BitePulse AI: Real-Time Eating-Pace Feedback from Meal video

Temporal Deep Learning for Bite Detection

Aktham Almomani

Master of Science in Applied Artificial Intelligence

Shiley Marcos School of Engineering

University of San Diego

aalmomani@sandiego.edu

ABSTRACT

Most of us have no idea how fast we really eat until a doctor, a coach, or a bad stomach reminds us. BitePulse AI asks a simple question: can a short phone video give people that feedback in real time, without human scoring or sharing their data? To explore this idea, we train a sequence of temporal deep learning models on a labeled meal dataset to detect intake events (bites) and estimate eating pace. We start with a pose-based Temporal Convolutional Network (TCN) as a lightweight baseline, then apply Hyperband tuning to the same architecture, and also train an RGB 3D-CNN on short frame clips to inject appearance cues. Finally, we move to a frame-level Multi-Stage TCN (MS-TCN) over MediaPipe pose sequences, which clearly dominates all prior models in macro precision, recall, F1, ROC AUC, and especially PR AUC for the rare INTAKE class. However, to keep latency, memory, and deployment complexity within the constraints of a browser-based Streamlit demo, the current app uses a lighter MediaPipe intake detector, with the MS-TCN serving as the "gold standard" offline model that informs the design and target behavior of a future on-device pace coach.

KEYWORDS

Eating pace, bite/intake detection, temporal action recognition, frame-level modeling, MS-TCN, pose-based modeling, 3D-CNN, MediaPipe, event-level evaluation, class imbalance, real-time inference, on-device / privacy-preserving AI.

1 Introduction

Eating rate is an overlooked behavioral risk factor. Experimental and observational studies show that rapid eating is associated with higher energy intake, weaker subjective satiety, and adverse gastrointestinal symptoms (Andrade et al., 2008). In controlled meal studies, asking participants to eat slowly reduces how much they eat when allowed to serve themselves freely and increases post-meal fullness ratings (Andrade, Greene, & Melanson, 2008). Faster ingestion, on the other hand, is linked to greater postprandial reflux and gastric distension in clinical cohorts (Su et al., 2000). Meta-analytic work also connects fast eating with higher odds of metabolic syndrome and overweight (Zhu et al., 2020). Despite this evidence, most people receive little or no feedback about how fast they eat in everyday settings.

Researchers already treat eating pace as a measurable signal. In laboratory and free-living studies, teams annotate "intake events" such as bites on video, then compute metrics like bites per minute or burst patterns to study self-control, comfort, and energy intake (Rouast, Heydarian, Adam, & Rollo, 2020). However, these analyses typically depend on manual labeling and are not accessible to consumers or digital-health partners. There is a gap between what research systems can measure about eating behavior and the kind of timely, privacy-preserving feedback that could help individuals make small but meaningful changes in daily life.

BitePulse AI investigates whether modern temporal deep learning can close part of this gap. The project explores if short meal videos, captured on a phone, can be converted automatically into bite detections and an interpretable eating-pace summary that is returned to the user in under one minute. The primary end users are individual consumers who might benefit from gentle, real-time coaching about pace, and wellness organizations who may want objective but non-intrusive indicators of eating behavior for their programs. For consumers, the envisioned experience is a simple application that accepts a brief clip and returns a pace score, a timeline of detected intake events, and one sentence of neutral guidance. For partners, the same signals can be exposed through an API without exposing or storing identifiable video.

To support this investigation, we use the EatSense dataset, a public collection of real-world meal videos with anonymized faces and frame-level labels for eating, chewing, drinking, and resting (Raza et al., 2023). These annotations enable supervised learning of frame-level "intake versus non-intake" predictions and aggregation into event-level bite detections. In a deployed system, analogous data would come from user-recorded clips, processed either entirely on device or in a secure backend that discards raw video after inference.

The technical approach is to construct time-aligned sequences from annotated meal videos and compare several temporal modeling approaches for binary intake detection. Specifically, we evaluate a pose-based Temporal Convolutional Network (TCN), an RGB-based 3D convolutional neural network (3D-CNN), and a frame-level Multi-Stage Temporal Convolutional Network (MS-TCN) on the task of distinguishing intake from non-intake behavior under strong class imbalance. Model predictions are then aggregated into event-level bite detections and summary measures of eating pace.

Finally, we prototype a live experience through a Streamlit web application that runs in the browser. For deployment constraints, the app uses MediaPipe-based landmark extraction with a lightweight intake heuristic for real-time feedback, while the MS-TCN serves as an offline "gold standard" model that informs the target behavior and metrics of the system. This demonstrates that the same pipeline can power both research-grade evaluation and a practical phone or web experience without persisting raw video (Lugaresi et al., 2019).

The central hypothesis is that temporal models trained on labeled meal videos can achieve practically useful precision and recall for intake detection and produce stable, interpretable measures of eating pace that are suitable for real-time feedback. If this hypothesis is supported, BitePulse AI points toward a privacy-respecting, deployable "eating-pace coach" that brings methods currently used only in research labs into everyday life for consumers and wellness programs.

2 Data Summary

The BitePulse AI prototype is built on the EatSense dataset, a public collection of 135 real-world meal videos with anonymized faces and frame-level activity labels, totaling roughly 14 hours of footage and averaging about 11 minutes per clip (Raza et al., 2023). Each recording contains RGB video of a person eating at a table and a set of time-aligned annotations describing what the person is doing in each moment. For this project, the most important labels are eating, drinking, chewing, and resting, which we use to define intake versus non-intake behavior.

At a high level there are four layers of variables. At the session level, each meal has an identifier, basic context (for example, lunch versus snack), and camera setup. At the frame level, each image has a timestamp, a frame index, and pre-computed 2D body-pose key points for head, torso, arms, and hands. At the segment level, the dataset provides start and end times for labeled activities, such as a chewing phase or a drink. On top of this, our capstone introduces a window level: we slide a fixed-length window (for example, 0.5 seconds) over time with a fixed stride and assign each window a binary label indicating whether it contains an intake event.

The window representation is a key part of the novelty in our approach. Rather than train directly on variable-length segments, we create a uniform "grid" over time that can be used consistently across pose and RGB models. Each window carries (a) a short sequence of pose features, engineered from the 2D key points as relative positions to the head and simple velocities, and (b) an index into the original video frames so that the same window can be mapped to an RGB clip for the 3D-CNN. The target label for each window is derived from the segment annotations using an overlap rule: a window is positive if its time span overlaps an eating or drinking segment beyond a chosen threshold, and negative otherwise. This produces a single table of training examples that sits between the raw dataset and our models and makes it straightforward to compare pose-only and RGB baselines.

For the MS-TCN experiments, we move one step closer to the raw annotations and operate directly at the frame level. The 16 original EatSense action labels are collapsed into a binary target, with INTAKE corresponding to the "eat it" action and NON_INTAKE for all other labels, yielding long sequences of frame-wise pose features and labels for each session. Approximately 5% of frames are labeled as INTAKE, which is higher than in the window-based dataset where windows are labeled positive only if their temporal overlap with an intake segment exceeds a predefined threshold rather than for any overlap; as a result, only about 0.4% of windows are labeled as INTAKE. Although the frame-level representation remains highly imbalanced, it provides a denser and more direct positive signal. Four of the 135 videos contain no intake frames at all; we retain these sessions as realistic "no-intake" examples, which force the model to correctly predict zero bites when appropriate. Variable-length sequences are padded with an ignore index so that batches can be formed without discarding context at the start or end of meals.

Like most behavioral datasets, EatSense is not perfectly regular. We occasionally see missing or low-confidence pose estimates, small gaps in the activity labels, and slight misalignments between annotation timestamps and video frame times. Some participants drift partially out of frame or briefly occlude their arms and face with utensils or hands. Intake events are also relatively rare

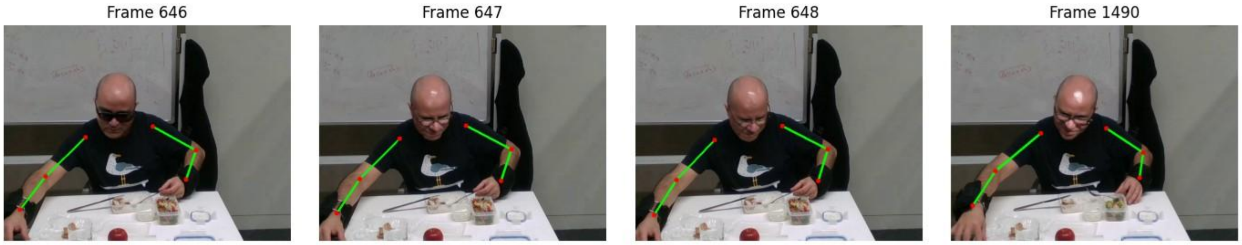


Figure 2.1 Example of anatomically correct but "noisy" poses: the participant fully extends their arms, producing unusually long shoulder–wrist distances that our distance-based heuristic flags as outliers.

compared with background motion, so the resulting window table is highly imbalanced.

To better understand pose reliability, we computed shoulder–to–wrist distances for both arms and treated the largest values (e.g., the top 0.5 % of this distribution) as candidate outliers. Many of these frames correspond to genuinely noisy or off-body detections, but some simply reflect participants fully extending their arms while still being tracked correctly. Figure 2.1 illustrates this latter case: the pose is anatomically plausible, yet the stretched arms produce unusually long limb lengths that our distance-based heuristic flags as outliers.

To handle these issues, we derive labels using timestamps rather than frame indices, drop windows that have no valid pose information, and merge very short gaps in intake segments so that a single bite is not split into multiple tiny labels. On the modeling side we compensate for class imbalance with a combination of class weighting in the loss and balanced sampling of positive and negative windows during training.

Exploratory analysis of the window table reveals several patterns that connect directly to the project goal. Figure 2.2 summarizes the duration of contiguous runs in this table.

Positive intake windows are short: most intake episodes last less than 15 seconds, with a median duration of about 9 seconds. In contrast, non-intake runs are long and highly variable, with a median length of roughly 480 seconds and many stretches that continue for thousands of seconds without a single intake label. This stark difference highlights both the rarity of intake behavior and

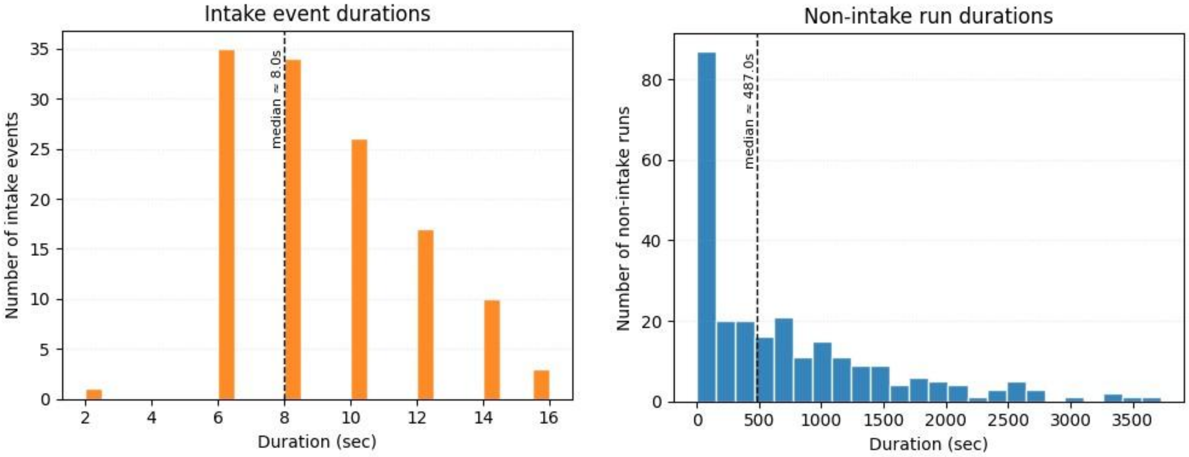


Figure 2.2 Histogram of contiguous run durations in the window table

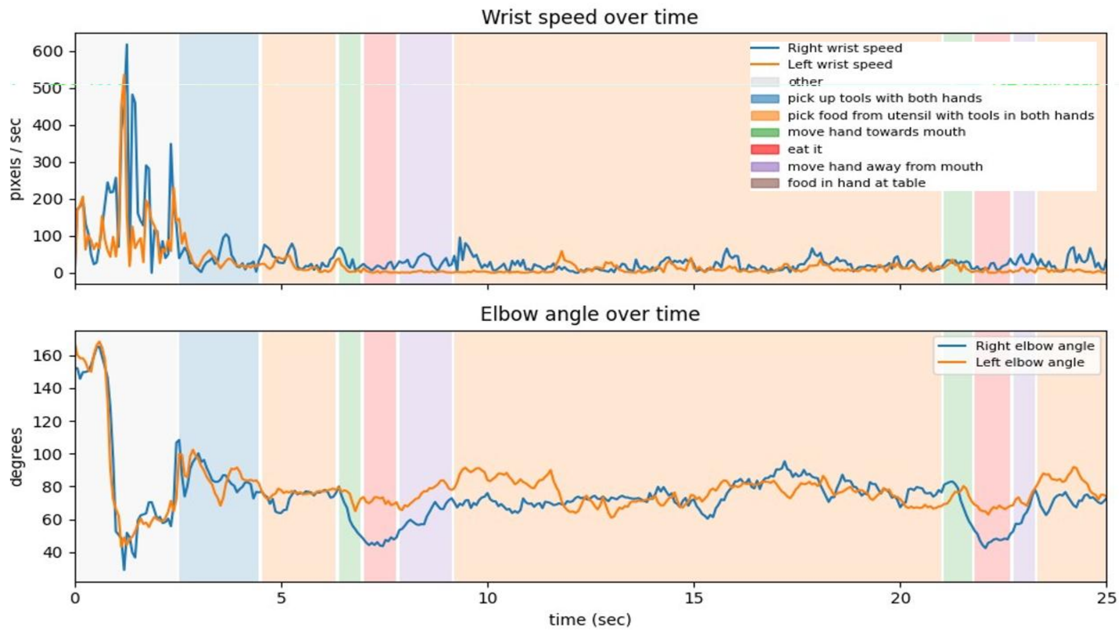


Figure 2.4 Wrist speed (top) and elbow angle (bottom) over a 25-second meal segment.

the strong temporal structure around it, motivating models that focus on brief, well-localized positive segments within very long background sequences.

Figure 2.3 illustrates wrist trajectories over a short time interval for a representative meal, plotted separately for intake and non-intake segments. During intake, the wrist trajectory forms a compact path directed toward the face, reflecting a purposeful hand-to-mouth motion.

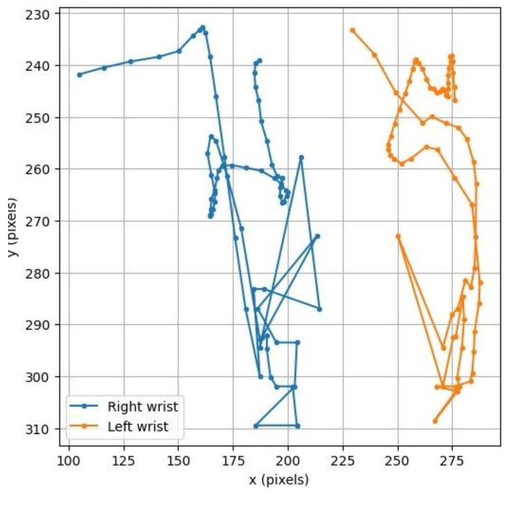


Figure 2.3. Wrist XY paths for left and right hands during the first 5 seconds of a sample meal.

In contrast, during non-intake periods such as resting or chewing without new intake, the wrist trajectory remains more diffuse and does not consistently converge toward the face. The figure highlights this contrast by showing distinct trajectory shapes across the two activity types rather than relying on subtle differences in instantaneous speed or joint angle.

When we look at the full sequence in time, we see that intake moments are not isolated spikes but short, structured episodes. Figure 2.4 plots right and left wrist speed (top) and elbow angle (bottom) over the first twenty-five seconds of the clip, with shaded bands indicating the ground-truth actions (for example, pick up tools with both hands, move hand towards mouth, eat it, and move hand away from mouth). Brief bursts of high wrist velocity appear precisely inside the green and red bands where the utensil moves toward the mouth and food is consumed, and they are accompanied by rapid elbow flexion followed by a more gradual extension as the arm returns to the table. In contrast, the long orange “food in hand at table” regions show relatively flat wrist speeds and slowly varying elbow angles.

These consistent temporal patterns across pose features motivate the use of temporal convolution rather than purely frame-wise models, and they suggest that short pose-only windows should already provide a strong baseline for detecting intake events. We also observe correlations and redundancies within the raw pose coordinates. Neighboring key points and their absolute image positions are strongly correlated, especially within each limb. To keep the model focused on behavior rather than camera geometry, we express pose features in a head-centered coordinate frame and include simple temporal derivatives instead of a large set of raw coordinates. This reduces input dimensionality and helps the Temporal Convolutional Network focus on relative motion patterns that generalize across subjects and camera setups. For the RGB path we use the same window index to extract short clips of frames, which theoretically allows a 3D-CNN to learn complementary appearance cues such as utensil type, cup orientation, or partial occlusions that are not visible in pose.

Taken together, the dataset and these engineered variables give us three views of the same underlying behavior: (1) raw video, (2) anonymized, structured pose sequences, and (3) a regular grid of labeled windows that bridge between them, plus (4) a frame-level representation used to train MS-TCN directly on long pose sequences. This design enables fair comparison between pose-based and RGB models, event-level evaluation built from window outputs, and a direct mapping from model predictions to the bite timeline and pace metrics that drive the BitePulse AI user experience.

3 Literature Review

Research on eating behavior has explored multiple sensing methods for detecting intake events and characterizing eating pace. Early work relied heavily on manual video annotation in controlled laboratory settings, where raters marked each bite and computed summary measures such as bites per minute, total intake, and temporal “burst” patterns to study satiety and self-control (Rouast et al., 2020; Raza et al., 2023). These methods produced high-quality labels but were labor-intensive and impractical for day-to-day feedback outside the lab.

Parallel streams of work investigated instrumented utensils, wearable bite counters, and multimodal systems that combine wrist motion, audio, and inertial signals, again with the goal of estimating intake frequency and speed with minimal user burden (Rouast et al., 2020). This literature treats eating pace as a measurable behavioral signal and shows that objective intake measures can be linked to energy intake, gastrointestinal symptoms, and metabolic risk, but most systems remain research prototypes rather than deployable tools for consumers or digital-health programs.

In the last several years, video-based datasets have made it possible to study intake detection at larger scale and with more realistic meal scenarios. Rouast et al. (2020) introduced the OREBA dataset, which provides synchronized multi-view video, audio, and detailed annotations of eating, drinking, and associated intake in semi-naturalistic settings, enabling frame- and segment-level recognition of eating behavior. Raza et al. proposed EatSense, a human-centric dataset of anonymized meal videos with fine-grained labels for eating, drinking, chewing, resting, and related activities, designed specifically for action recognition and localization in the context of eating behavior (Raza et al., 2023). These datasets demonstrate that deep models can successfully localize intake episodes in complex table scenes, but most published work focuses on offline analysis rather than real-time feedback, and typically reports performance at the segment level rather than at the level of user-facing pace metrics such as bites per minute or stable intake timelines.

On the modeling side, recent sequence-modeling research has established Temporal Convolutional Networks (TCNs) as a strong alternative to recurrent architectures for time-series and event-detection tasks. Bai et al. (2018) conducted a broad empirical evaluation and showed that causal convolutional stacks can match or outperform recurrent neural networks on a wide range of sequence problems while offering stable gradients and high parallelism during training. This work highlights several properties that are particularly relevant for real-time eating-pace analysis: TCNs can model long temporal contexts via receptive fields, maintain order information through causal filtering, and run efficiently on modern hardware, including resource-constrained devices. Building on this, multi-stage temporal convolutional architectures (MS-TCN–style models) have been proposed for frame-level action segmentation, where successive stages refine dense per-frame predictions and help handle strong class imbalance—an attractive property for rare intake events.

In parallel, the video understanding community has developed 3D convolutional neural networks (3D-CNNs) that operate directly on short RGB clips, learning joint spatiotemporal features that capture both appearance and motion cues. Tran et al. (2015) demonstrated that compact 3D-CNN architectures can learn expressive features for human action recognition across diverse video benchmarks, providing a general backbone for many downstream applications. Combined with modern pose-estimation systems such as MediaPipe, which provide fast, anonymized 2D landmarks from commodity cameras, these models enable hybrid pipelines in which pose sequences drive temporal models while RGB clips contribute complementary appearance information when needed (Lugaresi et al., 2019).

Together, these lines of work outline a research trajectory from manually annotated, offline analyses of eating behavior toward automated, model-based intake detection from video. Eating-behavior datasets like OREBA and EatSense show that intake events can be reliably labeled and recognized in realistic meal settings, while the broader sequence-modeling and video-action-recognition literature provides mature architectures such as TCNs/MS-TCNs over pose sequences and 3D-CNNs over short RGB clips that are well suited for fast, window- or frame-based intake classification. The combination of structured pose representations, event-level evaluation, and efficient temporal models therefore represents a natural, research-grounded foundation for building systems that could eventually deliver real-time, privacy-preserving feedback on eating pace in everyday contexts, rather than only within specialized research labs.

4 Methodology

Our modeling pipeline starts from time-aligned intake labels and produces predictions that can be aggregated into event-level feedback and pace metrics for users. We train a sequence of complementary deep learning models: a pose-only Temporal Convolutional Network (TCN), a Hyperband-tuned variant of the same architecture, an RGB-based 3D convolutional network (3D-CNN), and finally a frame-level Multi-Stage TCN (MS-TCN) over pose sequences.

The pose TCNs operate on joint trajectories extracted from each meal video and are designed as fast, privacy-preserving baselines, while the 3D-CNN consumes short clips of raw frames to capture appearance and motion cues that are not visible in pose alone (Bai et al., 2018; Tran et al., 2015). The MS-TCN builds on these ideas by predicting intake at every frame rather than at the window level, providing a denser and more expressive temporal model for evaluating intake timelines and pace.

For the window-based models (baseline TCN and RGB 3D-CNN), we operate on the fixed-length temporal windows constructed as described in section 2. Each window is assigned a binary label indicating intake or background based on its overlap with annotated intake events.

To avoid leakage, all windows from a given eating session are assigned exclusively to either the training, validation, or test split. This windowed representation enables the models to learn short, localized temporal patterns associated with intake while operating on uniformly sized inputs.

The pose-based branch uses a strong TCN backbone. For each window, we load the full sequence of 2D joint coordinates from the EatSense pose files, subsample or interpolate them to a fixed number of timesteps, and standardize the features with a per-window z-score. These sequences have shape (T, F), where T is the number of timesteps and F is the number of pose features.

We propose PoseTCNPro, a pose-only Temporal Convolutional Network that operates on fixed-length temporal windows and serves as our primary pose-based model. PoseTCNPro is designed to capture short- to medium-range temporal dependencies in wrist and arm motion while remaining lightweight enough for efficient inference.

The model first applies a 1D convolution to project the F-dimensional pose feature vector into a base channel width, followed by a stack of dilated temporal convolutional blocks with depth-wise separable convolutions, residual connections, and squeeze-and-excitation (SE) attention over channels. Dilation factors increase across blocks (e.g., 1, 2, 4, 8, 16), allowing the network to achieve a large temporal receptive field while keeping the number of parameters modest. An attention-based pooling layer aggregates the temporal dimension into a single feature vector, which is passed through a small fully connected head to produce a single logit for binary intake prediction.

A second implementation of this architecture in Keras is tuned with Hyperband, exploring a range of base widths, dropout rates, and learning rates while keeping the overall structure fixed.

For the RGB baseline, we use a compact residual 3D convolutional neural network, which we refer to as VideoResNet3D. This model serves as the RGB 3D-CNN in our comparison.

For each temporal window, we load the corresponding frames from disk, resize them to a square resolution, and uniformly sample a fixed-length clip (e.g., 16 frames) across the window interval. This produces an input tensor of shape (3, T, H, W).

The VideoResNet3D architecture begins with a 3D convolutional “stem” and then applies several stages of residual 3D blocks with spatial and temporal down-sampling. Each block includes two 3D convolutions, batch normalization, GELU activations, dropout, and a squeeze-and-excitation (SE) module that re-weights channels based on global spatiotemporal context. After the final stage, we apply an attention pooling layer over time (averaging over the spatial dimensions) so that the model can focus on the most informative frames, followed by a small fully connected head that outputs a single logit. This architecture is intentionally smaller than typical video backbones to keep training feasible on a single GPU while still capturing motion-texture patterns relevant to eating behavior (Tran et al., 2015).

For the MS-TCN, we move from fixed-length windows back to the frame level and operate directly on frame-wise pose features. The model follows the multi-stage refinement architecture commonly used in temporal action segmentation: an initial stack of dilated temporal convolutional layers produces per-frame logits, and two subsequent stages iteratively refine these predictions by operating on the outputs of the previous stage.

This design enables the model to smooth predictions over time, correct local inconsistencies, and better capture the temporal structure of short intake events embedded within long non-intake sequences. To account for the strong class imbalance, training uses a class-weighted cross-entropy loss that up-weights intake frames. Model performance is evaluated at the frame level using confusion matrices, ROC curves, and precision–recall curves.

We train all models using a similar supervised learning procedure. For each split, we construct PyTorch datasets that stream windows or frame sequences from disk and apply light data augmentation: horizontal flips and mild brightness/contrast jitter for RGB frames, and per-window or per-sequence normalization for pose. To address the strong class imbalance between intake and non-intake behavior, we combine class-balanced sampling with either a weighted binary cross-entropy loss for the window-based models or class-weighted cross-entropy for the frame-level MS-TCN. Models are optimized with the AdamW optimizer using mini-batches of 8–16 examples, learning rates on the order of 3×10^{-4}, and small weight decay. Training runs for a fixed number of epochs (e.g., 10–12), with a checkpoint saved whenever the validation F1 or PR AUC improves. Mixed-precision training is used where available to reduce memory usage and speed up iterations.

During model optimization, we explore a small set of hyperparameters for each architecture. For the pose TCNs, we vary the base channel width (e.g., 64 vs. 128), kernel size, dilation schedule, and dropout rate, along with the learning rate; the Keras implementation uses Hyperband to search this space more systematically. For the 3D-CNN, we experiment with different base channel widths, depth of residual stages, clip length, image resolution, and dropout, as well as the learning rate and strength of data augmentation. For the MS-TCN, we adjust the number of stages, depth of each stage, dilation schedule, and class-weighting scheme. For all models, we also tune the decision threshold on the validation set by sweeping over possible probability cutoffs and selecting the one that maximizes F1 subject to minimum precision and recall floors. The final configuration for each branch, including the chosen architecture, training hyperparameters, and operating threshold, is then carried forward to the Results section, where we compare window-based and frame-level performance and discuss their implications for a future fused, on-device BitePulse AI system.

5 Result

The BitePulse pose dataset is extremely imbalanced at the window level. In the original validation split there are 21,184 windows, but only 57 are labeled as intake (about 0.27%). On top of these window-based experiments, the MS-TCN is

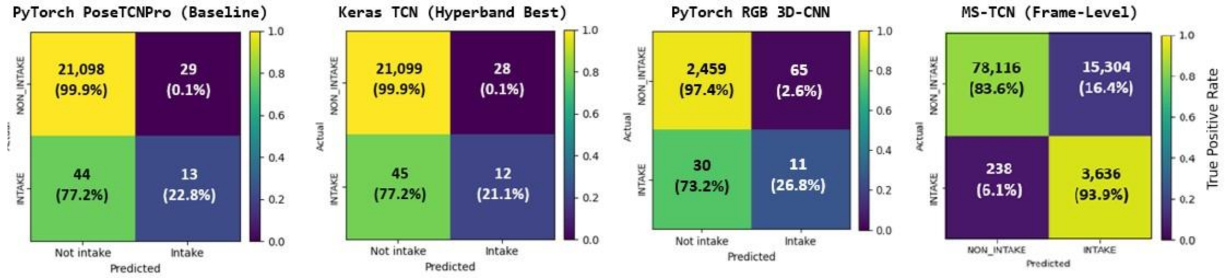


Figure 5.1 Confusion matrices for BitePulse AI Models

trained and evaluated at the frame level, where roughly 5% of frames are labeled as INTAKE.

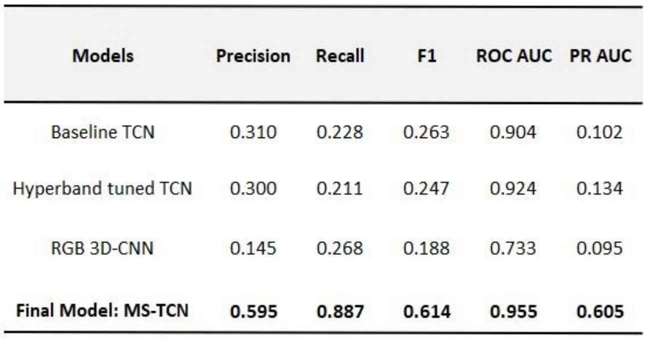
Figure 5.1 presents the confusion matrices for all BitePulse AI models. For the window-based pose TCNs (baseline and Hyperband-tuned), the top-left cells dominate, indicating that the models correctly classify the vast majority of non-intake windows while detecting only a small number of intake events. This reflects very high specificity but low sensitivity, consistent with the extreme class imbalance at the window level.

The RGB 3D-CNN shows a different error profile. While it detects more intake windows than the pose-based TCNs, it also produces substantially more false positives, resulting in lower overall precision. This trade-off suggests that appearance-based cues alone are insufficient to reliably distinguish intake from background motion at short temporal scales.

In contrast, the frame-level MS-TCN exhibits a markedly different confusion pattern, with a much stronger balance between true positives and true negatives. This indicates that modeling longer temporal context at the frame level substantially improves the detection of intake events while maintaining reasonable specificity.

Table 5.1 summarizes precision, recall, F1 score, ROC AUC, and PR AUC for all evaluated models. Among the window-based approaches, the baseline and Hyperband-tuned TCNs achieve similar performance, with strong ROC AUC values but limited precision–recall performance, reflecting the difficulty of detecting rare intake windows.

Table 5.1 Macro precision, recall, F1, ROC AUC, and PR AUC for all BitePulse AI models



Models	Precision	Recall	F1	ROC AUC	PR AUC
Baseline TCN	0.310	0.228	0.263	0.904	0.102
Hyperband tuned TCN	0.300	0.211	0.247	0.924	0.134
RGB 3D-CNN	0.145	0.268	0.188	0.733	0.095
Final Model: MS-TCN	**0.595**	**0.887**	**0.614**	**0.955**	**0.605**

The RGB 3D-CNN exhibits weaker overall ranking performance, consistent with its higher false-positive rate observed in the confusion matrix. Across all window-based models, PR AUC remains low, underscoring the challenge of maintaining practical precision when positive examples are sparse.

The frame-level MS-TCN clearly outperforms all window-based baselines across metrics. Its substantially higher precision–recall performance highlights the benefit of directly modeling frame-wise temporal structure rather than relying on short, fixed-length windows.

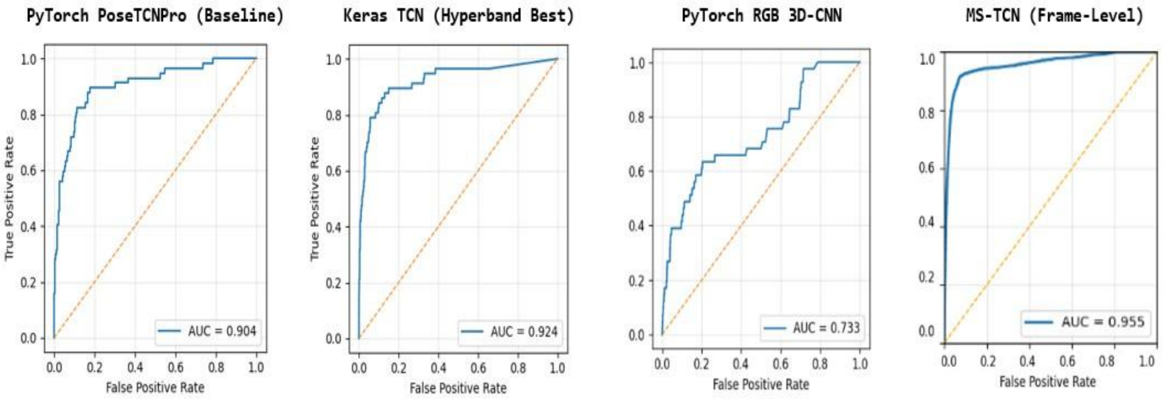


Figure 5.2 ROC curves for BitePulse AI Models

Figure 5.2 shows the ROC curves for all models. The baseline and Hyperband-tuned pose TCNs trace smooth curves well above the diagonal, indicating that they learn a meaningful ranking of intake versus non-intake windows despite severe class imbalance. The RGB 3D-CNN curve lies noticeably closer to the diagonal, consistent with its weaker ranking performance.

The frame-level MS-TCN dominates the ROC space, achieving consistently higher true positive rates across false positive thresholds. This further demonstrates the advantage of frame-level temporal modeling for distinguishing intake behavior from background motion.

Figure 5.3 provides a more informative view of model performance under class imbalance through precision–recall curves. All window-based models exhibit very low precision across most recall levels, indicating that even when intake windows are detected, false positives remain frequent.

In contrast, the frame-level MS-TCN achieves substantially higher precision over a wide range of recall values. This improvement reflects its ability to leverage longer temporal context to reduce spurious detections and better localize intake events. The precision–recall curves therefore reinforce the conclusion that frame-level modeling is critical for achieving practical intake detection performance in highly imbalanced settings.

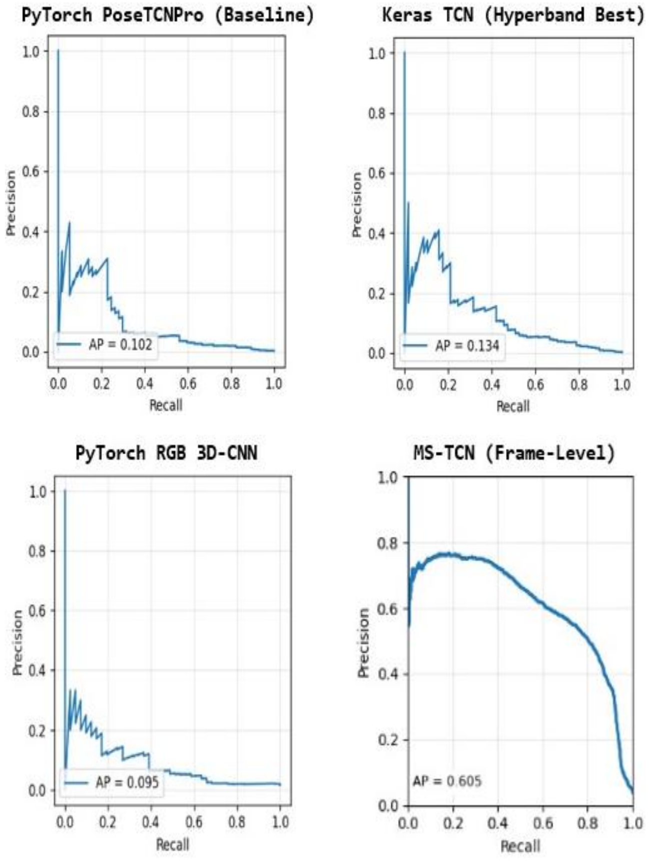


Figure 5.3 Precision recall curves for BitePulse AI Models

6 BitePulse AI Application

The BitePulse AI prototype is delivered as a Streamlit web application and is publicly accessible here. The application runs directly in the browser using the webcam and provides live bite detection and eating-pace feedback without uploading or storing video. When a user opens the app and grants camera access, WebRTC creates a temporary media stream for the session. All computer-vision processing happens within that session; only an annotated overlay and summary statistics are rendered back to the browser, and no raw frames are written to disk or sent to external services.

Once the user clicks Start, the app begins a recording session and continuously updates three views: the live video feed with a mouth box, wrist markers, and an INTAKE / NON_INTAKE status indicator; a panel of numeric statistics; and a short textual "coach" describing the current pace as Slower, Typical, or Faster. A session is defined simply as the time between Start and Stop, and is intended to feel like a low-friction mirror rather than a formal test.

All metrics are computed online and reset at the beginning of each session. The app tracks session duration and uses detected bite events to compute bites per minute over the entire session as well as a rolling 30-second bites-per-minute curve that highlights short-term accelerations or slowdowns. It maintains a running count of bites, the proportion of frames classified as intake, and the number of long gaps between bites that qualify as pauses (for example, gaps longer than ten seconds). From the sequence of bite timestamps, the application computes inter-bite intervals, summarizes them with the median and 75th percentile, and uses the session-average bites-per-minute to assign a categorical pace label: slower below a low threshold, typical within a mid-range, and faster above a higher threshold. The same timestamps are split into the first and second halves of the session to show whether the user speeds up or slows down over time. At each bite, the nearer wrist to the mouth is recorded, which allows the app to estimate the proportion of bites taken with the left versus the right hand, which can serve as a proxy for eating consistency and potential hand-dominance shifts during a meal.

Under the hood, the application relies on a geometric detector built on MediaPipe. Each incoming frame from the WebRTC stream is processed with MediaPipe Face Mesh and MediaPipe Pose. Lip landmarks are used to define a mouth bounding box and center, and pose landmarks provide the locations and visibility scores for the left and right wrists. The app computes the Euclidean distance between the mouth center and the closer wrist. If this distance falls below a calibrated pixel threshold at the current resolution, the frame is tagged as an intake frame; otherwise, it is considered non-intake.

Rather than counting every intake frame as a bite, the system maintains a small state machine that tracks runs of consecutive intake frames. When a run exceeds a minimum length and then ends, the app registers a single bite event. This temporal smoothing reduces spurious detections from brief occlusions or incidental hand movements and produces a cleaner bite timeline.

As bite events are detected, the analytics module updates all derived statistics in real time and refreshes the charts, while the overlay module draws the mouth and wrist markers on the live video. All computation happens inside the Streamlit process, and the only output that leaves the runtime is the rendered web page, which keeps the privacy story simple and avoids the need for a separate model-serving stack.

To contextualize this heuristic baseline, we evaluated the MediaPipe-based intake detector offline against the same labeled EatSense data used to assess our learned models. As shown in Section 5, the heuristic achieves reasonable precision under controlled conditions but is substantially outperformed by the frame-level MS-TCN in both recall and overall precision–recall behavior, particularly for subtle or temporally extended intake events.

Although the frame-level MS-TCN over pose sequences delivers substantially better precision–recall performance in offline evaluation, this model is not deployed in the current application. The MS-TCN is trained on EatSense and has not yet been validated across the full range of camera positions, lighting conditions, utensils, and eating styles that a public-facing app would encounter. In addition, its architecture is better suited to processing longer temporal windows than to making causal predictions from a small number of recent frames, which complicates low-latency streaming use cases.

In contrast, the MediaPipe-based approach attains practically useful accuracy on typical laptop webcams with minimal compute overhead and a clear, easily explainable privacy model. In the current design, the MS-TCN therefore serves as an offline reference model, defining what high-quality intake timelines and pace metrics should look like, while the Streamlit application focuses on delivering a robust, fully on-device experience accessible from any modern browser. Future iterations may narrow this gap by introducing a causal, lightweight temporal model informed by real-world usage data, but the present version already demonstrates that live, privacy-preserving eating-pace feedback is technically feasible.

7 Conclusion

BitePulse AI started from a simple question: can short meal videos, captured on an everyday device, be converted into reliable bite detections and eating-pace feedback fast enough to coach someone in real time? Using the EatSense dataset and a series of temporal deep learning models, the project shows that the answer is largely yes. Window-based pose TCNs and an RGB 3D-CNN provided reasonable ranking of intake versus non-intake behavior, but the frame-level MS-TCN over pose sequences emerged as the strongest model. It achieved substantially higher recall and PR AUC than all other architectures and produced dense, stable intake timelines that are well suited for computing user-facing metrics such as bites per minute, pause structure, and inter-bite intervals.

The most significant result is the performance gap between MS-TCN and the window baselines. At the frame level, collapsing the 16 EatSense actions into INTAKE versus NON_INTAKE and training a multi-stage temporal convolutional network yielded strong macro scores and a PR AUC above 0.60, compared with values near 0.10 to 0.13 for the window-based TCNs and the 3D-CNN. This shows that treating eating behavior as a dense sequence labeling problem, rather than as isolated windows, is crucial when intake events are short and rare. It also suggests that temporal refinement stages can recover a large fraction of true intake frames without sacrificing specificity. In this sense, MS-TCN serves as a gold-standard model for the project and provides clear evidence that modern temporal models can turn raw meal videos into accurate, interpretable pace signals.

Several findings were unexpected. The compact RGB 3D-CNN underperformed the pose-only TCNs even though it had access to full visual context. This suggests that, for the current data and training regime, structured pose features carry most of the discriminative information for intake detection, and that taking advantage of appearance cues may require larger models, stronger pretraining, or explicit fusion with pose. The extreme imbalance at the window level, with only about 0.27 percent intake windows in the validation split, also made it very difficult for any window-based model to achieve high precision and recall at the same time, even when ROC AUC looked healthy. Moving to frame-level labels with roughly 5 percent intake frames helped, but also highlighted how sensitive evaluation is to the chosen representation. Finally, in the deployed Streamlit app, a simple geometric heuristic based on MediaPipe landmarks delivered surprisingly usable bite counts and pace trends, which reinforced the idea that an online system does not always need the most sophisticated model to create value for users (Lugaresi et al., 2019).

If this work were to continue, the next step would be to connect the offline MS-TCN with the live BitePulse application. One natural path is to distill MS-TCN into a lighter, causal variant that can run online, using the current MediaPipe-based detector as a teacher and the EatSense labels as ground truth (Lugaresi et al., 2019). This kind of "TCN-lite" could be integrated as an optional backend in the app, either on device for powerful phones and desktops or behind a small GPU-backed service, with careful attention to latency and privacy. In parallel, the label space could be expanded beyond "eat it" to include chewing and sipping actions, enabling richer feedback such as bite-chew ratios, sip patterns, or alternation between food and drink. That extension would require multi-class frame labeling, additional annotated data that captures a wider variety of utensils and foods, and updated evaluation focused on how these extra signals change the interpretation of pace.

From an application perspective, BitePulse can grow from simple pace labels into a more personalized coach. With repeated sessions, the system could estimate each user's typical range of bites per minute and inter-bite intervals, then frame recommendations relative to that personal baseline rather than to fixed global thresholds. Session summaries could translate raw statistics into concrete suggestions, such as encouraging an extra pause between bites, pointing out when the second half of a meal consistently speeds up, or noting when sip patterns indicate rushing drinks instead of food. To productionize the system, it would also be necessary to harden telemetry and monitoring, build robust device-compatibility tests, and run user studies that evaluate not only detection accuracy but also perceived usefulness, comfort, and actual behavior change.

Overall, BitePulse AI demonstrates that research-grade models like MS-TCN and pragmatic webcam-friendly heuristics can work together to bridge the gap between eating-behavior science and everyday digital coaching. The current prototype validates the feasibility of extracting bite timelines and pace metrics from real meal videos. Future work lies in broadening the behaviors detected, strengthening generalization across environments, and turning these signals into gentle, adaptive recommendations that help people experiment with eating more mindfully in their daily lives.

Works Cited

Andrade, A. M., Greene, G. W., & Melanson, K. J. (2008). Eating slowly led to decreases in energy intake within meals in healthy women. *Journal of the American Dietetic Association, 108*(7), 1186–1191. https://www.jandonline.org/article/S0002-8223(08)00518-X/abstract

Bai, S., Kolter, J. Z., & Koltun, V. (2018). *An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv*. https://arxiv.org/abs/1803.01271

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C. L., Yong, M. G., Lee, J., Chang, W. T., Hua, W., Georg, M., Grundmann, M., & Google AI. (2019). *MediaPipe: A framework for building perception pipelines*. arXiv. https://arxiv.org/abs/1906.08172

M. A. Raza, L. Chen, L. Nanbo, R. B. Fisher (2023). *EatSense: Human Centric, Action Recognition and Localization Dataset for Understanding Eating Behaviors and Quality of Motion Assessment, Image and Vision Computing*. https://groups.inf.ed.ac.uk/vision/DATASETS/EATSENSE/

Rouast, P. V., Heydarian, H., Adam, M. T. P., & Rollo, M. E. (2020). *OREBA: A dataset for objectively recognizing eating behaviour and associated intake. arXiv*. https://arxiv.org/abs/2007.15831

Su, Y. C., Wang, W. M., Wang, S. Y., Lu, S. N., Chen, L. T., Wu, D. C., Chen, C. Y., Jan, C. M., & Horowitz, M. (2000). The association between *Helicobacter pylori* infection and functional dyspepsia in patients with irritable bowel syndrome. *American Journal of Gastroenterology, 95*(8), 1900–1905. https://pubmed.ncbi.nlm.nih.gov/10950033/

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). *Learning spatiotemporal features with 3D convolutional networks.* In *2015 IEEE International Conference on Computer Vision (ICCV)* (pp. 4489–4497). IEEE. https://arxiv.org/abs/1412.0767

Zhu, B., Haruyama, Y., Muto, T., Yamazaki, T., Sobue, T., & Koyama, W. (2015). *Association between eating speed and metabolic syndrome in a three-year population-based cohort study.* Journal of Epidemiology, *25*(5), 332–336. https://pubmed.ncbi.nlm.nih.gov/25787239/