IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 October 2026, accepted 21 October 2026, date of publication 26 October 2026, date of current version 30 October 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3746970

RESEARCH ARTICLE

Density-Guided Repetition Cycle Localization

DONGHOON KANG, (Member, IEEE)
Intelligence and Interaction Research Center, Korea Institute of Science and Technology (KIST), Seoul 02792, South Korea
e-mail: kimbab.moowoo@gmail.com

This work was supported in part by the Korea Institute of Science and Technology (KIST) under Grant 2V09141, Grant 2V09232 (KIST K-DARPA Project), and Grant 26E0211 (KIST Institutional Project); and in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) under Grant RS-2022-NR070020.

ABSTRACT Repetitive action counting in videos has been widely studied, whereas localizing the start and end of each repetition has received less attention. Most existing counting models predict a frame-level density map and obtain the repetition count by summing its values over time. Gaussian density targets centered at cycle midpoints provide useful cues for center estimation but do not explicitly represent cycle boundaries. Based on this observation, we propose CycLocNet (Cycle Localization Network), a density-guided, class-agnostic network for localizing individual repetition cycles. CycLocNet estimates the number and approximate centers of cycles from the cumulative density produced by a frozen, pretrained counting model. At the temporal token nearest to each estimated center, it predicts the distances to the start and end boundaries as separate probability distributions over discrete distance candidates. We evaluate CycLocNet on RepCount and UCFRep using Period-mAP and AP at different temporal IoU thresholds. The proposed boundary prediction head improves cycle localization over density-only decoding. Code and models are available at https://github.com/kimbabmoowoo/CycLocNet

INDEX TERMS Class-agnostic repetition counting, repetition cycle localization, density map, temporal boundary prediction.

I. INTRODUCTION

Repetitive action counting in videos, which automatically counts repetitive human actions [1], [2], [3], [4], [5], has been widely studied. This task has also been explored using sensor sequences [6], including wearable sensor data [7]. However, compared with repetition counting, localizing the start and end of each repetition has been less explored. Recent studies, including DeTRC [8] and MultiCounter+ [9], have investigated the localization of individual repetition cycles.

In DeTRC, each dynamic action query predicts a repetitive-action score as well as the midpoint and duration of a cycle. Queries classified as repetitive actions correspond to detected cycles, and the number of these queries determines the predicted repetition count. Although DeTRC provides temporal cycle intervals, its experiments focus on counting accuracy and do not provide a separate quantitative evaluation of cycle localization. In contrast, MultiCounter+ quantitatively evaluates repetition localization using spatiotemporal tubes in multi-person videos. Its primary objective is to detect and track multiple human instances and count the repetitions of each individual. Although it is action-agnostic, its formulation explicitly models person-level bounding boxes and tracks instance identities over time.

Repetition cycle localization also differs from conventional temporal action localization. General temporal action localization methods identify action instances using predefined semantic categories and often separate them from background segments [10], [11]. In repetition videos, however, consecutive cycles usually contain similar motion patterns and may have little or no background between them. Their durations and temporal boundaries may also vary within the same video. These characteristics make it difficult to distinguish adjacent cycles using semantic or background cues alone.

We consider class-agnostic temporal cycle localization without requiring person detection or tracking. In this setting, cycle boundaries are inferred from temporal repetition patterns rather than semantic action labels. This allows the same localization framework to be applied across different repetitive actions without training a separate detector for each category.

To localize the start and end of each cycle, we use the density map predicted by a repetition-counting model as an initial temporal cue. In density-based counting, each

The associate editor coordinating the review of this manuscript and approving it for publication was Inês Domingues.

© 2026 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

cycle is represented by a Gaussian density target centered at its temporal midpoint [12], [13]. The cumulative density therefore provides estimates of the number and approximate centers of individual cycles, which serve as reference points for boundary prediction. However, the density map does not explicitly represent the start and end boundaries, making precise cycle localization difficult.

Based on this observation, we propose CycLocNet, a density-guided, class-agnostic framework for localizing individual repetition cycles. The main contribution of this work is a formulation that connects density-based cycle initialization with learned boundary refinement. Cumulative density provides the estimated number and approximate centers of cycles, while temporal features from the same frozen counting model are used to predict their boundaries. At the temporal token nearest to each estimated center, the boundary head predicts separate probability distributions over start and end distances. Their expected values determine the cycle interval. Predicting the two distances separately allows asymmetric intervals without requiring the estimated center to coincide with the true cycle midpoint.

The counting model remains frozen during cycle-localization training, and only the boundary prediction head is optimized using annotated cycle intervals. Distribution focal loss (DFL) [14] supervises the start- and end-distance distributions, while interval IoU loss supervises the overlap between the decoded intervals and the target intervals. The head learns boundary distances at all temporal tokens within annotated cycles, rather than only at their midpoints. It therefore learns boundary information that the density map does not explicitly encode.

We evaluate CycLocNet on RepCount and UCFRep using the annotated start and end positions of individual repetition cycles. Localization performance is measured using Period-mAP and AP at different temporal IoU thresholds. Ablation studies compare distributional boundary prediction with density-only decoding, symmetric-duration prediction, and direct distance regression.

The main contributions of this work are summarized as follows:

- We introduce a density-guided formulation for repetition-cycle localization that combines cumulative-density-based cycle initialization with boundary refinement learned from temporal features.
- We predict separate start- and end-distance distributions, supervised by DFL and interval IoU loss throughout each annotated cycle. This formulation supports asymmetric intervals decoded from approximate cycle centers.
- Experiments on RepCount and UCFRep show that CycLocNet achieves higher localization accuracy than density-only decoding. Boundary-representation and loss ablations further show the benefit of distributional boundary prediction trained with DFL, particularly at the stricter temporal IoU threshold.

II. RELATED WORK

A. REPETITIVE ACTION COUNTING

For class-agnostic repetition counting in videos, RepNet [15] introduced a temporal self-similarity matrix as an intermediate representation and predicted the period length and periodicity of each frame. RepNet does not require individual cycle-boundary annotations and is trained with synthetic repetition videos that provide period-length and periodicity targets. Most subsequent methods, however, use cycle-boundary annotations to learn a density map whose sum gives the repetition count.

TransRAC [12] introduced a density-based approach that encodes multi-scale temporal correlations using transformers and predicts the repetition count through density map regression. ESCounts [16] used action exemplars during training while enabling exemplar-free inference. TBAN [17] combined boundary-aware features with actionness-guided density estimation to handle varying cycle durations and motion interruptions. LMRL [18] introduces foreground localization to distinguish repetitive-action segments from background frames and improve repetition counting. Unlike our method, it does not localize the start and end boundaries of individual repetition cycles. HTRM-Net [19] integrated diverse temporal self-similarity matrices with local temporal context to improve density prediction. SkimFocusNet [13] uses a skim branch to guide density prediction in a focus branch and serves as the pretrained counting model in our framework.

B. TEMPORAL ACTION LOCALIZATION AND BOUNDARY REGRESSION

Temporal action localization aims to identify action instances in videos by predicting the start and end frames of each action interval. Accurate boundary prediction is important because localization quality depends on the temporal overlap between predicted and ground-truth intervals. BMN [10] constructs temporal proposals by matching candidate start and end boundaries and estimating their confidence scores. ActionFormer [11] instead adopts an anchor-free approach that directly regresses the distances from each temporal position to the action boundaries.

The exact boundary may be ambiguous when an action gradually transitions into another state. Representing each boundary as a single value cannot adequately capture this uncertainty. Generalized Focal Loss [14] addresses this issue in object detection by representing bounding-box locations as distributions over discrete values. TriDet [20] applies a related formulation to temporal action localization by modeling relative boundary locations as probability distributions. The proposed CycLocNet builds on this distributional boundary prediction approach through a density-guided formulation for individual cycle localization. CycLocNet is distinguished by its use of both the density map and temporal features from a frozen counting model. Cumulative density

TABLE 1. Comparison of repetition-cycle localization methods.

Method	Cycle intervals	Quantitative evaluation	Category-independent	Person annotations
DeTRC [8]	Yes	No	Yes	Not required
MultiCounter+ [9]	Yes	Yes	Yes	Required
Ours (CycLocNet)	Yes	Yes	Yes	Not required

determines the number and approximate centers of cycles, while the temporal features are used to predict separate start- and end-distance distributions for boundary decoding at those centers.

C. REPETITION CYCLE LOCALIZATION

DeTRC [8] formulated class-agnostic repetition counting as a set-prediction problem using dynamic action queries. Each query predicts a repetitive-action score and a cycle interval defined by its midpoint and duration; the number of queries classified as repetitive actions determines the predicted count. Although DeTRC provides explicit cycle intervals, its evaluation focuses on counting accuracy without separately presenting quantitative results for cycle-boundary localization.

MultiCounter+ [9] quantitatively evaluates cycle localization in multi-person videos by jointly detecting, tracking, and counting the repetitive actions of multiple people. Its frame-level period-velocity and periodicity predictions are used to construct spatiotemporal tubes. MultiCounter+ uses person bounding boxes and identity annotations to distinguish individuals.

In this work, we present a class-agnostic model that leverages the density map used for repetition counting. The cumulative density provides estimates of the number and approximate centers of cycles, which guide boundary prediction. Unlike MultiCounter+, our method does not require person detection or tracking. We quantitatively evaluate the predicted start and end boundaries of individual cycles.

Table 1 summarizes the differences among these approaches. Unlike MultiCounter+, the proposed CycLocNet performs temporal cycle localization without person-level annotations. It also quantitatively evaluates the predicted cycle intervals, which are not separately evaluated in DeTRC.

III. PROPOSED METHOD

A. OVERVIEW

CycLocNet consists of a frozen density-based counting model and a trainable boundary prediction head. As illustrated in Fig. 2, the boundary prediction head is trained to predict start- and end-distance distributions using annotated cycle intervals. At inference, as shown in Fig. 5, the cumulative predicted density is used to estimate the number and centers of individual cycles. The predicted distance distributions are then used to decode the corresponding cycle intervals.

B. TRAINING

1) FROZEN DENSITY-BASED REPETITION COUNTING MODEL

CycLocNet uses SkimFocusNet [13] for repetition counting. Fig. 1 illustrates the SkimFocusNet architecture adapted to our cycle-localization framework. SkimFocusNet consists of skim and focus branches. The skim branch processes a contextual view and selects informative frames, whereas the focus branch uses these frames to guide density prediction from fine-grained views.

We make two modifications to the original SkimFocusNet. First, we replace the VideoSwin backbone in both branches with a pretrained VideoMAE v1 encoder [21]. VideoMAE represents every two consecutive frames as a 768-dimensional temporal token. The skim and focus branches apply separate projection blocks to the frozen VideoMAE features. Each block projects the features from 768 to 512 dimensions, followed by layer normalization and GELU activation. In the skim branch, a learned positional embedding and a two-layer Transformer encoder further process the projected contextual features before the tiny decoder. These added layers are trained with the counting model and then frozen during CycLocNet training. Second, we construct the contextual view by resampling the full-video VideoMAE token sequence to 256 temporal tokens using linear interpolation. The skim branch then selects the top-k temporal tokens according to the scores predicted by the tiny decoder.

Following SkimFocusNet [13], the focus branch uses the informative frames to guide density prediction from 64-token fine-grained fragments. For each fragment, it produces a feature matrix $\mathbf{X}_{CF}^{n} \in \mathbb{R}^{64\times512}$ and a density map $\mathbf{d}^{n} \in \mathbb{R}^{64}$. The fragment outputs are combined in temporal order. Features and density values at overlapping positions are averaged, and padded positions are removed. The resulting full-video outputs are $\mathbf{X}_{CF} \in \mathbb{R}^{T\times512}$ and $\mathbf{d} = [d_1, \ldots, d_T] \in \mathbb{R}^{T}$ where T denotes the number of temporal tokens in the video. Since each VideoMAE v1 temporal token represents two consecutive frames, T is approximately half the number of video frames. The predicted repetition count is

$$\hat{c} = \sum_{i=1}^{T} d_i. \tag{1}$$

Hereafter, we refer to this adapted version of SkimFocusNet as the counting model.

2) BOUNDARY PREDICTION HEAD

Most density-based counting models, including TransRAC [12] and SkimFocusNet [13], construct the density

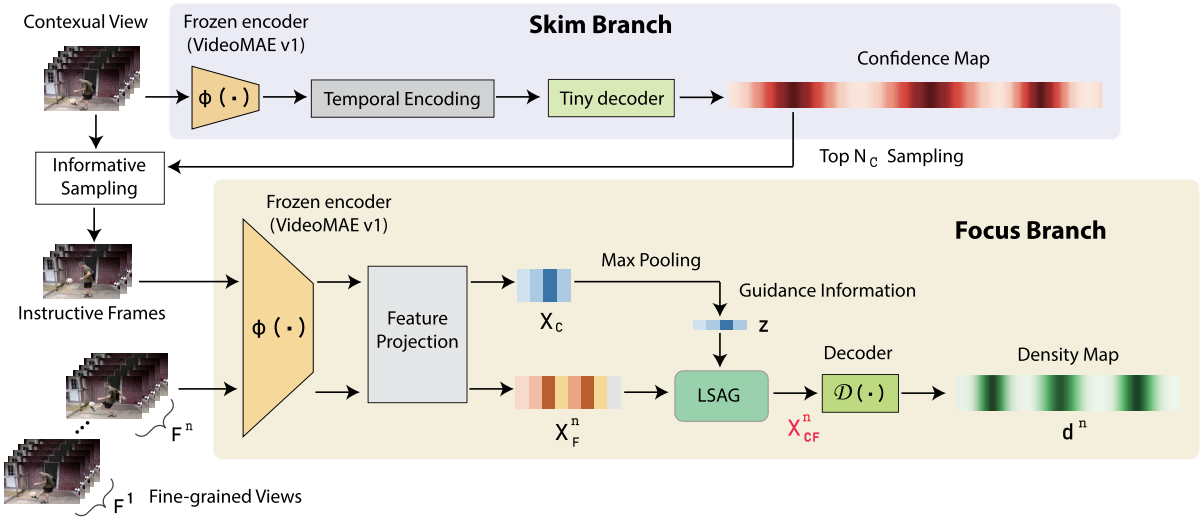


FIGURE 1. Architecture of SkimFocusNet [13], originally designed for repetition counting. In our framework, SkimFocusNet is adapted to provide a repetition density map and temporal features for cycle-boundary localization. We replace the original VideoSwin backbone with a pretrained VideoMAE v1 and construct the contextual view by temporally resampling the full-video token sequence to 256 tokens.

target for each cycle as a Gaussian centered at its temporal midpoint. The cumulative density thus provides cues about the number and approximate center locations of individual cycles. However, because the start and end boundaries are not explicitly represented in the density target, they may not be precisely recovered from the cumulative density alone. This limitation is further amplified by density prediction errors and, when present, background intervals between consecutive cycles. As demonstrated in Section IV, directly decoding cycle intervals from the cumulative density yields substantially lower localization accuracy, particularly at strict temporal IoU thresholds.

To address this limitation, the boundary prediction head uses $\mathbf{X}_{CF}$ to predict the distances to the start and end boundaries at every temporal token. The density map is not used as input to the head. During training, the annotated cycle intervals determine the supervised tokens and their target distances to the corresponding boundaries.

Fig. 2 presents the training pipeline of CycLocNet, while Fig. 3 details the architecture of the boundary prediction head. Let $\mathbf{X}_{CF} \in \mathbb{R}^{T\times 512}$ denote the temporal feature matrix produced by the frozen counting model, whose i-th row $\mathbf{x}_i^\top$ represents the i-th temporal token, $i = 1, \ldots, T$. The boundary prediction head applies two fully connected hidden layers of width 512, each followed by a ReLU activation, to each temporal token independently using shared parameters. These hidden layers preserve the matrix dimension of $T \times 512$. A final fully connected layer produces the boundary-logit matrix $\mathbf{U} \in \mathbb{R}^{T\times 2R}$. For each temporal token i, the corresponding logit vector $\mathbf{u}_i \in \mathbb{R}^{2R}$ is divided into two vectors, $\mathbf{u}_i^s, \mathbf{u}_i^e \in \mathbb{R}^R$, representing the start- and end-distance logits, respectively. Separate softmax operations convert them into distance distributions:

$$\boldsymbol{\alpha}_i = \mathrm{softmax}(\mathbf{u}_i^s), \qquad \boldsymbol{\beta}_i = \mathrm{softmax}(\mathbf{u}_i^e). \tag{2}$$

Here, $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$ represent the start- and end-distance distributions, respectively, over $R = 32$ distance bins.

3) BOUNDARY SUPERVISION AND TRAINING LOSS

During training, only temporal tokens within annotated cycles receive boundary supervision. Background tokens are excluded because they have no target cycle boundaries. Each supervised token has two targets: its distances to the cycle's start and end boundaries. The head therefore learns to predict boundaries from positions throughout each cycle, rather than only from its midpoint. Let $[s_k, e_k]$ denote the annotated start and end positions of the k-th cycle in temporal-token coordinates. For the i-th token within this cycle, let p_i denote its temporal position. Its relative distances to the start and end boundaries, denoted by l_i^s and l_i^e, respectively, are computed as

$$l_i^s = p_i - s_k, \qquad l_i^e = e_k - p_i. \tag{3}$$

Both distances are nonnegative and measured in temporal tokens. Target distances are clipped to the maximum supported distance $B_{\max}$ as follows: $\tilde{l}_i^s = \min\{l_i^s, B_{\max}\}$ and $\tilde{l}_i^e = \min\{l_i^e, B_{\max}\}$.

As illustrated in Fig. 3, the boundary prediction head outputs R logits for the start distance and another R logits for the end distance at each temporal token. Applying softmax separately to the two sets of logits produces the start- and end-distance distributions $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$, respectively. Each clipped target distance is encoded by linear interpolation between two adjacent distance bins, using weights computed from the actual bin spacing. If the target equals a bin value, its target probability is one for that bin and zero for all other bins. We use distribution focal loss (DFL) [14] to train the predicted distributions to match these targets.

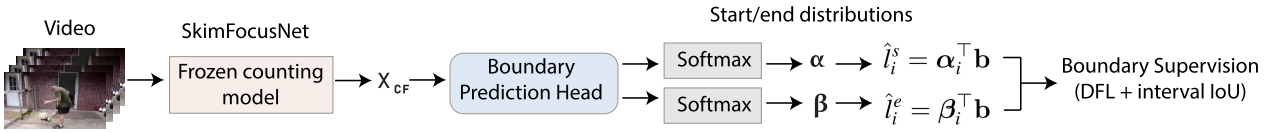


FIGURE 2. Training pipeline of CycLocNet. The frozen counting model produces the temporal feature matrix $\mathbf{X}_{CF} \in \mathbb{R}^{T\times 512}$, from which the boundary prediction head predicts start- and end-distance distributions at every temporal token. The boundary prediction head is trained using distribution focal loss (DFL) and interval IoU loss.

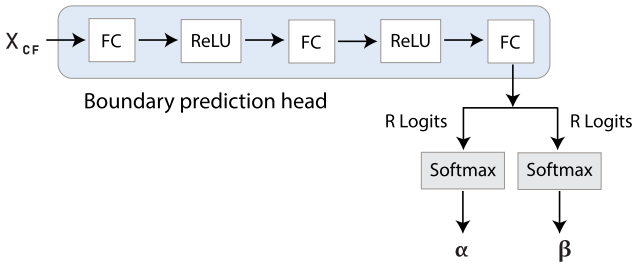


FIGURE 3. Architecture of the boundary prediction head. The two fully connected hidden layers preserve the input dimension of $T \times 512$, while the final fully connected layer produces $2R$ logits at every temporal token. The logits are divided into two groups and normalized separately to obtain the start- and end-distance distribution maps $\alpha, \beta \in \mathbb{R}^{T\times R}$.

Let $\mathbf{b} = [b_0, \ldots, b_{R-1}]^\top$ denote the fixed distance-bin values shared across all temporal tokens. The predicted start and end distances are obtained as the expected values of the corresponding distributions:

$$\hat{l}_i^s = \boldsymbol{\alpha}_i^\top \mathbf{b}, \qquad \hat{l}_i^e = \boldsymbol{\beta}_i^\top \mathbf{b}. \tag{4}$$

These distances define the predicted cycle interval as $[p_i - \hat{l}_i^s,\ p_i + \hat{l}_i^e]$, where $p_i - \hat{l}_i^s$ and $p_i + \hat{l}_i^e$ are the predicted cycle start and end positions, respectively.

The interval IoU loss is $1 - \text{tIoU}$, where tIoU measures the overlap between the predicted interval and the target interval $[p_i - \tilde{l}_i^s,\ p_i + \tilde{l}_i^e]$. Minimizing this loss encourages greater overlap between the two intervals. The overall training loss is

$$\mathcal{L} = \lambda_{\text{DFL}}\mathcal{L}_{\text{DFL}} + \lambda_{\text{IoU}}\mathcal{L}_{\text{IoU}}. \tag{5}$$

The start- and end-distance DFL terms are averaged separately over valid tokens within annotated cycles in each training batch and then summed to obtain $\mathcal{L}_{\text{DFL}}$. The interval IoU loss is averaged over the same tokens to obtain $\mathcal{L}_{\text{IoU}}$. Background and padding tokens are excluded from both losses. The combined loss supervises both the distance distributions and the decoded interval overlap. The hyperparameters λ_{DFL} and λ_{IoU} control the relative contributions of the two loss terms. During training, only the boundary prediction head is optimized, while the counting model remains frozen.

C. INFERENCE

1) DENSITY-BASED CYCLE CENTER ESTIMATION

Following TransRAC [12], the density target for each cycle is a unit-normalized Gaussian centered at its temporal midpoint. Because the Gaussian is symmetric, half of the density lies before and half lies after the cycle center. Thus, when the density is accumulated from the beginning of the video, the centers of consecutive cycles correspond approximately to cumulative values of 0.5, 1.5, 2.5, ….

At inference, the predicted count $\hat{c}$ in Equation (1) is rounded to the nearest integer to obtain the predicted number of cycles $\hat{K}$. We then estimate $\hat{K}$ cycle centers from the cumulative predicted density. Let $\hat{m}_k$ denote the estimated center of the k-th cycle in temporal-token coordinates. We determine $\hat{m}_k$ as the temporal position where the cumulative predicted density reaches $k - \frac{1}{2}$. Fig. 4 illustrates how the thresholds $k - \frac{1}{2}$ and $(k+1) - \frac{1}{2}$ identify $\hat{m}_k$ and $\hat{m}_{k+1}$, respectively.

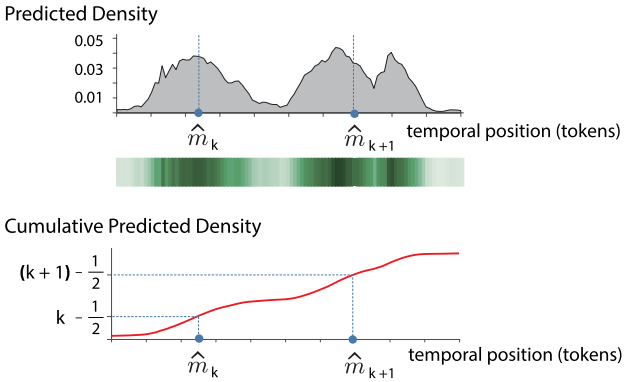


FIGURE 4. Illustration of cycle center estimation from cumulative predicted density. The estimated cycle centers $\hat{m}_k$ and $\hat{m}_{k+1}$ are the temporal positions where the cumulative density reaches $k - \frac{1}{2}$ and $(k+1) - \frac{1}{2}$, respectively.

2) CYCLE BOUNDARY DECODING

The inference pipeline is illustrated in Fig. 5. During training, the boundary head is supervised at all temporal tokens within annotated cycles to predict boundary distances relative to each token. Consequently, at inference, the estimated center $\hat{m}_k$ need not coincide with the true cycle midpoint for boundary decoding.

The estimated center $\hat{m}_k$ of the k-th cycle may lie between two consecutive temporal tokens because it is obtained by linearly interpolating the cumulative predicted density. We therefore round $\hat{m}_k$ to the nearest token index, $j_k = \text{round}(\hat{m}_k)$. At token j_k, the boundary prediction head produces two sets of R logits, one for the start distance and one for the end distance. We apply softmax to each set to obtain the corresponding distributions, $\boldsymbol{\alpha}_{j_k}$ and $\boldsymbol{\beta}_{j_k}$,

respectively. The expected start and end distances are then computed using the distance-bin vector $\mathbf{b}$ defined during training:

$$\hat{l}_k^s = \boldsymbol{\alpha}_{j_k}^\top \mathbf{b}, \qquad \hat{l}_k^e = \boldsymbol{\beta}_{j_k}^\top \mathbf{b}. \tag{6}$$

Since these distances are measured relative to token j_k, the start and end boundaries of the k-th cycle are obtained by subtracting $\hat{l}_k^s$ from j_k and adding $\hat{l}_k^e$ to j_k, respectively:

$$\left[j_k - \hat{l}_k^s, \quad j_k + \hat{l}_k^e \right]. \tag{7}$$

During both training and inference, boundary distances are measured from the position of the token at which they are predicted. Although j_k is an integer, the expected distances $\hat{l}_k^s$ and $\hat{l}_k^e$ can be non-integer, so the predicted start and end positions can also be non-integer in temporal-token coordinates.

IV. EXPERIMENT

A. EXPERIMENTAL SETUP

1) DATASETS

We conduct our cycle-localization experiments on RepCount [12] and UCFRep [22]. Both datasets provide annotated start and end frames for each repetition cycle, enabling the training and evaluation of cycle-localization models. RepCount Part-A contains 758 training, 131 validation, and 152 test videos. UCFRep contains 421 training and 105 validation videos across 23 action categories, with no separate test split. We use the provided splits without repartitioning either dataset. We evaluate cycle localization on the RepCount test split and the UCFRep validation split.

2) EVALUATION METRICS

We evaluate cycle localization using temporal intersection over union (tIoU) between predicted and ground-truth intervals. Because localization is guided by the predicted density map, we also present the counting accuracy of the frozen counting model. Counting accuracy evaluates the total repetition count, whereas cycle-localization accuracy evaluates the temporal alignment between predicted and ground-truth cycle intervals. An accurate count does not necessarily imply accurate cycle boundaries. The evaluation metrics are defined as follows:

- AP@.50 and AP@.75 denote average precision (AP) at tIoU thresholds of 0.50 and 0.75, respectively, following the standard temporal action localization evaluation procedure [11], [23]. Predictions are evaluated in descending order of confidence, and each ground-truth cycle can be matched only once. A prediction is considered correct if its tIoU with the matched cycle meets the corresponding threshold.
- MultiCounter+ [9] introduced Period-AP for repetition localization based on the spatiotemporal IoU between predicted and ground-truth tubes. We adapt this metric to temporal cycle localization by using tIoU between cycle intervals. Period-mAP is the mean AP over tIoU thresholds from 0.50 to 0.95 in increments of 0.05.
- MAE and OBO denote mean absolute error and off-by-one accuracy, respectively, for repetition counting [12].

For AP computation, we rank each predicted interval using $q_k = 1/\left(1+(\sigma_k^s+\sigma_k^e)/\hat{L}_k\right)$, where $\hat{L}_k = \hat{l}_k^s+\hat{l}_k^e$ is the predicted interval length, and σ_k^s and σ_k^e are the standard deviations of the start- and end-distance distributions at token j_k:

$$\sigma_k^s = \sqrt{\sum_{r=0}^{R-1} \alpha_{j_k,r}(b_r - \hat{l}_k^s)^2}, \tag{8}$$

$$\sigma_k^e = \sqrt{\sum_{r=0}^{R-1} \beta_{j_k,r}(b_r - \hat{l}_k^e)^2}. \tag{9}$$

Here, $\alpha_{j_k,r}$ and $\beta_{j_k,r}$ are the probabilities assigned to distance bin b_r. Each standard deviation measures the probability-weighted spread around the corresponding expected distance. A higher score indicates lower boundary uncertainty relative to the interval length. The score is used to rank predictions, rather than to estimate the probability of correct localization. Unless otherwise specified, all results are presented as the mean $\pm$ standard deviation over three experiments. Localization metrics are expressed as percentages.

3) IMPLEMENTATION DETAILS

For the counting model (*i.e.*, the modified SkimFocusNet shown in Fig. 1), we use a VideoMAE v1 ViT-B backbone initialized from the publicly released checkpoint fine-tuned on Kinetics-400 [21]. The backbone remains frozen, and its precomputed features are reused by both the skim and focus branches. The counting model is trained for 150 epochs using AdamW with a weight decay of 5×10^{-2}, a cosine annealing learning rate schedule, and a batch size of 32. The learning rate is 1×10^{-4} on RepCount and 5×10^{-5} on UCFRep. After training, the entire counting model remains frozen during boundary prediction head training (Fig. 2) and inference (Fig. 5).

The distance-bin vector $\mathbf{b}$ in Equation (4) contains $R = 32$ bins with a maximum distance of $B_{\max} = 32$ temporal tokens. The r-th bin is defined as

$$b_r = B_{\max}\left(\frac{r}{R-1}\right)^\gamma, \qquad r = 0, \ldots, R-1, \tag{10}$$

where $\gamma = 1.5$. This power-law spacing provides finer resolution for shorter distances. The maximum distance $B_{\max}$ is a hyperparameter that controls the representable distance range. With R fixed, increasing $B_{\max}$ expands this range but increases the spacing between bins. We use $B_{\max} = 32$ for both datasets. Target distances exceeding $B_{\max}$ are clipped to $B_{\max}$ before computing both DFL and interval IoU loss.

The loss hyperparameters in Equation (5) are set to $\lambda_{\text{DFL}} = 1$ and $\lambda_{\text{IoU}} = 1$. The start- and end-distance DFL terms are averaged separately over temporal tokens within annotated

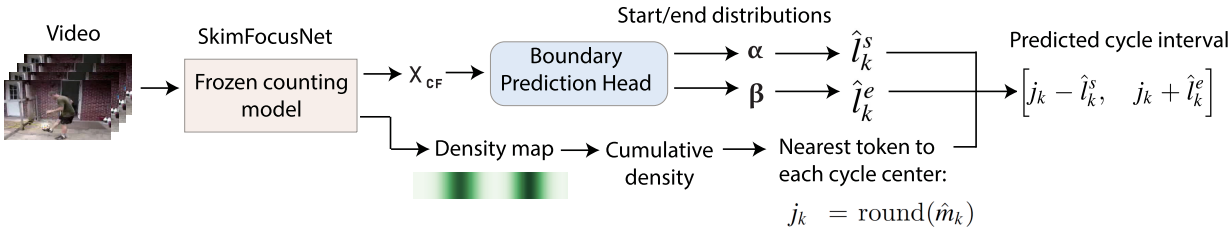


FIGURE 5. Inference pipeline of CycLocNet. The frozen counting model produces the temporal feature matrix $\mathbf{X}_{CF}$ and density map. The boundary prediction head predicts the start- and end-distance distributions, while the density map provides the estimated cycle centers. These outputs are combined to decode the individual cycle intervals.

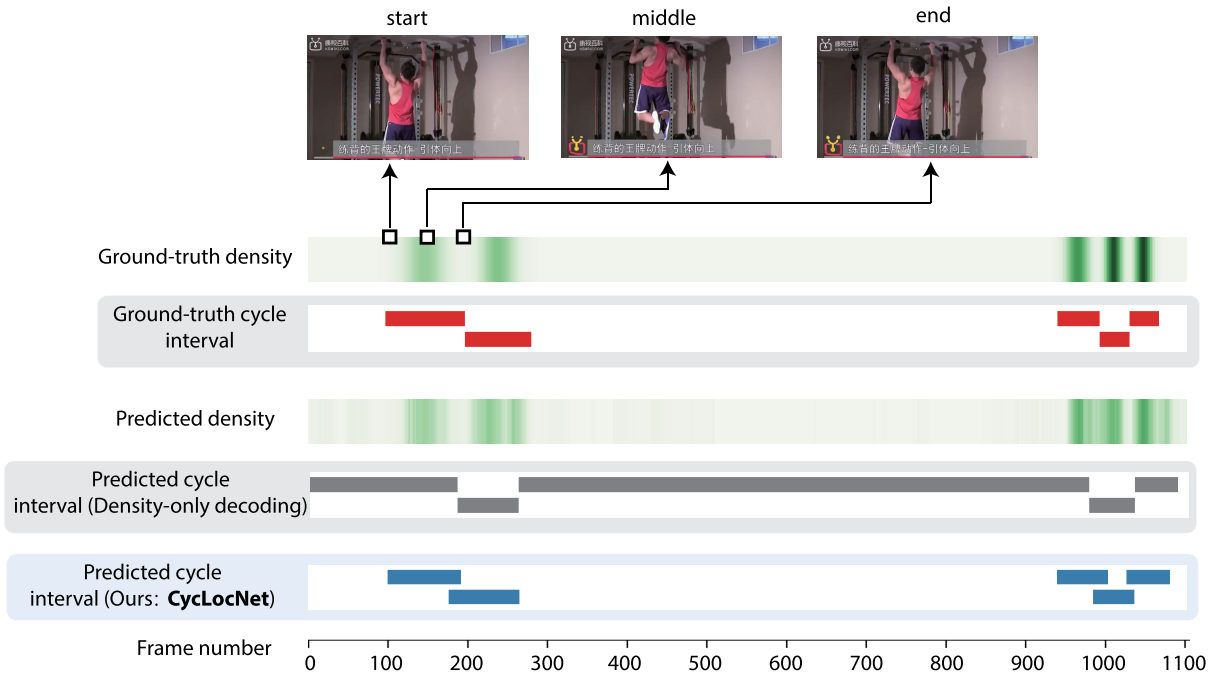


FIGURE 6. Qualitative cycle-localization result on RepCount. Ground-truth and predicted cycle intervals are shown in red and blue, respectively.

cycles and then summed. The interval IoU loss is averaged over the same tokens.

We train the boundary prediction head in Fig. 3 using Adam with a learning rate of 1×10^{-4} and a batch size of 32. Training proceeds for up to 150 epochs and stops early if Period-mAP on the validation split of the training dataset does not improve for 40 consecutive epochs. We select the checkpoint with the highest validation Period-mAP. For UCFRep, the provided validation split is used for both checkpoint selection and final evaluation because no separate test split is available. For cross-dataset evaluation (see Table 2), we use the boundary-head checkpoint selected on the validation split of its training dataset, without further checkpoint selection on the evaluation dataset.

All experiments are conducted on a single NVIDIA GeForce RTX 4070 Ti SUPER GPU. We measure computational cost on RepCount using precomputed VideoMAE v1 features, excluding feature extraction. The boundary head has 0.56 million parameters and requires 0.50 GFLOPs per video. During inference, running the frozen counting model and boundary head, followed by cycle-interval decoding, takes 5.9 ms per video on average. Peak allocated GPU memory is approximately 84 MB.

B. CYCLE LOCALIZATION RESULTS

Existing methods use evaluation metrics and experimental settings that differ from ours. DeTRC [8] predicts the mid-point and duration of each cycle, but presents only counting metrics (MAE and OBO), without a separate quantitative evaluation of cycle localization. MultiCounter+ [9] evaluates cycle localization using spatiotemporal tube IoU, which requires per-person bounding-box and identity annotations that RepCount and UCFRep do not provide. In contrast, we evaluate class-agnostic temporal cycle localization on these datasets using tIoU between predicted and ground-truth cycle intervals.

Table 2 shows that CycLocNet achieves better localization performance on RepCount than on UCFRep when the

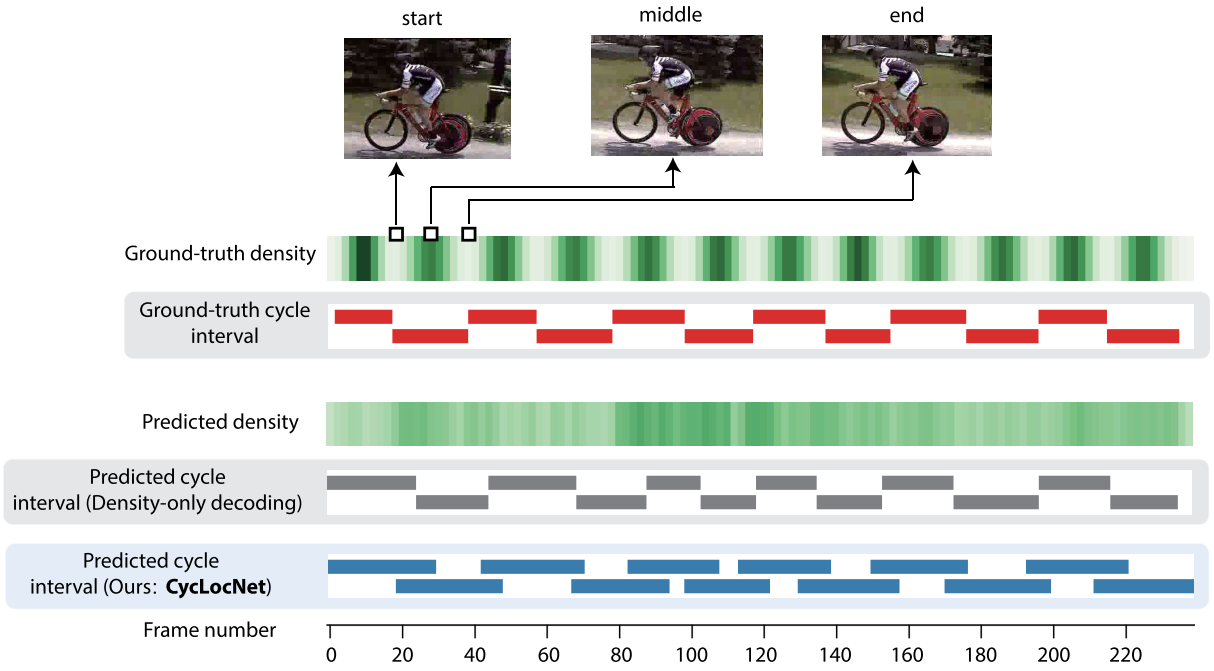


FIGURE 7. Qualitative cycle-localization result on UCFRep. Ground-truth and predicted intervals are shown in red and blue, respectively. CycLocNet performs worse than density-only decoding on this video but achieves better overall localization performance on UCFRep (see Table 4).

boundary head is trained and evaluated on the same dataset. On UCFRep, AP@.75 (4.08%) is considerably lower than AP@.50 (31.19%), indicating that the model can identify approximate cycle locations but has difficulty precisely localizing their boundaries. This may be attributed to the short, consecutive cycles in UCFRep, with little or no background between cycles. In addition, UCFRep has a smaller training set of 421 videos [22], compared with 758 training videos in RepCount [12].

We evaluate the boundary head trained on one dataset on the other without adaptation, using the frozen counting model trained on the evaluation dataset. Performance decreases in both directions, indicating limited cross-dataset generalization. AP@.75 is much lower than AP@.50 in both cross-dataset settings, suggesting difficulty in precisely localizing cycle boundaries.

Table 3 presents the counting accuracy of the frozen counting model, whose predictions remain unchanged during cycle-localization training. The higher MAE on UCFRep (0.409) than on RepCount (0.264) demonstrates the lower counting accuracy of the model on UCFRep. Because CycLocNet derives the number and approximate centers of cycles from the predicted density map, this larger counting error may also contribute to its lower localization performance on UCFRep.

C. ABLATION STUDIES

We conduct all ablation studies on RepCount because it has a larger training set than UCFRep. We evaluate the contribution of the boundary prediction head and compare three boundary representations. We also examine the loss terms and the distance-bin configuration. Finally, we evaluate sensitivity to cycle center estimation errors.

TABLE 2. Within-dataset and cross-dataset cycle-localization results on RepCount and UCFRep. Training dataset refers to the dataset used to train the boundary head. Evaluation uses the frozen counting model trained on the evaluation dataset, without adapting the boundary head. Results are percentages (mean ± standard deviation) over three experiments.

Training	Evaluation	Period-mAP	AP@.50	AP@.75
RepCount	RepCount	32.18 ± 2.34	58.84 ± 2.83	32.21 ± 3.06
UCFRep	RepCount	8.08 ± 0.45	26.74 ± 1.25	3.11 ± 0.42
UCFRep	UCFRep	9.68 ± 0.84	31.19 ± 2.03	4.08 ± 0.61
RepCount	UCFRep	4.64 ± 0.52	13.44 ± 1.36	2.26 ± 0.23

TABLE 3. Within-dataset counting accuracy of the counting model shown in Fig. 1 on RepCount and UCFRep. Each model is trained and evaluated on the corresponding dataset.

Dataset	MAE ↓	OBO ↑
RepCount	0.264 ± 0.023	0.479 ± 0.041
UCFRep	0.409 ± 0.022	0.521 ± 0.022

1) EFFECT OF THE BOUNDARY PREDICTION HEAD

One simple approach is to obtain the k-th cycle directly from the cumulative density. The levels $k-1$ and k determine its boundaries, whereas $k-\frac{1}{2}$ determines its center. The boundaries are obtained by linear interpolation, and the number of intervals is $\hat{K}$. Because this approach does not produce boundary distributions, each interval is ranked by its maximum density value. We refer to this approach as *density-only decoding*. Density-only decoding does not explicitly

TABLE 4. Comparison of density-only decoding and CycLocNet on RepCount and UCFRep. Results are mean ± standard deviation over three experiments.

Dataset	Method	Period-mAP ↑	AP@.50 ↑	AP@.75 ↑
RepCount	Density-only	5.47 ± 0.57	22.76 ± 1.79	1.04 ± 0.27
	CycLocNet	$\mathbf{32.18} \pm 2.34$	$\mathbf{58.84} \pm 2.83$	$\mathbf{32.21} \pm 3.06$
UCFRep	Density-only	6.25 ± 0.37	23.85 ± 2.21	2.06 ± 0.32
	CycLocNet	$\mathbf{9.68} \pm 0.84$	$\mathbf{31.19} \pm 2.03$	$\mathbf{4.08} \pm 0.61$

TABLE 5. Effect of the loss terms on RepCount. CycLocNet uses $\lambda_{\text{DFL}} = 1$ and $\lambda_{\text{IoU}} = 1$. Results are percentages (mean ± standard deviation) over three experiments.

λ_{DFL}	λ_{IoU}	Period-mAP	AP@.50	AP@.75
1	1	$\mathbf{32.18} \pm 2.34$	$\mathbf{58.84} \pm 2.83$	$\mathbf{32.21} \pm 3.06$
1	0	32.11 ± 2.53	58.67 ± 2.82	31.94 ± 3.42
0	1	25.67 ± 0.90	52.41 ± 1.33	22.59 ± 1.36

separate repetitive intervals from background, so its predicted cycle intervals may include non-repetitive frames (see Figs. 6 and 7). It uses the same frozen counting model, predicted cycle count, and estimated centers as CycLocNet. CycLocNet achieves higher Period-mAP, AP@.50, and AP@.75 than density-only decoding on both datasets (see Table 4). These results support the use of the boundary prediction head to refine cycle boundaries beyond those obtained from cumulative density alone.

2) EFFECT OF LOSS TERMS

We compare DFL alone, interval IoU loss alone, and their equally weighted combination using the same distributional boundary head, by setting λ_{DFL} or λ_{IoU} in Equation (5) to 0 or 1. The results are presented in Table 5. Removing DFL reduces Period-mAP from 32.18 to 25.67 and AP@.75 from 32.21 to 22.59, whereas DFL alone yields results similar to the combined loss. Thus, DFL provides the main contribution to localization accuracy, while interval IoU loss has a relatively small additional effect.

3) EFFECT OF THE DISTANCE-BIN CONFIGURATION

We evaluate the effects of the number of distance bins R and the maximum distance $B_{\max}$ on the localization performance of CycLocNet. The results are presented in Table 6. Among the evaluated configurations, $R = 32$ with $B_{\max} = 32$ yields the highest mean AP@.50 and AP@.75.

4) COMPARISON OF BOUNDARY REPRESENTATIONS

We compare three boundary representations using the boundary prediction head in Fig. 3, with its output layer adapted to each representation. All three methods use cycle centers estimated from the cumulative density.

- *Symmetric duration* predicts a single cycle duration for each estimated center. The start and end boundaries are placed symmetrically around the center.
- *Direct regression* predicts two scalar distances: one from the estimated center to the start boundary and the other to the end boundary. This representation allows asymmetric cycle intervals.
- *Distributional prediction (CycLocNet)* predicts the start and end distances as separate probability distributions over discrete distance candidates, as illustrated in Fig. 2. The expected start and end distances are then used to decode the cycle interval.

TABLE 6. Effect of the number of distance bins R and the maximum distance $B_{\max}$ on RepCount. CycLocNet uses $R = 32$ and $B_{\max} = 32$. Results are percentages (mean ± standard deviation) over three experiments.

R	$B_{\max}$	Period-mAP	AP@.50	AP@.75
8	32	26.51 ± 1.34	53.78 ± 2.03	23.47 ± 1.38
16	32	30.04 ± 1.41	57.07 ± 2.03	28.73 ± 1.62
32	32	$\mathbf{32.18} \pm 2.34$	$\mathbf{58.84} \pm 2.83$	$\mathbf{32.21} \pm 3.06$
64	32	$\mathbf{32.18} \pm 2.29$	58.38 ± 2.82	32.19 ± 2.88
32	16	26.42 ± 1.43	53.24 ± 1.64	23.50 ± 2.28
32	64	31.57 ± 2.13	58.49 ± 2.55	30.96 ± 2.75
32	128	29.24 ± 1.42	57.20 ± 1.90	26.98 ± 1.88

TABLE 7. Effects of boundary representation and DFL on RepCount. DFL requires distributional outputs. Results are percentages (mean ± standard deviation) over three experiments.

Representation	Loss	Period-mAP	AP@.50	AP@.75
Symmetric duration	IoU	17.57 ± 1.00	51.75 ± 2.12	8.10 ± 0.85
Direct regression	IoU	27.08 ± 1.30	56.02 ± 1.74	23.80 ± 1.87
Distributional	IoU	25.67 ± 0.90	52.41 ± 1.33	22.59 ± 1.36
Distributional (CycLocNet)	DFL + IoU	$\mathbf{32.18} \pm 2.34$	$\mathbf{58.84} \pm 2.83$	$\mathbf{32.21} \pm 3.06$

To compare boundary representations under the same loss, we train all three heads using interval IoU loss alone. We then evaluate the effect of adding DFL to the distributional head, yielding the CycLocNet loss in Equation (5). With interval IoU loss alone, direct regression achieves higher scores than symmetric duration and distributional prediction. DFL requires probability distributions over discrete bins and therefore cannot be directly applied to the scalar outputs of symmetric duration or direct regression. We thus evaluate its effect only for the distributional head. Compared with training this head using interval IoU loss alone, adding DFL increases Period-mAP from 25.67 to 32.18, AP@.50 from 52.41 to 58.84, and AP@.75 from 22.59 to 32.21. These results demonstrate the benefit of DFL for distributional boundary prediction.

5) SENSITIVITY TO CYCLE CENTER ESTIMATION ERRORS

During inference, CycLocNet estimates cycle centers from the cumulative predicted density, so inaccurate density predictions can lead to center-position errors. To examine sensitivity to these errors, we apply the same fixed shift to all estimated centers. We test four shifts: -4, -2, $+2$, and $+4$ temporal tokens, where negative and positive values indicate earlier and later positions, respectively. Each interval is re-decoded using the boundary predictions at the token nearest to the shifted center, and its confidence score is computed at the same token. The model is not retrained,

TABLE 8. Sensitivity of CycLocNet to cycle center estimation errors on RepCount. Results are percentages (mean ± standard deviation) over three experiments. Negative and positive shifts indicate earlier and later center positions, respectively.

Center shift (tokens)	Period-mAP	AP@.50	AP@.75
−4	29.22±0.53	57.69±1.20	27.21±0.98
−2	31.21±1.29	58.10±1.47	30.77±1.85
0	**32.18**±2.34	**58.84**±2.83	**32.21**±3.06
+2	31.55±2.17	58.81±2.32	30.73±2.91
+4	29.46±1.73	57.05±1.83	27.75±2.74

and the predicted count remains unchanged. Table 8 shows that AP@.75 decreases as the shift magnitude increases in either direction. The decrease is larger for AP@.75 than for AP@.50, indicating that localization at the stricter tIoU threshold is more sensitive to center-position errors.

D. QUALITATIVE RESULTS

Figs. 6 and 7 show qualitative cycle-localization results on RepCount and UCFRep, respectively. Ground-truth and predicted intervals are shown in red and blue. Adjacent cycles are displayed on alternating rows to distinguish their start and end boundaries. The CycLocNet predictions align well with the ground-truth intervals in the RepCount example. CycLocNet performs worse than density-only decoding in the UCFRep example shown in Fig. 7. However, it achieves better overall localization performance on UCFRep, as shown in Table 4.

E. LIMITATIONS

When the boundary head is trained and evaluated on the same dataset, CycLocNet performs better on RepCount than on UCFRep (see Table 2). The low AP@.75 on UCFRep indicates that precise boundary localization remains challenging at a strict tIoU threshold of 0.75. This may be related to its short, consecutive cycles, which have little or no background between them. The smaller UCFRep training set may further limit localization accuracy. Table 3 shows higher counting error on UCFRep, which may affect the estimated number and centers of cycles. CycLocNet relies on both the density map and temporal features $\mathbf{X}_{CF}$ provided by the frozen counting model for cycle initialization and boundary prediction, respectively. Cross-dataset generalization of the boundary head also remains limited. Table 2 shows that evaluating the head on another dataset without adaptation reduces performance in both directions, even when using the counting model trained on the evaluation dataset.

V. CONCLUSION

In this paper, we presented CycLocNet, a density-guided, class-agnostic model for localizing individual repetition cycles. CycLocNet estimates the number and approximate centers of cycles from the cumulative density of a frozen counting model. It then uses the temporal features at these centers to predict the start and end distances as probability distributions over discrete distance candidates. Experiments on RepCount and UCFRep demonstrate the effectiveness of CycLocNet. Ablation studies on RepCount show that the boundary prediction head improves localization performance over density-only decoding. Distributional boundary prediction also outperforms symmetric-duration and direct-regression alternatives, particularly at strict temporal IoU thresholds.

REFERENCES

[1] H. Liu, J. Ruan, and H. Zheng, "MMACA-RAC: A multi-stage multimodal fusion framework with auto-correlation attention for repetitive action counting," *Comput. Vis. Image Understand.*, vol. 270, Aug. 2026, Art. no. 104846.

[2] X. Wang, H. Wang, Y. Liu, H. Hu, and B. X. B. Yu, "D^2-STX: Decoupling Spatial–temporal cross-attention for dual-branch repetitive action counting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPRF)*, Jun. 2026, pp. 8208–8217.

[3] S. Gao, J. Li, and Q. Li, "Count what repeats: Period-adaptive multi-scale consistency for self-supervised repetitive action counting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPRF)*, Jun. 2026, pp. 8146–8155.

[4] J. Yu, Z. Zhou, X. Wu, and H. Wang, "Periodicity-aware AutoEncoder for unsupervised repetitive action counting," *IEEE Trans. Multimedia*, early access, Jul. 20, 2026, doi: 10.1109/TMM.2026.3715327.

[5] J. Lim, D. Kang, K. Ryu, and J. H. Hong, "SSTRAC: Skeleton-based dual-stream spatio-temporal transformer for repetitive action counting in videos," *IEEE Access*, vol. 13, pp. 184046–184058, 2025.

[6] S. Cao, Y. Wu, Y. Tang, D. Ge, Y. Ma, and C. Xiao, "SenCounter: Towards category-agnostic action counting in open sensor sequences," *Inf. Fusion*, vol. 118, Jun. 2025, Art. no. 102957.

[7] D. D. Nguyen, L. T. Nguyen, Y. Huang, C. Pham, and M. Hoai, "Class-agnostic repetitive action counting using wearable devices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 6, pp. 4984–4995, Jun. 2025.

[8] X. Ma, Z. Li, Q. Shang, W. Zhu, H. Ci, Y. Qiao, and Y. Wang, "Efficient action counting with dynamic queries," *Int. J. Comput. Vis.*, vol. 134, no. 8, Aug. 2026, p. 364.

[9] Y. Tang, D. Zhang, W. Luo, J. Zhang, W. Huang, R. Jing, and Y. Zhang, "MultiCounter+: Toward efficient and robust multi-instance repetitive action counting," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 36, no. 7, pp. 9462–9476, Jul. 2026.

[10] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen, "BMN: Boundary-matching network for temporal action proposal generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3888–3897.

[11] C.-L. Zhang, J. Wu, and Y. Li, "ActionFormer: Localizing moments of actions with transformers," in *Proc. ECCV*, 2022, pp. 492–510.

[12] H. Hu, S. Dong, Y. Zhao, D. Lian, Z. Li, and S. Gao, "TransRAC: Encoding multi-scale temporal correlation with transformers for repetitive action counting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18991–19000.

[13] Z. Zhao, X. Huang, H. Zhou, K. Yao, E. Ding, J. Wang, X. Wang, W. Liu, and F. Bin, "Skim then focus: Integrating contextual and fine-grained views for repetitive action counting," *Int. J. Comput. Vis.*, vol. 133, no. 9, pp. 6347–6361, Sep. 2025.

[14] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. J. Yang, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," in *Proc. NeurIPS*, 2020, pp. 21002–21012.

[15] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, "Counting out time: Class agnostic video repetition counting in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10384–10393.

[16] S. Sinha, A. Stergiou, and D. Damen, "Every shot counts: Using exemplars for repetition counting in videos," in *Proc. ACCV*, 2024, pp. 384–402.

[17] Z. Zhang, K. Li, S. Tang, Y. Wei, F. Wang, J. Zhou, and D. Guo, "Temporal boundary awareness network for repetitive action counting," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 21, no. 4, pp. 1–22, Apr. 2025.
[18] S. Wang, X. Shen, Y. Tang, X. Dong, W. Geng, and L. Chen, "Localization-aware multi-scale representation learning for repetitive action counting," 2025, *arXiv:2501.07312*.
[19] K. Li, X. Peng, D. Guo, X. Yang, and M. Wang, "Repetitive action counting with hybrid temporal relation modeling," *IEEE Trans. Multimedia*, vol. 27, pp. 3844–3855, 2025.
[20] D. Shi, Y. Zhong, Q. Cao, L. Ma, J. Lit, and D. Tao, "TriDet: Temporal action detection with relative boundary modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 18857–18866.
[21] Z. Tong, Z. Tong, J. Wang, and L. Wang, "VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training," in *Proc. NeurIPS*, 2022, pp. 10078–10093.
[22] H. Zhang, X. Xu, G. Han, and S. He, "Context-aware and scale-insensitive temporal repetition counting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 667–675.
[23] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 961–970.



DONGHOON KANG (Member, IEEE) received the B.S. and M.S. degrees in mechanical engineering from Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 1997 and 1999, respectively, and the Ph.D. degree in mechanical and aerospace engineering from Seoul National University, Seoul, South Korea, in 2018.

Since 2000, he has been with Korea Institute of Science and Technology (KIST), South Korea, where he is currently a Senior Researcher. Since 2020, he has also been an Adjunct Professor with the Division of AI-Robotics, KIST School, University of Science and Technology (UST). His research interests include latent variable optimization in robotics and machine learning, and video understanding using deep learning.

Dr. Kang received the Gold Medal from IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT for his outstanding performance as one of the most productive reviewers over the past seven years, from 2014 to 2020. He was also selected as one of the Outstanding Reviewers for IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, from 2017 to 2020. From 2021 to 2024, he was an Associate Editor of IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, specializing in robotics and artificial intelligence.