, Stockholm, Sweden, pp. 949-953, August 2017. https://doi.org/10.21437/Interspeech.2017-1296

[3] C. Raffel, M.-T. Luong, P. J. Liu, R. J. Weiss, and D. Eck, “Online and Linear-Time Attention by Enforcing Monotonic Alignments,” in *Proceeding of the 34th International Conference on Machine Learning (ICML '17)*, Sydney, Australia, pp. 2837-2846, August 2017.

[4] C.-C. Chiu and C. Raffel, “Monotonic Chunkwise Attention,” arXiv:1712.05382, 2018. https://doi.org/10.48550/arXiv.1712.05382

[5] H. Inaguma, M. Mimura, and T. Kawahara, “Enhancing Monotonic Multihead Attention for Streaming ASR,” arXiv:2005.09394, 2020. https://doi.org/10.48550/arXiv.2005.09394

[6] S. Zhang, Z. Gao, H. Luo, M. Lei, J. Gao, Z. Yan, and L. Xie, “Streaming Chunk-Aware Multihead Attention for Online End-to-End Speech Recognition,” in *Proceeding of Interspeech 2020*, Shanghai, China, pp. 2142-2146, October 2020. https://doi.org/10.21437/Interspeech.2020-2248

[7] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, ... and R. Pang, “Conformer: Convolution-augmented Transformer for Speech Recognition,” in *Proceeding of Interspeech 2020*, Shanghai, China, pp. 5036-5040, October 2020. https://doi.org/10.21437/Interspeech.2020-3015

[8] Y. He, T. N. Sainath, R.Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, ... and A. Gruenstein, “Streaming End-to-End Speech Recognition for Mobile Devices,” in *Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton: UK, pp. 6381-6385, May 2019. https://doi.org/10.1109/ICASSP.2019.8683416

[9] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, “Transformer Transducer: A Streamable Speech Recognition Model with Transformer Encoders and RNN-T Loss,” in *Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2020*, Barcelona, Spain, pp. 7829-7833, May 2020. https://doi.org/10.1109/ICASSP40776.2020.9053896

[10] J. Li, Y. Wu, Y. Gaur, C. Wang, R. Zhao, and S. Liu, “On the Comparison of Popular End-to-End Models for Large Scale Speech Recognition,” arXiv:2005.14327, 2020. https://doi.org/10.48550/arXiv.2005.14327

[11] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, ... and M. Grundmann, “MediaPipe: A Framework for Perceiving and Processing Reality,” arXiv:1906.08172, 2019. https://doi.org/10.48550/arXiv.1906.08172

[12] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, “MediaPipe Hands: On-Device Real-Time Hand Tracking,” arXiv:2006.10214, 2020. https://doi.org/10.48550/arXiv.2006.10214

[13] B. Shi, A. Martinez Del Rio, J. Keane, J. Michaux, D. Brentari, G. Shakhnarovich, and K. Livescu, “American Sign Language Fingerspelling Recognition in the Wild,” in *Proceeding of the IEEE Spoken Language Technology Workshop (SLT)*, Athens, Greece, pp. 145-152, December 2018. https://doi.org/10.1109/SLT.2018.8639639

[14] B. Shi, A. Martinez Del Rio, J. Keane, D. Brentari, G. Shakhnarovich, and K. Livescu, “Fingerspelling Recognition in the Wild with Iterative Visual Attention,” in *Proceeding of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, pp. 5399-5408, October 2019. https://doi.org/10.1109/ICCV.2019.00550

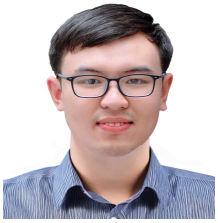
[15] M. Georg, G. Tanzer, E. Uboweja, S. Hassan, M. Shengelia, S. Sepah, ... and T. Starner, “FSboard: Over 3 Million Characters of ASL Fingerspelling Collected via Smartphones,” in *Proceeding of the 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* Nashville: TN, pp. 13897-13906, June 2025.

[16] Kaggle and Google Research. Google – American Sign Language Fingerspelling Recognition [Internet]. Available: https://www.kaggle.com/competitions/asl-fingerspelling.

[17] C. Deotte. 2 Lines of Code Change (LB 0.760+) [Internet]. Available: https://www.kaggle.com/code/cdeotte/2-lines-of-code-change-lb-0-760.

유승수(Seung-Su Yu)

2024년 : 국립한밭대학교 창업경영대학원 (창업학 석사)
2026년 : 대전대학교 대학원 (기술경영 박사수료)
2011년~현 재: ㈜멀틱스 유승수 대표
※관심분야 : AI, Barrier Free, 수어(Sign Language), Multi Modal Communication Technology 등



Khoa Nguyen

2023년 : 베트남-독일대학교 컴퓨터공학(학사)
2024년~현 재: 충북대학교 정보통신공학부(석박사 통합과정)
※관심분야 : Research Interests, Federated Learning, Time Series Forecasting

Thong-Nhat Tran

2023년 : 홍익대학교 전기전자공학(박사)
2023년~현 재: 충북대학교 정보통신공학부 연구원 및 계약교수
※관심분야 : Research Interests, Wireless Communications, Optimization, Computer Vision

서영욱(Young-Wook Seo)

2000년 : 성균관대학교 경영학(석사)
2008년 : 성균관대학교 경영학(박사)
2014년~현 재: 대전대학교 일반대학원 융합컨설팅학과 교수
※관심분야 : 정보경영, IT컨설팅, 경영컨설팅, 창의성, 컨설턴트